Classes of Defense for Computer Systems

by

Josephine Wolff

A.B., Princeton University (2010)
S.M., Massachusetts Institute of Technology (2012)

Submitted to the Engineering Systems Division
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Engineering Systems: Technology, Management, and Policy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

©Massachusetts Institute of Technology 2015. All rights reserved.

Author . . . .
Engineering Systems Division
June 1, 2015

Certified by . . . .
David D. Clark
Senior Research Scientist
Thesis Supervisor

Certified by . . . .
Kenneth A. Oye
Associate Professor, Political Science and Engineering Systems
Committee Chair

Certified by . . . .
Frank R. Field, III
Senior Research Associate

Accepted by . . . .
Munther Dahleh
William A. Coolidge Professor of Electrical Engineering and Computer Science
Acting Director, Engineering Systems Division

Classes of Defense for Computer Systems

by

Josephine Wolff

Submitted to the Engineering Systems Division
on June 1, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Engineering Systems: Technology, Management, and Policy

Abstract

Computer security incidents often involve attackers acquiring a complex sequence of escalating capabilities and executing those capabilities across a range of different intermediary actors in order to achieve their ultimate malicious goals. However, popular media accounts of these incidents, as well as the ensuing litigation and policy proposals, tend to focus on a very narrow defensive landscape, primarily individual centralized defenders who control some of the capabilities exploited in the earliest stages of these incidents. This thesis proposes two complementary frameworks for defenses against computer security breaches—one oriented around restricting the computer-based access capabilities that adversaries use to perpetrate those breaches and another focused on limiting the harm that those adversaries ultimately inflict on their victims. Drawing on case studies of actual security incidents, as well as the past decade of security incident data at MIT, it analyzes security roles and defense design patterns related to these broad classes of defense for application designers, administrators, and policy-makers. Application designers are well poised to undertake access defense by defining and distinguishing malicious and legitimate forms of activity in the context of their respective applications. Policy-makers can implement some harm limitation defenses by monitoring and regulating money flows, and also play an important role in collecting the data needed to expand understanding of the sequence of events that lead up to successful security incidents and inform which actors can and should effectively intervene as defenders. Organizations and administrators, meanwhile, occupy an in-between defensive role that spans both access and harm in addressing digital harms, or harms that are directly inflicted via computer capabilities, through restrictions on crucial intermediate harms and outbound information flows. The comparative case analysis ultimately points to a need to broaden defensive roles and responsibilities beyond centralized access defense and defenders, as well as the visibility challenges compounding externalities for defenders who may lack not only the incentives to intervene in such incidents but also the necessary knowledge to figure out how best to intervene.

David D. Clark
Senior Research Scientist
Thesis Supervisor

Kenneth A. Oye

Associate Professor, Political Science and Engineering Systems
Committee Chair

Frank R. Field, III
Senior Research Associate

Acknowledgments

I came to MIT five years ago knowing very little about computer networks and even less about policy. That I believed I would be able to do research at the intersection of those two topics was frankly laughable. That anyone else believed it was nothing short of incredible.

I was fortunate to be advised by David Clark, whose clear thinking and patience (especially in the face of my impatience) were essential every step of the way. In response to my first attempt at drawing out general lessons from the case studies in this thesis, he sent me an email with the subject line: "generalization" chapter. I have never received feedback defter or more discerning than those quotation marks.

Ken Oye helped me understand the ways in which defensive considerations were absolutely, inextricably tied to institutional interests, and served as a constant source of encouragement, as well as educational and entertaining anecdotes, throughout graduate school. Frank Field was the rare engineer who read this thesis and encouraged me to make it more philosophical, not less.

I am grateful also to Alan Davidson, Susan Landau, Fred Schneider, and Bruce Schneier for crucial conversations about security and defense, and to the National Science Foundation and Northrop Grumman for funding this work.

Members of MIT's Information Systems & Technology group were exceedingly generous with both their time and their security records, and played a vital role in helping me understand the practical considerations involved in making defensive decisions.

The advanced network architecture group in CSAIL has been a wonderful academic home for the past five years, and I am especially grateful to Shirley Hung, Susan Perez, and Jesse Sowell for support and guidance throughout that period. Other ANA members whom I have been lucky enough to share an office with include Albert Domingo, Alex Gamero-Garrido, Rubén Garcia, Cecilia Testart, and Steve Woodrow.

Beth Milnes in the Engineering Systems Divison and Ed Ballo and Barb DeLaBarre in the Technology Policy Program were unfailingly friendly presences in E40 and cheerfully helped me navigate the administrative side of MIT.

Dara Fisher made a crucial contribution to this thesis by suggesting, during an especially cold and snowy month of writing, that I should start watching *The Bachelor*. Tommy Leung and Nathan Perkins helped me keep a sense of humor and a sense of perspective about the ups and downs of graduate school.

During the 2014-15 academic year, the Berkman Center for Internet & Society provided me with a remarkable community of people interested in understanding the Internet in the context of social and political issues.

Paul Mitchell and Kevin Sullivan hosted me at Microsoft during the summers of 2012, 2013, and 2014, providing an up-close look at the endlessly fascinating and occasionally agonizing process of security policy-making.

Torie Bosch at *Slate* allowed me to test out several ideas that eventually made their way into this thesis in a series of columns about computer security and current events.

Cooper Lloyd and Mary Schnoor made MIT feel just a little bit like middle school (in the best possible way). The 2013 World Champion Red Sox made the penultimate fall of graduate school feel just a little bit like the penultimate fall of high school (in the best possible way). Email correspondence with Alexis Levinson, in which we vowed on a daily basis to put our lives in order and take the world by storm just as soon as we managed to change out of pajama pants, made the purple house feel just a little bit like our college dorm rooms (in the best possible way).

Sheila Solomon Klass hated computers almost as much as she loved writing, and she believed, with the wholly unwarranted and unconditional faith of a grandmother, that I had a great genius for both. I wrote the last chapters of this thesis on her computer and no one would have been prouder to see me finish, or more bewildered by the final product. The summer before my fourth year of graduate school, after I retrieved a web browser icon she had accidentally deleted from her desktop, she sent me an email with the subject line: You have made me the happiest Jewish grandmother-writer-with-a-big-mouth-who-should-have-gotten-married-but-instead-roasted-peanuts-in-Planter's-Duffy-Square-window-and-went-to-college! I was so lucky to know her.

Orlando and Anatol Klass let me fill out forms, make calendars, and file paperwork for them when graduate school proved to have too few deadlines and bureaucratic demands to satisfy my need for color-coded charts and recreational organizational activities. Larry Wolff, who lived with that organizational mania for many years, understood how daunted I was by the prospect of a project that couldn't be reduced to checklists and meticulous planning. The month after my general exams, as I struggled to figure out what came next, he sent me an email with the advice I most needed to hear to embark on this thesis. He wrote: remember that you can NOT lay the whole thing out in advance, exactly what you're going to read, write, and show; if you could do that then the research wouldn't really be worth doing. Perri Klass instilled in me early on the belief that life was too short for wide-ruled paper, and supplied graph-paper notebooks and pads with appropriately small squares from all over the world for my checklists and meticulous planning. Throughout graduate school, she reminded me of the joys of non-academic writing and urged me to write about the topics in this thesis for a wider audience. Together, they supported the thesis-writing process primarily by making fun of me and encouraging me to drink more (not necessarily in that order). It would have been weird for them not to have done that.

These ambiguities, redundancies and deficiencies remind us of those which doctor Franz Kuhn attributes to a certain Chinese encyclopaedia entitled ‘Celestial Empire of Benevolent Knowledge.’ In its remote pages it is written that the animals are divided into: (a) belonging to the emperor, (b) embalmed, (c) tame, (d) sucking pigs, (e) sirens, (f) fabulous, (g) stray dogs, (h) included in the present classification, (i) frenzied, (j) innumerable, (k) drawn with a very fine camelhair brush, (l) et cetera, (m) having just broken the water pitcher, (n) that from a long way off look like flies.

The Bibliographic Institute of Brussels exerts chaos too: it has divided the universe into 1000 subdivisions, from which number 262 is the pope; number 282, the Roman Catholic Church; 263, the Day of the Lord; 268 Sunday schools; 298, mormonism; and number 294, brahmanism, buddhism, shintoism and taoism. It doesn’t reject heterogene subdivisions as, for example, 179: “Cruelty towards animals. Animals protection. Duel and suicide seen through moral values. Various vices and disadvantages. Advantages and various qualities.”

I have registered the arbitrarities of Wilkins, of the unknown (or false) Chinese encyclopaedia writer and of the Bibliographic Institute of Brussels; it is clear that there is no classification of the Universe not being arbitrary and full of conjectures.

“The Analytical Language of John Wilkins,” Jorge Luis Borges

Did I ever tell you that Mrs. McCave
Had twenty-three sons and she named them all Dave?
Well, she did. And that wasn't a smart thing to do.
You see, when she wants one and calls out, "Yoo-Hoo!
Come into the house, Dave!" she doesn't get one.
All twenty-three Daves of hers come on the run!
This makes things quite difficult at the McCaves'
As you can imagine, with so many Daves.
And often she wishes that, when they were born,
She had named one of them Bodkin Van Horn
And one of them Hoos-Foos. And one of them Snimm.
And one of them Hot-Shot. And one Sunny Jim.
And one of them Shadrack. And one of them Blinkey.
And one of them Stuffy. And one of them Stinkey.
Another one Putt-Putt. Another one Moon Face.
Another one Marvin O'Gravel Balloon Face.
And one of them Ziggy. And one Soggy Muff.
One Buffalo Bill. And one Biffalo Buff.
And one of them Sneepy. And one Weepy Weed.
And one Paris Garters. And one Harris Tweed.
And one of them Sir Michael Carmichael Zutt
And one of them Oliver Boliver Butt
And one of them Zanzibar Buck-Buck McFate . . .
But she didn't do it. And now it's too late.

"Too Many Daves," Dr. Seuss

Contents

1 Introduction **17**

1.1 Defense in Depth 19

1.2 Classes of Defense & Classes of Attack 20

1.3 Cases & Units of Analysis 22

1.4 Access & Harm 24

1.5 Classes of Defenders 26

1.6 Thesis Organization 27

2 Origins of Defense in Depth **29**

2.1 Defense in Depth in Military Strategy 30

2.2 Defense in Depth for Nuclear Security 34

2.3 Information Assurance Through Defense in Depth 36

2.4 Computer Defense Catalogs & Taxonomies 40

2.4.1 NIST Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations . . . 41

2.4.2 ISO/IEC 15408: Common Criteria for Information Technology Security Evaluation 43

2.4.3 Twenty Critical Security Controls for Effective Cyber Defense 45

2.4.4 High-Level Information Security Frameworks 48

2.5 Definitions of Defense in Depth in Computer Security 50

2.5.1 Perverse Effects in Defense 55

2.5.2 Independent Defenses 59

3 Access and Harm **65**

3.1 Attack Trees and Kill Chains 72

3.1.1 Narrowing of Options 73

3.2 Access Capabilities 74

3.2.1 Capabilities for Unauthenticated Users 77

3.2.2 Capabilities for Authenticated Users 79

3.2.3 Insider Threats 81

3.3 Harms and Attacker Goals 81

3.3.1 Digital Harms 83

3.4 Intermediate Harms 85

4 Case Studies in Defense **87**
4.1 TJX Companies, Inc. Breach (2005–2007) 88
4.1.1 Access Capabilities & Defense 91
4.1.2 Harm Defense 93
4.2 DigiNotar Compromise (2011) 96
4.2.1 Access Capabilities & Defense 97
4.2.2 Harm Defense 100
4.3 PLA Unit 61398 Espionage (2013) 105
4.3.1 Access Capabilities & Defense 106
4.3.2 Harm Defense 108
4.4 Spamhaus Denial-of-Service Attacks (2013) 109
4.4.1 Access Capabilities & Defense 111
4.4.2 Harm Defense 113
4.5 Defender Interests 114

5 Application Design as Defense **117**
5.1 Access Defense at the Application Layer 119
5.1.1 Restricting Non-Credentialed Capabilities 119
5.1.2 Restricting Credentialed Capabilities 128
5.2 Harm Defense at the Application Layer 132

6 Management as Defense **135**
6.1 Administrative Access Defense 136
6.1.1 Tailoring Application Restrictions 136
6.1.2 Protecting Authentication Credentials 138
6.2 Defending Against Intermediate Harms 141
6.2.1 What Security Looks Like to an Institution 142
6.2.2 Outbound Traffic 142
6.3 Institutional Harm Defense 144
6.3.1 Overlooking Harm Defense 145
6.3.2 Disruption of Physical Service 147
6.3.3 Disruption of Digital Service 149
6.3.4 Espionage 150
6.3.5 Financial Loss 151

7 Policy as Defense **153**
7.1 Targeting Attackers Versus Targeting Defenders 153
7.2 Defender-Oriented Policy Levers 155
7.2.1 Tailoring Policies to Different Defenders 157
7.3 Policy for Access Defense 158
7.4 Policy Harm Defense 161
7.5 Security Reporting Policies 164

8 Conclusion **169**

8.1 Two Frameworks for Defense 169

8.2 Roles of Different Defenders 171

8.3 Revisiting Defense in Depth 173

References **175**

List of Figures

3-1 The number of compromised accounts reported to IS&T during the year leading up to the implementation of a new password policy in July 2013 and for the period during the implementation (July 1, 2013–July 1, 2014) and following its implementation, through December 2014. 66
3-2 The number of compromised hosts reported to IS&T during the year leading up to the implementation of a new firewall in Fall 2013, as well as the following months during the gradual roll out across campus. . . 67
3-3 The number of different types of security incidents recorded by IS&T yearly from 2005 through 2014. 68
3-4 A possible division of different computer system capabilities according to their potential to be used for legitimate and malicious purposes. . 77

4-1 Diagram of DigiNotar's network security zones. Source: Hoogstraaten et al. (2012). 98
4-2 Screenshot of message left on DigiNotar's computers by the perpetrator of the CA's compromise. 101
4-3 Mandiant's Attack Lifecycle Model. Source: Mandiant (2013). 106
4-4 The design of the DDoS attacks directed at Spamhaus. Source: Markoff and Perlroth (2013). 112

5-1 An email sent to MIT email accounts prompting recipients to visit a website for security purposes. 121
5-2 The website linked to by the email in Figure 5-1 prompting visitors to enter their MIT username and passwords. 121
5-3 Potential of different email capabilities afforded to users with unknown credentials to be used for malicious and legitimate purposes. 122
5-4 Potential of different web browser capabilities afforded to users with unknown credentials to be used for malicious and legitimate purposes. 124

List of Tables

2.1 Levels of defense in depth for nuclear plant safety. Source: *Defence in Depth in Nuclear Safety, INSAG-10* (1996). 35

2.2 The eighteen families of security controls identified in NIST 800-53 Revision 4, as well as their corresponding classes, as designated by Revision 3 of the same document. 42

2.3 The eleven classes of functional security requirements listed in the Common Criteria. 44

2.4 The twenty critical security controls for effective cyber defense identified in version 4.0 of the CSC. 45

2.5 Controls from NIST 800-53 mapped onto each of the Twenty Critical Security Controls. 47

2.6 Technology Summary Table from "Information Assurance Through Defense in Depth." Source: Woodward (2000, p. 14). 49

2.7 Definitions of defense in depth presented in 2000 U.S. military report on "Defense in Depth for Information Assurance." 56

6.1 Comparison of different divisions of attacks into stages. 146

7.1 Different classes of defenders and the scope of their control within the security ecosystem. 159

7.2 Different purposes of security incident reporting policies. 165

Chapter 1

Introduction

In April 2013, the Massachusetts Institute of Technology (MIT) announced several new computer security measures, ranging from changes in the treatment of incoming traffic and the operation of the university's Domain Name Service (DNS) servers to new password complexity and expiration requirements to access restrictions for major administrative applications. Because of my interest in how people defend computer systems, and the interactions among the various technical, managerial, and policy decisions that constitute those defensive strategies, I spent some time over the course of the following year trying to figure out why the Institute had chosen this particular set of changes—how they fit into MIT's broader security agenda and what overarching logic or goals had dictated this combination of defenses rather than any other. Members of MIT's Information Systems and Technology (IS&T) group and its governing committee offered a variety of different rationales for the policy changes. Most of these explanations centered on how embarrassed and concerned the Institute's leadership had been following the suicide of Internet activist Aaron Swartz in January 2013, when MIT's homepage had been briefly redirected to a makeshift memorial webpage commemorating Swartz and condemning MIT for the role it played in bringing charges against him for downloading large volumes of academic articles on the university's network. According to several interviewees, the homepage debacle had spurred the university to update its security practices, an explanation that aligned with survey research suggesting that organizations often employ a "wait-and-see" approach to investing in online security, waiting until after breaches to invest in new security measures (Gordon, Loeb, & Lucyshyn, 2003).

But while this explanation shed some light on why MIT had chosen that particular moment to change its security mechanisms, it offered very little insight into why MIT had chosen that particular set of security mechanisms—only one of the new measures (a contract with Akamai to improve resilience of mit.edu) would have helped defend against the redirection attack that had so embarrassed MIT, and none of the new measures were in any way related to defending against the high-volume downloading activity perpetrated by Swartz. So, if they weren't aimed at addressing the kinds of behavior that had, apparently, prompted their implementation, what, then, were these new defenses supposed to accomplish? And more importantly, how had MIT landed on those goals to drive the selection of their new defenses?

Several IS&T employees alluded to the Institute's desire to keep up with "best practices" and match the security postures of other universities. Often, interviewees suggested that MIT was "catching up" with others in implementing changes that many of its peers had made several years earlier. No one I spoke with was able to articulate a defensive strategy that went beyond the basic premise of "everyone else is doing this, so we probably should, too." No one was able to speak to MIT's broader security goals (besides keeping up with peer institutions), or any framework or logic that had governed the specific selection of the new defenses. Finally, one IS&T employee asked me point-blank of the security changes: "What makes you think there's any overarching plan or logic?"

This thesis is motivated by that lack of logic—by the inability of so many capable and experienced computer security practitioners, at MIT and elsewhere, to articulate a high-level rationale or set of organizing principles for not just an individual defensive decision, but an entire complex, expensive array of them. MIT is not alone in struggling to make sense of the defensive landscape. Writing that computer defense mechanisms are "proliferating and becoming increasingly complex—exceeding our ability to understand how best to configure each mechanism and the aggregate of mechanisms," Saydjari (2004) argues that "Today, the process of designing a well-defended system is a matter of black art."

This notion of computer defense as something more akin to magic than science calls to mind the most notoriously ill-fated and unteachable of defense classes, Hogwarts' Defense Against the Dark Arts curriculum, in which Professor Snape, during his brief stint as instructor, cautions students (in language reminiscent of Saydjari's) that:

> the Dark Arts are many, varied, ever-changing and eternal. Fighting them is like fighting a many-headed monster, which, each time a neck is severed, sprouts a head even fiercer and cleverer than before. You are fighting that which is unfixed, mutating, indestructible . . . Your defenses . . . must therefore be as flexible and inventive as the Arts you seek to undo (Rowling, 2005).

Defense—whether against computer-based attacks or mythical monsters—is presented as an essentially unteachable art, a skill gifted to the lucky few but beyond the grasp of most, a talent unattainable through mere hard work and diligent study. This mindset is perhaps partly responsible for the copycat mentality of computer defenders, the inclination to imitate the actions of others rather than trust one's own abilities, and it is suggestive of how thoroughly we have failed to come up with satisfying, comprehensive frameworks for understanding computer system defense. This idea of transitioning from an "art" to a practice reducible to competent analysis also speaks to issues at the core of broader notions of engineering science and design—fields that are largely concerned with translating ill-defined, qualitative design problems into well-defined, quantitative specifications and solutions (Layton, 1976; Vincenti, 1990).

The questions at the heart of this work center on how we think about different computer system defenses in aggregate, the mental models that govern our understanding of how those defenses interact with and relate to each other, and what a particular set of them does—and does not—collectively protect against. More specifically, this thesis focuses on drawing out different classes of computer system defenses—and defenders—to look at how different ways of categorizing and organizing the vast assortment of available defensive options can yield insight into how we—as application designers, as managers, and as policy-makers—can make better decisions for computer security. The goal of this analysis is to provide a framework that enables us to talk about MIT's DNS changes, password policies, and firewall in the same sentence—to say something about how and why they work in concert, what threats they do and do not address, and which other classes of defense, and defenders, might prove useful to further reinforce those protections; to say something that is not mired in bleak wait-and-see or keeping up with the Joneses attitudes about computer security, much less mystical notions of black art or magic. This objective is not specific to MIT; the design patterns provided in this thesis are aimed at helping a variety of actors think through their respective roles in defending against different threats. Every set of defenders discussed here has opportunities to defend against multiple types of threats, just as every class of threat described can be defended against by multiple different actors. Individual security decisions are made in the context of this broader defensive ecosystem, ideally by people with some understanding of where and how they, and the measures they implement, fit into it. The point of classifying computer defenses—or defenders, or attacks, for that matter—lies in providing a lens that somehow clarifies or changes that understanding.

As a research agenda this is, at once, aggressively arrogant—to say something that radically changes how you think about computer defense!—and rather laughably unambitious—to say something about defense slightly more illuminating than an appeal to best practice or black magic. Perhaps that is only to say that the philosophy of computer defense, in contrast to the technology of it, is a relatively untrod area, and the mental models correspondingly flimsy. This thesis begins by trying to understand where those mental models derive from—what influences shape the ways we think about defending computer systems—and why they often prove unsatisfactory. At my least ambitious, this seems to me the central contribution of this work—a detailed deconstruction of the inadequacies of our current frameworks for understanding defense, a deconstruction that I hope makes a compelling case for why we need to spend some time thinking about how we think about computer system defense in search of better frameworks. At my most ambitious, I hope I have made some headway here towards finding one.

1.1 Defense in Depth

We use a variety of tools and techniques to defend ourselves against computer-based threats—encryption, anti-virus programs, passwords, firewalls, software patches, and network traffic monitoring, to name a few—on the understanding that a well-defended system requires more than just one line of defense. The idea that computer defenses should be combined and layered to reinforce each other is often invoked under the umbrella notion of "defense in depth," a term so vaguely defined and inconsistently applied in computer security that it sometimes seems to signify only that more defense is better than less. As a guiding philosophy, this is both untrue and unhelpful—untrue because layering more defenses in a system may actually make it easier for an attacker to compromise if the functions of different defenses are at odds with each other (Kewley & Lowry, 2001), and unhelpful because it offers little direction for defenders operating under various constraints trying to choose among the array of different security options. Developing a more nuanced understanding of what defense in depth means in the context of computer systems is closely related to the problem of defining classes of computer defense—both questions center on the characterization of different types of computer defense in relation to each other and in relation to the types of threats they thwart, and both are aimed at helping defenders navigate an ever-increasing space of security choices.

Understanding the different ways that ideas about defense in depth have been appropriated from other fields—including military history and nuclear safety—and applied to computer security sheds some light on the inconsistencies and confusion that underlie existing classification schemes for computer security controls. This historical perspective on defense in depth also explains some of the ways in which defenses for computer systems differ from defenses in other fields, and why the language and organizing frameworks of physical system security have proven difficult to translate into the digital realm. Untangling the origins of defense in depth and its different meanings in the context of computer security is therefore a crucial first step in making sense of the existing perspectives on different classes of defense and how the current landscape got so muddled.

We know we need multiple defenses protecting our computer systems—that one isn't enough because each has vulnerabilities of one form or another—but it's difficult to know where to begin when it comes to combining them, and notions of defense in depth appropriated from other fields have done little to clarify this space, instead serving primarily to add to the confusion. At this point, defense in depth is not a meaningful idea in the context of computer security, though it continues to be regularly invoked, often as little more than a guise for advocating more and more security controls of any form whatsoever. Our inability to imbue defense in depth with any consistent meaning or more specific definition in this context is closely tied to our inability to divide computer defenses into consistent classes that can inform some definition of defensive depth and its construction. Every classification of defenses, in some sense, invokes its own notion of how defenses should be combined, by determining which different buckets defenders should draw from when assembling security protections, and, accordingly, what it means to design a defensive strategy that is stronger—deeper, even—than a single defense.

1.2 Classes of Defense & Classes of Attack

Inevitably, thinking about defense means thinking about attacks, or rather, about what is being defended against. In fact, classifying defenses is essentially a means of classifying attacks according to how they can be prevented, or interrupted. The process of identifying classes of defense is essentially the process of identifying classes of attack—but the reverse is not necessarily true. There are several existing taxonomies of computer attacks and intrusions that organize such incidents according to a range of characteristics, including their technical modes, targets, impact, sources, and exploited vulnerabilities (Lindqvist & Jonsson, 1997; Hansman & Hunt, 2005; Kjaerland, 2006; Harrison & White, 2011). However, such taxonomies are typically designed for the purpose of helping information and incident response bodies keep track of and categorize reported incidents. In other words, attack taxonomies are usually intended to help with attack trend data collection rather than defense decisions and, as such, their classes often do not map clearly onto specific sets of defenses.

For instance, Kjaerland (2006) proposes ten different "methods of operation" for classifying cyber incidents: misuse of resources, user compromise, root compromise, social engineering, virus, web compromise, trojan, worm, recon, and denial of service. But the same types of defenses may be used to protect against multiple of these categories—for instance, anti-malware programs may target viruses, trojans, worms, reconnaissance efforts, and web compromises alike—leaving the defensive implications of such taxonomies unclear. Similarly, Hansman and Hunt (2005) propose four dimensions for classifying attacks, including one that identifies "the main means by which the attack reaches its target" with eight overarching categories of attack vectors: viruses, worms, buffer overflows, denial of service attacks, network attacks, physical attacks, password attacks, and information gathering attacks. Here, again, there are several attack classes with limited relevance to defensive decisions, including information gathering attacks, viruses, and worms. These are taxonomies that emerge from looking at attack and incident data primarily from a reporting perspective rather than a defensive one, a perspective focused on how to consistently describe and record attacks for the purposes of posterity and analysis, rather than how to prevent or address them.

This emphasis on reporting divorced from defensive considerations is echoed, to some extent, even in the foundational information security framework of confidentiality, integrity, and availability. These three seminal security properties may help organize security incidents, each of which can be sorted according to the property (or properties) it violates, but they are of much less use when it comes to categorizing defenses, since many common defenses (e.g., anti-malware programs, firewalls, and passwords) do not correspond directly to any particular element of the triad. There is a tendency in computer security classification efforts to start from attacks and the ways they are typically thought about—worms, viruses, buffer overflow—and then attempt to apply those frameworks onto defenses. But many defenses block (or monitor, or mitigate) certain classes of action that do not correlate directly with those embedded attack models and could instead constitute one component of multiple different types of attacks as defined by such taxonomies. Using the language of attacks to inform or organize the ways we think about defenses therefore requires some careful consideration of the relationship between what specific action or capability a defense constrains, and the different contexts in which that action might be carried out for malicious purposes.

This analysis also starts by looking at security incidents, but it aims to do so from a perspective that is explicitly defensive, a perspective that takes as its primary aim the division of the computer defense landscape in a way that maps clearly onto actual security tools and techniques—hence the emphasis on classes of defense rather than classes of attack.

The inextricable relation of attack and defense classes also relates to discussions of defense in depth. Kewley and Lowry (2001) define defense in depth as "multiple mechanisms against a particular attack class," in contrast to "defense in breadth," which they view as "multiple mechanisms across multiple attack classes." But defining the classes of defense that constitute defense in depth (or breadth, as the case may be) in terms of attack classes merely shifts the onus for developing a logic and taxonomy to the attack space, especially since many of the ways in which computer attacks and intrusions are commonly classified do not clearly correspond to specific defenses. In the absence of a clear and consistent attack taxonomy, this distinction between depth and breadth is entirely arbitrary, and the circularity of defining defense classes in terms of unspecified attack classes does little to clarify how defenses should be combined or what such combinations protect against in aggregate.

1.3 Cases & Units of Analysis

Tackling security incidents from a defensive perspective means setting aside the familiar categories of worms and viruses and trojans, or confidentiality, integrity, and availability breaches, and trying to focus solely on the defensive opportunities and interventions that occur across different incidents, and how those defensive patterns can be used to divide up the attack space. That is the primary analytical lens applied to the security incident cases discussed in this thesis: what opportunities did they present for defensive interventions and what impact would such defenses have on the work required of the perpetrators. Specifically, this work analyzes the 2007 TJX Companies Inc. breach, the 2011 compromise of the Dutch DigiNotar certificate authority, the 2013 espionage efforts of Unit 61398 of the Chinese People's Liberation Army, and the 2013 distributed denial-of-service attacks directed at the organization Spamhaus. While many other incidents are referenced and discussed in passing, these were selected to span four different classes of harm inflicted through computer security breaches: financial theft, political espionage, economic espionage, and digital service disruption (a fifth class of harm—physical service disruption—is also discussed, though too few known cases exist, beyond the infamous Stuxnet worm, to allow for much comparative analysis).

Individual security incidents are the focal unit of analysis for this work, and these cases allow for comparison across the different defensive interventions that might impact such incidents and the variety of actors poised to carry out such interventions. A secondary unit of analysis in this work is institutional, taking as a case study roughly a decade's worth of security incidents and defensive decisions made by a single organization, MIT. While this case is also built predominantly on incident records, bolstered by some interview data and administrative documents, it looks at the impact of defensive decisions made by a single entity on a wide range of incidents—rather than the impact of defensive decisions made by a wide range of entities on a single incident. The MIT incident records, spanning 2004 through 2014, are drawn directly from IS&T's security queue, in which individual incidents are logged in separate "tickets" that typically include any notes, documents, or email correspondence relating to the incidents and how they were addressed. Additionally, security-related incidents logged in IS&T's accounts and help desk queues were added to those in the security queue, totaling roughly 17,000 separate incidents over a period of ten-and-a-half years.

In contrast to the in-depth incident case studies, the majority of MIT's recorded incidents were brief, unsophisticated, and resulted in minimal—if any—actual harm, so far as MIT was able to track them. However, while these more mundane records offer less insight into the complicated and clever ways attackers perpetrate threats over computer networks than cases like the TJX breach and the DigiNotar compromise, they provide a useful supplement by revealing an individual defender's day-to-day security concerns and decisions. More importantly, they offer some idea of how limited any individual defender's window into a particular incident is—what a tiny snapshot an institution like MIT is afforded of the complex chain of events that lead up to potentially serious breaches, and how little ability that institution may have to assess what harm, if any, is being inflicted, who the victims are, or how best to intervene.

This analysis is built on records of real security events, analyzed through the lens of individual incidents and individual entities. These two units of analysis are meant to provide different, complementary perspectives on how we think about different classes of defense in the context of computer systems—the former sheds light on how different actors and defenses can work in concert to thwart individual incidents, the latter focuses on how individual actors grapple with defensive decisions that span different threats and incidents that they themselves often have very limited visibility into. The goal of combining these two frames is to be able to understand classes of computer system defense in the context of both a wide range of incidents with different aims and a diverse set of different defenders with slightly different capabilities and windows into security breaches.

Using data about real incidents has the advantage of not requiring any assumptions about or simulations of ill-understood areas such as threat actor behavior or incident cost, but it also has limitations. The data itself is in short supply since detailed data about security incidents—including how they were perpetrated and what defensive measures were and were not in place to protect against them—is not available for the vast majority of computer security breaches. This limits the selection of cases for in-depth analysis to the few instances where detailed information is made public, either through legal battles, investigative reporting, independent analysis, or voluntary disclosure on the part of the affected parties. Similarly, it is extremely rare for an organization to make the entirety of its security incident records available for research, thus the institutional lens of this analysis is limited to a single institution—the only one willing to grant that access.

The limited data available about incidents and institutions in this area presents serious challenges to generalizing this analysis of the roles of different defenses and defenders, and wherever possible additional cases and examples are offered to bolster the general arguments at the heart of this work. Undoubtedly, this research would benefit tremendously from greater availability of information about actual security incidents and institutions. In a climate where organizations are deeply reluctant to reveal voluntarily even the smallest details about breaches for fear of liability and negative publicity, however, it does not seem likely that that information is imminent. This is, ultimately, a thesis that draws broad conclusions about defense from a relatively small window onto the security landscape. As that window of available security information expands in the coming years—if, in fact, it does expand—it will be interesting to see how well these conclusions hold up.

1.4 Access & Harm

One of the central arguments of this thesis is that computer security incidents involve both access and harm components—sometimes clearly separable from each other, and sometimes less so—and that defenses tend to be oriented along one of these dimensions. The notion of access to a computer or a network is not binary—rather, it encompasses a wide range of capabilities that people may exercise in the context of a computer system, including everything from being able to send an email to or establish a connection with a targeted system to installing malicious code or exfiltrating data. Access capabilities are the building blocks attackers use to perpetrate some kind of harm—be it financial theft, espionage, or service disruption—and many of the defenses that we rely on are aimed at cutting off, or narrowing, those access pathways.

Often, though not always, there is a distinction between those digital capabilities and the actual damage a perpetrator hopes to inflict. For instance, using stolen payment card numbers to perpetrate financial fraud requires both the theft of financial information from a targeted computer system (digital access) as well as the successful sale and use of that information for fraudulent charges *outside the context of the targeted computer system* (harm). In another example, Levchenko et al. (2011) analyze sales of spam-advertised pharmaceuticals, replica luxury goods, and counterfeit software and determine that the most effective defensive intervention in these markets is to crack down on the few banks that provide merchant services for the large majority of these transactions. The intervention advocated by the authors, in which credit card companies refuse to settle certain transactions with banks known to work with spam-advertised retailers, occurs well beyond the context of the computer systems used to send (or receive) spam emails and is distinct from access defense interventions that target those systems by, for instance, regulating spam-sending bots or improving spam filtering. Similarly, using computer access capabilities to alter operations of a physical system, as in the case of Stuxnet, requires both access to a protected computer system and the ability to use that access in ways that affect not just the targeted network but also the physical world beyond its boundaries. In such cases, access is what happens within the context of a targeted computer system and harm is what happens in the greater world beyond it.

But this distinction between digital access capabilities and the physical (or financial) harms inflicted via that access is not always so clear cut. Many threats—for instance, denial-of-service attacks and political espionage—harm victims in ways that are entirely digital, at least in their direct effects (second-order effects, such as organizations' loss of income when they are hit by denial-of-service attacks, may be more likely to ripple beyond the confines of computers). These digital harms blur the line between access and harm because, in such cases, harm is inflicted solely through the access capabilities acquired within the context of a computer system—no translation or leap into the outside world is necessary. The access-harm framing is still useful in understanding the defensive opportunities in such cases: for instance, denial-of-service attacks often feature an access stage which centers on building up botnets by acquiring the capability to send unwanted outbound network traffic from a large number of protected machines, as well as a harm stage, in which those botnets are used to bombard a victim's servers with high volumes of traffic. Just as there is a distinction between defending against credit card fraud by protecting stored payment card numbers (access defense) versus protecting against manufacturing of fraudulent cards (harm defense), so, too, there is a difference between trying to protect against denial-of-service attacks by protecting against bot infections versus filtering high-volume malicious traffic. In the former case, access defense happens in the context of a computer system and harm defense happens beyond that border; in the latter case, both access and harm defense happen in the context of computer systems, but require the intervention of different actors and the protection of separate systems.

Defending the access pathways that attackers use and protecting against the harms they aim to inflict through use of those capabilities can be thought of as two distinct undertakings, often involving different actors and mechanisms. Because a single access capability (e.g., phishing emails, malware) can be used to inflict multiple different types of harm, and because a single class of harm (e.g., financial theft, espionage) can be inflicted through the use of multiple different access capabilities, it is important to separate out these two defensive goals: constraining access and mitigating harm. This distinction allows for a clearer understanding of the actual impact—and limitations—of any individual defense. Access defenses such as firewalls do not protect against classes of harm, and harm mitigation defenses, such as monitoring anomalous payment card activity, do not constrain attackers' access to protected computer systems. Understanding how classes of defense relate to classes of attacks requires acknowledging this disconnect between access- and harm-oriented defenses: they protect against two very different things which do not map clearly onto each other.

Access and harm represent two fundamentally different ways of thinking about security threats—the latter focused on what threat actors ultimately aim to do, the former focused on how, specifically, they use computers to do it—and, as such, they call for separate defensive treatment. Access defense requires a more broad-based, general approach to restricting computer system capabilities that might be used for any form of malicious activity, while harm mitigation defenses are more focused and tailored to interrupt a specific type of malicious activity. At the root of this divide is a deeper question relating to what computer security actually means: is a computer secure only "if you can depend on it and its software to behave as you expect" (Garfinkel, Spafford, & Schwartz, 2003) or is it secure so long as it cannot be used to inflict any harm? Access defense speaks to the former, more general notion of security, in which anything unexpected is automatically viewed with suspicious. Harm defenses speak to the latter idea, in which computer security matters only insofar as it affects or leads to damaging consequences for people. One framing implies that the purpose of computer system defenses is to prevent anything unusual or unanticipated, the other suggests a much narrower goal of preventing harmful outcomes. Applying both access and harm lenses to computer defense also raises key questions for defenders about when access in the absence of harm matters and is worth protecting against, and whether there are ways of designing or managing their systems so that access matters less because it affords so few opportunities for harm.

1.5 Classes of Defenders

Computer system defenses are designed, implemented, and supported by a variety of people and organizations with different agendas and interests. Just as there are infinite different ways to categorize and classify defenses, so, too, there are endless ways to group the myriad defenders involved in making these security decisions. This analysis aims to draw out the different defensive roles that certain types of actors are best positioned to undertake, and understand the relationship between the capabilities of each of these different groups. The three classes of defenders discussed are application designers, organizations, and policy-makers—all three are intentionally broad in scope, each encompassing a large swath of diverse actors with different security perspectives and priorities. The purpose of the general design patterns for defense laid out in this thesis is not to prescribe specific steps that all members of any one of these groups should take, but rather to elucidate the role that each plays in relation to the others, as well as the unique defensive stance and capabilities of actors involved in the design, operation, and regulation of computer systems. The focus is on what each of these three groups can—and cannot—do to defend against threats, both in terms of constraining potentially malicious access and mitigating harmful consequences, and also, returning to the theme of combining different defenses and understanding their interactions, what these different groups can do to bolster or reinforce each other's defensive efforts.

Just as the access and harm framings of defense invoke a more general and a more specific notion of computer security, the different classes of defenders identified here operate along a spectrum of security specificity and generality. For instance, application designers can focus on the security of specific applications—with well-defined functions—but often have to allow for a very broad, general base of users, who may use the same application in a variety of different scenarios. By contrast, managers typically operate with a narrower user base in a more clearly defined organizational setting, but, at the same time, they are responsible for defending a wide range of different applications used by that group of people. Policy-makers, meanwhile, are generally faced with the broadest populations of users to defend and the most diverse set of threats to defend them against. General security is harder than specific security—just as it is harder to defend against anything unexpected than it is to defend against anything harmful, it is also harder to defend against all threats to all people than it is to defend against specific threats to certain target populations and activities. Accordingly, the actors with the most general security agendas, such as policy-makers, tend to be most effective in the context of more specific defenses, such as those targeting classes of harm, because they need the narrowness of the defensive context to shape and balance out what would otherwise be an overly broad security mandate. By the same logic, defenders with more specific security agendas, such as those designing particular applications, are best able to implement broader, access-oriented defenses that deal with more general framings of security exactly because they can interpret those framings in the relatively narrow context of a single application. Organizations and managers operate in between these two extremes, and are often poised to defend against threats from both access and harm perspectives—though their ability to do the former is often strongly shaped by the decisions of application designers, and the latter heavily influenced by policy decisions.

1.6 Thesis Organization

This thesis proposes two complementary framings of computer system defense focused on access and harm and applies these frameworks to several case studies of security incidents as well as one case of an institution actively engaged in defending against threats. Through comparative case analysis, general design patterns are identified describing different defensive roles and capabilities of application designers, managers, and policy-makers across a range of computer-based threats. The primary contributions of the thesis are the proposed dual access and harm framings as a means of organizing broad classes of computer system defenses, as well as the defensive models they suggest for different actors when applied to actual security case studies.

Chapter 2 reviews the shortcomings of existing computer defense frameworks through an examination of the historical origins of the notion of defense in depth in the fields of military history and nuclear safety. The analysis explains how the term came to be applied in inconsistent—and sometimes conflicting—ways to the area of computer security, as well as the lasting impact of those inconsistencies on current computer defense catalogs and classifications. Chapter 3 then proposes the access and harm frameworks for computer defense, and in Chapter 4 those frameworks are applied to a series of security incident case studies. Chapters 5, 6, and 7 propose general patterns for defense extracted from these case studies and other security incidents, with Chapter 5 focusing on the defensive roles of application designers, Chapter 6 highlighting opportunities for managers and organizations, and Chapter 7 discussing potential policy levers for strengthening computer security. Finally, Chapter 8 summarizes the key conclusions of the thesis and revisits the question of how we can shed some of the misguided mental models we have come to rely on for understanding computer security and think more clearly about defending computer systems without resorting to imitation, analogy, or dark art.

Chapter 2

Origins of Defense in Depth

One of the distinct ironies of the literature on defending computer systems, and especially defense in depth, is the extent to which it turns out we rely on the language and ideas of medieval castles and the Imperial Roman Army to explain and understand a technology several thousand years their junior. The appropriation of concepts from physical security and military history is so pervasive and consistent that it almost seems to belie the artificial nature of computers and computer networks as artifacts of our own construction—we seem to grapple with them as foreign entities or strange discoveries, rather than the products of our own deliberate design, as if unable to understand them without appealing to more familiar terrain (despite the fact that, for most of us, the personal computer is a far more familiar presence than a fortified castle). The confusion that surrounds our current understanding of how computer defenses fit together stems both from the imperfect analogies imposed by each of these older realms, and from the gradual conflation of multiple such metaphors in ways that give rise to our current inconsistent and incoherent computer defense classification schemes and catalogs. The roots of these catalogs—of our empty notions of best practice and defense in depth—run as far back as the third century, so grasping how we approach the newest security challenges of the twenty-first century requires turning back the clock accordingly.

Two distinct definitions of defense in depth arise in discussions of military history and nuclear safety, each with different implications for what defensive depth might mean for a computer system and what its primary purpose should be. A third meaning derived from the design of medieval castles, apparently invented or adopted by computer scientists purely for its metaphorical power, has come to dominate computer security discussions of defense in depth, further complicating the picture. These analogies are limited in their ability to inform defensive strategies for computer systems at a more than superficial level, and attempts to translate ideas from the domain of military strategy or nuclear safety for computer security have yielded some ambiguous and unhelpful results in the form of security control catalogs with no coherent structure or organizational principles. In practice, elements of these different existing definitions are often muddled and conflated in discussions of computer system defense in depth, leaving defenders with an unclear and inconsistently defined principle to guide their security strategies. Examining the historical origins of the term defense in depth and its different meanings in other contexts, as well as the ways those meanings have been applied to and appropriated by computer security, provides a window into understanding the inconsistencies and shortcomings of existing defense catalogs.

2.1 Defense in Depth in Military Strategy

The phrase "defense in depth" is used to describe military strategy as early as the 1950s, in an article by Beeler (1956) on the ways castles were used for defensive purposes in medieval England. Beeler argues that castles were located at strategic points within the kingdom to help fight off invaders, and points out that almost half of the twenty-one Norman castles in the English county of Kent were clustered together along the path that would serve as the "natural approach to London for any invader landing on the coast of Kent or Sussex" so that invaders approaching the capital would find themselves "in the midst of a veritable defense in depth" (1956, p. 595). Similar arrangements of castles in Herfordshire and Essex were intended to protect against invasions along the two primary northern routes to London, as well as the roads connecting them, Beeler notes, explaining:

> A hostile force advancing on London from the north would be on the horns of a dilemma upon encountering this network. If the advance were continued, a dozen garrisoned castles had to be left in the rear. If it were decided to reduce the obstacles, the enemy would be faced with the prospect of interminable siege work, during which, if his own army did not melt away, a relieving force could be collected from garrisons not threatened. It was essentially the same strategic device which can be seen in the location of the Kentish castles—defense in depth at a distance from London. (1956, p. 596)

Beeler does not explicitly define defense in depth, but he hints at a meaning centered on the strategic location of fortresses so as to surround invading parties and prevent attackers from advancing towards the heart of the kingdom and the seat of the reigning ruler.

In 1976, twenty years after Beeler applied the term to the military practices of William the Conqueror in the tenth century, Luttwak published his seminal work on defense in depth, using the term to describe the military strategy adopted by the Roman Empire during the third century. Luttwak argues that from the first century to the third, the Roman Empire shifts from using a "forward defense" strategy, focused on keeping enemies from entering their territory, to a system of defense in depth, which emphasized fighting off enemies after they had already crossed Roman borders. Luttwak offers a much more detailed and extensive explanation of defense in depth than Beeler, but the underlying principles are notably similar. "Forward defense demands that [the enemy] be intercepted in advance of the frontier so that peaceful life may continue unimpaired within," Luttwak explains. By contrast, a defense in depth strategy "provides for [the enemy's] interception only inside imperial territory, his ravages being meanwhile contained by the point defenses of forts, towns, cities, and even individual farmhouses." In other words, Luttwak's notion of defense in depth is one in which defenders cede the perimeter and focus on battling back intruders on their own land. The "depth" he refers to is essentially geographic in nature—attackers are permitted to move deeper into the defender's territory and defenses are relocated to this geographic depth as well, rather than being concentrated on the border.

Luttwak further distinguishes between elastic defense and defense in depth, writing that an elastic defense "entails the complete abandonment of the perimeter with its fortifications" so that defense relies "exclusively on mobile forces, which should be at least as mobile as those of the offense" (1976, p. 130). This strategy has both advantages and disadvantages for the defender, Luttwak argues, because while the defense can be as concentrated in a single area as the offense, it also "sacrifices all the tactical advantages normally inherent in its role (except knowledge of the terrain) since neither side can choose its ground, let alone fortify it in advance" (1976, p. 131). While an elastic defense relies only on mobile defense forces, Luttwak defines defense in depth as specifically involving a combination of mobile defense forces deployed around self-contained strongholds (the forts, towns, cities, and farmhouses).

This combination of soldiers and fortresses is the central characteristic of Luttwak's defense in depth—and it is consistent with Beeler's use of the term in his analysis of the role of medieval castles as "self-contained strongholds" or "point defenses" that helped the English kingdom stave off enemies who had already reached their shores. The adoption of this combination of mobile and stationary internal defenses meant that war was "no longer a symmetrical contest between structurally similar forces," Luttwak argues. Defense in depth put the attackers and defenders on less equal footing than elastic defense because it left the attackers with greater flexibility to move their forces wherever they wished, but gave defenders some support for their mobile troops in the form of self-contained strongholds.

Luttwak identifies three criteria for successful defense in depth. First, the strongholds must be sufficiently resilient to withstand attacks even without the assistance of the mobile forces; second, the mobile forces must similarly be able to withstand attacks without the shelter of the strongholds; and third, taking the strongholds must be essential to the victory of the attackers. If all three of these conditions are met, then "sooner or later, the offense will be faced by the superior strength of both fixed and mobile elements acting in combination," and the defenders will triumph. The first of these criteria led to the development of more sophisticated and effective defenses for castles themselves, since these often served as the defensive strongholds.

Later, when the notion of defense in depth began to be applied to computer systems and information assurance, these defenses for individual castles, which were serving as part of a larger defense in depth strategy, were appropriated by some in the computer security field as the essential core of defense in depth. In fact, these castle fortifications were a byproduct of defense in depth strategies but not, themselves, defense in depth as Beeler and Luttwak defined it. In this interpretation of defense in depth, the castles are transformed from one element of a larger defensive approach designed to protect the capital into the object of defense themselves—a reframing that has implications for whether or not defending the castle is just one means of constraining attackers' access (in this case, access to the capital city) or the sole mission of the defense, and correspondingly, whether breaching the castle defenses is a means to an end for attackers, or is itself their end goal. With this subtle shift, the meaning of defense in depth is muddled and the conflation of access and harm defense begins to take hold.

Also important for understanding where notions of military and computer defense in depth diverge is the rationale Luttwak offers for why an army would adopt a defense in depth strategy, namely, cost. Luttwak writes of the Empire's motivation for shifting from forward defense to defense in depth that the former was "obviously superior" but "impossibly costly to maintain." Since attackers could concentrate their forces at any point on the perimeter, defenders employing a forward defense would have to have sufficient troops stationed at every point on their border to hold off the full force of their enemy, and this would presumably require a vast number of troops, particularly when faced with protecting a large border against powerful enemies. By contrast, defense in depth allowed defenders to do more with fewer resources, because they could concentrate their mobile forces wherever the enemy itself was concentrated, rather than trying to cover the entire border. An important assumption here is that, given enough soldiers and resources (and a sufficiently weak enemy), an effective forward defense is possible—just expensive. In this context, defense in depth is not the best or most effective defense strategy, it is simply the cheapest one. In the context of computer security, the idea of defense in depth is usually invoked to encourage spending more money on defense, since it often means little beyond "more defense," and it is therefore motivated not by cost-savings but rather the absence of an effective alternative or single line of defense—there is no "obviously superior" forward defense strategy for computer systems.

The tension between forward defense and defense in depth in Luttwak's analysis also gives rise to a trade-off between the two strategies that has no real analogy when it comes to computer security. For the Imperial Roman Army, defense in depth, in some sense, precluded a strong forward defense because reallocating troops away from the perimeter to the interior mobile defense forces meant that attackers "could no longer be prevented by interception on the frontier line itself, for its garrisons were thinned out" (1976, p. 132). This type of defense in depth does not serve to reinforce a strong boundary defense, but rather replaces—or at the very least, weakens—it.

The crucial advantage of defense in depth for the defender, therefore, lay not in adding more layers of protection to help support the existing defenses, but rather in reconfiguring where those defenses were deployed and, simultaneously, redefining what constituted a successful attack. "Meeting only static guardposts and weak patrol forces on the frontier, the enemy could frequently cross the line virtually unopposed, but in the context of defense in depth, this no longer meant that the defense system had been 'turned' and overrun," Luttwak explains. "Instead, the enemy would find itself in a peripheral combat zone of varying depth" (1976, p. 132). This, too, is reminiscent of Beeler's analysis of the castles that surrounded the pathways to London. An enemy that landed on the north or south shore of England could not successfully conquer the kingdom unless it was able to reach London with sufficient forces to defeat the rulers in the capital—and that pathway to London could, itself, be part of the defensive strategy to wear down invaders before they accomplished their ultimate goal. So defense in depth as it is understood in the military strategy literature is not about adding more defense to a kingdom, but rather about reconfiguring and diversifying defensive resources to adjust the terms of victory and defeat.

The notion of defense in depth is invoked much less frequently in military history than in discussions of computer security, perhaps in part because it has this much more specific—and consistent—meaning for historians. Undoubtedly, some parallels can be drawn between Luttwak's definition and the defense of computer systems, particularly his emphasis on ceding the perimeter and anticipating that at least some attackers will be powerful enough to breach preliminary lines of defense. Computer system defense techniques similarly include an array of detection tools aimed at identifying attacks in progress, after intruders have already breached the system's "boundary" but before they have achieved their final goals. Indeed, proponents of "prevention-detection-response" computer security strategies often defend their framework by arguing that tactics to detect and respond to intrusions are needed to complement prevention measures, which are ultimately bound to fail in some cases (Bejtlich, 2013). This line of reasoning has something in common with the military strategy of allowing enemies to encroach slowly on protected land, aiming to cut them off en route the capital.

However, in other ways, the military history definition seems to have little bearing on computer systems. For instance, the combination of mobile forces and self-contained strongholds that is so central to Luttwak's definition has no obvious counterpart in computer defense. Perhaps more importantly, detection and late-stage mitigations are not, as a general rule, implemented in place of perimeter defenses for computer systems, but rather in addition to those "forward" defense elements. Accordingly, implementing internal defenses to detect and respond to intruders is generally not a cheaper defense strategy in the context of computer security because it is often layered on top of defenses meant to prevent access, instead of serving to replace or weaken those lines of defense.

This gets at a more fundamental difference between the computer security and military history notions of defense in depth. Luttwak's analysis assumes that a defender with limitless resources would elect a forward defense strategy since it offers greater benefits to society, but forward defense is simply not a viable option when it comes to protecting computer networks. There is no virtual equivalent of surrounding a kingdom with an army so large and so strong that it can hold off all invasions—indeed, the notions of "large" and "strong" are not even clearly applicable to computer defenses. When Bejtlich (2013, p.5) writes of computer systems that "determined adversaries will inevitably breach your defenses" he is not arguing for later-stage detection measures as a means of cutting costs or a replacement for early-stage prevention defenses, but rather as an additional type of defense to be implemented even—perhaps especially—by defenders with the greatest resources. Effective forward defense is basically impossible in this context, and perimeter and internal defense are qualitatively different, so strengthening one doesn't automatically weaken the other.

The crucial insight of the strategic military notion of defense in depth as applied to computer systems lies in its emphasis on blocking attackers' essential access pathways—the idea that varied and complementary defensive resources should be positioned along the key routes to attackers' end goals, and that each successful step closer to those goals should be treated not as a defeat by defenders but rather as a foray deeper into the "peripheral combat zone." This is the foundation of access defense—the principle of trying to constrain attackers' ability to acquire the necessary capabilities to inflict harm. For the Kings of England and Emperors of Rome this meant constraining the ability of adversaries' armies to reach the capitals where they ruled; for the defenders of computer systems it means tackling a more varied set of access capabilities and less literal routes to harm, often without the benefit of the far-reaching power and perspective of their historical counterparts.

2.2 Defense in Depth for Nuclear Security

If military history provides a foundation for understanding defensive strategies aimed at blocking access pathways and capabilities, the field of nuclear safety offers a contrasting picture of defense in depth centered on harm mitigation. Nuclear safety specialists, like military historians (and unlike computer scientists), have a consistent and clearly defined concept of defense in depth, but while military historians describe a notion of defense in depth based around a geographic notion of depth, nuclear regulators define the term along a temporal dimension. According to the International Nuclear Safety Advisory Group's (INSAG) 1996 report on "Defence in Depth in Nuclear Safety," defense in depth of nuclear plants centers on two key aims: "first, to prevent accidents and, second, if prevention fails, to limit their potential consequences and prevent any evolution to more serious conditions." More specifically, nuclear plant defense in depth involves five successive levels of defense, described in Table 2.1, structured in such a way that "should one level fail, the subsequent level comes into play" (*Defence in Depth in Nuclear Safety, INSAG-10*, 1996).

The defining characteristic of each layer of "depth" in the nuclear safety model is the point at which a defense interferes with an incident. Defense in depth in this context does mean layering on more defenses, but more than that, it means having defenses that operate at several different stages along the path from a properly functioning nuclear plant to an all-out crisis. Unlike Luttwak's military defense in depth strategy, the nuclear safety version does not preclude other defensive approaches so much as it unifies them into a single framework. That is, the defenses that can be implemented at level 5 of the nuclear framework (e.g., emergency response teams) may be substantively different from those that operate at level 1 or 2, so adding more defenses at one level does not diminish those at others in the manner that increasing mobile troops automatically depletes the stationary troops defending the border. Still, both notions of defense in depth focus on the question of how defensive resources should be allocated. Those resources may take more different forms in the nuclear industry than they did in the Roman Empire, but just as Luttwak's notion of defense in depth entails shifting resources from the forward defense along the perimeter to the internal mobile forces, so, too, the nuclear defense in depth framework emphasizes investing resources in a particular class of defenses—those operating at level 1.

The INSAG report notes that "accident prevention is the first priority" of defense

Table 2.1: Levels of defense in depth for nuclear plant safety. Source: *Defence in Depth in Nuclear Safety, INSAG-10* (1996).

Defense level	**Objective**	**Essential means**
Level 1	Prevention of abnormal operation and failures	Conservative design and high quality in construction and operation
Level 2	Control of abnormal operation and detection of failures	Control, limiting and protection systems and other surveillance features
Level 3	Control of accidents within the design basis	Engineered safety features and accident procedures
Level 4	Control of severe plant conditions, including prevention of accident progression and mitigation of the consequences of severe accidents	Complementary measures and accident management
Level 5	Mitigation of radiological consequences of significant releases of radioactive materials	Off-site emergency response

in depth because:

> Provisions to prevent deviations of the plant state from well known operating conditions are generally more effective and more predictable than measures aimed at mitigation of the consequences of such a departure, because the plant's performance generally deteriorates when the status of the plant or a component departs from normal conditions.

In some ways, this idea has more in common with Luttwak's notion of forward defense, which involved relying heavily on the first line of defense at the border, than his conception of defense in depth, in which defenders conceded the perimeter would be breached and reorganized their forces accordingly. In this regard, the nuclear and military history conceptions of defense in depth are actually somewhat at odds with each other. The former prioritizes preventative measures and therefore encourages investment in the first line of defense, while the latter prioritizes concentration of defensive forces and in doing so weakens the first line of defense, or the defenses focused on early-stage prevention of an attack or incident.

These differences may stem in part from the contrast between safety- and security-oriented defense. Early-stage prevention measures, such as the nuclear "level one" defenses or the military forward defense, may be less effective at stopping malicious actors intentionally trying to circumvent or evade those defenses than they are at addressing accidental incidents or natural disasters with more predictable patterns (of course, the same may also be true of reactive defenses). Another possibility is that the preventative measures for military attacks on the Roman Empire were simply more prohibitively expensive to the Empire than nuclear safeguards are to those responsible for implementing them. Luttwak, after all, acknowledges that the downfall of forward defense is that it is "impossibly costly"—the implication being that the Romans, like the nuclear scientists, might have preferred to prioritize early-stage or perimeter defenses but simply could not afford to do so. The focus of nuclear safety conceptions of defense in depth is not on cost-savings or access—after all, access capabilities are less meaningful in the absence of actual adversaries who pursue them—but rather on harm mitigation. The five-level model of defense in depth centers on the defenders' ultimate fear (harmful release of radioactive materials) and works backward from there to devise consecutive safeguards protecting against that outcome. Unlike the military conception in which defenses are aimed at blocking adversaries' steady progress towards acquiring useful terrain, the nuclear notion of defense in depth is not interested in the sources of danger or paths they take to nuclear plants, only in how to minimize the impact of that danger once it arrives.

Both the nuclear safety and military strategy notions of defense in depth deal with combining different classes of defense, but those classes relate to each other in very different ways. The military model involves combining two classes of defense—mobile forces and self-contained strongholds—that serve to actively reinforce each other, or provide "mutual support," operating simultaneously. The five nuclear safety classes (or levels) of defense, by contrast, do not function concurrently; rather, each is triggered only when the previous one has failed. This ties back to the different kinds of depth—or dimensions of defensive interaction—involved in the two models. For military defenders, geographic defensive depth is intended to help slow the physical advance of an attacking army and its access to a protected capital; for nuclear incidents, temporal depth is aimed at stemming escalating problems in nuclear plants. The former focuses on blocking, or obstructing, attackers' access pathways to their intended target (e.g., London, or Rome, as the case may be), while the latter emphasizes the need to limit harmful consequences of an incident. Both models have some important implications for computer system defense, but rather than recognizing these as two distinct frames for defense, computer scientists have tended to conflate and confuse elements of both—and further compounded that confusion by inventing their own pseudo-historical touchstone for discussions of defense in depth: the castle.

2.3 Information Assurance Through Defense in Depth

While military historians and nuclear safety experts have defined accepted, field-specific notions of defense in depth, computer security specialists have instead seized on the term and invoked it so often, so inconsistently, and in such vague terms that it generally amounts to little more than a reassuring label for "lots of defense" or "more than one defense." This confusion is not just a matter of linguistic laziness or sloppy vocabulary—it has profound implications for the ways we classify and categorize computer system defenses and contributes to the challenges of understanding those diverse security controls in an organized and comprehensive manner. Perhaps the one thing that computer scientists seem to agree on when invoking defense in depth is that it has something to do with castles—but not in any way that would have looked familiar to Luttwak or Beeler.

It would be difficult to overstate the role of medieval castles in shaping computer security discussions of defense in depth. Because this analogy does not rely on any actual, defined historical notion of defense in depth, it has been applied and appropriated in a variety of different ways to the landscape of computer security, serving as a useful foil for just about any point or argument to be made about defense. For instance, Markowsky and Markowsky (2011) propose eight distinct lessons for cybersecurity drawn from castles and castle warfare:

- Start with a good overall plan for the castle and all other entities that must be defended.
- Elements of the defense must be active. A completely passive defense will not survive the challenges and repel attackers.
- The cyber castle must be adequately staffed.
- Use defense in depth and make sure that the inner defenses also support the outer defenses. Be sure to have the equivalent of drawbridges and removable planks. Identify points in the security topology that can be used to quickly isolate zones from the network and from from other zones.
- Make sure that the cyber castle has a solid foundation.
- Use every means possible to make the attacker's job more challenging.
- Know your attackers. It is important to get some idea of the sophistication of your primary attackers.
- Find a balance between security and service. Castle designers faced this problem and found many successful solutions.

Discussions of "cyber castles" aside, Kewley and Lowry (2001) summarize the guiding philosophy derived from the castle metaphor by many computer security practitioners:

> [T]he more layers of protection you add, the more secure a target will be. For example, you can lock the doors and windows, put a moat around the castle, put alligators in the moat and a guard dog in the yard, a fence around the moat, and an armed guard in the watchtower to create layers of physical security.

Even researchers who, like Kewley and Lowry, challenge the assumptions implicit in this model do so explicitly in reference to the castle, underlining its status as the cornerstone of common defense in depth vocabulary in computer science.

How did the medieval castle attain this exalted status among computer scientists as the pinnacle of defensive achievement, the much-vaunted technology from which all future thinking about defense in depth should be derived? It's an attitude that dates back at least as far as a 2000 U.S. military report on "Information Assurance Through Defense in Depth." The report, with a cover picture of a medieval castle and its page numbers printed on small turret icons, codified the variation on the military historians' notion of defense in depth that would become a recurring theme in discussions of computer security. Where military historians followed Luttwak's definition of defense in depth as a strategy in which defenders ceded the perimeter of their territory and focused on fighting off invaders with a mix of mobile forces and stationary strongholds inside their own kingdom, the 2000 information assurance report focused just on the role of those strongholds—the castles. The report states:

> The dynamically evolving defenses of the medieval castle offer a valuable analogy. In that violent age, castles offered secure bases for armed forces to control key terrain. In response to changing threats, they evolved from simple to complex and very strong fortifications, following two principles: (1) increase and strengthen the defensive barriers, and (2) provide means to fight back actively. Castles on strong foundations, often on higher ground, employed successive barriers such as water obstacles, ditches, rings of strong and high walls with overhangs, and towers. Improvements to the walls allowed defenders to engage the attacker, and multiple gates enabled local counterattacks and raids. A small force could hold out against a much larger adversary. Just as the castle protected critical resources, now we must defend our vital military information and actively fight back with appropriate responses. (2000, p. 1)

Note the departure from earlier notions of defense in depth by Beeler and Luttwak, in which castles were themselves a part of a larger defense in depth strategy intended to protect a kingdom, rather than the primary asset being defended. That the two principles of defense in depth identified by the report represent a departure from the ways in which the term is used in other fields is not, in itself, necessarily problematic, but given the reliance of the report on castle analogies, the failure of those analogies to illuminate any concrete guidance for defenders of computer systems is troubling.

The closest the 2000 report comes to defining defense in depth for computer security is as an "approach [that] integrates the capabilities of people, operations and technology to establish multi-layer, multi-dimension protection—like the defenses of a castle." This combination of people, operations, and technology as defensive components is at the crux of the report's notion of defense in depth, suggesting yet another potential dimension of depth—the type or mode of defensive mechanisms. Castle defense in depth, as understood by the report's authors, involved the combination of defenses provided by people (sentries, spies, and informants), as well as operational activities (sorties and raids), and technology (arrows, spears, swords, axes, clubs, pikes, flung rocks, and hot or burning liquids). Information defense in depth, according to the report, entails the combination of updated versions of these same resources to defend four major elements of the "information environment:" local computing environments or enclaves (end-systems, LANs, and relay systems), enclave boundaries, networks that link enclaves, and supporting infrastructures (defined as "organized capabilities to provide special support," such as cryptographic logistics). Defense in depth for computer security, as defined by the 2000 report, thus involves using three defensive modes (people, operations, and technology) to defend each of four elements of computer systems (enclaves, enclave boundaries, networks linking enclaves, and supporting infrastructures)—and that's not all. According to the report, defense in depth also means that these defenses should also be arranged sequentially, so that "an adversary who penetrates or breaks down a barrier . . . promptly encounter[s] another defense in depth barrier, and another, until the attack ends." And furthermore, to be defense in depth, the report contends, "no critical sector or avenue of approach into the sensitive domains of the information system should be uncontested or unprotected" and "the weaknesses of one safeguard mechanism should be balanced by the strengths of another."

In other words, there are at least six different definitions of computer system defense in depth tangled up in the Department of Defense information assurance report:

- Defense in depth involves increasing and strengthening defensive barriers as well as providing targets with the means to fight back actively;

- Defense in depth is when multiple different types of defensive mechanisms are deployed in concert (people, operations, technology);

- Defense in depth is when multiple different elements of computer systems are protected (enclaves, enclave boundaries, networks linking enclaves, and supporting infrastructures);

- Defense in depth is when several defenses are arranged to be encountered sequentially so that an attacker must overcome all of them in order to be successful;

- Defense in depth is when every means of attacking a computer system is protected against;

- Defense in depth is when the vulnerabilities of each defense are reinforced by other defenses with different vulnerabilities that cannot be exploited in the same manner.

These different ideas of defense in depth hold echoes of both the military strategy and nuclear safety definitions—for instance, defending "every means of attacking a computer system" relates to the strategic placement of strongholds to prevent attackers' approaching along key routes, and the idea of arranging several defenses "to be encountered sequentially" has much in common with the five levels of defense in depth for nuclear plants identified by INSAG. The 2000 military report mashes together these—and several other—ideas about defense in depth into a jumbled non-definition which evolved into the incoherent framework upon which later defense catalogs and classifications were based.

The six different defensive strategies laid out in the report are not mutually exclusive, and there may be value in considering all of the different types of depth alluded to by them—reactive measures, defensive mechanisms, computer system elements, sequential layering of defenses, attack pathways, and independent vulnerabilities—but they do constitute distinct, if partially overlapping, notions of what defense in depth means. Furthermore, each of these definitions is far from straightforward when applied to computer systems. Identifying what it means to "fight back," or the relevant different classes of defensive mechanisms, or different elements and access pathways for computer systems, presents considerable challenges, as does defining sequential steps of attackers, or sets of defenses with complementary strengths and weaknesses. The conflation of all of these different, ambiguously defined elements into computer system defense in depth yields a concept that is, at once, so stringent as to be unattainable (what defensive strategy could conceivably claim to cover all modes of defense, all forms of attack, all elements of a computer system, all stages of intrusion, and plug all weaknesses of every defense with corresponding strengths from others?) and so broad as to allow defenders to pull out of it whichever piece best suits their purposes.

This has been the enduring legacy of the information assurance report—a multitude of meanings for computer system defense in depth with no clear consensus around a specific definition or the relationship between all the different interpretations. It forms the shaky foundation of future discussions of defense in depth in the context of computer and information security, imbuing those conversations with a confusion that is reflected in several existing catalogs and taxonomies for computer system defenses which simultaneously categorize defenses along several different dimensions in parallel.

2.4 Computer Defense Catalogs & Taxonomies

The existing mappings of the landscape of defensive measures, developed to understand what tools are available and what boxes ought to be checked by a diligent information security team, yield surprisingly little consensus on the critical, concrete components of a defensive strategy. This suggests significant divergence in the ways authors of these standards and catalogs conceptualize and define the roles of these security measures in relation to each other and in the broader context of system-wide security. This divergence—the fact that different groups working on this problem cannot agree on the crucial categories of computer defense, that these groups are so dissatisfied with the existing categorizations that they keep developing new ones in hopes of hitting on something more useful, and that they then, inevitably, undertake the convoluted process of mapping their new framework onto the older, existing ones—hints at a larger problem underlying the division of defenses into a set of neatly demarcated buckets.

Computer security measures, like computer security incidents, cannot be meaningfully characterized along a single dimension. In the world of security incident research and reporting, there is general agreement that the category of "data breach" incidents is neither useful nor informative but must instead be looked at and understood across a whole host of related yet distinct dimensions, including who perpetrated the breach and how, what data they accessed and why they wanted it, and so on. Indeed, resources such as the CERT Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) Taxonomy of Operational Risk and the Verizon Vocabulary for Event Recording and Incident Sharing (VERIS) have been developed and adopted specifically to provide a common vocabulary for identifying the important elements of computer security risks and incidents, respectively. This notion of dimensionality is often missing from computer defense taxonomies, many of which either focus on categorizing along a single dimension, such as time, or conflate several different dimensions into confusing and overlapping categories presented as cohesive and consistent groupings. Tellingly, when such taxonomies, including NIST 800-53, ISO/IEC 15408, and the Twenty Critical Security Controls, are looked at side by side, their organizational structures bear little resemblance to each other and suggest no coherent framework, reflecting instead the confusing mess of conflated defensive properties outlined in the 2000 information assurance report.

2.4.1 NIST Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations

Though NIST 800-53 was originally drafted to help U.S. government agencies defend their computer systems, it is often referenced and used by other organizations as the definitive catalogue of information security strategies. The fourth revision, issued in April 2013, identifies 18 families of security controls, listed in Table 2.2. All of these families except for "Program Management" are taken directly from the Federal Information Processing Standards (FIPS) Publication 200, in which they are identified as the seventeen "minimal security requirements for federal information and information systems." But while they are all originally formulated as security "requirements," the NIST 800-53 families, like many other attempts to map the information security world, comprise a mix of security objectives, such as access control or media protection, and operational mechanisms, such as contingency planning and program management. The families also divide the controls along different axes, with some groupings (e.g., Media Protection and System and Communications Protection) determined by *what component* of the system is being protected, others (e.g., Incident Response and Audit and Accountability) organized according to *what stage* of an incident they relate to, and still others (e.g., Access Control and Identification and Authentication) defined by *how* they provide protection.

In the third version of NIST 800-53, the eighteen families were themselves further grouped into three broad classes—technical, operational, and management—as shown in Table 2.2. These classes—reminiscent, perhaps, of the people, process, and technology notion of defense in depth proposed by the 2000 military report—and the subsequent decision to exclude them from the fourth revision of the document hint at some further problems with the catalog's classification scheme. The three classes combine statements about who (or what) is executing a control with others about the primary focus, or purpose, of those controls. Technical controls are defined as safeguards implemented and executed primarily by the information system through

Table 2.2: The eighteen families of security controls identified in NIST 800-53 Revision 4, as well as their corresponding classes, as designated by Revision 3 of the same document.

Identifier	**Family**	**Class**
AC	Access Control	Technical
AT	Awareness and Training	Operational
AU	Audit and Accountability	Technical
CA	Security Assessment and Authorization	Management
CM	Configuration Management	Operational
CP	Contingency Planning	Operational
IA	Identification and Authentication	Technical
IR	Incident Response	Operational
MA	Maintenance	Operational
MP	Media Protection	Operational
PE	Physical and Environmental Protection	Operational
PL	Planning	Management
PS	Personnel Security	Operational
RA	Risk Assessment	Management
SA	System and Services Acquisition	Management
SC	System and Communications Protection	Technical
SI	System and Information Integrity	Operational
PM	Program Management	Management

mechanisms contained in the hardware, software, or firmware components of the system, while operational controls are those primarily implemented and executed by people, and management controls “focus on the management of risk and the management of information system security.” In other words, the technical and operational classes are defined by whether they are executed by people or machines, while the management class is defined according to the focus of the controls. These muddled categories are further confused by the fact that, according to the report, a control family is associated with a given class “based on the dominant characteristics of the controls in that family,” implying that there may not be a perfect, or one-to-one mapping of these families to the specified classes.

The fourth version of 800-53 acknowledges this confusion, removing the classes from the classification “because many security controls within the security control families . . . have various combinations of management, operational, and technical properties.” Still, the Fourth Revision suggests that “organizations may determine that the responsibility for system-specific controls they have placed in the management class belong to the information system owner, controls placed in the operational class belong to the Information System Security Officer (ISSO), and controls placed in the technical class belong to one or more system administrators.” Here, the focus is not on the capabilities of each class of control but rather on who is responsible for its implementation. NIST 800-53 aims to sort defenses according to too many different characteristics simultaneously—who is responsible for its implementation, what element of the system it protects, what stage of an attack it interferes with—without explicitly drawing out and disentangling each of these threads. In doing so, the catalog leaves readers uncertain how to reconcile all the different, conflicting organizing frameworks it is based on, and which to use to ground their thinking about defense.

2.4.2 ISO/IEC 15408: Common Criteria for Information Technology Security Evaluation

Defense taxonomies are often developed with slightly different agendas. For instance, the stated purpose of NIST 800-53 is to “provide guidelines for selecting and specifying security controls,” while the ISO/IEC Common Criteria (CC) is, by contrast, intended to provide a “common set of requirements for the security functionality . . . and for assurance measures” of IT products. In other words, NIST 800-53 is supposed to help organizations select defenses, while the CC are supposed to help with evaluating and certifying the security provided by a set of defenses. Thus, where NIST 800-53 defines families of controls, the CC identifies classes of security functionality and assurance requirements. (This distinction is muddled somewhat by the fact that seventeen of the eighteen NIST 800-53 families are named for the FIPS 200 “minimal security requirements.”)

The overarching security attributes identified by the CC are confidentiality, integrity, and availability, but the CC translates these high-level requirements into classes of security requirements that, in many cases, do not clearly correspond to one of those three, hinting at their inadequacy. The eleven classes of security functional re-

Table 2.3: The eleven classes of functional security requirements listed in the Common Criteria.

Identifier	**Class**
FAU	Security Audit
FCO	Communication
FCS	Cryptographic Support
FDP	User Data Protection
FIA	Identification and Authentication
FMT	Security Management
FPR	Privacy
FPT	Protection of Security Functionality
FRU	Resource Utilization
FTA	Access
FTP	Trusted Path/Channels

quirements identified by the CC for evaluating information technology security, shown in Table 2.3, are largely distinct from the eighteen NIST 800-53 families, though they exhibit many of the same problems, including a mix of declarative objectives (privacy, user data protection) with operational mechanisms (cryptographic support, security audit). Only one category, "Identification and Authentication," appears in both lists identically, along with a few other closely related pairs—"Security Audit" and "Audit and Accountability," for instance, as well as "Access" and "Access Control"—while others seem to have no clear counterpart, including "Incident Response" in NIST 800-53 and "User Data Protection" in the CC. This may be partly explained by the fact that the CC explicitly limits itself to addressing "IT countermeasures" (perhaps the rough equivalent of the NIST 800-53 technical class), declaring all "non-IT countermeasures (e.g., human security guards, procedures)" beyond its scope, while NIST 800-53 takes a more comprehensive view.

For a more in-depth understanding of the divergence between the encyclopedic security catalogues compiled by NIST and the ISO, one need go no further than the 22-page table in Appendix H of the fourth version of NIST 800-53 which maps the CC requirements onto the NIST 800-53 controls. The elaborate mapping exercise confirms that the two documents cover largely overlapping material and have similar aims in helping organizations define and achieve desirable security requirements for computer systems, but it also highlights the extent to which the two diverge when it comes to thinking through a framework for understanding how different defensive elements fit together. Many of the CC requirements map to several NIST 800-53 controls across multiple different families, some have no mapping, and the authors explicitly warn readers that "The table represents an informal correspondence between security requirements and security controls (i.e., the table is not intended to determine whether the ISO/IEC 15408 security requirements are fully, partially, or not satisfied by the associated security controls)." The problem of trying to articulate whether or not a security requirement or function is fully or partially met—and if so, how—is dodged entirely.

2.4.3 Twenty Critical Security Controls for Effective Cyber Defense

The list of Twenty Critical Security Controls (CSC), compiled by a consortium of more than 100 contributors from US government agencies, private industry, and research institutions, are intended to "strengthen the defensive posture of your organization's information security; reduce compromises, recovery efforts, and associated costs; and protect critical assets and infrastructure." Assembled to provide firms with practical advice and clear defensive guidance, the CSC, listed in Table 2.4, represent yet another way of slicing up and reshuffling the landscape of computer defenses and categorizing them along several different axes simultaneously.

Table 2.4: The twenty critical security controls for effective cyber defense identified in version 4.0 of the CSC.

Inventory of Authorized and Unauthorized Devices
Inventory of Authorized and Unauthorized Software
Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers
Continuous Vulnerability Assessment and Remediation
Malware Defenses
Application Software Security
Wireless Device Control
Data Recovery Capability
Security Skills Assessment and Appropriate Training to Fill Gaps
Secure Configurations for Network Devices such as Firewalls, Routers, and Switches
Limitation and Control of Network Ports, Protocols, and Services
Controlled Use of Administrative Privileges
Boundary Defense
Maintenance, Monitoring, and Analysis of Security Audit Logs
Controlled Access Based on the Need to Know
Account Monitoring and Control
Data Loss Prevention
Incident Response Capability
Secure Network Engineering
Penetration Tests and Red Team Exercises

Just as the NIST 800-53 and CC categories mix declarative objectives and operational mechanisms, the CSC categories comprise specifications ranging from where defenses should be implemented (e.g., at the boundary), to what they should protect against (e.g., malware), how they should be assessed (e.g., red team exercises), and what actions they should be used to prevent (e.g., data loss). Furthermore, when the CSC, like the CC, are mapped onto the language of NIST 800-53 (as shown in Table 2.5) many of the listed critical controls end up corresponding to multiple NIST 800-53 controls across several different families. Controls from some of the 800-53 families, such as Access Control and System and Communications Protection, turn out to span more than a third of the CSC categories, while other 800-53 families, including Planning, Maintenance, Physical and Environmental Protection, Personnel Security, do not map onto any of the CSC. Notably, all four of the 800-53 families which are not represented in the mapping to the CSC fall into either operational or management classes, while both of the two 800-53 families that are represented across the most CSC categories are classified as technical. This may indicate a greater emphasis on technical controls by the CSC authors, as compared to the NIST 800-53 authors, or may suggest some greater confusion or inconsistency as to how to meaningfully classify these technical controls. Most likely, as in the case of the CC, the authors of the CC view certain people-oriented elements of computer security as beyond the scope of their document, though the omission is not articulated as clearly as it is in the CC, nor is it entirely consistent with the CSC's inclusion of inventory, configuration, and maintenance tasks.

The same tensions underlie the frameworks defined by the CSC, the CC, and NIST 800-53: in each, the high-level categories used to organize the world of computer system defense suggest no logical structure, no consistent form, no clear concept of completeness or coherence. Categories like "boundary defense," "data loss prevention," "penetration tests," "wireless device control," and "secure configuration" are presented in parallel, as if it makes perfect sense, in a single taxonomy, to switch seamlessly back and forth between classifying defenses by what piece of the network they protect, what they aim to protect against, how they are tested, what type of devices they apply to, and whether or not they are properly configured. Again and again, the authors of these frameworks run into the same problem—that there are several moving parts, or important elements, or dimensions—to thinking through a strategy for defending a computer system, and over and over they seem to sidestep that challenge by jumbling all of these pieces together into a single set of labels that serves to bewilderingly abstract and fragment the seemingly straightforward role of individual defenses.

These taxonomies reinforce the underlying message of the 2000 defense in depth report: computer defense is messy; it can't be neatly divided into mutually exclusive or even complete covering categories. When we formulate high-level goals of defense they often have very little relation to the specific tools and techniques we use to achieve those goals, and when we formulate lists of specific tools and techniques they are often devoid of any coherent organizational framework or consistent categorization scheme. In practice, this means that there is no clear or systematic way to design defensive strategies composed of multiple elements or assess the cumulative effect of those elements from an attacker's perspective.

Table 2.5: Controls from NIST 800-53 mapped onto each of the Twenty Critical Security Controls.

Critical Security Control	**Corresponding NIST 800-53 Controls**
Inventory of Authorized and Unauthorized Devices	CM-8 (a, c, d, 2, 3, 4), PM-5, PM-6
Inventory of Authorized and Unauthorized Software	CM-1, CM-2 (2, 4, 5), CM-3, CM-5 (2, 7), CM-7 (1, 2), CM-8 (1, 2, 3, 4, 6), CM-9, PM-6, SA-6, SA-7
Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers	CM-1, CM-2 (1, 2), CM-3 (b, c, d, e, 2, 3), CM-5 (2), CM-6 (1, 2, 4), CM-7 (1), SA-1 (a), SA-4 (5), SI-7 (3), PM-6
Continuous Vulnerability Assessment and Remediation	RA-3 (a, b, c, d), RA-5 (a, b, 1, 2, 5, 6)
Malware Defenses	SC-18, SC-26, SI-3 (a, b, 1, 2, 5, 6)
Application Software Security	CM-7, RA-5 (a, 1), SA-3, SA-4 (3), SA-8, SI-3, SI-10
Wireless Device Control	AC-17, AC-18 (1, 2, 3, 4), SC-9 (1), SC-24, SI-4 (14, 15)
Data Recovery Capability	CP-9 (a, b, d, 1, 3), CP-10 (6)
Security Skills Assessment and Appropriate Training to Fill Gaps	AT-1, AT-2 (1), AT-3 (1)
Secure Configurations for Network Devices such as Firewalls, Routers, and Switches	AC-4 (7, 10, 11, 16), CM-1, CM-2 (1), CM-3 (2), CM-5 (1, 2, 5), CM-6 (4), CM-7 (1, 3), IA-2 (1, 6), IA-5, IA-8, RA-5, SC-7 (2, 4, 5, 6, 8, 11, 13, 14, 18), SC-9
Limitation and Control of Network Ports, Protocols, and Services	CM-6 (a, b, d, 2, 3), CM-7 (1), SC-7 (4, 5, 11, 12)
Controlled Use of Administrative Privileges	AC-6 (2, 5), AC-17 (3), AC-19, AU-2 (4)
Boundary Defense	AC-17 (1), AC-20, CA-3, IA-2 (1, 2), IA-8, RA-5, SC-7 (1, 2, 3, 8, 10, 11, 14), SC-18, SI-4 (c, 1, 4, 5, 11), PM-7
Maintenance, Monitoring, and Analysis of Security Audit Logs	AC-17 (1), AC-19, AU-2 (4), AU-3 (1,2), AU-4, AU-5, AU-6 (a, 1, 5), AU-8, AU-9 (1, 2), AU-12 (2), SI-4 (8)
Controlled Access Based on the Need to Know	AC-1, AC-2 (b, c), AC-3 (4), AC-4, AC-6, MP-3, RA-2 (a)
Account Monitoring and Control	AC-2 (e, f, g, h, j, 2, 3, 4, 5), AC-3
Data Loss Prevention	AC-4, MP-2 (2), MP-4 (1), SC-7 (6, 10), SC-9, SC-13, SC-28 (1), SI-4 (4, 11), PM-7
Incident Response Capability	IR-1, IR-2 (1), IR-4, IR-5, IR-6 (a), IR-8
Secure Network Engineering	IR-4 (2), SA-8, SC-7 (1, 13), SC-20, SC-21, SC-22, PM-7
Penetration Tests and Red Team Exercises	CA-2 (1, 2), CA-7 (1, 2), RA-3, RA-5 (4, 9), SA-12 (7)

2.4.4 High-Level Information Security Frameworks

Further complicating this picture is the fact that the classic pillars of information security—confidentiality, integrity and availability—do not produce a correspondingly clear map of defenses, as suggested by the CC's relative abandonment of them as organizing principles. While confidentiality, integrity and availability (CIA) remain desirable qualities of a secure computer system, we can't actually sort out defenses that address each of those components individually. Landwehr et al. (2012) note, "There is currently no theory about why these properties [confidentiality, integrity, and availability] are considered security properties. In addition, there is no standard way to decompose a given property into confidentiality, integrity, and availability components." Other attempts to define lists of security properties have typically built on CIA, which is used extensively in instructional, regulatory, and standards-setting documents to define the high-level goals of information security. For instance, ISO/IEC publication 7498-2 on security architecture for information processing systems lists the crucial elements of security as identification and authentication, access control, data integrity, data confidentiality, data availability, auditability, and non-repudiation. Parker (1998) also expands on the CIA triad, proposing a "Parkerian hexad," which includes utility ("usefulness of information for a purpose"), authenticity ("validity, conformance, and genuineness of information"), and possession ("the holding, control, and ability to use information"), in addition to the original confidentiality ("limited observation and disclosure of knowledge"), integrity ("completeness, wholeness, and readability of information and quality being unchanged from a previous state"), and availability ("usability of information for a purpose") criteria.

But these high-level frameworks offer relatively little guidance when it comes to organizing or understanding the more detailed catalogues of computer defenses assembled by NIST and others. For instance, consider a table included in the 2000 military defense in depth report, and recreated in Table 2.6, that summarizes the technology components of its people, process, and technology framework for defense in depth, mapping those technical defenses against a framework of five such security properties (availability, confidentiality, integrity, identification and authentication and non-repudiation), identified as "security services." Each of the five services maps to at least six of the sixteen listed defenses (some to as many as ten), and most of the individual defenses correspond to multiple services as well. Furthermore, the report explicitly notes that "Implementation of any combination of measures supporting a security service does NOT necessarily ensure the security service." Overall, the chart suggests the extent to which the five high-level services offer minimal insight into the specific functions of the different defenses—and those defenses offer equally little assistance in ascertaining what properties they have provided.

Beautement and Pym (2010) observe that, like the previously discussed defense taxonomies, many information security frameworks, including the Parkerian hexad and the ISO-IEC reference model, confusingly combine declarative objectives of information security, such as confidentiality or integrity, with operational mechanisms implemented to achieve those objectives, such as access control. They argue, "This situation is problematic not only from the conceptual point of view—because declara-

Table 2.6: Technology Summary Table from “Information Assurance Through Defense in Depth.” Source: Woodward (2000, p. 14).

	Availability	Confidentiality	Integrity	Identification and Authentication	Non-Repudiation
Cryptography		x	x	x	x
User Name/ID, Password, PIN, Token, Biometrics				x	x
Digital Signatures				x	x
Firewall				x	x
Intrusion Detection		x	x	x	x
Malicious Code/Virus Detection and Removal	x	x	x		
Vulnerability Checker	x	x	x	x	
Guard		x			
Proxy Server		x			
System Monitoring Tools	x		x		
Transmission Security	x	x	x		
Control of Compromising Emanations		x			
Anti-tamper	x	x	x	x	
Protected Distribution Systems	x	x	x		
Redundant/ Multiple Data Paths	x		x		
Backup	x		x		x

tive and operational concepts must be treated differently in order to understand how objectives are delivered (or not) by making (in)appropriate implementation choices—but also from the economic and management points of view—because we are concerned with how the objectives of information security measures trade off against one another." Categorization errors of this sort, in which a set of seemingly very different classes or types of defenses are presented as consistent, logical, and even complete taxonomies are endemic in discussions of information security at every level of specificity.

Just as the high-level frameworks for information security like the CIA triad cannot be easily translated into an operational understanding of how to implement security controls, so too, that operational understanding laid out in NIST and ISO catalogues cannot be easily mapped back to a coherent, consistent high-level framework. There is a fundamental disconnect between the literature describing high-level information security frameworks and low-level information security controls. Stolfo, Bellovin, and Evans (2011) summarize this problem, writing:

> In recent years, much research has provided a strong understanding of a particular vulnerability or the security issues involved in designing a given system. However, we've seen little success in collecting the knowledge from this research into a general, systematic framework. . . . One goal is to establish a common framework for classes of defenses, categorized according to the policies they can enforce and the classes of attacks those policies can thwart.

2.5 Definitions of Defense in Depth in Computer Security

Existing defense taxonomies and security frameworks echo many of the same inconsistencies and confused, contradicting notions as the 2000 "Information Assurance Through Defense in Depth" report, conflating too many different ways of thinking about defense into disorganized, amalgamated catalogs and categories. The six different definitions of defense in depth implied by the report branch into as many—if not more—different mental models for organizing defenses in other work on classifying defenses and understanding what defense in depth actually means in the context of computer security. As different people have adopted and extended certain elements of the convoluted notion of defense in depth laid out in the 2000 report, it has become clearer which of these definitions have the potential to offer the most meaningful and relevant contributions to the defenders of computer systems. Correspondingly, it has also become easier to identify which elements of the extended castle metaphor employed by the U.S. military are least applicable to computer systems and fail to offer any significant insights.

One key defensive principle cited in the report—providing the means to "fight back actively"—has evolved into the notion of "active defense" of computer security, largely separate from discussions of defense in depth. Active defense is, itself, not always clearly or consistently defined in this space and may be used to mean anything from reacting to new threats to attacking one's attackers. For instance, Lynn (2010) describes three "overlapping lines of defense" used by the U.S. military to protect defense and intelligence networks: one intended to provide "ordinary computer hygiene" by keeping security software and firewalls updated, another composed of "sensors which detect and map intrusions" and a third that "leverages government intelligence capabilities to provide highly specialized active defenses." Those active defenses "automatically deploy defenses to counter intrusions in real time," according to Lynn, who adds that "they work by placing scanning technology at the interface of military networks and the open Internet to detect and stop malicious code before it passes into military networks." While Lynn's version of "active" means responding to threats in real time, Kesan and Hayes (2011) characterize active defense rather differently as "offensive actions undertaken with the goal of neutralizing an immediate threat rather than retaliating." In their construction, active defense includes three types of technology: intrusion detection systems, technology to trace the source of an attack, and counterstrike capabilities that "involve some method of sending data back at the attacker to disrupt the attack."

While there may not be clear consensus on what active defense entails, or indeed what the equivalent of castle-age "fighting back" is in the context of computer systems, this debate is not central to defense in depth, which instead focuses on the first principle of medieval castle defense referenced in the 2000 report: increase and strengthen the defensive barriers. By itself, though, that principle makes for a poor definition since it offers no insight into how a defender should increase or strengthen those barriers.

The combination of people, operations, and technology all being used to further defensive purposes is central to the idea of defense in depth put forth by the 2000 military report, but this definition has received relatively little attention or adoption among other discussions of the concept. Perhaps the closest equivalent of the people-operations-technology classification appears in the technical, operational, and management classes used to divide up the control families in the (now obsolete) third version of the NIST 800-53. But the combination of these components does not, in itself, constitute defense in depth. If it did, the criteria for defense in depth would be very easy to meet, very vague, and barely more helpful than the formulation in which it involves only increasing and strengthening defensive barriers.

Another notion of defense in depth alluded to in the 2000 report centers on protecting all the different elements, or layers, of a computer system. The report identifies four such elements—enclaves, enclave boundaries, networks linking enclaves, and supporting infrastructures—but it is possible to imagine any number of other ways to subdivide a computer system into its constituent components. For example, P. Anderson (2001) identifies the perimeter, network, and hosts as the crucial elements of a computer system to be defended by a comprehensive defense in depth strategy. Others add to those three categories the computer applications and the data stored on systems as additional components requiring protection (Lyons, 2011), while Ware (1979) lists five groups of "leakage points" requiring protection: physical surroundings, hardware, software, communication links, and organizational (personnel and procedures). At the heart of these list-making endeavors lies one of the fundamental tensions in discussions of computer system defense in depth: the desire to use the phrase to mean "everything is protected" and the inability to articulate "everything" in a computer system, or even every level of abstraction at which defenses may be needed. Once again, this exercise may well be worthwhile for defenders—just as it may be useful to consider the role of different defensive roles played by people, operations, and technology—but it does not provide a clear or stable definition for defense in depth because it is impossible to enumerate a complete list of a computer system's components. If defense in depth entails defending all elements of a computer system it is unachievable—and meaningless.

Though the dominant metaphor of defense in depth discussions is undoubtedly the castle, the nuclear safety notion of defense in depth is also echoed in some of the castle-centric 2000 report, which notes that "constructing successive layers of defense will cause an adversary who penetrates or breaks down a barrier to promptly encounter another defense in depth barrier, and another, until the attack ends." This definition of defense in depth as a series of sequential barriers, or defenses, appears in other places, as well, for instance, with defense in depth for computer systems described as "a sequence of controls that will be serially encountered by a perpetrator who is working toward a specific target. . . . The violation of one control station should immediately alert a security administrator and result in reinforcement of the other controls" (Parker, 1998). But this notion of layering computer defenses along the dimension of time—as soon as an intruder gets past one, he comes up against another—assumes a fairly linear and static progression, in which defenders can always be certain they know exactly how a perpetrator will progress through the target system. This is an assumption drawn more from the world of physical security than computer security, however. "The concept of outer and inner rings of defense has no real meaning when attackers can choose the attack, and hence defense, sequence," argues Stytz (2004), noting that "attackers can strike at almost any application defensive measure first (making it the outer defensive layer) and any other measure last (thereby making it the innermost layer of defense)." The concluding report from a 2011 National Science Foundation workshop on defense in depth echoed these concerns, noting that defense in depth "implies a sense of direction, which may not apply in cyberspace. Layered defenses may be a more appropriate term given the temporal, spatial, and other dimensions of the operating environment."

There is a difference between defensive strategies designed to force attackers through a specific sequence of barriers and those that simply require them to go through several barriers—in any order—and those that attempt to address escalating damage with a sequence of defensive measures. The fact that computer systems make it difficult for defenders to force attackers to encounter a set of defenses in a particular order does not mean that it is impossible to implement multiple lines of defense, rather it means that those lines of defense should not rely on being triggered in sequence. Furthermore, the nuclear safety strategy of defense in depth centers on containing damage during an escalating incident, or mitigating the impacts of a breach, a strategy which may in some cases be applicable to computer security. For instance, if financial information is stolen from a computer system, it may still be possible for banks and credit card companies to mitigate the potential fraud and theft activities that that information can be used for—notice, however, that the sequential nature of the defenses relies on elements of the attack outside the scope of the targeted computer system itself. That is, within the context of a computer system it may be very difficult to force an attacker through a chain of defenses in a specific order, but if the ultimate objective of that attacker goes beyond the system itself—to acting on stolen information or using access to the system to affect the physical world in some way, for instance—those later stages of the attack may be more vulnerable to sequential barriers put in place by defenders to contain damage. But here, again, though there is some relevant insight for defenders, it is difficult to discern a clear definition for what constitutes computer system defense in depth. Certainly, the idea that an attacker should encounter multiple lines of defense is central to what defense in depth is, but taken by itself it offers little by way of concrete guidance.

Another definition of defense in depth suggested by the 2000 report is that "no critical sector or avenue of approach into the sensitive domains of the information system should be uncontested or unprotected." Similar to the second definition, in that it categorizes defenses based on whether all pieces of a computer system are protected, this meaning shifts more towards the attacker's perspective, emphasizing the means by which an attacker can approach, or access, a computer system instead of the system's individual components. The idea that defense in depth is when every means of attacking a computer system is protected against presents several challenges—most immediately the problem of enumerating all possible means of attack—and may be considered distinct from the idea that defense in depth involves building up several layers of defense against an individual type of attack. Does defense in depth mean defending against many different threats or building many different layers of defense against an individual threat? Or, as the information assurance report seems to imply, both? Kewley and Lowry (2001) define the former as "defense in breadth" (or "multiple mechanisms across multiple attack classes"), in contrast to defense in depth, which they define as "multiple mechanisms against a particular attack class." But many defenses block (or monitor, or mitigate) certain classes of action that do not correlate directly with individual or entire modes of attack and could instead constitute one component of multiple different types of attacks. Using the language of attacks to inform or organize the ways we talk about defenses therefore requires some careful consideration of the relationship between what specific action a defense constrains and the different contexts in which that action might be carried out for malicious purposes.

While it is difficult to define defense in depth as defense against every attack class since defenses don't necessarily correspond to individual attack classes, the 2000 report actually provides a more specific description focused on the protection of the "avenue[s] of approach into the sensitive domains of the information system." These access points, each of which may be used for multiple different types of attacks, offer one way of organizing defenses, related to the idea of "leakage points" except focused instead on how intruders might enter a system, rather than how information might leave it. But as with the definition of defense in depth that focuses on components of a computer system, this emphasis on access points, or avenues of approach, though useful for thinking through what types of defense a computer system may require, presents an insurmountable completeness problem. The impossibility of creating a comprehensive list of all possible ways to access a computer system renders this a weak basis for a general definition of defense in depth.

The 2000 U.S. military report never quite makes up its mind about what defense in depth is—or is not—in the context of information assurance. It is, at once, all combinations of multiple different types of defense, where types of defense are determined by any number of different classification schemes simultaneously. Each of those classification schemes has something to recommend it, and something to contribute to the considered construction of a multi-component defense strategy, but taken either individually or together they fail, at some very fundamental level, to provide a clear or meaningful definition of defense in depth on par with those found in military history or nuclear safety. One of the notable attributes of those definitions is that they can be achieved—that defense in depth in those fields is not merely synonymous with "lots of defense" or "defense that covers everything," but rather outlines a clear and articulable strategy that an army, or a nuclear plant, can either be said to be implementing or not.

It is hard to imagine making a comparable assessment as to whether an organization had successfully implemented a defensive strategy that left "no critical sector or avenue of approach into the sensitive domains of the information system ... uncontested or unprotected." Indeed, any definition of defense in depth that entails exhaustively listing every component, or avenue of approach, or mode of attack, or interface of a computer system is similarly doomed to encounter unresolvable completeness problems. For defense in depth to be meaningful, much less achievable, in the context of computer systems, it cannot rely on defenders being able to exhaustively catalogue everything that requires protecting, or protecting against.

Eliminating these definitions, that require exhaustive cataloging of computer system access pathways, makes it easier to hone in on one that makes more sense in this space. Of the six characterizations of defense in depth presented by the 2000 U.S. military report, only two are sufficiently specific and feasible as to suggest a principle that could actually be attainable for defenders, as described in Table 2.7. It is these two characterizations—one in which adversaries encounter multiple defenses as they attempt to penetrate a computer system, and another in which the weaknesses of individual defenses are countered by the strengths of others—that therefore present the most promising basis for a definition of defense in depth for computer systems.

The first of these, with its emphasis on "successive layers of defense," bears some relation to the nuclear safety notion of defense in depth in which defenses correspond to escalating damage, as well as discussions of cyber attack "kill chains" (Hutchins, Cloppert, & Armin, 2011) and stages of attacks (Skoudis & Liston, 2006). Where these analyses highlight the sequential nature of attack steps and corresponding defenses, however, the crucial point for defense in depth is not that adversaries must penetrate a series of defenses in a specific order but rather that they must, at some point, in some order, get past multiple lines of defense. As Stytz (2004, p. 72) points out, "different attack profiles can attack and defeat independent defenses in different sequences. As a result, the concept of outer and inner rings of defense has no real meaning when attackers can choose the attack, and hence defense, sequence. So, in the cyberworld, we gain little or no advantage by arraying defenses . . . pseudosequentially."

2.5.1 Perverse Effects in Defense

Simply ensuring that an attacker must get through many different defenses is not, in itself, defense in depth, however—in some cases, depending on the interactions between those defenses, more layers may even prove less difficult and time-consuming for an attacker to penetrate than fewer layers. "Traditional thinking lends itself to the philosophy that the more layers of protection you add, the more secure a target will be," Kewley and Lowry (2001) write of their experiments, in which red teams were, in some circumstances, able to get past four layers of defense faster than they were able to penetrate a more scaled-back defensive set-up. "In the cyber world, however, multiple layers of defense do not necessarily add together to create a higher level of assurance," they conclude, explaining that "individual layers may have dependencies on other layers that must be enforced, otherwise they can be exploited by the adversary." Too much defense, in other words, can be just as dangerous as too little.

Computer defenses can have perverse effects either due to unexpected interactions with people or with other technical defenses. Both types of perverse effects fit the general definition of unanticipated consequences proposed by Merton (1936) in which "consequences result from the interplay of the action [the new defense] and the objective situation, the conditions of action [the users and existing defenses]." It is worth distinguishing between the two classes for the purposes of discussing computer security controls, however, because the human conditions and the technical conditions of the environment a new defense is introduced to are subject to different degrees of external control and predictable behavior. Furthermore, separating the scenarios in which multiple technical defenses specifically interfere with each other may contribute to the as-yet murky understanding of how to construct effective multi-layer defense strategies (or, at least, how not to construct very ineffective ones).

The interactions between an individual defense mechanism and end-users can cause perverse effects in a variety of different ways due to both the behavior of non-malicious and malicious users. Non-malicious end-users who do not intend to subvert the utility of a newly implemented defense may, nonetheless, negate any potential benefits of a given defense by altering their behavior due to either a disproportionate perception of the increased security afforded them by the new control or the challenges associated with using the new defense mechanism properly. Mitnick and Simon (2002) argues, "A security code improperly used can be worse than none at all because it gives the illusion of security where it doesn't really exist." In other words, a new layer of defense may backfire simply by presenting an illusion of security so compelling as to make non-malicious users lower their guard.

Another possible avenue for perverse effects of defense is a change in user behavior due to usability challenges. For instance, Komanduri et al. (2011) find a correlation between the use of higher-entropy passwords and storage of those passwords (either written down or in a computer file). Thus, a password policy intended to make users'

Table 2.7: Definitions of defense in depth presented in 2000 U.S. military report on "Defense in Depth for Information Assurance."

Defense in depth definition	**Depends on**	**Specificity and feasibility**
Increasing and strengthening defensive barriers as well as providing targets with the means to fight back actively	Being able to characterize "strengthened barriers" and "fighting back" in the context of computer security	The vagueness of this definition renders it difficult to implement and assess
Multiple different types of defensive mechanisms (people, operations, technology) deployed in concert	Being able to categorize the different types of defensive mechanisms and their interactions with each other	It is unclear what is meant by "in concert" here; definition #6 offers a possible clarification
Multiple different elements of computer systems all being protected	Being able to list and protect the different elements of computer systems	This definition depends on an unattainable requirement and is therefore unfeasible
"No critical sector or avenue of approach into the sensitive domains of the information system" is unprotected	Being able to list all of the critical sectors and avenues of approach into computer systems	This definition depends on an impossible requirement and thereby renders defense in depth unattainable
Adversaries who penetrate one defense "promptly encounter another . . . and another, until the attack ends"	Being able to ensure attackers must pass through multiple defenses to achieve their ultimate goals	This definition depends on design elements that are both reasonably specific and feasible
The weaknesses of one safeguard mechanism are "balanced by the strengths of another"	Being able to characterize the weaknesses and strengths of defenses in a common framework	This definition is both specific and feasible but requires further analysis of defense weaknesses to build such a framework

accounts significantly more secure might backfire by causing more people to write down their difficult to guess and equally difficult to remember passwords on easily accessible post-it notes. The impact of user behavior on defense effectiveness is not limited to passwords; Wool (2004) notes that nearly 80 percent of corporate firewalls exhibited “gross mistakes” in their configuration.

The tendency of non-malicious users to alter their behavior in ways that counteract security controls is an unwitting, if not insignificant, source of perverse results. By contrast, malicious actors may be able to exploit new defensive technologies in much more intentional and direct ways. Geer (2004) notes that several security products, including firewalls, anti-spam software, and intrusion prevention and detection software “have been found to have potentially dangerous flaws that could let hackers gain control of systems, disable computers, or cause other problems.” He describes three categories of vulnerabilities in security products: vulnerabilities that give malicious actors exploitable information about a system, vulnerabilities that allow intruders to enter a system, and vulnerabilities that let successful intruders expand the access they’ve gained to system resources. These vulnerabilities are due to rushed production timelines, inadequate debugging, and increasing complexity, Geer argues, noting that as security firms add functionality to their products to “gain a competitive edge and meet the demands of users who assume that more features make their systems safer,” the resulting “feature bloat” may introduce more vulnerabilities since “a combination of functions might cause problems that individual ones would not.”

The increasing complexity of corporate defenses and combination of more security controls may also spur organizations to implement centralized defense management systems that allow for easier monitoring and management of a diverse set of defenses. Such mechanisms may facilitate the jobs of not only internal systems security workers but also external intruders looking for an access point to a protected system. By centralizing the management and configuration of all security controls under a single system, an organization may create “a new control surface for the adversary to exploit” (Kewley & Lowry, 2001) that can actually be used against the defenders. In other words, centralizing control of layers of diverse defenses can negate the security value of that diversity by creating a single, centralized point of failure for all of them.

While software vulnerabilities and centralized defense management systems can backfire by helping intruders figure out how to gain access to the systems they’re intended to protect, other forms of defense may provide additional information that attackers can exploit to their advantage. In some cases this information may simply be the existence of new layers of defense, indicating to the intruder that there is something worth protecting, and therefore worth stealing. This may not be relevant for intruders motivated by financial gain seeking to expend as few resources as possible in their pursuit of profitable information, since they may simply look for less well-protected data. On the other hand, well-funded espionage organizations may have precisely the opposite reaction to encountering strong defenses since they may have significant resources and their interest is in identifying and accessing the most useful and relevant intelligence information, rather than making money.

Malicious actors may also be able to take advantage of additional information provided by security mechanisms to perpetrate a completely different type of attack than the defense was intended to prevent. For instance, the Domain Name System Security Extensions (DNSSEC) implemented to prevent the use of forged or manipulated DNS data required the addition of a new type of DNS record, RRSIG records, which contain the digital signatures that can be used to verify that the provided DNS data has not been forged. These digital signatures were intended to protect users from cache poisoning attacks, but they have also raised concerns in its potential to worsen a different type of attack—DNS amplification attacks. By spoofing the source address of DNS queries, an attacker can direct the response to that query to a target server and flood it with the amplified response of the DNS records. While these amplification attacks existed before the implementation DNSSEC, the addition of the RRSIG records can increase the amplification factor even more, leading to higher volume attacks. It's uncertain to what extent DNSSEC actually exacerbates the problem of amplification attacks, though estimates of how much DNSSEC increases the amplification effect range from a factor of two (Kaminsky, 2011) to seven (Lindsay, 2012).

Individual technical defenses can create perverse effects in a number of ways, but it is rarer to see multiple defenses actually counteract each other and layers of defense serve to lessen the cumulative protection. One category of these negative reactions concerns defenses tagging other defenses as malicious and trying to disable them. Consider for instance the United States Computer Emergency Readiness Team (US-CERT) Security Tip (ST06-009) on Coordinating Virus and Spyware Defense which explicitly advises users not to install more than one anti-virus program because "in the process of scanning for viruses and spyware, anti-virus or anti-spyware software may misinterpret the virus definitions of other programs. Instead of recognizing them as definitions, the software may interpret the definitions as actual malicious code. Not only could this result in false positives for the presence of viruses or spyware, but the anti-virus or anti-spyware software may actually quarantine or delete the other software." Anti-virus programs' propensity to attack each other could stem from several factors, including the possibility that the manufacturers of these products are trying to squash their competition by removing competing products from machines, as well as US-CERT's assertion that the signature detection engines may trigger each other. Furthermore, many of the capabilities and characteristics of anti-virus and anti-malware software are also suggestive of malware—for instance, their ability to scan and delete other programs and the fact that they cannot be easily switched off—so it is not altogether surprising that they might be regarded with suspicion by other anti-malware programs.

Defense mechanisms can also interfering with each other's ability to function properly. For instance, an intrusion detection or prevention system responsible for monitoring a system's traffic for anomalies or intruders may be effectively foiled by using a strong encryption scheme on internal traffic. This relates to the findings of Kewley and Lowry (2001) that IPsec actually degrades the utility of firewalls in their red team experiments. They note, "The inclusion of IPsec in this case provided an avenue for the adversary to exploit and thus get through the boundary firewall." In other words, while both encryption and intrusion detection can serve a valuable defensive purpose individually, in combination they may actually aid intruders, who can take advantage of the encryption to hide from an intrusion detection or prevention mechanism.

Understanding the interactions between different types of defense—and between technical defenses and end-user behavior—is not just important for figuring out how best to strengthen computer system protections, therefore. It is also central to ensuring that defenses don't inadvertently serve to undermine a system's security. This makes it all the more critical to have some notion of how defenses are composed that goes beyond a vague invocation of defense in depth to get at the functions of these defenses in relation to each other and their broader environment.

2.5.2 Independent Defenses

If more is not necessarily better when it comes to defense of computer systems, it stands to reason that something is needed beyond just multiple, non-sequential layers of defense—something about how defenses should be combined to create defense in depth, something akin to the five nuclear safety levels but less dependent on a static, linear progression of damage. The 2000 report offers a starting point for this discussion in its characterization of defense in depth as a strategy in which "the weaknesses of one safeguard mechanism should be balanced by the strengths of another," and others have expressed versions of this same idea in discussing the ideal for computer system defense in depth. For instance, P. Anderson (2001) describes the aim of defense in depth thus: "If the tools or techniques fail at one layer, the safeguards implemented at the other layers will compensate to prevent system compromise." Schneier (2006) offers a very similar definition of defense in depth as "overlapping systems designed to provide security even if one of them fails."

This idea of trying to use defenses that reinforce each other's weaknesses predates the 2000 report. Tirenin and Faatz (1999) describe defense in depth as the layering of multiple defensive techniques "in such a manner that the weaknesses of some can be mitigated by the strengths of others." The authors argue that the key to this defense strategy is implementing "orthogonal" security controls, that is, controls with independent, or different, vulnerabilities. But it is not immediately apparent how to either characterize the full set of vulnerabilities associated with a specific control or determine whether that set is fully covered by the other layers. In other words, it is not easy to identify orthogonal controls.

Two 1999 reports from the National Research Council (NRC), one on "Realizing the Potential of C4I: Fundamental Challenges" and another on "Trust in Cyberspace," also reference this definition of defense in depth, albeit somewhat inconsistently. The former draws heavily on Luttwak's terminology, presenting defense in depth as and alternative to "perimeter defenses." It states, "A perimeter strategy is less expensive than an approach in which every system on a network is protected (a defense in depth strategy) because defensive efforts can be concentrated on just a few nodes (the gateways)." Interestingly, in this appropriation of Luttwak's language, the very rationale for abandoning a perimeter defense in favor of defense in depth—the cost—is reversed, since the authors argue that defense in depth is more expensive. The idea that defense in depth is a function of where defenses are placed (i.e., at gateways versus individual systems) is abandoned later in the report, however, when the term is defined as “a strategy that requires an adversary to penetrate multiple independently vulnerable obstacles to have access to all of his targets.” The report continues:

> The property of “independent vulnerabilities” is key; if the different mechanisms of defense share common-mode vulnerabilities (e.g., all use an operating system with easily exploited vulnerabilities), even multiple mechanisms of defense will be easily compromised. When the mechanisms are independently vulnerable and deployed, the number of accessible targets becomes a strong function of the effort expended by the attacker.

This latter definition aligns with the “Trust in Cyberspace” NRC report, which articulates the underlying principle of defense in depth as “one mechanism covers the flaws of another.” It, too, cautions readers “an attack that penetrates one mechanism had better not penetrate all of the others.”

If this then, is the core of defense in depth—an array of defenses such that an attacker must tackle and get past each one individually because they cannot be defeated with one single maneuver but must all be addressed to achieve some end goal—it presents two crucial questions: How do we ensure that an attacker must go through every one of a set of defenses? Or that multiple of those defenses cannot be defeated by a single attack? Neither presents entirely straightforward answers, but they do allow for a reasonably concrete—and, in some fashion, attainable—definition of defense in depth for computer systems. The clearest articulation of this definition is given by Schneider (2007), who writes of defense in depth:

> No single mechanism is likely to resist all attacks. So the prudent course is that system security depend on a collection of complementary mechanisms rather than trusting a single mechanism. By complementary, we mean that mechanisms in the collection
>
> - exhibit independence, so any attack that compromises one mechanism would be unlikely to compromise the others, and
> - overlap, so that attackers can succeed only by compromising multiple mechanisms in the collection.

Two of the characterizations of the 2000 report are clearly visible in the two pillars of this definition, but more importantly, several of the peripheral attributes attached to defense in depth at various points have been shed. Notions of defending every means of access or component of a computer system—though still important considerations for defenders—have been removed. Similarly, the idea that an attacker must pass through a set sequence of defenses in a given order is gone, as is Luttwak’s language in which defense in depth is presented as mutually exclusive with a perimeter defense. In their place, Schneider focuses on two characteristics—independence and overlap of defensive mechanisms—as the cornerstone of defense in depth. Even with this definition, however, important questions remain to be answered about the nature of defense in depth in computer systems, including, most crucially, what it means for an attack to be “unlikely” to compromise multiple defenses, or rather, how to characterize the independence exhibited by a set of defenses.

If independent defenses are defenses that cannot be simultaneously compromised, then no two defenses can ever be entirely independent. That is why, when defining defense in depth, it is necessary to use the rather unsatisfactory "unlikely" modifier—we cannot simply assert that any two mechanisms which can be compromised by the same attack are not independent because then we would have no independent defenses at all. In fact, in some sense, this is indeed the case. As Schneider puts it, "mechanisms deployed in the same system will necessarily have points of similarity" and therefore "an attack that exploits vulnerabilities present in a point of similarity could compromise both mechanisms." Any defenses for a computer system share, at some level, a common dependency—whether it's a dependency on the same organization, the same person, the same machine, the same operating system, the same application—and can therefore, at that level, be circumvented, or defeated, by an attacker who has gained control of that shared basis. Correspondingly, the broader—or more difficult to gain control of—their common dependency is, the more independent two defenses are.

In this context, independence cannot be characterized as binary or absolute. Rather, it implies a range of common dependencies that defensive mechanisms may share, each of which corresponds to the limitations of those mechanisms' independence. Defenses cannot therefore simply be independent, they can only be independent up to a certain point—the point of their common dependency. For instance, Schneider (2007) presents as an example of defense in depth the common ATM withdrawal system which requires both a physical bank card and a PIN. These two mechanisms satisfy his definition for independence because "the bank considers it unlikely that somebody who steals your bank card will deduce your PIN" (again, that nebulous notion of unlikeliness is key here), but, he adds, "both mechanisms have the card holder in common and there is a trivial attack that subverts both mechanisms: abduct the card holder and use coercion to get the bank card and learn the PIN." In other words, both the PIN and the bank card share a common dependency on an individual person. If abduction didn't seem sufficiently unlikely, it's possible to imagine trying to broaden that common sphere of dependency so that withdrawing money required the consent of a second person, and a successful attack would now require gaining control of twice as many people. Or perhaps instead of multiple people, an action would require the consent of multiple machines—and just as those people might share common attributes that would make them easier to compromise en masse (working in the same building, living in the same house), so, too, might machines (running the same operating system, connecting to the same network). At what point can it be said these different mechanisms are unlikely to be compromised by a single attack?

In many ways, this seems to bear little resemblance to the notions of defense in depth discussed earlier. Certainly, work on defense in depth in military history and nuclear safety does not explicitly reference ideas of independence or common dependency, but there is a common thread running through both of them related to establishing an array of defenses that are likely to bolster each other and require efforts to overcome. In the context of Luttwak's definition, there are only two such forms of defense: mobile troops and fixed fortresses. Recall that the combination of these two defensive forces is the defining feature of his formulation of defense in depth (as distinct from elastic defense, another alternative to the prohibitively expensive forward defense strategy, which features only mobile troops and no stationary strongholds). The central feature of Luttwak's defense in depth, in other words, is not that it moves defensive forces into a protected territory, away from the perimeter, but rather that it presents attackers with two distinct—independent, if you will—forms of defense which must be overcome: stationary castles and mobile forces. These two lines of defense can be considered independent not because there is no interaction between them (quite the opposite, in fact, as Luttwak points out, since the mobile troops may rely on strongholds for temporary respite, restocking supplies, and other functions) but because a concentrated attack in a single, fixed location to take control of a castle is unlikely to also defeat a mobile troop of soldiers with less impressive fortifications but much greater degree of influence over the time and location of battles.

Returning to the 2000 military report's notion of balancing the weaknesses and strengths of different safeguards, the mobile troops and fortresses of Luttwak's description can be said to balance each other in a similar sense. The strength of the troops is their mobility—they can concentrate wherever attackers strike and position themselves strategically depending on where and when they wish to fight. The strength of the castles, by contrast, is in their significant fortification—possible only because they are fixed, massive structures which cannot move to meet (or, indeed) to flee their opponents. That fortification balances the comparatively low level of protection afforded troops (e.g., armor, hand-held weapons), while those troops' ability to position themselves balances the fortresses' immobility and together, Luttwak, argues they present a much more effective strategy than either would on its own. Note also that Luttwak incorporates Schneider's notion of overlap into his discussion, with the third of his criteria for successful defense in depth—that taking the strongholds must be essential to the victory of the attackers. In other words, defense in depth does not work if the attackers need only get through one line of defense (the mobile troops) to meet their goal—the depth of defenses depends on having to pass through all of them. Undoubtedly, there are also significant differences between the ways defense in depth is applied to military strategy and computer security, including the trade-off with strong perimeter defenses in the military version and the related idea that defense in depth would be cheaper than a strong forward defense. Still, when it comes to defining defense in depth specifically, as distinct from elastic defense, the notions of independent lines of defense that could not be easily defeated by a single attack emerges are central to Luttwak's analysis.

In the nuclear safety version of defense in depth, too, there is an underlying notion of independence. Implicit in the five layers of defense for nuclear plants is the idea that a malfunction or accident that bypasses one will not compromise the next. That is, an incident which succeeds in breaching the design and construction safety features (level 1) is expected not to also overcome the independent protection and surveillance systems operating at level 2, much less the off-site emergency response teams at level 5. In some sense, independence is the crucial criteria for distinguishing between these different levels because they are defined by the notion that "should one level fail, the subsequent level comes into play." In other words, should one level be defeated or overcome in some manner, the next is likely to remain intact (otherwise it would be of little use as a follow-up mechanism). This idea that the different lines of defense in depth are determined by how likely they are to be simultaneously compromised mirrors Schneider's definition of independence. Here, again, there are points of departure—the strictly ordered, sequential set of protections is difficult to imagine emulating precisely in the context of a computer system where detection is often more difficult and damage may be less immediately obvious and linear, or predictable, in its progress. This difference impacts the issue of overlap as well, since the nuclear defense in depth relies on the detection of any malfunction to trigger each successive layer of protection. So long as each escalation of an incident is identified, an incident will, indeed, have to defeat all five of the layers to become a full-blown nuclear disaster, but the lines of defense do not operate independently of each other in the sense of all being active at all times, regardless of whether the others have been bypassed. Rather, they rely on the defeat of the earlier lines—and, accordingly, the ability to detect the defeat of the earlier lines—to initiate the later ones.

Assembling a combination of independent defenses—that is defenses which are not likely to be compromised or overcome by a single maneuver—is central to all of these definitions of defense in depth. Perhaps the primary difference between the military and nuclear notions relates directly to this question of sequential ordering of defenses and whether the arrayed independent defenses can be assaulted in any order an assailant chooses in his steady progression to the capital, or are instead defined by a clear progression of stages intended to limit damage as threats gain traction within the targeted system and grow increasingly dangerous. Both of these models can be applied to computer systems, the former to efforts to limit access to protected machines and information and the latter to mechanisms intended to limit the damage that can be perpetrated if that access is achieved. Access to computer systems is difficult to break down into set pathways because, as Stytz (2004) points out, attackers have so much flexibility in how they choose to approach these systems and can therefore often dictate which lines of defense they encounter first. In these cases, a defense in depth set-up modeled after the nuclear definition would provide little help because without knowing which defenses will be breached first it is impossible to rely on them to trigger the others. On the other hand, defenses that overlap in the manner of Luttwak's troops and castles—that is, defenses which coexist and operate simultaneously regardless of each other's status—can slow intruders regardless of their chosen points of entry so long as it is necessary to pass through all of them to achieve the desired degree of access.

Defenses intended to limit the ability of attackers to achieve their ultimate aims and inflict damage on their targets, particularly when those targets lie beyond the scope of the breached computer system, may correspond to more clearly sequential stages of an incident. Once an attacker has achieved some desired set of access capabilities in the context of a computer system and tries to exploit them for some malicious purpose outside that computer system—stealing money, or moving data, or disrupting physical services, for instance—it may be possible to pin down the attacker's next steps more precisely, and those steps may follow a more set sequence. Limiting access requires a more general notion of security for a computer system, it forces defenders to make assumptions about behaviors and capabilities which are not obviously or directly harmful but may be used for malicious purposes, and find ways of preventing them regardless of their intended aim. Limiting damage, on the other hand, requires a more specific sense of what attackers will ultimately be after should they seek to access a system, or what kind of damage they will try to inflict, and requires tailoring layers of defenses designed specifically to prevent perpetrators from achieving that aim, even after they've successfully gained access capabilities in the context of their targeted system.

Ultimately, what we can take away from the mess of historical and invented influences and muddled definitions and taxonomies of computer system defense in depth is this distinction between defenses aimed at restricting access and those designed to minimize harm, as well as the different types of independence and overlap that each of these classes of defense allows for. The language of defense in depth is by now an unsalvageable casualty of overuse and slip-shod appropriation in computer security—an odd exercise in casting back to the Middle Ages to justify principles of modern day technology, of imagining castles in the image of computers. Still, returning to some of the original ideas underlying that language—before they were melded, mangled, and misinterpreted—draws out defensive distinctions that allow us to start straightening up the current landscape.

Chapter 3

Access and Harm

Let us leave behind the Roman Empire and the Middle Ages and return to 2013, when MIT was trying to strengthen computer security on campus in the wake of a series of availability attacks on its infrastructure in January 2013, including a denial of service attack and a DNS-redirect for all sites in the mit.edu domain (Kao, 2013). MIT Executive Vice President and Treasurer Israel Ruiz announced several new cybersecurity measures the following April: password complexity requirements and expiration policies, a firewall that would by default block most inbound traffic originating from non-MIT IP addresses, a VPN requirement for accessing certain administrative applications from off campus, restricting recursive access to MIT DNS servers to clients on MIT's network, and a contract with content delivery network Akamai to protect against future denial-of-service attacks (Ruiz, 2013). The university later restricted access to MIT's non-guest wireless networks to authenticated users with MIT login credentials, as well. Most of these measures target access capabilities that, while not harmful in and of themselves, could serve as stepping stones for malicious activity. For instance, the password complexity requirement restricts attackers' ability to guess MIT user passwords, while the firewall and VPN access requirements similarly cut off capabilities that adversaries might take advantage of to compromise hosts or administrative applications. Restricting recursive access to MIT DNS servers serves the same function with regard to access capabilities that might aid denial-of-service attacks.

All of these are forms of access defense—security controls intended to make it more difficult for attackers to acquire the capabilities they need to exploit a computer system in some harmful fashion. This is the notion of defense rooted in the military strategy conception of defense in depth: slow the invaders' onslaught, anticipate the routes they will take in their approach and station diverse protections at several points along those paths to stave them off. But predicting the routes by which attackers will approach is a much messier business in the context of computer systems than kingdoms, and enemy advances much more difficult to distinguish from innocent ones. Accordingly, MIT's efforts at access defense appeared to be largely futile in the months following their implementation. As an increasing number of user passwords were made more complex, the number of compromised accounts reported to IS&T also rose, as shown in Figure 3-1. Similarly, the gradual implementation of MIT's

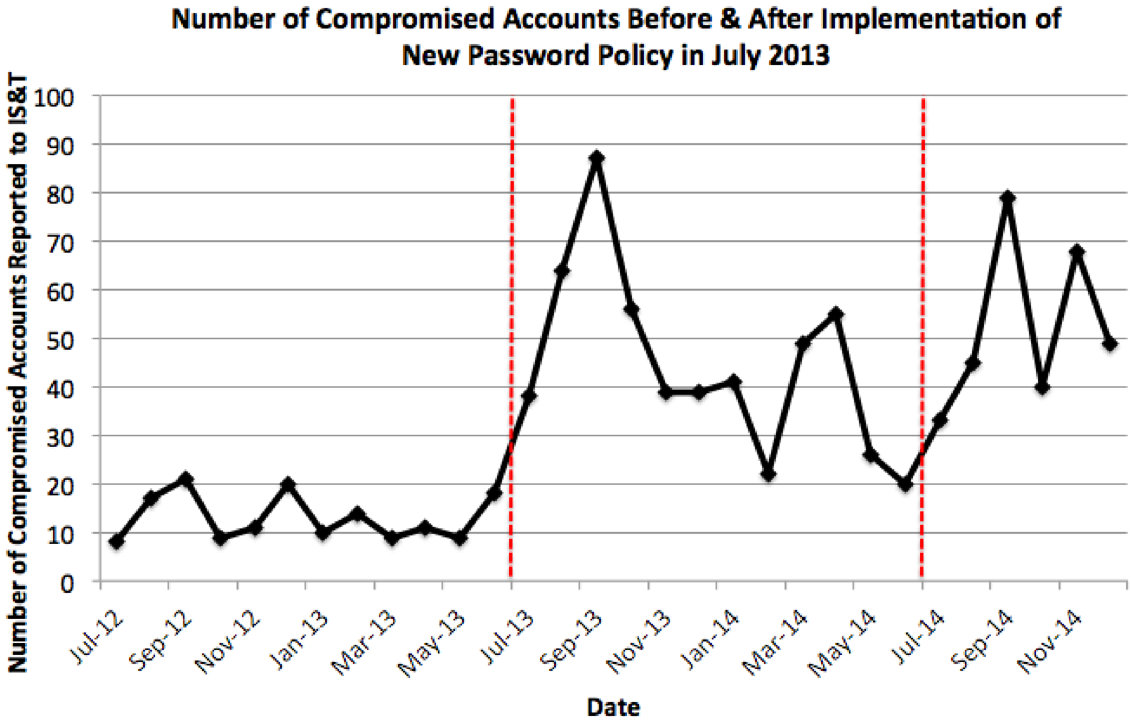


Figure 3-1: The number of compromised accounts reported to IS&T during the year leading up to the implementation of a new password policy in July 2013 and for the period during the implementation (July 1, 2013–July 1, 2014) and following its implementation, through December 2014.

firewall, building by building beginning in late 2013, showed little evidence of reducing the number of compromised hosts on campus, as shown in Figure 3-2.

This apparent failure of MIT's defensive measures to effectively address compromised accounts and hosts hints at the disconnect between the access-based defenses IS&T implemented and the harms that they actually wished to mitigate. IS&T had good reason to be concerned about compromised hosts and accounts on campus—as shown in Figure 3-3, both of these classes of incidents had been on the rise in the years leading up to the security changes—but the capabilities MIT cracked down on as proxies for, or pathways to, these incidents appeared to be inadequate surrogates. For instance, one path to compromising an MIT account is to guess a user's password through brute force—a capability that MIT rendered significantly more difficult to attain by requiring users to choose more complicated passwords. But, crucially, this is not the only access capability that enables attackers to compromise accounts. In fact, the increase in compromised accounts following the announcement of the new password policy was so significant that IS&T began asking people how they thought their accounts had been compromised during the summer and fall of 2013. The vast majority of people they asked appeared to have fallen prey to email phishing attacks (this also aligned with an increase in phishing emails reported to IS&T during the

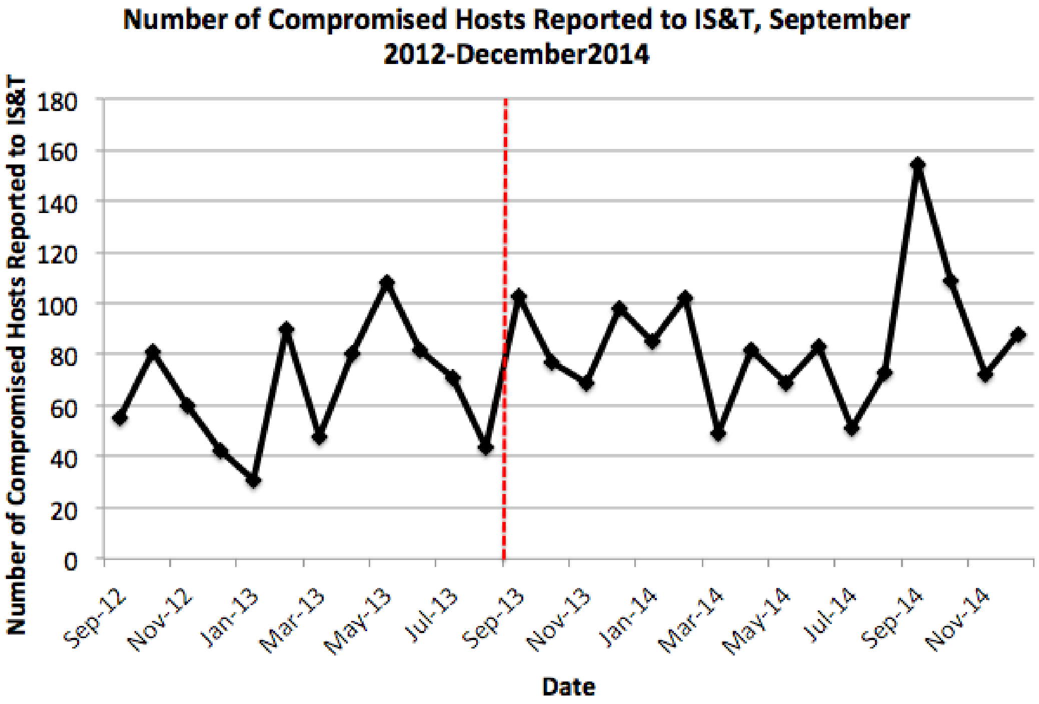


Figure 3-2: The number of compromised hosts reported to IS&T during the year leading up to the implementation of a new firewall in Fall 2013, as well as the following months during the gradual roll out across campus.

same period). Thus, the two access pathways addressed by the new password complexity requirements and expiration policy—brute-force attacks, and the use of old or shared passwords—were largely irrelevant to the capabilities actually being exploited by attackers at the time, namely, the capability to send emails impersonating IS&T with links to malicious password reset websites. In August 2013, one IS&T staff member even indicated, in response to a coworker's concerns about the "dramatic increase" in compromised accounts, that he wondered whether "our new policy—where we actually email you and tell you to change your password—is working against us here."

On the surface, MIT's struggle to implement effective defenses is a fairly routine story of well-intentioned security efforts that offer too little too late—their focus on yesterday's dictionary attacks rather than today's phishing threats (recall the sentiment expressed by several IS&T staff members that the new measures were a form of long overdue catching up, rather than a proactive stance to address current or emerging threats). Undoubtedly, the IS&T records suggest, MIT is defending against the wrong things—or, at any rate, failing to defend against the right things—but the Institute's inability to put a dent in the number of reported compromised hosts and accounts speaks to a much deeper problem than simply being behind the times. There is a disconnect between the access capabilities that MIT's defenses constrain (e.g., dictionary attacks, exploitation of open ports) and the actual campus

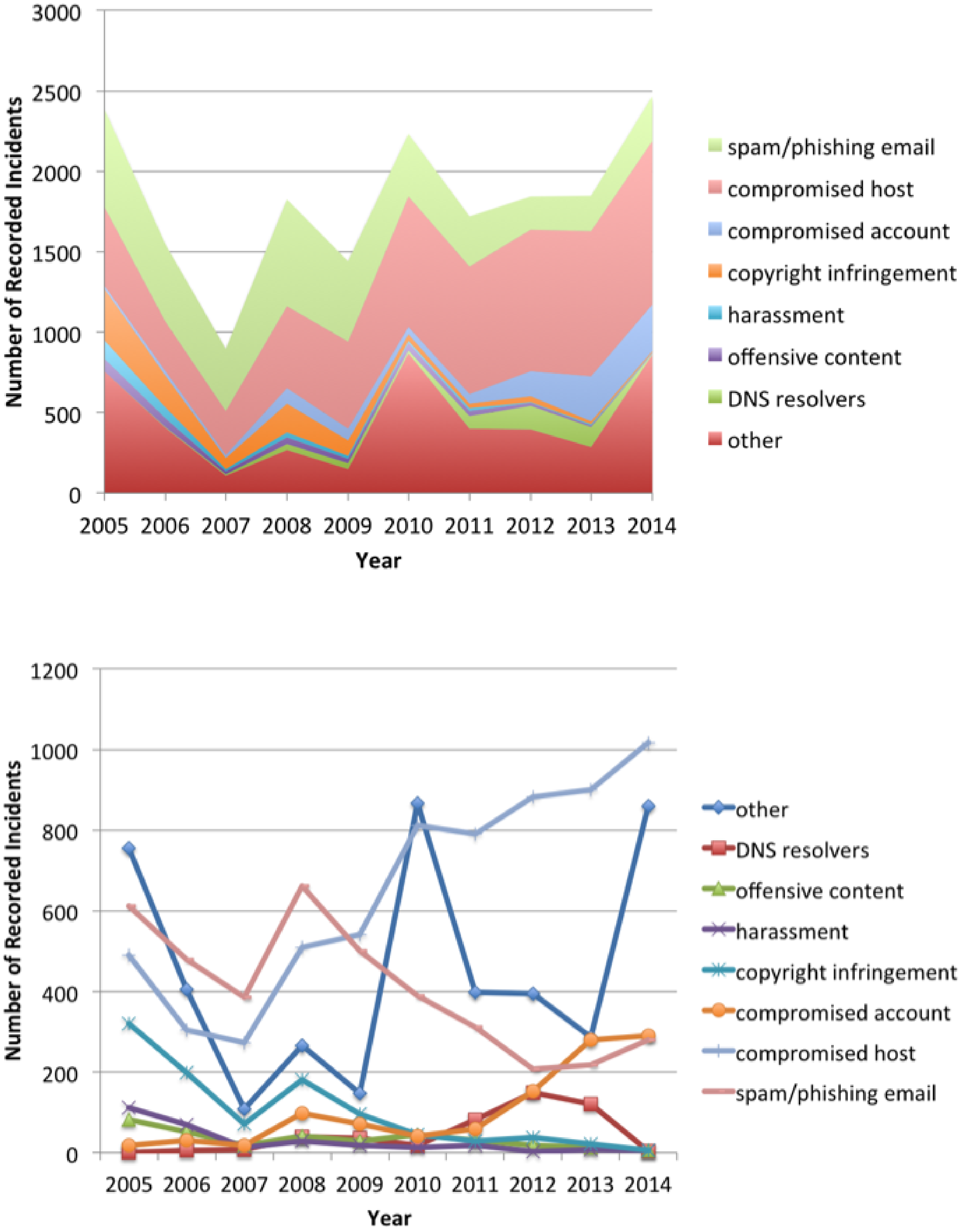


Figure 3-3: The number of different types of security incidents recorded by IS&T yearly from 2005 through 2014.

resources that threat actors exploit in harmful ways (e.g., compromised accounts, compromised hosts). Put another way, MIT is relying primarily on access defenses to do the work of harm mitigation. In fact, the only one of MIT's new suite of security measures directed at mitigating harm was the contract with Akamai intended to increase resilience of the university homepage and external DNS service. The one-year password expiration policy also served a secondary harm defense function by enabling IS&T to update the encryption schemes used to protect stored passwords—a task that can only be performed when a password is reset—and thereby mitigating to some extent the potential consequences of an IS&T data breach.

Taking steps to shield users from the consequences of a denial-of-service attack or data breach in this manner is a form of defense rooted in the principles of nuclear safety defense in depth: worry about the outcomes, not the access pathways by which they arrive. Such defenses target the ultimate harms attackers may wish to inflict—harms like the service disruption MIT experienced in January 2013 when users could not reach its homepage—rather than the intermediate harms, like compromised hosts and accounts, which serve as a means to a malicious end, but not an end in themselves. Distinctions between ultimate and intermediate harms are very context dependent: the ultimate aim of one attacker may be only an intermediate step for another—for instance, stealing stored data might be one person's ultimate goal, but stealing user passwords would more likely serve as a tactic for compromising accounts, circling back to the class of intermediate or micro-harms associated with an access capability but not themselves the central malicious intent of the attacker.

So, MIT is faced with defending against a set of access capabilities that may or may not be used for malicious purposes (e.g., email, login attempts), a set of intermediate micro-harms that are almost certainly being used for malicious purposes of some variety but offer little indication of what ultimate harm they serve (e.g., compromised hosts and accounts), and also a class of ultimate harms that motivate its adversaries (e.g., service disruption, financial fraud, espionage). And at each step of the way, the capabilities they select to defend against seem to serve as poor proxies for the intermediate harms they can monitor—and those intermediate harms may themselves not prove effective proxies for the broader harms that MIT has much less visibility into. Because of this limited visibility, MIT itself occupies a somewhat uncomfortable intermediate position in the context of many of the security incidents it witnesses: unable to determine the access vectors by which adversaries compromise hosts and accounts, except by surveying victims, and equally unable to determine the purposes for which those compromises are exploited, except for when victims lodge complaints. This is because much of the access work of compromise—whether it entails sending phishing emails and collecting credentials through malicious web forms, or using shared passwords stolen from other sites—happens outside the context of MIT's computing environment and beyond the scope of what IS&T can see or control.

Note that the one access capability IS&T tried to defend against to prevent account compromises—brute force log-in attempts—is one of very few such capabilities that relies explicitly on the use of their network and can therefore be monitored directly by IS&T. Similarly, much of the work of directly inflicting harm—whether it involves stealing money, stealing secrets, or launching denial-of-service attacks from large botnets—also occurs outside MIT's narrow window of visibility and scope of control. Perhaps compromised hosts and accounts on MIT's network are being used for espionage—certainly, some of them are being used in denial-of-service attacks and spam-sending operations, according to complaints lodged with IS&T—but those are activities MIT has limited unilateral ability to see or to stop because they occur primarily beyond the borders of its network. It is no coincidence that the most direct mode of harm defense undertaken by IS&T to provide homepage resilience required contracting with a third party: defending against ultimate harms requires a more global perspective than a single university is likely to possess. Nearly everything MIT knows about the kinds of threats it faces—and supports—is contained in Figure 3-3. It's not much of a basis for designing a defensive strategy.

Constrained by its own limited window into the larger arc of each security incident it deals with, MIT is trapped somewhere between being able to implement effective access defenses and meaningful harm mitigations, lacking a clear picture of which access capabilities are being exploited on its network and how those capabilities are evolving into bad outcomes. Teasing out the involved processes by which seemingly innocent capabilities such as sending email or downloading web-based content become the basis for enormously lucrative or disruptive crimes, and understanding the difference between defending against those capabilities and defending against those outcomes, therefore requires going beyond the perspective of an individual institution.

The distinction between access and harm is perhaps clearest in the context of computer security incidents when the harm that threat actors ultimately wish to inflict requires going beyond the boundaries of the computer system they target. For instance, many data breach incidents involve the theft of financial information (e.g., payment card numbers, billing addresses, etc.) from systems belonging to retailers, or other organizations. That stolen information is often then sold and used to steal money—through forged cards, identity theft, card-not-present transactions, or other means—inflicting financial harm on the victims (or, ultimately, significant losses on the credit card companies). This means that there are two distinct stages that attackers must accomplish in order to be successful: first, accessing financial information stored on a computer system, and second, using that information to steal money. The first of those stages centers on the question of access capabilities—how do threat actors acquire the necessary digital capabilities to get their hands on financial information—and the second focuses on harm, or how those actors then translate those digital capabilities into actual, financial damage.

Both the access and harm elements of a security incident provide meaningful opportunities for defensive intervention—clearly, an attacker with no way to access stored payment card numbers does not pose a threat to their owners, but similarly, an attacker with access to millions of payment card numbers who has no way to use that information to steal money cannot inflict any financial harm and therefore, in some sense, also poses no threat to the victims. This framing builds on the prevention-detection model, extending its philosophy of attack interruption prior to "ultimate mission accomplishment" beyond the boundaries of a single computer system. Bejtlich (2013, p. 5) asks: "if adversaries gain unauthorized access to an organization's computers, but can't get the data they need before defenders remove them, then what did they really achieve?" By expanding our window into computer security incidents beyond the computer system of an individual organization, it becomes possible to take that question even further: if adversaries get the data, or access, they need but can't decipher it, can't act on it, can't profit from it, can't leverage it to inflict their ultimate, desired harm, then what did they really achieve?

The access-harm framing of defenses is intended to draw out two different ways of defending against computer-based security incidents: constraining attackers' ability to acquire unexpected or unauthorized capabilities in the context of computer systems (access), and constraining their ability to use that access to inflict harm on others. Data breaches that enable financial fraud suggest a fairly straightforward division between an access phase aimed at stealing information from a computer system and a harm phase aimed at stealing money outside the context of that system, but in other cases, such as denial-of-service attacks or espionage, the access and harm elements of security incidents are more blurred. More generally, for security incidents that do not aim to inflict harm beyond the confines of the targeted computer system and therefore do not involve a component of direct physical or financial damage, harm is often synonymous with some specific computer access capabilities. This means that both classes of defense—those aimed at constraining access capabilities and those intended to mitigate harm—occur only in the context of computer systems. So, access defense might mean constraining attackers' ability to compromise hosts and build bots, while harm defense would involve finding ways to filter or block large volumes of malicious traffic, or mitigate its impact through partnerships like MIT's contract with Akamai. While these are all computer-based defenses, they have in common with the credit card fraud model a reliance on distinct entities to implement access and harm protections: the machines and networks being compromised are responsible for access defense, while targets, service providers, and third-party cloud providers are better poised to tackle harm.

Framing defense in terms of access restriction and harm mitigation sets up security incidents to be addressed from both their beginnings and their ends: the initial steps that enable threat actors to access computer systems, and the end goals that those actors are driving towards. Ideally, defending from both ends in this manner means that the access defenses reinforce weaknesses in the harm defenses, and vice versa, but, as MIT's experience juxtaposed between the two realms suggests, these two classes of defense do not "meet in the middle" and generally cannot be characterized in relation to each other. The classes of behavior constrained and defined by access defenses—password requirements, firewalls, email filters—do not map onto the classes of harm that attackers seek to inflict, including financial loss, espionage, and physical or digital service disruption. A password-guessing dictionary attack, for instance, might grant an attacker access capabilities that could be used towards any one of these kinds of harm—as might a phishing email or a malicious website. Similarly, each individual harm may originate from a variety of different access capabilities. Therefore, designing defenses that target specific access pathways requires defining attacks (or behaviors that need to be protected against) in terms of completely different characteristics from those used to design defenses that address specific classes of harm. Just as the two framings invoke different definitions of security, they invoke different attack taxonomies (one defined by attackers' initial access modes, another defined by attackers' ultimate goals) and protect against behaviors defined along fundamentally different axes. That does not mean that access defenses and harm defense cannot reinforce each other, but it does mean that they must be considered and implemented independently: that defending against access capabilities is no proxy for defending against harm.

3.1 Attack Trees and Kill Chains

Existing threat modeling tools, including attack trees and kill chain models, can help draw out the differences between access capabilities and harm, as well as the processes by which one leads to the other. Both of these techniques frame individual security incidents as the culmination of a series of different steps, involving a variety of actors, that provide multiple opportunities for defensive intervention. Attack trees are created by identifying possible goals of attackers and using each goal as the "root" of a tree, which then attempts to break down, step by step, possible different ways that goal might be achieved by the attacker (Schneier, 1999). Attack trees, depending on the specificity of the goal at their root, can capture both harms and access pathways, by tracing specific attacker goals all the way down to the different access capabilities that may be used to achieve those goals. Multiple trees can share the same subtrees and nodes, however, since the same capabilities or sub-goals may apply to multiple different attacker goals. Applying the access and harm-oriented frameworks of defense to computer-based threats is analogous to approaching threats from both the top and the bottom of these trees—from the attacker's end goal and initial access vector—but since there is significant overlap between the lower nodes on different trees, those two processes are distinct in what they target. Pruning a branch close to the root may provide strong protections against the goal, or harm, at the root node but still offer no defense against the access nodes that sprout from it, while defenses intended to prune off those access capability nodes may be relevant to a number of different trees, and harm classes, but do little to protect against the broader attacker goals that anchor those trees.

A more strictly sequential model for attacks is offered by Hutchins et al. (2011), who propose a "kill chain" model for advanced persistent threats, which they contend consist of seven distinct phases: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. "The adversary must progress successfully through each stage of the chain before it can achieve its desired objective; just one mitigation disrupts the chain and the adversary," the authors write, urging defenders to "move their detection and analysis up the kill chain [i.e., to earlier stages] and more importantly to implement courses of actions across the kill chain." However, these seven stages are too broadly defined to provide clear direction to defenders about which countermeasures to implement, and their proposed sequential progression is misleadingly linear. Especially during the access phases of attacks, there is no set order in which individual capabilities must be acquired and exploited. Stealing credentials via phishing emails and malicious websites can be a means of stealing sensitive information, or, conversely, attackers may steal sensitive data containing credentials as a means of accessing accounts to send phishing emails and spam. Similarly, stolen credentials can be used to deliver and install malware—or malware can be used to steal credentials. The potential circularity of these access capabilities makes it difficult for a defender to know, at any given moment, where they are interfering in an adversary's attack chain, or, indeed, what that adversary's ultimate intention is. It also gives that adversary considerable freedom to improvise and adjust course during these early stages of a compromise when capabilities can be so easily reordered and reassembled in response to defensive interventions.

This freedom from a set sequence of steps diminishes, however, as attackers reach the end of their kill chains and prepare to actually carry out their intended aim. Just as the harm defense model adopted by the nuclear safety community deals with a strictly sequential set of barriers, so, too, the harm infliction stages of computer-based security incidents follow a much more rigid order than do the earlier access stages. The structure of attack trees also illustrates the extent to which attackers' options—and, correspondingly, the number of pathways defenders must cover—narrows as adversaries get closer to their goal. There are lots of different ways for a thief to initiate the very early stages of credit card fraud, and defending against any one of those access modes individually may do little to hinder his progress, so long as there is another. By contrast, there are relatively fewer different ways to profit from stolen payment card numbers, so defenses aimed at that stage can be more narrowly focused. Implementing access defenses is still useful in many cases—both because harm defense often requires the cooperation of third parties who may or may not be inclined to offer assistance, and because access restrictions may still help dissuade at least some attackers from pursuing threats by increasing the work required of them. However, this narrowing of options as threat actors come closer to reaching their goals suggests that, counter to the assertion by Hutchins et al. (2011) that defenders should focus on moving "up the kill chain," some of the most fruitful opportunities for defensive intervention may occur in the latest stages of attacks.

3.1.1 Narrowing of Options

Initial access capabilities are likely to be largely interchangeable to threat actors, making it necessary for defenders to protect against a much wider array of different actions than is needed later on, when the intruders close in on their specific, motivating goals, leaving them fewer alternative paths. The interchangeable access modes reflect the "weakest link" philosophy of security, which dictates that any unprotected pathway is enough to allow for an attacker to be successful. The final, essential harm-inflicting phases of an attack, by contrast, resonate with the "kill-chain" security mentality, in which blocking any individual stage is sufficient to stop an attack entirely.

As attackers come closer to actually inflicting the harm that is their ultimate goal, the available options for how to achieve those malicious ends are likely to narrow and their behavior is also likely to become more unambiguously malicious. This is the crucial feature of attackers' options as they narrow—that coming closer to inflicting harm doesn't just mean fewer options, it also means there are fewer ways to mask activity as legitimate or non-malicious. Initial access capabilities such as sending emails, connecting to wireless networks, setting up websites, or physically using computers are all activities that non-malicious users engage in constantly for entirely legitimate purposes. By contrast, that is decidedly not the case for selling large blocks of credit card numbers—or actively inflicting harm of almost any nature. There may be any number of ways to access and steal sensitive information, for instance, or assemble large botnets—but once that information has been obtained or those botnets built and it comes time to put them to use, their owners' motivations are likely to dictate a fairly limited set of ways in which they can be exploited, and those uses are not likely to resemble legitimate, non-malicious behavior, precisely because they cause harm to the victims.

Maliciousness is defined by harm—which is part of what makes it so difficult to identify and defend against the access elements of attacks that occur prior to the infliction of that harm: there's often no clear way to distinguish between legitimate and malicious activity at that point. Thus, defending against initial access to capabilities on computer systems often presents a very different set of challenges from defending against the use of those capabilities to cause harm. In some ways, the former type of defense may actually be more straightforward—for instance, it may involve fewer third parties and therefore less complicated coordination problems and alignment of incentives among different actors. But in other ways, defense gets easier as attackers get closer to their end goals because the closer someone comes to actually inflicting harm, the easier it is to recognize his behavior as malicious.

Many of the access capabilities exploited by attackers—email, web queries, physical drives—are the same ones used by employees, customers, and non-malicious contacts, so access defense is primarily an exercise in how to make it as difficult as possible for malicious actors to disguise their activity as legitimate. Or, put another way, how to force malicious actors to do as much work as possible before they can acquire the essential capabilities needed in order to inflict harm. This means finding ways of distinguishing between malicious and non-malicious activity that offer reliable clues about whether or not they originate from someone intending harm, even before it is clear what that harm may be.

3.2 Access Capabilities

We often speak of attackers getting "into" (and, conversely, defenders keeping intruders "out of") computers, but drawing on this language from the physical world, where inside is a well-defined and binary state, can obscure the fact that the only way to define what it means to be "in" a computer system is in reference to a specific capability. Similarly, a number of computer security controls are focused on keeping intruders and thieves "out" of the systems they protect, but what they actually block can take a number of different forms, from malicious code and infected USB drives to misleading emails and unwanted Internet queries—and for each of these different vectors, "out" has a slightly different meaning. Thinking through the types of defense that can be used to block intruders and malicious access attempts on computer systems therefore requires recasting the notion of access not as a binary idea of either "in" or "out," but rather as a range of capabilities that an intruder may seek to acquire in the context of that system. Defenses, correspondingly, are not intended to keep those intruders "out" of the system so much as they are designed to restrict the capabilities permitted to those intruders. When breaching computer systems, an attacker does not access those systems (except in the rare cases where he physically procures them), he accesses specific capabilities within the context of those systems that can be exploited for harmful purposes.

This framing has implications for what "independence" means in the context of access defense. Recall that independent defenses are those that are unlikely to be simultaneously compromised by the same attack. For access defenses, whose primary function is distinguishing between malicious and legitimate activity on computer systems, independence is a matter of looking for different malicious (or legitimate) indicators. For instance, a defense that filters incoming emails sent from blacklisted domains and another that filters incoming emails sent with executable attachments would be independent because one uses domain names to identify malicious activity and the other uses attachment type—they also satisfy the overlap criteria, since to access a system via email an attacker faced with these defenses must both use a legitimate sending domain and disguise or dispense with any executable attachments (for instance, by phishing for credentials or including a link to a malicious website).

These sorts of indicators of malicious behavior are especially important for interfaces like email that are typically accessible to all users, whether those users are known to the system or not. Often, however, access defenses operate not by identifying specific behaviors as malicious but instead by restricting capabilities to a specific set of known users by means of authentication—on the assumption that users known to be non-malicious can be trusted with capabilities that, in the wrong hands, could be quite harmful. One way to understand the role of access defense then, is as a set of three related goals for constraining attackers' access: constraining the access capabilities granted to external (unauthenticated) users so that those capabilities are more difficult to wield maliciously, constraining potentially malicious capabilities so that they can only be acquired by users who provide authentication credentials, and constraining authentication credentials so they can only be provided by the authorized users to whom they correspond.

Every computer-based access capability—from receiving email to downloading web-based content or connecting to a wireless network—that is associated with potential harms it could be used to inflict suggests three possible options for defense: eliminating the mode of access altogether, identifying the harmful modes in the access capability and blocking them (i.e., determine features that distinguish malicious use of the capability from legitimate use), or restricting its use to authorized users and take steps to insure that they are trustworthy (i.e., distinguishing between malicious and legitimate users, rather than uses). For instance, defenders concerned about the exploitation of email capabilities for phishing (or other malicious purposes) could choose to block email entirely, block only messages with certain suspicious characteristics, such as links or attachments, or block messages from senders outside their domain. These approaches can be applied to more granular capabilities nested within broader access modes, as well. So, for instance, defenders concerned specifically with the capability of emailing attachments could instead apply those three strategies to that particular capability by blocking attachments, screening attachments for malicious indicators (e.g., file type or known malware), or only allowing attachments from known, authenticated senders.

All three of these defense tactics speak to the fundamental challenge of access defense in distinguishing between legitimate and malicious behavior. Taken together, they force defenders to divide computer system access capabilities fall into one of four categories: capabilities that are not associated with any harm and therefore require no defense, capabilities whose harmful potential outweighs any legitimate purpose and are therefore blocked outright, potentially malicious capabilities that can be constrained in some way so they pose less of a threat regardless of who exercises them, and potentially malicious capabilities that are granted only to a restricted set of trusted users. Dealing with the first two sets of capabilities is relatively straightforward; determining which capabilities fall into each of the latter two classes—and how they can be effectively constrained—is the work of access defense. Capabilities entrusted to all users call for defenses that constrain their malicious uses (e.g., using email to deliver malware or phishing messages, or targeting public web servers with denial-of-service attacks) through restrictions on their nature and extent. Capabilities entrusted only to authenticated users can almost always be put to malicious use—otherwise there would be no reason to restrict them to authorized users—so access defenses for these capabilities must instead focus on strengthening the authentication credentials required to acquire them.

Enumerating all of the different capabilities users can acquire in the context of a computer system is, all by itself, a Herculean task. Inevitably, a defender will fail to account for some of them, or attackers will discover ways of acquiring new ones that defenders did not anticipate. But those discoveries of truly novel access pathways seem to be the exception, not the rule—many security incidents, including the cases discussed in Chapter 4, make use of access pathways that are well known and readily identifiable, including email, USB drives, passwords, wireless networks, websites, and web server queries. So setting aside the impossibility of creating an exhaustive list of access capabilities, access defense begins by dividing these known capabilities according to their potential for both legitimate and malicious use. Those assessments are highly dependent on the nature of the system being protected—a crucial capability in one context might be utterly unnecessary in another; for instance, an open wireless network may make perfect sense in a coffee shop but be entirely inappropriate in an office. One set of possible designations for a hypothetical system is shown in Figure 3-4.

These distinctions then dictate the kinds of access defense that need to be implemented for each capability. Capabilities that have high potential for both malicious and legitimate uses (i.e., those in the upper right quadrant of the maliciousness-legitimacy axes in Figure 3-4) should be restricted to authorized users or constrained in some fashion that reduces their potential for malicious use. Those in the upper left quadrant, which have only low potential for malicious use in the context of the

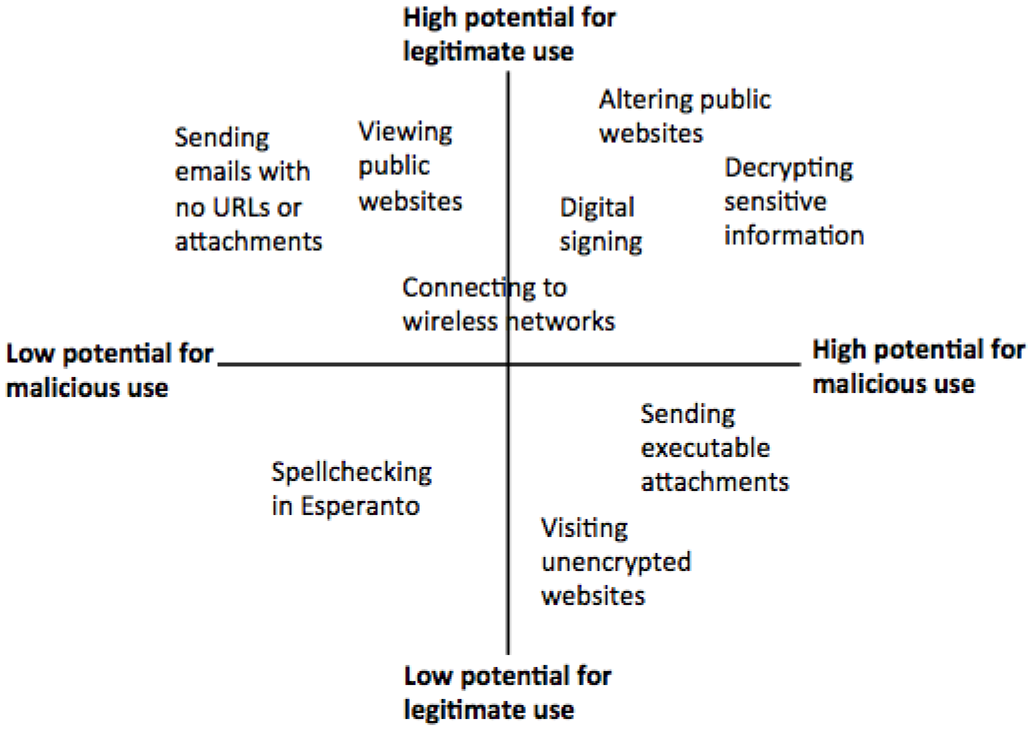

High potential for legitimate use
Altering public websites
Sending emails with no URLs or attachments
Viewing public websites
Decrypting sensitive information
Digital signing
Connecting to wireless networks
Low potential for malicious use
High potential for malicious use
Sending executable attachments
Spellchecking in Esperanto
Visiting unencrypted websites
Low potential for legitimate use

Figure 3-4: A possible division of different computer system capabilities according to their potential to be used for legitimate and malicious purposes.

defended system but still serve a legitimate function, should be afforded to all users. Finally, capabilities in the lower two quadrants, that serve little or no legitimate purpose, should be accessible to no one.

3.2.1 Capabilities for Unauthenticated Users

Many of the things we do on computers, and especially online, require credentials of one form or another. Accordingly, many computer security incidents—including 76 percent of data breaches, according to *Verizon Data Breach Investigations Report* (2013)—involve exploiting weak or stolen credentials to access protected capabilities. But not every application or capability is protected by authentication credentials—email and the web both allow for regular interaction among strangers without the support of any centralized repository of credentials. While these applications may be the exception rather than the rule in enabling unauthenticated capabilities, those capabilities often play crucial roles in security incidents—either as an alternative to or a means of acquiring credentials—and present their own set of defensive challenges. Fundamentally, designing defenses for these capabilities is a matter of reducing them to their most minimal, least harmful components—and then requiring credentials to exercise all of the other components, or sub-capabilities, they encompass. So in some sense, defending against unauthenticated capabilities involves transforming them into authenticated capabilities or, rather, separating out the elements that pose the greatest risk and transforming those into distinct authenticated capabilities.

The underlying principle of this process is simply that any capability that relies on absolutely no credentials should be very limited since it will be so easily obtained by everyone and difficult to trace back to an individual. There may be perfectly legitimate, non-malicious reasons to make certain capabilities easily obtainable and anonymous, but that puts the onus on defenders to figure out ways of making those capabilities difficult to use to inflict harm. If it makes sense to allow everyone to connect to a wireless network, or send email to an account, or query a web server, how can these capabilities be narrowed, or limited in scope, so that they can only be easily used for non-malicious purposes? These constraints are dictated by the potential malicious uses of those specific capabilities, so they may take many different forms, from restricting the websites devices may load to defining the characteristics of permissible emails or limiting the volume of allowed traffic. Restrictions of this nature aim to retain the legitimate functions of a capability while lowering its potential to be used for malicious purposes, essentially trying to drive capabilities further to the left on the "maliciousness" axis in Figure 3-4 or, more precisely, to define related but slightly more restrictive subcapabilities that afford fewer opportunities for misuse.

One means of designing access defenses for unauthenticated user capabilities is to restrict the capabilities for everyone in exactly the same manner; another model is to create an escalating set of capabilities that can be garnered by unknown users through the investment of money or time, even though the system in which they are granted these capabilities has no knowledge of their real identities. (This is still a means of transforming an unauthenticated capability into one that requires credentials—but it allows for those credentials to not necessarily be tied to a real person or even a trusted issuer). This approach targets the ease with which unauthenticated capabilities can be required rather than the extent of the capabilities themselves. So access defenses may apply to users who are not known to a system either by restricting their capabilities to completely exclude potentially malicious uses or by forcing them to invest more heavily in acquiring those potentially malicious uses. The latter model only makes sense if the potentially malicious uses also serve legitimate functions in the hands of strangers (or people whose credentials are not necessarily issued by a trusted entity), but this is not uncommon for access capabilities.

Picking apart the malicious and legitimate elements of a computer access capability is not easy, especially at an institutional level where administrators are faced with monitoring and defending against capabilities in a wide range of different applications not necessarily of their own design. In fact, applications that enable interaction with people who do not possess credentials, or do not possess credentials issued by a trusted, centralized authority, are in some sense the applications that allow for the least unilateral tinkering and implementation by organizations. If part of the point of an application is that it facilitates communication and interaction with the outside world, then it stands to reason an institution will want to use an application that is already in widespread use by others. When it comes to applications intended exclusively to facilitate internal interaction among their own members, organizations may have greater freedom to design or customize their own security features, but decentralized applications like email and the web, that are intended to facilitate communication beyond a single, specific community, require that institutions make use of commonly used and independently designed applications. This means that constraints on unauthenticated capabilities (or lack thereof) are often already built into applications and users have relatively few means of unilaterally tailoring those defenses that still allow for widespread interaction with other users. So the responsibility for subdividing unauthenticated capabilities into their more and less malicious components, and defending against the former through use of credentials, falls largely on application designers. Since the design of unauthenticated access defenses depends heavily on the specific nature of individual capabilities and their potential legitimate and malicious uses, these types of defenses are often most effective in the context of specific applications, designed with a particular function (or functions) in mind.

3.2.2 Capabilities for Authenticated Users

Anyone can send an email or set up a webpage, but many computer system capabilities are restricted to a set of people whose identities are known to the systems' owners and administrators (e.g., employees, family members, students). These capabilities typically serve valuable, legitimate purposes when wielded by non-malicious users but, in the wrong hands, have the potential to further malicious goals. Lots of common computer capabilities fall into this category, from reading emails to updating online content to deleting files. Generally, these sorts of activities are restricted through an authentication process in which users provide credentials—passwords, biometrics, one-time codes—that are intended to verify their identities. For these capabilities, the challenge of access defense is not distinguishing between malicious and legitimate activity—since that distinction is entirely dependent on who is performing the activity—but rather distinguishing between malicious and legitimate users. With the exception of insider threats, discussed briefly in Section 3.2.3, distinguishing between malicious and legitimate users is about trying to ensure that authentication credentials cannot be used or imitated by anyone other than the individual to whom they are issued.

To some extent, tying credentials to specific people reverses the challenges of defending against the malicious use of unauthenticated capabilities. Strengthening authentication defenses means looking for multiple, independent indicators (passwords, tokens, biometrics, etc.) associated with a legitimate identity. Defenses that constrain unauthenticated capabilities, by contrast, revolve around screening computer activity for the absence of multiple, independent indicators associated with malicious activity (e.g., executable attachments, malware, suspicious keywords or content). In other words, defenses for authenticated capabilities derive from some definition of how to identify trusted users, while defending against misuse of unauthenticated capabilities instead requires assumptions about how to identify malicious activity. Correspondingly, evading the former mode of access defense requires attackers to collect all the attributes of a trusted user, while circumventing the latter only requires attackers to avoid any actions that are explicitly forbidden.

Despite these differences, authenticated and unauthenticated capabilities are closely intertwined, both because unauthenticated capabilities (e.g., websites or emails) often serve as the means by which attackers acquire credentials needed to exploit authenticated capabilities, and also because every authentication defense requires the creation of a new unauthenticated interface that can also be exploited. That is, implementing any form of authentication requires creating an authentication interface of some form (usually a login screen) that is accessible to everyone, including those without trusted credentials, since there is no way of discerning between people who do and do not possess those credentials until they have had a chance to provide them to that interface. Put another way: the act of trying to authenticate oneself is itself, fundamentally, an unauthenticated capability. So defending against brute force password guessing attacks by limiting the permitted frequency of login attempts is both a defense that constrains an unauthenticated capability (logging in) and a defense against credentials being acquired by someone other than their rightful owner.

Authentication interfaces can also undermine the relative independence of multiple credentials. For instance, a common construction for authentication entails requiring users to provide both a password and a code from a physical token or phone. These two credentials are independent in that the process of guessing or phishing users' password is unlikely to also yield their phones (and stealing phones is similarly unlikely to reveal their owners' passwords), however these independently accessed credentials are often entered into a centralized authentication interface, creating an opportunity for attackers to intercept and reuse credentials, as in the case of the Gameover Zeus botnet (Krebs, 2014a), or by initiating malicious activity in real time, during an authenticated session (Lemos, 2009). Some authentication efforts go beyond requiring multiple independent credentials at a single "gateway" interface, by analyzing behavioral indicators—such as typing patterns or eye movements—throughout authenticated sessions, as well (Bergadano, Gunetti, & Picardi, 2002; Komogortsev, Karpov, & Holland, 2012).

Authentication defense is complicated by the crowded landscape of different defenders that issue and store credentials. Some credentials are issued by individual applications for their users (maybe based on knowledge of those users' real identities, maybe based on no more than an email address), others are distributed by employers to their workers, and still others are created by individuals for themselves. Similarly, credentials may be stored on servers administered by applications or organizations, or kept locally on devices and administered by the operating system, or even stored in browsers or centralized credential managers protected by their own credentials. This makes it difficult to separate out specific actors who are responsible for authentication defenses—some credential management is done at the application level, some at an institutional or organizational level, some at a strictly personal level. It also reinforces the limited visibility of individual defenders. The strength of MIT's users' credentials does not rely on MIT alone. If users store those passwords in browsers or password managers, or reuse them for other accounts, then the security of those credentials is no longer up to just MIT. This dependence on third party systems that MIT has no control over and no insight into adds to the challenges of access defense, leaving organizations like MIT with limited ability to protect their own credentials.

3.2.3 Insider Threats

The overarching goals of access defense for unauthenticated user capabilities and authenticated users, respectively, are to make it more difficult for users without credentials to acquire potentially malicious capabilities in the context of a computer system, and to make it more difficult for users who have not legitimately been issued credentials to acquire someone else's credentials. What these defensive goals do not account for is the possibility that someone who has been legitimately issued credentials may use the capabilities those credentials afford towards malicious ends. The cases discussed here do not, for the most part, involve significant malicious insider involvement, but it is worth noting that these threats can be described in terms of the proposed access defense framework of constraining user capabilities.

Insider threats demonstrate the need for there to be restrictions placed on authenticated users' capabilities as well as on unauthenticated users' capabilities. Some of these capabilities, which have no clear legitimate function in the context of the system being defended (i.e., those that fall in the bottom right quadrant of Figure 3-4), may be blocked from all users, authenticated and unauthenticated alike, as part of an access defense approach. Insider threats illustrate the crucial importance of not neglecting this quadrant of capabilities—some access capabilities may simply serve too little legitimate purpose (or conversely, provide too great a threat) to be allowable even to authenticated users.

Other capabilities may serve important legitimate functions but still be sufficiently rarely used or associated with such great potential harm as to require the concurrence of multiple authenticated users, each of which serves as an independent line of defense against misuse. The National Security Agency's 2013 implementation of a "two-man rule" requiring "a second check on each attempt to access sensitive information" following the leaks by Edward Snowden is an example of precisely this kind of defensive independence (Drew & Sengupta, 2013). These additional lines of independent authentication defense don't just serve to protect against insider threats—they also drive up the work factor for external attackers trying to exploit authenticated access capabilities by stealing or forging credentials.

3.3 Harms and Attacker Goals

Ultimately, what defenders care about is not what computer capabilities users have but rather what kinds of harm can be inflicted through the use of those capabilities. That harm is what makes activity malicious—and it is the anticipation of that harm that motivates access defenses. If access defense involves starting with different computer system capabilities and trying to anticipate how they may be used for malicious means, damage defense involves starting from the other end, with the ultimate harms attackers aim to inflict, and then working backwards to understand which capabilities are most essential to their infliction. To some extent, identifying the harms inflicted using computer security breaches is easier than listing computer systems capabilities—the classes of harm that constitute attackers' end goals tend to be relatively stable over time, unlike computer capabilities which proliferate rapidly.

Malicious actors aim to inflict some measure of physical, economic, or psychological harm on their victims, often in combination (for instance, embarrassing revelations of sensitive information may be both psychologically scarring and costly for targets; similarly, physical disruption of a service can lead to economic harms if it interrupts an organization's business). These classes of harm are not specific to computer security incidents, and as with attack categorizations around breaches of confidentiality, integrity, and availability, these harms are too broad to imply clear defensive interventions because each can be achieved in so many different, and sometimes overlapping, ways. In considering computer security incidents, it is therefore helpful to identify more specific and distinct classes of common desired "end results" or goals of attackers.

Charney (2009) proposes four categories of attacks: conventional cyber crimes, in which "computers are targeted for traditional criminal purposes, such as fraud," military espionage cases, in which "nation-states intrude into and exfiltrate large amounts of sensitive military data from government agencies," economic espionage cases, and cyber warfare. Focusing in on the types of harm each of these different categories implies offers a more concrete starting point for considering what classes of harm damage defenses must cover. These include:

- financial harm inflicted directly through:
    - payment card fraud,
    - unauthorized direct transfer of funds, or
    - ransom attacks used to extort authorized transfer of funds,
- disclosure of sensitive political or military information,
- economic harm inflicted indirectly through theft of intellectual property or sensitive market information, and
- disruption of physical operations.

Two other common forms of harm that cut across Charney's attack categories are:

- disruption of digital service, and
- the infliction of reputational harm through public embarrassment and negative publicity (which also sometimes leads to indirect economic harms).

These classes of harm lend themselves to very different mitigation defenses—even though the same computer system capabilities can be leveraged to achieve many of them.

Harm defense often involves a wider range of different defenders than access defenses because harms are not limited to the context of computer systems or applications. Many attackers try to translate computer system access capabilities into harms whose impacts are not just digital, but also financial or physical. This means that a variety of third parties, distinct from the one that owns or operates a breached computer system, may have opportunities to limit the harms that can be inflicted through that translation process. Depending on the type of harm being defended against, these actors may range from payment card processors to law enforcement officers and policy-makers—and each additional third party that contributes to harm mitigation defenses can be thought of as adding another degree of independence to the defense.

While this institutional independence has the potential to strengthen combined defense constructions, it can also present significant coordination problems. Often, defensive actors poised to protect against different stages of the same class of harm end up blaming each other for security incidents and avoid taking active defensive steps to address threats for fear that in doing so they will end up increasingly held responsible for any successful breaches. Accordingly, they may prefer to argue with each other about whose responsibility defense ought to be, rather than focusing on trying to address it collectively. For instance, in the aftermath of a major data breach of retail chain Target, five banks pursued lawsuits against the company to recoup their losses from the resulting payment card fraud (Perlroth, 2014). The challenges associated with assigning harm defense responsibilities to multiple different actors in such cases present opportunities for regulatory intervention, implying a significant role for policy-makers in implementing harm defense effectively.

Many types of harmful behavior—selling payment card information, manufacturing fraudulent credit cards, selling products based off stolen intellectual property, disrupting physical infrastructure and operations—occur outside the context of protected computer systems. However, in some cases, a security incident can cause harm without ever going beyond the virtual realm—perpetrating denial-of-service attacks, for instance, involves no harmful behavior beyond a barrage of digital communications, nor does military and political espionage, if there is no further harm inflicted beyond the successful acquisition of information stored on a protected computer system. Similarly, incidents aimed at defacing public websites, or altering or erasing stored data, offer little recourse to non-computer-based defenses. For these "digital harms," harm defense often means defending against the capabilities acquired in the context of the targeted system—the same role of access defense in many other situations, where these capabilities were a means to an end, not an end in themselves. This means that access defense shifts as well, to defending against the capabilities acquired earlier, in the context of other systems, that were used to target the ultimate victim.

3.3.1 Digital Harms

Incidents whose scope is solely digital and that never extend beyond computer systems are constrained in some ways as to how devastating their impact can actually be on people. They may be immensely inconvenient or embarrassing—and result in loss of business or trust or valuable information—but they cannot, taken by themselves, be the cause of physical harm. They may, in some fashion, contribute to a larger physical or financial goal—but actually inflicting such harm usually means moving from a purely digital form of harm to one more grounded in the physical world, and that transition creates new opportunities for defense. Still, the consequences are sufficiently serious to warrant some attention to the modes of possible defense, and while these harms present fewer opportunities than non-digital harms for defensive intervention outside the context of targeted computer systems, they also feature more clear-cut malicious behavioral indicators within the context of those systems. This is because the narrowing of options that occurs as attackers come closer to inflicting harm happens almost entirely in the course of acquiring access capabilities for attackers aiming to inflict digital harms. So instead of having a variety of different options for viable access capabilities that can all be used towards the same malicious ends, attackers are eventually forced to acquire fairly specific access capabilities—the ones that directly enable the infliction of some digital harm.

For instance, two classes of common digital harms—denial-of-service attacks and political espionage—suggest two types of malicious indicators that may be valuable for distinguishing between malicious and legitimate activity (of both access capabilities and harmful behavior, in these cases): volume of web server queries and data exfiltration. Neither of these behaviors is necessarily always malicious, but there is no way to cause a denial-of-service attack without the capability to initiate a large volume of traffic directed at a single server, nor to perform espionage on a protected system without an exfiltration capability. In other words, these capabilities are *essential* to the infliction of these harms—in fact, it is the very act of exercising that capability that inflicts the harm in these cases. Compared to the classes of harm that can be achieved via a range of different capabilities, digital harms are unique in this regard; they rely on irreplaceable, essential access capabilities that can serve as valuable defensive bottlenecks, since attackers must acquire these capabilities in order to achieve their ultimate goals.

Many access capabilities do not meet this criteria of essentialness. For instance, an intruder faced with strong email or phishing defenses might resort to other means of gaining the capabilities he is ultimately after—this might mean guessing credentials, or intercepting them, or connecting to unprotected networks to find stored credentials, or tricking a credentialed user into transmitting malicious code via a physical drive, or even foregoing credentials entirely and instead making use of unauthenticated capabilities. There is no type of harm for which the successful execution of a phishing attack—or a dictionary attack, or a buffer overflow attack, or an SQL injection, or any number of other common malicious behaviors—is essential. It may still be well worth defending against these behaviors—just because they are not essential to attackers doesn't mean they aren't often useful—but it is a mistake to believe that defending against any (or even all) of these behaviors is an effective strategy for addressing classes of harm.

This is part of what makes it difficult to map many defenses onto consistent classes of attacks. The success of an attack is not determined by the perpetrator's specific technical maneuvers but rather by his ability to inflict some specific harm on the target. Therefore, the defining characteristic of that attack from a defender's point of view should be the harm it imposes—and how to guard against that harm. But many technical defenses instead map to specific technical vulnerabilities—vulnerabilities that may be exploited by attackers after a variety of different aims—obscuring the fact that their implementation guards only partially against any class of harm, even though it may fully cut off a class of technical maneuvers. The more essential a capability is to the type of ultimate harm an adversary aims to cause, the more closely a defense against that capability will map onto that class of harm and provide protection against it. So defending against digital harms is tricky in that it must rely entirely on computer-based defenses—but because attackers' narrowing of options occurs entirely within the context of those computer systems, it is also easier to identify capabilities that will actually address an entire class of digital harm.

3.4 Intermediate Harms

Computer security incidents do not always divide neatly into distinct access and harm stages but, generally, the shift from one to the other accompanies a shift from one defender's scope of control to another. For instance, in the context of payment card fraud the access components might occur in the context of a retailer's computer system and the harm components in the context of a global black market for credit cards. By contrast, for a denial-of-service attack, access capabilities might be used to gain control of people's computers while the actual harm infliction would center on targeting a third party's servers. These stages represent distinct shifts in the burden of defense—from the retailer to law enforcement and credit card companies, from the owners of inadequately protected computers to the targets of denial-of-service attacks—and highlight the limited control of any individual defender.

Nested within these larger chains of events that span multiple actors, each individual defender may also experience a more micro-level or personal version of both access and harm, in the context of their own respective systems. Even intermediary defenders who are not the targets of the "ultimate harm" the attacker intends to inflict (e.g., theft, espionage), and therefore may not have visibility into those outcomes, may be able to see and stop "intermediate harms" on their own systems. The compromised accounts and hosts on MIT's network are examples of such intermediate harms—most of them are, presumably, intended for use towards some larger malicious purpose that does not necessarily target MIT, and that MIT may therefore not be able to track. So, in and of themselves, these compromised resources may not be inherently harmful but they are still, indisputably, malicious. That is, there is no legitimate reason for accounts or hosts to be controlled by people without the explicit knowledge and permission of their owners. In this regard, these compromises differ from the access capabilities that attackers exploit to gain control of MIT resources, such as sending email and loading web pages, which can be used for both legitimate or malicious purposes.

For intermediary defenders who are not the targets of—and cannot easily detect—the ultimate harms inflicted through security breaches, and who are also not the designers of their own applications—and cannot easily dictate the capabilities afforded by the software they use—identifying these intermediate harms can be helpful for implementing security measures that fall somewhere between access and harm defense. At the same time however, these intermediate harms are difficult to defend against effectively without some clearer understanding of the access pathways through which they're achieved or the harmful purposes for which they're being used. In the absence of that broader perspective on the full chain of events leading up to a successful security incident, an intermediate defender is all too likely to find itself defending against the wrong thing, not necessarily because of any malice or stupidity or lack of adequate resources, but merely because that blindness to all but a tiny portion of the full narrative arc of a security incident makes it so difficult to identify what is actually being defended against. Individual defenders are limited in their decision-making by how much visibility they have into those narrative arcs, which typically span months and multiple different actors. To understand the roles of different defenders and defenses, it is therefore essential to reconstruct the full timelines of such incidents and the different steps and stages and lines of defense that attackers progress through in order to pull them off successfully.

Chapter 4

Case Studies in Defense

Understanding the access and harm components of actual security incidents requires a fairly detailed picture of these events, one that traces their journey across multiple systems from innocuous, initial access capabilities to full-fledged, headline-making disasters. But most defenders are understandably reluctant to disclose much information about the security incidents they witness for fear of incurring liability, generating bad publicity, or providing useful information to attackers. So our visibility into the specifics of how security breaches occur, and what measures were and were not in place to defend against them, is limited, and centered on certain types of incidents and information. For instance, the enactment of data breach notification laws in the European Union and many states in the United States has led to much more disclosure about the occurrence and size of incidents involving personally identifiable information. However, since the aim of these policies is consumer protection, rather than defensive assessment and analysis, often these disclosures are limited to what information was breached and how many people it affected—they rarely address how it happened or the target's defensive posture. Even less is known about security incidents that do not involve personally identifiable information, since organizations have not been required to disclose any information about these events. And if it is challenging to gather information on the defensive landscapes faced by successful threat actors, it is perhaps even harder to investigate those encountered by unsuccessful ones—specific incidents in which a defender's combination of protections successfully mitigate a threat are very rarely disclosed or discussed in public.

Occasionally, a detailed report on a particular incident is released publicly—either as a public service or a form of self-promotion by the investigating organization. For instance, following the 2011 compromise of the Dutch certificate authority DigiNotar, the European Network and Information Security Agency (ENISA) released the Fox-IT investigation report of the incident; the Governor of South Carolina did the same with a Mandiant report investigating the 2012 breach of the state's Department of Revenue records. Other reports on specific incidents, such as the Mandiant investigation of Chinese espionage efforts directed at the United States and the CloudFlare analysis of the distributed denial-of-service attacks directed at Spamhaus, are released by the investigating firm rather than the target, perhaps to generate attention and business. Finally, there are a small number of high profile incidents that generate so much media attention and so many legal disputes that it becomes possible to piece together detailed information from a variety of independent media stories, research investigations, and lawsuits, even in the absence of a single, focused report. Accumulating adequate information through these disparate outlets is usually a long process and therefore tends to be most helpful when dealing with incidents many years in the past, such as the 2007 data breach of retail chain owner TJX Companies, Inc.

This analysis focuses on four security incident case studies—the 2007 TJX breach, the 2011 DigiNotar compromise, the 2013 Mandiant investigation of Chinese espionage efforts, and the 2013 denial-of-service attacks directed at Spamhaus—for which there is detailed information available about not just what happened but also how it happened, and what defensive measures were involved. These cases span several motivations and classes of harm—including financial fraud, political espionage, economic espionage, and disruption of digital service—as well as different technical exploits, and targets. They offer glimpses into how the access and harm framings of defense can be applied to different types of incidents and defenders, the window each of those defenders has into the evolution of those incidents from computer access capabilities to harmful outcomes, and the implications of those framings for defensive decisions and responsibilities.

4.1 TJX Companies, Inc. Breach (2005–2007)

On January 17, 2007, retailer TJX Companies, Inc., which operates T.J. Maxx and Marshalls stores, announced that its computer systems had been breached, potentially resulting in the theft of millions of credit and debit card numbers. As the estimates of the quantity of stolen card numbers and the duration of the breach steadily increased over the following months, rising to a reported 90 million card numbers stolen over a period of 18 months, the case attracted significant attention as the largest such incident ever reported at the time. Though TJX reported in a 2007 filing with the SEC that they had engaged General Dynamics Corporation and IBM to assist in investigating the breach, no report on the investigation was ever made public by the company. However, the resulting lawsuits and media reports, which continued for several years following the 2007 announcement, as well as the subsequent arrest and prosecution of several of the perpetrators, make it possible to reconstruct an unusually detailed timeline of how the TJX breach was carried out.

The TJX breach turned out to be one in a series of credit card theft schemes, dubbed Operation Get Rich or Die Tryin' by their chief perpetrator, Albert Gonzalez. There is little doubt, given the name of the operation and the type of information that Gonzalez targeted, that he was motivated primarily by financial gain. In an online chat with one of his co-conspirators on March 7, 2006, he wrote: "I rather stay home and make money, I have a goal . . . I want to buy a yacht" (*USA v. ALBERT GONZALEZ, Indictment*, 2008). In 2003, using blank bank cards encoded with stolen debit card information at ATMs in upper Manhattan, Gonzalez would withdraw the maximum daily limit on each card right before midnight, and then repeat that transaction several minutes later, after the limit had reset for the next day. He was arrested for this in July 2003 by the New York Police Department and later recruited as an informant by the U.S. Secret Service (Verini, 2010). It was during his four-year association with the Secret Service, that Gonzalez initiated the TJX breach—even as he continued to help law enforcement identify and arrest other criminals.

The access dimension of the TJX breach began in July 2005, when Gonzalez, along with his friend Christopher Scott, identified potential targets by driving along the South Dixie Highway in Miami with a laptop and a high-power radio antenna. They were looking for vulnerable wireless networks that permitted open system authentication (OSA), allowing any device to connect to the network. Gonzalez and Scott identified and successfully compromised several commercial targets, including BJ's Wholesale Club, OfficeMax, Dave & Buster's Restaurant, and Marshalls, owned by TJX. Since the stores' networks were configured to allow OSA instead of shared key authentication, which would have required devices to provide a shared key before they were permitted to join the network, Gonzalez and his co-conspirators were able to join the stores' networks easily, just by sitting in the parking lot within range of the wireless signals. However, the traffic between devices on those networks was encrypted with Wired Equivalent Privacy (WEP) wireless encryption. This meant that while the thieves could monitor the encrypted traffic freely, if they wanted to understand it they needed the encryption key. This posed an obstacle to the intruders—but not an insurmountable one: an attack discovered in 2001 on the stream cipher used in WEP showed that it was possible to derive the key from large volumes of collected WEP wireless traffic (Fluhrer, Mantin, & Shamir, 2001). Using a packet sniffer, Gonzalez' team captured the encrypted traffic on the Marshalls store network and was able to exploit patterns in that encrypted traffic to identify the WEP key. Armed with the encryption key, they could then decrypt the traffic on the store's network, including store officials' password and account information, which enabled the conspirators to access different computer servers containing payment card data within the TJX corporate network, as well as "track 2 data," the information found on the magnetic stripes of credit and debit cards (*USA v. ALBERT GONZALEZ, Indictment*, 2008).

Using the employee credentials they had decrypted off the store's network, Gonzalez' team was then able to connect to the TJX corporate servers in Framingham, MA. With access to the corporate servers established, Scott and another member of the group, Jonathan James, rented rooms near the Marshalls store and used a six-foot radio antenna to capture the store's signal from their hotel so as not to attract attention by spending hours in the parking lot. From their hotel room, the men could then capture not just the bank card information for transactions processed in the single store in Miami, but also the track 2 data for tens of millions of credit and debit cards used in transactions at the thousands of stores operated by TJX worldwide. Card data from transactions prior to 2004 had been stored in cleartext, but the information relating to more recent transactions was encrypted, due to a change in TJX security practices. Concerned that the older information would be less valuable, since those cards would be more likely to have already expired, or be closer to expiring, than the ones used to make more recent purchases, Gonzalez enlisted help from accomplices in Eastern Europe to decrypt the more recent data.

This encryption added a considerable hurdle. On May 27, 2006, Gonzalez wrote in an online chat to his co-conspirator who sold the stolen information: "have patience please :) it took me 2 years to open pins [PIN numbers] from omx [OfficeMax]," adding, "2 years from the time I hack them, to download all data, the to find proper decryption method" (*USA v. ALBERT GONZALEZ, Governor's Sentencing Memorandum*, 2008). Since two years is half the lifespan of an average credit card, this delay would have been a serious setback for Gonzalez and his team. However, while downloading the encrypted data from the TJX servers, they had discovered that for a brief moment, while a transaction was being processed, between the time when a card was swiped and the credit card company network approved it, that card's data was available unencrypted on the TJX payment card transaction processing server. So Gonzalez recruited his friend Stephen Watt to program a custom packet sniffer, named blabla, to capture the unencrypted card information at this exact moment, obviating the need for time-intensive decryption. The blabla program was installed on the TJX corporate server on May 15, 2006, and immediately began capturing the unencrypted card data and compressing it into files stored on the corporate servers, which Gonzalez' team could then download through their access to the Miami Marshalls store. To relieve their dependence on the Marshalls access point, in May 2006 the group established a VPN connection between the TJX corporate payment processing server and a server in Latvia controlled by Gonzalez. During the latter half of 2006, Gonzalez downloaded the unencrypted card data from millions of current transactions in TJX stores all over the world directly to his server in Latvia.

Gonzalez had successfully acquired millions of credit and debit card details, but he still didn't have what he really wanted: money (or a yacht). Access capabilities had gotten him this far, but in order to turn stolen information into stolen money he had to move beyond the confines of the TJX computer systems—into the harm infliction stages of the attack. He enlisted the help of Maksym Yastremskiy, a Ukrainian black market card seller, who operated a website advertising stolen payment cards. Buyers would wire money to Yastremskiy who would then send them the purchased card information. Yastremskiy would then pay Gonzalez using ATM cards linked to accounts he set up in Latvia. These cards were delivered to "cashers," who were responsible for withdrawing the money from the Latvian accounts and then sending the cash (less a 10 percent commission) to Gonzalez' drop box in Miami. Sometimes, Gonzalez would also send couriers to collect cash directly from Yastremskiy's partners. Humza Zaman, a firewall security specialist at Barclays who acted as both a courier and casher, later testified that he was responsible for repatriating between $600,000 and $800,000 for Gonzalez.

In December 2006, a credit card company noticed that several compromised cards were linked to TJX stores and alerted the company to a possible breach. On the advice of law enforcement officials, the company waited a month to alert customers in hopes of being able to trace the thieves before scaring them off with a public announcement. The General Dynamics and IBM investigation did not go unnoticed by Gonzalez, however, and he decided to shut down the exfiltration of TJX information, writing by way of explanation "after those faggots at general dynamics almost owned me with 0day while I was owning tjx I don't want to risk anything" (*USA v. ALBERT GONZALEZ, Governor's Sentencing Memorandum*, 2008). The TJX investigation did not appear to succeed in identifying the responsible parties, but many of those involved—including Gonzalez, Yastremskiy, Scott, Watt, James, and Zaman—were later identified following Yastremskiy's capture in a nightclub in Turkey in July 2007.

In Yastremskiy's chat logs, the Secret Service agents found the username of someone who had been supplying many of the stolen card numbers. They could not immediately identify whose username it was, but the same person had asked Yastremskiy to provide a fake passport to help a casher who had recently been arrested leave the country. The investigators figured out that the anonymous supplier was Jonathan Williams, one of Gonzalez' cashers, who had recently been arrested while carrying $200,000 in cash and 80 blank debit cards. Searching a thumb drive that Williams had on him at the time of his arrest, the Secret Service found a photo of Gonzales, his credit report, and the address of his sister Maria—materials Williams said were meant to serve as insurance against Gonzalez ever informing on him (Gonzalez was, after all, a Secret Service informant). The investigators traced the cash Williams was in charge of delivering to a P.O. box in Miami registered to Jonathan James, and then discovered an arrest record for James from 2005, when he had been found late at night in a store parking lot, sitting in a car with another man, named Christopher Scott, along with laptops and a huge radio antenna. Finally, the Secret Service tracked down the registration information for the chat username of Yastremskiy's major supplier, and linked the email address, soupnazi@efnet.ru, to the online alias Gonzalez was known to use: soupnazi. Gonzalez was arrested on May 7, 2008, at the National Hotel in Miami Beach (Verini, 2010).

4.1.1 Access Capabilities & Defense

The TJX breach starts with Gonzalez and his co-conspirators acquiring a series of capabilities in the context of first a Marshalls store's computer systems, and later the computer systems of its parent company TJX. These escalate gradually from connecting to a single store's wireless network and collecting the encrypted traffic it carries, to decrypting that traffic and using it to connect to the TJX headquarters' servers, to intercepting real-time, unencrypted transaction information and exfiltrating that data from TJX servers in Framingham to a non-TJX server in Latvia. Some of these capabilities TJX could have monitored and restricted—for instance, by limiting access to its wireless network, or outbound traffic from its servers—but others, such as the decryption of wireless network traffic and stored payment card information, were acquired outside the company's purview. TJX could have made that decryption more difficult by using stronger encryption algorithms, especially for its store's wireless network, but it had no way of knowing that anyone was attempting to decrypt its data somewhere off in Eastern Europe.

The question of which stages of a security incident an individual organization like TJX can and can't see—which elements it does and does not have control over—is important because it speaks directly to that organization's capabilities and responsibilities as a defender. And TJX, it's worth noting, had not completely ignored those responsibilities—communication between devices on store wireless networks was encrypted (though not well), as was all payment card information stored on the corporate servers since 2004 (fairly well, since the decryption process was estimated at two years), and only authenticated users could access those servers. When the breach was brought to their attention, the company enlisted General Dynamics and IBM, as well as law enforcement officials, to investigate the incident and their efforts were sufficiently effective to scare off Gonzalez from continuing his operation. On the other hand, looking back over the capabilities the thieves exploited, it's easy, in retrospect, to identify several other access defenses that TJX chose not to implement, including protecting store networks by eliminating wireless access, or restricting it to known devices or devices with a shared key, or using stronger WPA encryption, as well as encrypting real-time transaction data, isolating card processing servers, and monitoring exfiltration of data from corporate and store servers.

In the aftermath of the breach, much was made of these missing defenses and TJX faced enormous criticism for its inadequate security. Lawsuits and media reports alike charged that TJX could have prevented the breach had it only implemented better access defenses, but both in print and in court, the company's critics largely avoided the question of which, specifically, of these missing defenses would have been adequate to stop the thieves. For instance, several class action suits were filed against TJX in state and federal courts in Alabama, California, Massachusetts and Puerto Rico, and in provincial Canadian courts in Alberta, British Columbia, Manitoba, Ontario, Quebec and Saskatchewan, on behalf of customers whose transaction data was compromised and financial institutions who issued credit and debit cards used at TJX stores during the breach (*The TJX Companies Inc. Form 10-K*, 2007). The Federal Trade Commission (FTC) summarized several of these criticisms in a complaint alleging that TJX had "failed to provide reasonable and appropriate security for personal information on its networks" because the company:

> (a) created an unnecessary risk to personal information by storing it on, and transmitting it between and within, in-store and corporate networks in clear text;
>
> (b) did not use readily available security measures to limit wireless access to its networks, thereby allowing an intruder to connect wirelessly to in-store networks without authorization;
>
> (c) did not require network administrators and other users to use strong passwords or to use different passwords to access different programs, computers, and networks;
>
> (d) failed to use readily available security measures to limit access among computers and the internet, such as by using a firewall to isolate card authorization computers; and
>
> (e) failed to employ sufficient measures to detect and prevent unauthorized access to computer networks or to conduct security investigations, such as by patching or updating anti-virus software or following up on security warnings and intrusion alerts. (*In the Matter of The TJX Companies, Inc., a corporation, Docket No. C-072-3055*, 2008)

The FTC complaint does not claim that any one of these decisions, individually, would have constituted inadequate security; instead, it emphasizes that these five practices "taken together" were responsible for the allegations against TJX. The complaint does not specify how many—or which—of these practices TJX would have needed to change in order to provide "reasonable and appropriate security," but the strong implication is that the company's failure to do so was not a result of any specific missing defense, or defenses, but rather a consequence of its failing to implement an adequate assortment of defenses. The TJX breach is undoubtedly a case of failed security—the protective measures the company had in place were unable to prevent the thieves from stealing and selling millions of dollars worth of payment card information—but it is not a straightforward story of a company that should have been using WPA encryption or requiring stronger passwords or storing less data. In fact, it's not clear that any of these measures would have succeeded at stopping Gonzalez' team—some of the practices the FTC mentions, particularly with regard to password strength, seem downright irrelevant. It's easy to go down the FTC's list and imagine how Gonzalez and his friends might have bypassed the "reasonable and appropriate" defenses TJX is chastised for not implementing: they could have circumvented WPA encryption by guessing or stealing the password of a store employee; the user passwords they stole from the store's network to access corporate servers would not have been any more difficult to decrypt and use if they were stronger; and much of the card data the team sold was accessed and stolen during current transactions, rather than decrypted from the company's stored, older records.

That does not mean TJX couldn't—or shouldn't—have done more to defend against the capabilities within its scope of control: the ease with which outsiders could connect to its wireless networks, the momentary storage of payment card information in cleartext, the outbound traffic flow to Latvia. But, as access defenses, even measures that targeted those capabilities might well have left the perpetrators room to maneuver and substitute different capabilities—clearly, for instance, the encryption of payment card data was not an insurmountable obstacle for the thieves since they were already planning to decrypt and sell the stolen data, even before they realized that the card numbers were briefly available unencrypted. They didn't necessarily *need* to be able to join an open wireless network or access cleartext card numbers in order to achieve their goal; exporting large volumes of data was a more essential capability for the attackers, though it was largely overlooked in the ensuing legal and media reports which focused primarily on TJX's failure to implement WPA encryption or encrypt data stored prior to 2004. Most essential of all, however, was the capability to turn those stolen card numbers into cash—a process TJX had no insight into, or control over, whatsoever.

4.1.2 Harm Defense

Gonzalez was not particularly worried about TJX interrupting his operation, but there were other defenders whom he viewed with greater trepidation. In an online chat with Yastremskiy on March 2, 2006, Gonzalez wrote:

> [Gonzalez] I hacked [major retailer] and i'm decrypting pins from their stores
> [Gonzalez] visa knows [major retailer] is hacked
> [Gonzalez] but they dont know exactly which stores are affected
> [Gonzalez] so i decrypt one store and i give to you
> [Gonzalez] visa then quickly finds this store and starts killing dumps
> [Gonzalez] then i decrypt another one and do the same
> [Gonzalez] but i start cashing with my guys
> [Gonzalez] visa then finds THAT store and kills all dumps processed by that [major retailer] store
> [Gonzalez] understand?
> [Gonzalez] its a cycle
> [Yastremskiy] yes
> [Gonzalez] this is why i'm telling you to sell them fast fast
> [Gonzalez] also some banks just said fuck waiting for the fraud to occur, lets just reissue EVERY one of our cardholders which shopped at [major retailer] for the last 4 years

Gonzalez knew that his real challenge was not evading TJX defenses but rather evading the credit card providers, like Visa, that had insight into payment card fraud patterns and could tie those cases back to individual retailers. Setting aside the question of whether TJX could have had stronger access defenses in place, the company had no way of knowing what had happened, no means to detect or monitor the harm inflicted on its customers, no visibility into the consequences of its decisions. Like MIT, which often learns about its security breaches from third-party complaints and reports, TJX learned about the breach from a credit card company which had precisely the perspective it lacked to piece together the common link of widespread financial fraud cases.

In fact, the third-party harm defenses implemented by banks, credit card companies, and law enforcement officials in the TJX case were hugely important. It was a credit card company's detection of patterns in fraudulent charges that led to the breach's discovery, regulations surrounding large international financial transactions that forced the perpetrators to repatriate their profits in numerous, smaller increments through use of multiple cashers and couriers, bank restrictions on maximum ATM withdrawals that forced those cashers and couriers to carry suspiciously large numbers of cards (recall that Williams was arrested carrying 80 cards), and credit card expiration policies that forced the thieves to discard the older, unencrypted data stored on TJX's servers and instead spend time decrypting more recent data and finding ways to compromise real-time transactions. Banks, credit card companies, and law enforcement officers exercised considerable control over the extent to which credit card fraud could be carried out using the stolen data, as well as the ease with which Gonzalez and his co-conspirators could reap the profits of those sales. While the barriers they imposed did not prevent the large-scale fraud, these defenses likely limited its scope and certainly forced Gonzalez to make some decisions, such as the recruitment of a network of cashers and couriers, which ultimately contributed to his arrest.

The payment card issuers and banks, responsible for covering the fraudulent charges and replacing their customers' cards, joined the FTC and TJX shoppers in condemning—and suing—the company's security practices, arguing, in particular, that TJX had failed to adhere to the Payment Card Industry Security Standards by using outdated WEP encryption and storing too much data. It's not clear that these would have been especially effective access defenses when it came to thwarting Gonzalez, but that was never the point. The harm defenders, faced with the costs of the large-scale fraud, wanted to blame the access defender for the breach—and vice-versa. After all, TJX was not the only party that could have done more to mitigate that damage caused by Gonzalez. In fact, the line of defense that Gonzalez himself expresses the greatest concern about in his chat logs is the possible pre-emptive cancelation of all credit cards used at a certain retailer by the issuing company. Other harm defenses that could have been implemented by non-TJX defenders include issuing chip and PIN credit cards, which would have required the thieves to acquire not just card numbers but also users' PINs, as well as setting spending limits on all possibly compromised cards (a less drastic pre-emptive measure than mass replacement), and monitoring withdrawals from foreign accounts at U.S. ATMs to make repatriation of profits from overseas sales more difficult. No one besides TJX could have stopped the thieves from accessing the payment card data, but lots of other defenders could—and did—play a role in limiting how much harm could be inflicted using that data, something TJX itself was ill-equipped to do.

TJX certainly played a large role in enabling the success of Gonzalez and his team, but the security controls it could have implemented—shared key authentication, WPA encryption, and data minimization, for instance—were not the lines of defense that Gonzalez was most concerned about. He could find other ways into the network, other decryption methods, and real-time data, but he couldn't do anything about the banks that, in his words, "just said fuck waiting for the fraud to occur, lets just reissue EVERY one of our cardholders." The effectiveness of the defensive measures available to these banks and credit card issuing companies stems both from their broad visibility into financial harm and the specificity of the threat they are responsible for defending against. Payment card fraud may begin with a poorly encrypted wireless network, a compromised point-of-sale terminal, or even a well-worded phishing email—and it is up to access defenders like TJX to protect against all possible access modes—but, ultimately, these schemes all take on a similar pattern as the perpetrators sell their stolen information, relying on the card issuers and processing banks to ensure its value and their profits. In this regard, those card issuers have a significant advantage over TJX when it comes to identifying and stopping financial fraud: they know exactly what the criminals will do because there are a very limited number of ways one can profit from stolen credit card information, even though there may be a great many ways to steal it. Financial theft, and perhaps especially payment card fraud, is notable among malicious motivations for involving an especially consistent and powerful set of third parties. Security incidents intended to cause other types of damage may go through other types of intermediaries, but few with the same degree of concentrated control and small number of alternatives as the major payment card issuers. This, along with the fact that financial fraud as a particularly involved and specific chain of events that need to be successfully undertaken by criminals in order to profit from their activities, makes these incidents easier to defend against, in some ways, than their more varied and flexible counterparts.

The TJX breach is primarily remembered as a devastating failure of computer security—and, indeed, the defenses in place did not prevent the loss of hundreds of millions of dollars—but it was also, in its way, an incredible success story about the identification, arrest, and imprisonment of an international ring of cybercriminals who were caught thanks to many of the constraints imposed on them by that same set of ineffective defenses. The Marshalls wireless network forced the thieves to sit in a parking lot for long periods with laptops and a radio antenna, attracting the attention of the police; selling the stolen data required the involvement of Yastremskiy, who eventually led investigators to Gonzalez' screenname, and the restrictions on international financial transactions meant Gonzalez had to employ cashers and couriers, one of whom would later reveal his identity to the Secret Service. That process took years and had little bearing on the fights playing out in court between access and harm defenders over who was responsible for the breach, despite the fact that it brought to light the men who were truly to blame.

4.2 DigiNotar Compromise (2011)

On July 19, 2011, a routine automated test run by Dutch certificate authority (CA) DigiNotar identified certificates bearing the company's signature that it had no administrative records of having issued. DigiNotar assembled an incident response team and revoked the rogue certificates, believing they had resolved the issue. Just over a month later, on August 27, a Gmail user in Iran posted on Google product forums that he had been blocked from accessing Google Mail by his browser, Google Chrome, because of an invalid certificate issued by DigiNotar. Similar reports surfaced from other users online and on August 29, the Dutch Government Computer Emergency Response Team (CERT) contacted DigiNotar, which promptly revoked the rogue Google certificate. Subsequent investigation revealed many more fraudulent DigiNotar-issued certificates that had gone unnoticed during the earlier July investigation. On September 3, the Dutch government took control of the company and on September 20, less than a month after the first forum postings, DigiNotar declared bankruptcy.

The lapses in DigiNotar's security that ultimately allowed an intruder to issue digital certificates with its signature finished the company, yet the year-long investigation of the incident by Dutch security firm Fox-IT suggested that the CA had taken considerable pains to protect its systems. Computers were kept in rooms requiring biometric scans, signing keys were stored on physical key cards locked in vaults, and the computer networks were segmented according to function sensitivity, with firewalls monitoring and restricting traffic between each zone, as well as an intrusion prevention system monitoring incoming Internet traffic. While there remain some open questions around how exactly the perpetrator compromised DigiNotar's systems and why, Fox-IT's 101-page report on the incident (Hoogstraaten et al., 2012), released by the Dutch government, sheds considerable light on what defensive interventions were used to prevent and mitigate the issuing of rogue certificates and how various holes in those defenses ultimately enabled the compromise.

4.2.1 Access Capabilities & Defense

The report provides a careful reconstruction of how the perpetrator entered DigiNotar's systems and bypassed its security controls. The intruder's first access to the company's network came on June 17, 2011, the investigators found, when two web servers on "the outskirts" of DigiNotar's network (that is, the least protected segment, or external demilitarized zone, which connected to the outside Internet) were compromised. The two servers were, at the time, running an outdated and vulnerable version of web content management system DotNetNuke, which enabled their compromise. An Intrusion Prevention System (IPS) monitored traffic that came through the DigiNotar router responsible for Internet connectivity, but it failed to block the initial intrusion, perhaps in part due to DigiNotar's decisions to run the IPS in its default configuration and position it in front of the company's firewalls where it registered a large number of false positives.

Once inside the external demilitarized zone (DMZ) of DigiNotar's network, however, there were still significant barriers to the production of certificates. Firewalls delineated several separate zones of the network, as shown in Figure 4-1, of which the external DMZ, called DMZ-ext-net, could only send traffic to an internal DMZ (called DMZ-int-net). The two DMZ zones, together, were intended to prevent direct connections between the Internet and the internal DigiNotar network. Neither DMZ could initiate connections to secure-net, where the certificate issuing systems resided. Instead, a secure-net service regularly collected the certificate requests that were sent by customers via the company's website and stored on a server in the internal DMZ. Each request then had to be vetted and approved by two people before being processed by DigiNotar's main production servers, which were located on secure-net and kept in a room protected by numerous physical security measures. The report states:

> This room could be entered only if authorized personnel used a biometric hand recognition device and entered the correct PIN code. This inner room was protected by an outer room connected by a set of doors that opened dependent on each other creating a sluice. These sluice doors had to be separately opened with an electronic door card that was operated using a separate system than for any other door. To gain access to the outer room from a publicly accessible zone, another electronic door had to be opened with an electronic card.

Following his initial compromise of the web servers in DMZ-ext-net, the intruder used these servers as "stepping stones" beginning on June 29 to tunnel traffic between DMZ-ext-net and the internal Office-net. On July 2, connections from the CA servers in secure-net were also initiated to the compromised servers in DMZ-ext-net. That

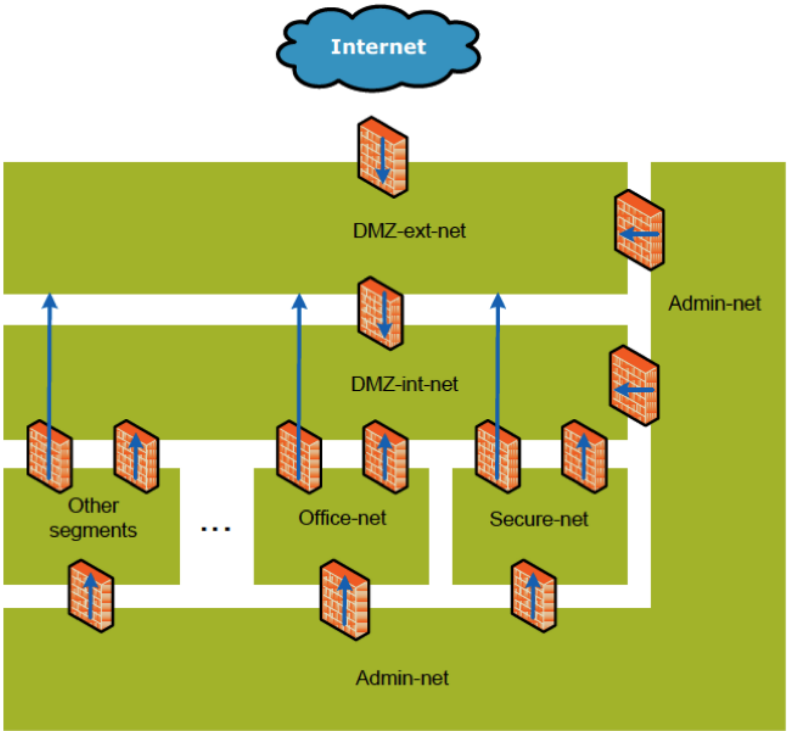


Figure 4-1: Diagram of DigiNotar's network security zones. Source: Hoogstraaten et al. (2012).

traffic was tunneled back to the intruder's IP addresses from DMZ-ext-net. The intruder appeared to have bypassed DigiNotar's security zone firewalls by taking advantage of their numerous exceptions (the Fox-IT investigators identified 156 rules in the firewalls detailing which interconnections were allowed and disallowed between zones). The network tunneling tools he left on the servers were used to set up a remote desktop connection using port 3389, which was then tunneled through port 443, generally used for HTTPS, so that the traffic could get past the firewall.

Having bypassed the zoning firewalls and accessed the secure-net servers, the intruder still could not issue a certificate from a DigiNotar CA server without activating the corresponding private key in the hardware security module (netHSM) using a physical smartcard. The activation process required an authorized employee to insert a smartcard into the netHSM, which was stored in the same highly secured room as the CA servers, and then enter a PIN code. This additional layer of defense, the report authors note, "meant that the unauthorized actions that might have taken place could not have included the issuing of rogue certificates if the corresponding private key had not been active during the period in which the intrusion took place." In their investigation, Fox-IT was unable to determine conclusively which private keys were activated when the rogue certificates were issued or why, since DigiNotar had not kept records of when the smartcards were used.

The investigator's hypothesis for how the intruder overcame this barrier was based on evidence from the server logs that showed some of the DigiNotar CA servers automatically generated Certificate Revocation Lists (CRLs), or lists of serial numbers of certificates that are no longer trusted. Since issuing a CRL also requires the appropriate private key be activated on the netHSM, the investigative team concluded that the private keys of those CA servers that generated CRLs automatically must always be activated, otherwise the servers would be unable to perform that function. Accordingly, any attempts to issue rogue certificates on those servers would be successful, even without physical access to the private key smartcards and netHSM. In other words, a few smartcards appeared to have been left in the netHSM permanently to allow for automatic generation of CRLs.

DigiNotar had in place several lines of access defense intended to prevent intruders from being able to issue DigiNotar certificates—an IPS meant to keep them off the company's network entirely, a set of firewalls intended to further restrict access to the CA servers, a four-eye principle that required certificate requests be approved by two people before being processed, and a physical key card that had to be activated from within a room protected by many other forms of security. The capabilities that the perpetrator exploited—connecting to an outward-facing web server, compromising the out-of-date DotNetNuke software to connect to other servers, navigating the maze of firewalls to find exceptions that would allow tunneling between the different network zones over specific ports, and using the always activated private keys intended to allow automatic generation of CRLs—seem to stem in large part from overly complex and arcane security measures, rather than too little defense.

By contrast to TJX, which simply did not implement several access defenses, DigiNotar actually had a fairly thorough—perhaps too thorough—suite of defenses, but designed those measures to be too complex (156 firewall rules!) or onerous (manual smartcard insertion) for employees to fully understand or comply with. And as with TJX it's not entirely straightforward to pinpoint what DigiNotar could have done by way of access defense to prevent the attack. The company could have updated its web content management software, but presumably DotNetNuke was not the only program on DigiNotar's systems with vulnerabilities; it could have simplified its firewall set-up and pared down the rules governing the partitions between different network segments, but it would still have needed to leave open some way for its secure-net servers to collect certificate requests submitted online; perhaps most compellingly, the company could have abided more strictly by its manual keycard protocol for secure-net servers, instead of leaving some of the keys permanently activated, but there would still have been windows of time when private keys were active to produce legitimate certificates that might have been exploitable. In other words, none of the defense vulnerabilities the perpetrator took advantage of were obviously absolutely essential to his ability to produce rogue certificates, though some (the always-on keycards, for instance) were likely more helpful and less easily replaceable than others (e.g., the DotNetNuke vulnerabilities). The DigiNotar story is a grim one for access defenders; it suggests that even an organization with the best intentions, that has devoted considerable time and energy to its security, is likely to end up with access defenses that are impossible to maintain or understand and, ultimately, inadequate. Hence the need for some degree of reinforcement in the form of harm defense.

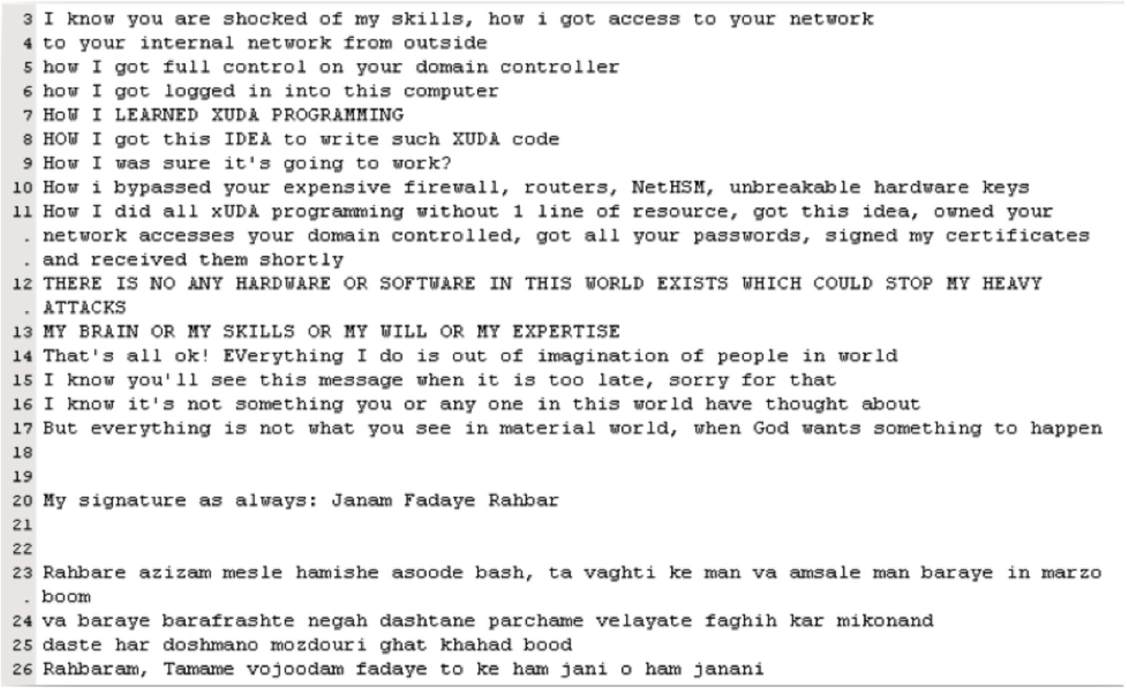
4.2.2 Harm Defense

Access defense in the DigiNotar case is a matter of trying to prevent the creation of rogue certificates, while harm defense entails limiting the extent to which those certificates can be used for malicious purposes. So to understand the role of harm defense in the DigiNotar breach we first need to know how the rogue certificates were intended to be used. Since the perpetrator has never been apprehended, his motivation for infiltrating DigiNotar and issuing hundreds of certificates remains somewhat unclear, as does the extent to which he was successful in achieving his ultimate aims. The only known significant consequence of the breach, beyond the damage done to DigiNotar, was a man-in-the-middle (MITM) attack that redirected visitors trying to access Google websites from 298,140 unique IP addresses, 95 percent of which originated from Iran. While any Google service could have been the focus of this redirection, and no specifics about what the users were trying to do or what happened on the websites they were redirected to can be gleaned from the rogue certificate logs, the Fox-IT report speculates as to the possible purposes of this attack:

> Most likely the confidentiality of Gmail accounts was compromised and their credentials, the login cookie and the contents of their e-mails could have been intercepted. Using the credentials or the login cookie, an attacker may be able to log in directly to the Gmail mailbox of the victim and read their stored e-mails. Additionally, the MITM attacker may have been able to log into all other services that Google offers to users, such as stored location information from Latitude or documents in GoogleDocs. Once an attacker is able to receive his targets' e-mails, he is also able to reset passwords of others services such as Facebook and Twitter using the lost password functionality.

So there is some reason to believe that the attacker's objective may have been to gather information about the email, location, and other online activity of Iranians but very little indication of what he intended to do with that information. In fact, the only clue left by the intruder—a message saved on a DigiNotar server, shown in Figure 4-2—suggests that the compromise was intended primarily to demonstrate his skills rather than to drive any specific malicious mission. "THERE IS NO ANY HARDWARE OR SOFTWARE IN THIS WORLD EXISTS WHICH COULD STOP MY HEAVY ATTACKS MY BRAIN OR MY SKILLS OR MY WILL OR MY EXPERTISE," he boasts in the message.

The motivation matters because the rogue certificates, on their own, cause no harm (except perhaps to DigiNotar)—like the stolen credit card numbers from TJX, the certificates are only harmful insofar as they can be used to steal from or spy on people, or otherwise disrupt their lives. If the sole aim of a threat actor is to show off then, in one sense, that actor succeeds just by managing to breach the protected system's defenses but, in another sense, that success hardly matters. The DigiNotar breach matters in part because there is some reason to believe that harm was inflicted as a result—it appears that at least part of the perpetrator's end goal was spying on Iranians, though the precise nature and extent of that espionage, as well as



```
I know you are shocked of my skills, how i got access to your network
to your internal network from outside
how I got full control on your domain controller
how I got logged in into this computer
HoW I LEARNED XUDA PROGRAMMING
HOW I got this IDEA to write such XUDA code
How I was sure it's going to work?
How i bypassed your expensive firewall, routers, NetHSM, unbreakable hardware keys
How I did all xUDA programming without 1 line of resource, got this idea, owned your network accesses your domain controlled, got all your passwords, signed my certificates and received them shortly
THERE IS NO ANY HARDWARE OR SOFTWARE IN THIS WORLD EXISTS WHICH COULD STOP MY HEAVY ATTACKS
MY BRAIN OR MY SKILLS OR MY WILL OR MY EXPERTISE
That's all ok! EVerything I do is out of imagination of people in world
I know you'll see this message when it is too late, sorry for that
I know it's not something you or any one in this world have thought about
But everything is not what you see in material world, when God wants something to happen


My signature as always: Janam Fadaye Rahbar


Rahbare azizam mesle hamishe asoode bash, ta vaghti ke man va amsale man baraye in marzo boom
va baraye barafrashte negah dashtane parchame velayate faghih kar mikonand
daste har doshmano mozdouri ghat khahad bood
Rahbaram, Tamame vojoodam fadaye to ke ham jani o ham janani
```

Figure 4-2: Screenshot of message left on DigiNotar's computers by the perpetrator of the CA's compromise.

its motivation, remain uncertain. Perhaps even more than that, the DigiNotar breach matters because of the potential harm that could have been caused by an unauthorized party with the ability to issue limitless trusted certificates. Online government services and financial sites might have been affected (cia.gov and Equifax.com were among the domains for which rogue DigiNotar SSL certificates were issued)—though there is no evidence to suggest that they were. It is possible to interpret the relatively minimal damage inflicted using DigiNotar rogue certificates as the deliberate choice of an attacker interested more in the access component of breaches than the infliction of actual harm—but it is also possible to interpret it as a triumph of harm defense.

Though several components of harm defenses relevant to the DigiNotar breach lay outside DigiNotar's control, as is typical of harm defense, the company did employ one security mechanism to check after-the-fact for the creation of rogue certificates. Besides the layers of access defense intended to prevent intruders from accessing its main production servers and issuing certificates, DigiNotar also ran regular, automated tests to confirm that it had records of issuing every certificate listed in its database of serial numbers—it was this test that first detected the existence of some rogue certificates in mid-July and led to their revocation. But while it detected several rogue certificate serial numbers for which there were no associated administrative records, the test failed to find and revoke many others, likely because the intruder was able to tamper with the database of serial numbers that the test verified. The Fox-IT investigation recovered versions of the serial_no.dbh database that had been removed from the DigiNotar servers and contained additional, unknown serial numbers—several of which corresponded to rogue certificates—that had been deleted from DigiNotar's records and were therefore not detected by the automated test.

Using these certificates to spy on Iranians—if that was, indeed, the end goal—required going through other parties besides DigiNotar. Digital certificates bind a public key to a particular entity using the digital signature of a trusted third party, such as DigiNotar. These signed certificates may be used for a variety of different purposes, to validate anything from a website to a software package to an individual. Rogue certificates can allow for impersonation of any of these entities, bypassing DigiNotar's vetting and approval process. So malicious websites can masquerade as google.com, malicious individuals can take on the identities of other people, malware can be attributed to legitimate software companies—all with the (unwitting) endorsement of DigiNotar, in the form of its digital signature. That signature ensured, for instance, that many web browsers, operating systems, and document readers—as well as many Dutch government services—would automatically trust the identity of anything bearing a DigiNotar certificate. In other words, the value of the rogue certificates—and their capability to cause damage—derived not from the certificates themselves but rather from the trust placed in them by numerous outside parties.

These actors—the operating systems and browsers and other applications that rely on certificates—therefore present another line of defense against rogue certificates, particularly if they have their own means of verifying certificate validity, independent of DigiNotar, as Google Chrome did. Since Google operates both a browser and a number of online services, its browser knows exactly what certificates the company holds for the domain google.com. These certificates are "pinned" in the Chrome browser, meaning that no other certificates for the google.com domain are accepted by the browser, regardless of whether they are signed by trusted Chrome CAs like DigiNotar. This enabled Chrome to warn users about malicious websites that used the rogue google.com certificate, and subsequently prevent them from accessing those sites. These warnings were what first tipped off users—as well as DigiNotar—to the existence of the rogue google.com certificate, leading ultimately to the company's public acknowledgment of the breach, the subsequent investigation, and the discovery of many other previously undetected fraudulent certificates.

Certificate pinning is somewhat limited by the extent to which individual entities, like Google, control both the major platforms that operate certificate verification as well as content that runs on those platforms. Since Google runs both Chrome and Gmail, it can tell its browser exactly which certificates should be trusted when attempting to access its email service—but it's unlikely to have that information for every website. Similarly, a company like Microsoft that develops both an operating system and software that runs on that operating system, may be able to dictate exactly which certificates the operating system should trust to indicate software that it developed—but it can't easily do the same for the software developed by other entities that may be downloaded onto machines running Windows. The visibility of individual defenders is crucial here—as it was in the TJX breach—to notice the harm inflicted with a rogue certificate, a defender needs to know both what certificate is being used for a service and what certificate should be used for that service. DigiNotar has no way of knowing the former, and most of the time browsers have no way of knowing the latter (other than to trust CAs). Just as the credit card companies' visibility into payment card fraud patterns brought the TJX breach to light, Google's broad visibility into not just a browser but also a suite of certificate-backed web services was essential for identifying the harm components of the DigiNotar breach—and very few defenders possess a comparably wide window into the certificate ecosystem.

Browsers were one possible place the DigiNotar intruder could have been—and ultimately was—thwarted in his attempts to spy on Iranian users. Those attempts relied on his being able to convince the users of Google services that they were checking their email (or searching for directions or watching videos) when, in fact, they were actually visiting a malicious site controlled by him. The DigiNotar certificate, trusted by all major browsers, would serve to persuade users that his site was legitimately operated by Google—but first he had to get them onto his site, otherwise the certificate was useless. This meant redirecting Iranian Internet traffic so that users who attempted to visit Google websites were redirected to the malicious sites without their knowledge—and this process involved yet another group of actors who had the potential to help prevent the perpetrator's success, even after he had completely penetrated DigiNotar's systems and successfully produced rogue certificates.

There are a few different ways to perpetrate an SSL MITM attack that redirects users' online traffic. One is to intercept traffic at users' upstream provider by inspecting their packets and redirecting those intended for certain destinations. In this case, the service provider would either have to be complicit in the attack or be extensively compromised, in which case the provider's security measures would offer another crucial line of defense. It is unlikely that this was the approach used by the DigiNotar intruder, however, since 5 percent of the IP addresses that were affected by the rogue google.com certificate originated outside of Iran—many of them corresponding to dedicated proxies, Tor, and VPN exit nodes. If the redirection had been done by (or through) Iranian service providers, these users would likely not have been affected. Another possibility is that the attacker altered the records belonging to a high-level DNS server in order to redirect visitors to certain domains. Since Tor and VPN users still query local DNS servers by default, this approach might explain why those users were impacted. However, the requests to validate the rogue certificate were extremely bursty according to Fox-IT's analysis—that is, at certain times the certificate's use would spike dramatically and then decline, over and over again. Had a high-level DNS server been responsible for the redirection, the requests to validate the rogue server should have instead increased steadily over the course of the attack, rather than rising and falling repeatedly. Therefore, the investigators conclude, the MITM attack was most likely carried out by DNS cache poisoning, or flooding DNS servers with forged responses for a certain domain, pretending they were sent by a higher-level DNS server. This technique "poisons" the targeted record for some period of time, until the responses expire and the DNS server makes another request to a higher-level server. The brief lifetime of these poisoning attacks might explain the erratic up-and-down volume of requests to validate the rogue google.com certificate over time, and would also explain the traffic from proxies, Tor, and VPNs. Redirecting traffic in this manner introduces a new set of defenders who can help protect against the successful exploitation of rogue SSL certificates: DNS operators. For instance, by implementing DNS Security Extensions (DNSSEC) to verify the senders of DNS records they receive, or disregarding records that are received in the absence of a particular query, DNS operators might reduce the ease with which someone could perpetrate a cache poisoning MITM attack.

Rogue certificates can be used for other purposes besides the imitation of existing websites—and even when used for that purpose, their creators can try to attract visitors by other means than MITM attacks. For instance, the URL of the malicious site could be distributed via email, online ads, or social media. In such cases, the lines of defense proposed for DNS operators would be useless—though browsers might still play an important role in trying to detect and block visitors to such sites by revoking their trust in the CAs issuing certificates to malicious sites. This was precisely what the major browsers did in the wake of the DigiNotar compromise: remove the company's root CA from their list of trusted CAs on the grounds that if someone had successfully issued one rogue DigiNotar certificate there might well be more—as indeed there were. Because there are a relatively small number of commonly used browsers, this was a fairly straightforward and quick way to mitigate the threat of rogue certificates for a large portion of Internet users. Browser operators retain significant discretion over what this revocation means for end-users and whether individuals will they be completely unable to access sites signed by revoked certificates or merely presented with a warning and then given the option of continuing to sites regardless.

Whatever role DNS operators and browsers may have played in defending against the DigiNotar intruder's espionage agenda, the security efforts of the CA itself undoubtedly drew the greatest scrutiny following the compromise. In particular, the investigators expressed concern that the intruder had installed tools to extract and crack password hashes and that, using these stolen credentials, the intruder had gained full administrative rights enabling him to delete and manipulate logs and databases. "The logging service for the CA management application ran on the same CA servers that were compromised by the intruder," the report states, adding that "database records on these CA servers were deleted or otherwise manipulated."

The report also proposes an alternative to using the serial_no.dbh database, stored on the CA servers, as the catalog of serial numbers that CA testing routinely verifies have been issued by checking against administrative records. Instead, the Fox-IT investigators recommended that the Online Certificate Status Protocol (OCSP) requests sent to a CA be used to accumulate a list of the serial numbers certified by the CA. OCSP requests allow users to verify whether specific certificates have been revoked by sending the serial number to the issuing CA, whose OCSP responder then checks the revocation status and sends it back to the requesting user in a signed OCSP response. DigiNotar's OCSP responder logs were what ultimately enabled Fox-IT to estimate the number of IP addresses that had confirmed the rogue google.com certificate's serial number, and also to assess the extent to which any of the other rogue certificates had been used. The investigators also relied on the OCSP responder logs to check whether additional, unknown certificate serial numbers were being verified by DigiNotar at the time of the compromise.

The "lessons learned" section of the DigiNotar investigation report emphasizes the importance of separating logging services from other parts of the system and separating critical systems from those that perform less critical processes or connect to the Internet. Yet, to some extent, this was already the approach taken by DigiNotar in dividing up its network according to function, and using firewalls to ensure unidirectional security gateways between the different segments. That doesn't mean DigiNotar had model security by any stretch. Like TJX, the company made some mistakes that, at least in retrospect, appear fairly glaring. Also like TJX, DigiNotar had very little visibility into the harm that was being inflicted due to its failed access defenses, and was forced to rely primarily on third parties both to detect and mitigate that harm. Despite having arguably much stronger defenses than TJX, and causing far less damage, DigiNotar paid much more dearly for its security lapses—its failed attempts at access defense were construed as grounds for going out of business, rather than an indication that stronger protections were needed to mitigate the risks posed by rogue certificates.

4.3 PLA Unit 61398 Espionage (2013)

It is unusual for the victim of a serious security breach to reveal the deals publicly—the decision in 2012 to release the DigiNotar report was ultimately made by the government, after the company had already folded, and for that reason made it possible to gain a rare insight into the mechanics of a specific breach. A report released the following year by security firm Mandiant detailed a series of espionage incidents it investigated, all of which it believes to have been perpetrated by Unit 61398 of the Chinese People's Liberation Army (PLA). By aggregating information about multiple incidents, Mandiant was able to conceal the identities of its clients, as well as the specific breaches they experienced, and offer another unusually comprehensive step-by-step deconstruction of an actual set of security incidents.

This incident aggregation makes it difficult to pinpoint specific motivations driving the perpetrators, or the actual harm they inflicted. The breaches were intended "to steal data, including intellectual property, business contracts or negotiations, policy papers or internal memoranda" that could be used, generally, to benefit the Chinese government and Chinese state-owned businesses, according to Mandiant's analysis (Mandiant, 2013). The data stolen by PLA Unit 61398 included information related to product development, designs, and manuals, as well as manufacturing procedures, processes, and standards, along with business plans, legal documents, records detailing contract negotiation positions, mergers, and acquisitions, meeting minutes and agendas, staff emails, and user credentials. The targets—primarily U.S.-based organizations—spanned 20 different sectors, though many victims were concentrated in the areas of information technology, satellites and telecommunications, aerospace, and public administration.

Given the large number of targets and the variety of different industry sectors they span, this stolen information could have served a number of different functions—revealing anything from how proprietary products were developed to sensitive financial statements—so it is difficult to say precisely what advantage the perpetrators gained by stealing it, or how they may have used it to harm others. The only specific example of harm included in the Mandiant report is an incident in which China negotiated a significant reduction in the price of a major commodity with a wholesale firm whose networks were compromised at the time. The report authors note: "This may be coincidental; however, it would be surprising if [Unit 61398] could continue perpetrating such a broad mandate of cyber espionage and data theft if the results of the group's efforts were not finding their way into the hands of entities able to capitalize on them."

Though the Mandiant report gives little detail about the ultimate use of the stolen information, it offers considerable insight into how that data was retrieved, describing several stages common across the incidents, shown in Figure 4-3, and how each was carried out. As is often the case in retrospective analysis of computer security incidents, the primary emphasis in this model is on access capabilities and defense opportunities. The harm infliction stages, encapsulated in Mandiant's vague "complete mission" phase, are largely glossed over—as are the specific harms themselves. This forces us to reframe our ideas about what access and harm defense look like, and particularly which of those forms of defense is most reliant on third-party defenders. The TJX and DigiNotar breaches offer glimpses into the ways that third parties can assist with harm defense, but the PLA Unit 61398 espionage cases suggest that for some times of espionage the opposite may be true: third-party defenders may have a vital role to play in restricting attackers' access capabilities.

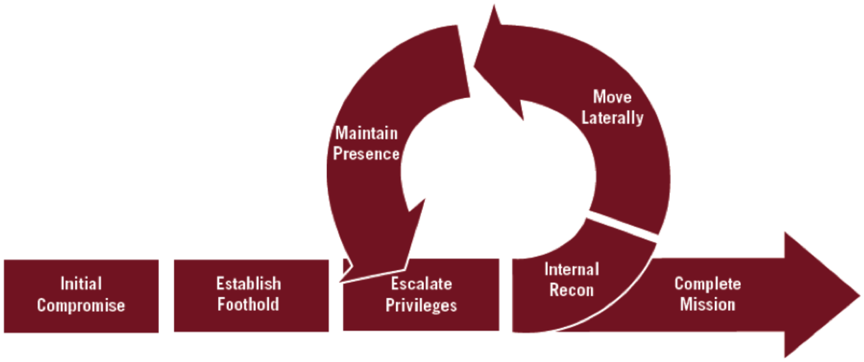


Figure 4-3: Mandiant's Attack Lifecycle Model. Source: Mandiant (2013).

4.3.1 Access Capabilities & Defense

Despite the variety of targets, the initial access steps took the same form in almost all of the incidents Mandiant investigated: phishing emails containing either a malicious attachment or hyperlink. To ensure recipients downloaded these attachments and followed these URLs, the perpetrators sent the emails from accounts created under the names of real employees at the company. They also disguised the attachments with appropriate file titles (e.g., 2012ChinaUSAviationSymposium.zip, Employee-Benefit-and-Overhead-Adjustment-Keys.zip, and North_Korean_launch.zip) and, in some cases, changed the attachment icons and names, hiding the actual file extensions from the recipient. When downloaded, these files established backdoors in the victims' systems that initiated outbound connections to command and control (C2) servers operated by the perpetrators. In some cases, these backdoors retrieved webpages from C2 and interpreted the data between special HTML tags in the page as commands, in others, the infected machines communicated with C2 servers directly, rather than through webpages, often using SSL encryption to shield those communications. Just these first two stages of Mandiant's attack lifecycle ("initial compromise" and "establish foothold") already implicate a variety of different defensive interventions (and associated defenders), ranging from the design of email and the treatment of attachments to the registration of domain names and regulation of outbound traffic.

Once those outbound connections were established between the backdoor and a C2 server, the intruders explored the network structure and searched for credentials stored on the breached system that would allow them to connect to shared resources, access protected servers, and execute commands restricted to system administrators. Mandiant notes that the PLA Unit 61398 intruders aimed to establish multiple means of entry into the systems they targeted. This way, even if the victim discovered and mitigated the initial backdoor delivered via e-mail, the intruder still had other available means of access and exfiltration. This was accomplished by establishing additional backdoors in different places on the compromised system, and using stolen credentials to log in through the target organizations' own VPNs and web portals. The use of stolen credentials made it difficult to distinguish between legitimate and malicious activity, forcing defenders to rely on the volume and sensitivity of outbound traffic to identify espionage. Outbound connections to C2 servers, on the other hand, could offer some other clues—for instance, if the connections are made to suspicious locations or on a suspiciously regular, unchanging schedule. But these clues, too, could be obscured by intruders—the intruders, whom the investigators ultimately trace back to networks in Shanghai, typically compromised third-party servers and used those servers to communicate with the targets of their espionage indirectly. This implies yet another set of potential intermediary defenders—the owners and operators of the infected third-party servers that are used as platforms for espionage attacks.

As before, there are some differences between what the defensive roles played by direct targets of security incidents and other, third-party defenders. The defensive mechanisms that can be implemented by the targets of espionage are primarily aimed at detecting or interrupting the exfiltration of sensitive information from their systems. Third-party defenders, who are not directly targeted by the espionage efforts, may be less able to disrupt the information theft directly, but may, in some cases, have opportunities to go after the perpetrators' infrastructure and profits. For instance, the Mandiant report notes that PLA Unit 61398 has a vast infrastructure supporting its espionage efforts which includes over a thousand servers, several encryption certificates, hundreds of domain names registered by the intruders themselves, and many more domains owned by compromised third parties. These domain names play a vital role in the group's espionage efforts, operating both as delivery vectors for initial backdoors via phishing emails and as embedded C2 server addresses in those backdoors. The report notes that by using domain names instead of specific IP addresses as C2 address, the intruders "may dynamically decide where to direct C2 directions from a given backdoor" so that "if they lose control of a specific hop point (IP address) they can 'point' the C2 FQDN address to a different IP address and resume their control over victim backdoors."

These domain names may sometimes present opportunities for additional defensive intervention, especially when they are designed to imitate a trusted third party (e.g., microsoft-update-info.com, cnndaily.com, and nytimesnews.net). In September 2011, Yahoo! took issue with one such domain, myyahoonews.com, and filed a complaint against the person who, using the name zheng youjun, had registered it with GoDaddy.com. The registrant did not respond to the complaint, and the National Arbitration Forum subsequently ruled that the domain be transferred from its current owner to Yahoo! following an investigation that showed it was being used as a "phishing web page . . . in an effort to collect login credentials under false pretenses" (*Yahoo! Inct v. zheng youjun, claim number: FA1109001409001*, 2011). This suggests another potential role for third parties in targeting the deceptive resources used for espionage, besides trying to identify and remediate infected machines that perpetrators route their intrusions through.

These roles for third-party defenders stand in stark contrast to the ones implied by the TJX and DigiNotar incidents. In those cases, third parties played vital harm defense roles—they had visibility into the harm being inflicted, and an ability to control some of the most essential stages of attacks, where perpetrators options had drastically narrowed. Moreover, that visibility and power was concentrated in the hands of a relatively small group of actors with broad global reach (e.g., credit card companies, browser manufacturers) who could conceivably coordinate their defensive efforts. The PLA Unit 61398 strategy relies in part on the diversity and dispersion of potential defenders across their espionage infrastructure. These potential third-party defenders have, in general, less visibility into the espionage efforts than the actual targets and can exercise control over the most replaceable stages of those attacks—myyahoonews.com can be replaced with another phishing site, and compromised hosts are presumably even more easily replaced. These are traits typical of access defenses—the capabilities they block are easily replaced and do not map directly onto blocking any class of harm—only in this case they apply to third parties rather than the victims themselves, placing the victims in the position of dealing directly with harm defense.

4.3.2 Harm Defense

Deconstructing the harms imposed by Unit 61398's espionage is tricky, given the limited detail in Mandiant's report. More generally, designing defenses to mitigate the harms imposed by cyberespionage, rather than the espionage itself, is tricky given how many different ways stolen information can be used—the intruders themselves may not even know beforehand exactly what information they will turn up or how they will use it. So the safest strategy in defending against espionage is to assume that the exfiltration of sensitive data is, itself, a form of harm. In certain cases, particularly when espionage efforts are geared towards the theft of stolen intellectual property, there may be some means of trying to limit those illicit profits through economic restrictions. For instance, in May 2013 a group of U.S. senators introduced the Deter Cyber Theft Act, aimed at fighting espionage efforts by blocking U.S. imports of products that benefitted from stolen intellectual property. But policy-based measures along these lines are limited by the extent to which foreign companies conducting espionage require the economic support of international customers, and their effectiveness depends entirely on being able to reliably identify the perpetrators of espionage—a problem the Deter Cyber Theft Act sidestepped by delegating that function to the Director of National Intelligence.

Policy may have some role to play in defending against certain forms of espionage, but for the most part victims are on their own when it comes to harm defense. They have two general means of trying to defend against the diverse harms posed by espionage: preventing the exfiltration of data from their systems and preventing the people who take that data from being able to use it. The first form of harm defense—monitoring and limiting outbound traffic to detect unwanted exfiltration—was considered an access defense in the TJX case, but only because of the specific nature of the data being stolen; nothing harmful could be done with the TJX transaction data until it could be processed through particular black market channels that offered additional avenues for defense. By contrast, the data stolen through espionage efforts like those undertaken by Unit 61398 offers no equally clear roadmap for how the thieves will act on it, so it is necessary to assume that their simply acquiring the information may, in itself, be harmful. Accordingly, restrictions on outbound data flows become a harm defense—and measures like encryption, or even planting false files, can serve to make stolen data more difficult for attackers to use, providing another layer of harm defense.

As with all digital harms, the harm defense options are much more limited because they are restricted to the targeted computer system. This also means that espionage targets have fewer third parties to rely on for harm defense—they, and likely they alone, have visibility into the infliction of espionage harm both because it is happening on systems they own and operate and because it is being inflicted on them. On the one hand, this means the targets have more capabilities and incentives to defend themselves; on the other, it means they have fewer lines of defense implemented by others to fall back on. In this regard the Unit 61398 espionage incidents are strikingly different from the TJX and DigiNotar breaches—both TJX and DigiNotar were victims of those breaches, but they were, in some sense, intermediate victims, the real targets were TJX customers and Iranian Google users. The organizations targeted by Unit 61398, by contrast, seem to bear most of the ill effects of the espionage efforts themselves, while compromised third-party hop-points and impersonated domains serve as intermediate victims. In other words, TJX and DigiNotar are primarily doing access defense, while the targets of Unit 61398 find themselves responsible for harm defense.

4.4 Spamhaus Denial-of-Service Attacks (2013)

Security incidents perpetrated for the purpose of espionage or financial fraud are both fundamentally concerned with theft—the theft of information and, in many cases, the money that can be obtained using that information. These types of incidents therefore present several opportunities for defense that focus on the final stages of theft, on preventing information or money from being taken from its rightful owners, or at the very least limiting the volume and duration of such thefts. But for security incidents that do not center on theft—incidents whose perpetrators seek only to inflict harm on the victims instead of taking anything for themselves—such defensive measures are irrelevant. For instance, the massive distributed denial-of-service (DDoS) attacks launched against anti-spam organization Spamhaus in March 2013 were apparently motivated neither by espionage or financial theft but instead intended as retaliation against the organization for blacklisting Dutch hosting company Cyberbunker. Spamhaus engaged the security firm CloudFlare to help it mitigate the DDoS attacks. With Spamhaus' permission, CloudFlare later published two blog posts detailing its defensive efforts and noting that its clients are usually "reluctant" to talk about the details of the attacks they face. "It's fun, therefore, whenever we have a customer that is willing to let us tell the story of an attack they saw and how we mitigated it," CloudFlare cofounder Matthew Prince writes of Spamhaus (2013b). It's also a rare window into the specifics of how such attacks are both carried out and combatted, as well as the range of possible lines of defense.

These lines of defense, by necessity, operate very differently from those used to mitigate espionage and fraud efforts since there are no data exfiltration or financial theft stages to interrupt. Instead, defensive efforts have to be pushed up to the very earliest stages of an incident because there are no later stages to fall back on—the damage, such as it is, is done almost in the same instant that communication with or access to the victim is achieved. Not all such incidents are denial-of-service attacks—tampering or deleting data, and defacement or redirection of a website might also be driven by similar motives—and not all denial-of-service attacks are intended solely to disrupt service, occasionally they are used for financial gain as a means of extorting money from victims. Still, denial-of-service attacks offer a useful model for thinking about how to defend against attacks that essentially begin and end with harm—rather than building to it through a series of spread out intermediate steps.

On March 18, 2013, the Spamhaus Project, a nonprofit organization that compiles and distributes lists of DNS servers, IP addresses, and domains known to be used by spammers, began experiencing an unusually large volume of traffic, around 10Gbps, to their website. The traffic saturated the organization's Internet connection, taking their site offline. The next day, Spamhaus contracted CloudFlare's services to help mitigate the attacks and CloudFlare used its 23 data centers to absorb and filter the traffic directed at Spamhaus. CloudFlare directed all traffic intended for Spamhaus to one of its data centers and then passed on to Spamhaus only the traffic that appeared to be legitimate. This required being able to distinguish between the malicious traffic and legitimate requests to Spamhaus' servers, a distinction that is not always clear when dealing with denial-of-service attacks routed through a large number of compromised machines. However, the techniques for perpetrating the largest such attacks are usually the ones that make it easiest to detect and filter malicious traffic. For instance, some DDoS attacks involve using compromised machines to issue a large volume of requests to the target's servers. In these cases, it can be very difficult to distinguish between the malicious and legitimate traffic until it is possible to identify which machines have been compromised, for instance, by observing which ones make unusually large number of requests. DDoS attacks that take this approach can reach a significant scale, but they are limited in size by the number of machines that their perpetrators have control over. They are therefore highly susceptible to third party efforts to detect and remediate compromised machines, and may be expensive to launch, requiring perpetrators to rent out extremely large botnets if they wish to incapacitate powerful targets with substantial connection capacity.

The DDoS attacks aimed at Spamhaus, however, were primarily comprised of DNS reflection traffic, in which the attackers used their compromised machines to issue queries to open DNS resolvers that appeared to come from a Spamhaus IP address. These DNS resolvers, in turn, responded to each query with a large zone file—each about 3,000 bytes, or 100 times larger than the 36-byte queries they were issued in response to—and sent these files not to the actual machines that generated the queries but instead to the spoofed IP address for Spamhaus in the queries. This process is illustrated in Figure 4-4.

Using these methods, the attackers were able to generate up to about 90Gbps of traffic directed at Spamhaus—much more than they likely would have been able to control using only the compromised machines, without the DNS amplification factor of 100. But the nature of the attack also changed the defensive landscape—and not just because of its size. In a more traditional DDoS attack, that is, one that does not make use of DNS queries, there are relatively few defensive options: either the target can filter traffic it receives (or hire someone else like CloudFlare to do it for them) and try to identify malicious packets by detecting high-volume senders or suspicious patterns, or the owners of the compromised machines sending that malicious traffic may notice (or be informed of) the large volume of outbound traffic and patch their systems. Moving earlier up the attack chain, it may be possible to go after the actors renting out botnets, but much of the defensive responsibility is likely to ultimately fall on the targets and the machines directly bombarding them with traffic.

4.4.1 Access Capabilities & Defense

By introducing the DNS as an intermediary for sending that traffic, attackers can greatly increase the volume of such attacks but they also create a new defensive opportunity for DNS operators. These operators can restrict which queries their DNS resolvers respond to, so that queries from unknown or unauthorized machines are ignored. When operators don't do this, leaving DNS resolvers "open" to respond to any query from anyone, it makes it much easier for DDoS attackers to use those resolvers for amplification because they can effectively generate queries from any compromised machines they have control over. Another option open to DNS operators is to rate limit their resolvers, rather than closing them completely to the public, limiting the amount of traffic that can be sent from them and therefore their value to a DDoS attacker. Following the Spamhaus attacks, in fact, the Open Resolver Project publicly released a list of 21.7 million open resolvers online in hopes of pressuring their operators to shut them down or further restrict them. (Recall that as part of their

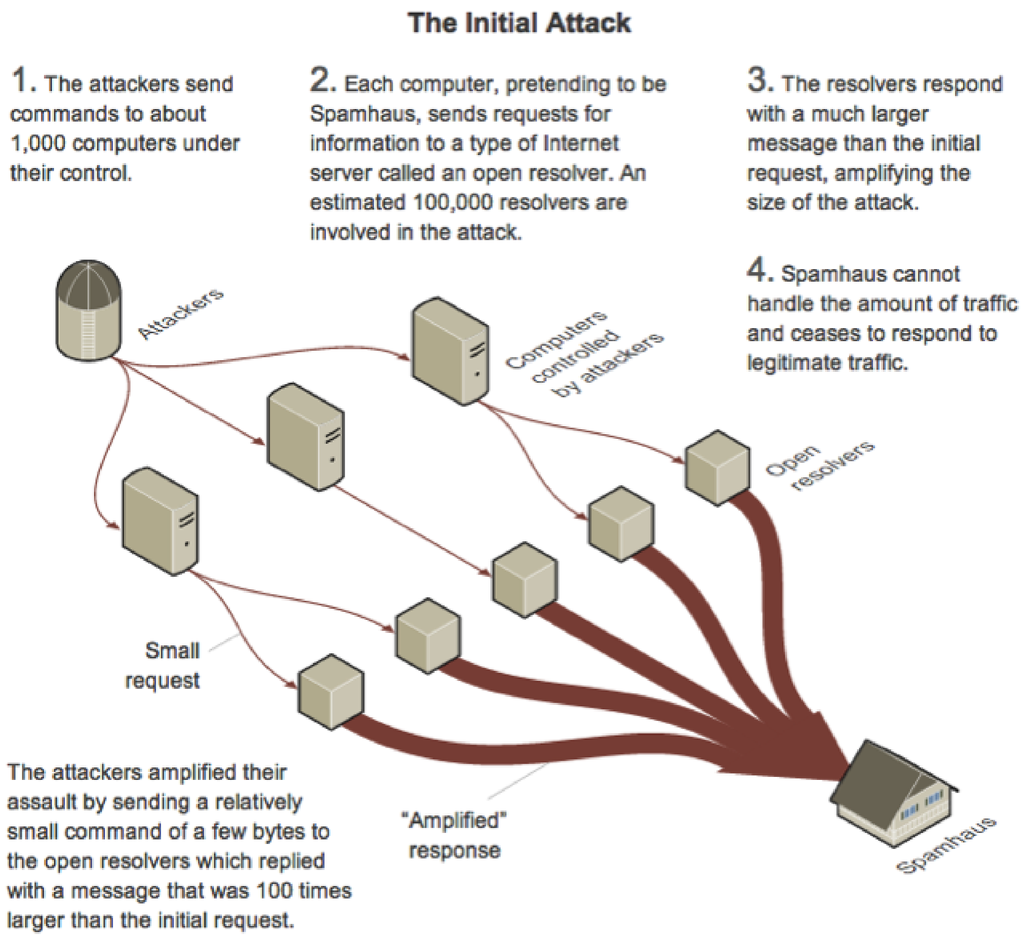

The Initial Attack
1. The attackers send commands to about 1,000 computers under their control.
2. Each computer, pretending to be Spamhaus, sends requests for information to a type of Internet server called an open resolver. An estimated 100,000 resolvers are involved in the attack.
3. The resolvers respond with a much larger message than the initial request, amplifying the size of the attack.
4. Spamhaus cannot handle the amount of traffic and ceases to respond to legitimate traffic.
Attackers
Computers controlled by attackers
Open resolvers
Small request
The attackers amplified their assault by sending a relatively small command of a few bytes to the open resolvers which replied with a message that was 100 times larger than the initial request.
"Amplified" response
Spamhaus

Figure 4-4: The design of the DDoS attacks directed at Spamhaus. Source: Markoff and Perlroth (2013).

2013 security changes, MIT stopped operating open resolvers.)

In addition to introducing a new defensive intermediary, DDoS amplification attacks also make it easier for targets to distinguish malicious and non-malicious traffic—in standard DDoS attacks both types of traffic may be of a similar nature, but in DNS amplification attacks the malicious traffic is likely to be of a very specific type (large DNS records) that can be recognized and dropped without affecting legitimate users. Another type of traffic involved in the attacks on Spamhaus was generated by an ACK reflection attack, in which the compromised machines initiated TCP sessions ostensibly from a Spamhaus IP address and the receiving servers therefore responded to Spamhaus with an ACK (acknowledgement) connection, acknowledging receipt of the session initiation. This model does not involve the amplification effect of the DNS queries but still helps with identifying malicious traffic, since targets can simply drop all unmatched ACKs that they know they did not request. Both DNS amplification and ACK reflection attacks however rely on the ability of senders to spoof the originating IP address of any traffic they send, however. Without being able to indicate that the DNS queries and TCP session initiation requests come from Spamhaus, these methods are of no use to the attackers. One possible defensive approach is therefore to target this spoofing of sender IP addresses by ingress filtering, or forcing routers to check source IP addresses of packets they receive. However, this defense would still leave open the potential for DDoS attacks in which compromised computers query targets directly.

Leveraging their capacity to collect and filter enormous volumes of traffic, CloudFlare helped Spamhaus mitigate the attacks for two days until they appeared to cease on March 21. On March 22, they resumed—but were no longer directed at Spamhaus. Instead, the attackers shifted their focus to the providers and Internet exchanges CloudFlare connects to. This change in targets shifted the defensive responsibilities upstream, forcing CloudFlare's bandwidth providers to filter traffic and, reportedly, causing some collateral congestion for other users (Prince, 2013a). As the attack—still, ostensibly, aimed at punishing Spamhaus for its listing of Cyberbunker—shifted focus, it again introduced new potential defenders and mitigation measures. For instance, CloudFlare, in its subsequent analysis, suggested two specific defensive steps that Internet exchanges could take to help mitigate such attacks—not announcing the IP addresses that Internet exchange members use to interchange traffic, and only permitting the receipt of traffic sent by those members.

4.4.2 Harm Defense

Access defenses, which quickly bleed into harm defenses when it comes to denial-of-service attacks, are intended to make it harder for attackers to generate large volumes of traffic. Harm defenses, then, are focused on preventing those large volumes of traffic from reaching, or incapacitating, their targets. This was the role CloudFlare played in the attacks on Spamhaus, absorbing the malicious traffic and filtering it before it hit Spamhaus' servers. It's a model built on the principle of third-party harm defense, the idea that protecting against harm is something victims don't have the visibility or resources to do themselves and must therefore rely on—or in Spamhaus' case actively insert—other intermediary parties to identify and mitigate harms. In some countries, service providers have been pinpointed by policy-makers as the intermediaries best equipped to serve this function for denial-of-service attacks—and bots more generally—given their unique visibility into traffic patterns and ability to identify malicious traffic. In both the United States and Australia, regulators have helped develop voluntary codes of conduct for service providers recommending measures that could help mitigate bots and, by extension, denial-of-service attacks.

Service providers are unique among the third parties capable of playing a role in defending against denial-of-service attacks because they are relatively concentrated (as compared to DNS operators or the owners of compromised machines, who could potentially assist with access defense but have little visibility to know when it is necessary, and even less incentive to bother). As harm defenders, service providers' visibility of malicious traffic is unparalleled, as is their ability to deliver—or not deliver—packets to their intended destination. Like other third-party defenders who do not directly bear the costs of security incidents, however, it is not clear that they have any incentive to assume responsibility for harm defense.

This lack of motivation on the part of the defenders best equipped to address these threats adds to the challenge of defending against security incidents that are intended only to humiliate, inconvenience, or disrupt targets rather than to extract their money or sensitive information. There are fewer steps involved in accomplishing this kind of harm—and therefore fewer opportunities for defensive intervention. In the Spamhaus case, law enforcement also ended up playing an important defensive role—two of the people believed to have been involved in perpetrating the attacks were later arrested, but this is often a tricky line of defense for a class of attack that rarely leaves a clear money trail or beneficiary. In the Spamhaus case it was likely only possible because Cloudbunker was so outspoken in assuming responsibility for the attacks. In an interview with *The New York Times*, Sven Olaf Kamphuis, who was later arrested, publicly acknowledged that Cyberbunker was using the DDoS attacks to punish Spamhaus for "abusing their influence" (Markoff & Perlroth, 2013). Attackers with a less eager spokesman might well be harder to identify and arrest. Sean Nolan McDonough, a teenager arrested in London in April 2013, later pled guilty to computer misuse and money laundering in connection with the attacks, as well. Still, Krebs (2014b) writes, "Putting spammers and other bottom feeders in jail for DDoS attacks may be cathartic, but it certainly doesn't solve the underlying problem: That the raw materials needed to launch attacks the size of the ones that hit SpamHaus and CloudFlare last year are plentiful and freely available online."

4.5 Defender Interests

Just because an actor is able to defend against some stage of a computer security incident or relevant capability does not mean that doing so will be in that actor's interests. Acting as a defender may be costly or viewed as an implicit acceptance of additional responsibilities and an invitation for others to hold you liable should your defenses fail. So broadening the landscape of potential defenders to include both access and harm dimensions of defense calls for more than just an analysis of what different actors can do to protect against security incidents—it also requires some understanding of those actors' interests and how those interests align with their defensive capabilities.

In general, it is in the interests of both access and harm defenders to shift defensive responsibility (and, accordingly, costs) to the other—hence the post-incident litigation patterns between actors such as TJX and the credit card companies and issuing banks. In other words, the interests of most private actor defenders are centered on not being blamed for security incidents, or rather, not bearing the costs of those incidents, instead of defending against them. The credit card industry, obliged by law to cover losses to their customers, therefore devoted its energy in the aftermath of the incident to recouping those costs from TJX in court, while also using those legal proceedings to reinforce the notion that defense against such breaches was the responsibility of retailers. This strategy was intended to protect the payment card industry's interests in both the short-term, by recovering the expenses of covering the fraudulent costs and reissuing compromised cards, and also the long-term, by perpetuating the belief that the lessons to be learned from this incident centered on defensive changes that retailers like TJX could make to their systems, rather than changes under the control of the payment card industry, such as new payment processing models like chip-and-PIN cards.

DigiNotar, to a much greater extent perhaps than any actor involved in the TJX breach, clearly deemed security—and therefore defense—an important interest in designing its certificate issuing processes. This makes sense for a company whose entire business model is dependent on being trusted by browsers and web users—and also for a company so profoundly affected by a computer security compromise that it ultimately shut down. Google also had an interest in detecting the use of fraudulent certificates for its websites, if not its browser, more generally, and may have harbored some concerns that a compromise of Gmail would have reflected more poorly on Google in popular opinion than it would on a certificate authority that many Google users would likely not have heard of or understood. The DigiNotar incident, like the TJX breach, presented externalities since the direct intended harm of the DigiNotar compromise still primarily affected third parties (i.e., the Iranian Gmail users whose accounts were apparently breached), but unlike the fraud protections for consumers there was no policy mechanism protecting those Google customers from harm. Still, despite the lack of formal mechanisms to internalize this harm for defenders, the consequences for DigiNotar of the breach were much more significant than those felt by anyone as a result of the TJX incident.

The espionage efforts of PLA Unit 61398 also took advantage of the indifference—or interests—of a number of potential defenders. The direct targets of that espionage clearly had a strong interest in defending against it, but the domains being impersonated to create phishing sites and servers and the users and organizations whose computers were compromised to use as platforms for that espionage had much less interest in exercising their abilities to defend against that espionage since it wasn't directly harming them. The owners of impersonated domains might, like Yahoo, see fit to protect their trademarks and brands, while the owners of compromised platforms used in espionage might similarly take some interest in protecting their systems—but these interests would almost certainly be less strongly felt than the direct targets' desire not to have their data stolen and used against them. This again points to the externalities plaguing computer security incidents and the potential role for policies internalizing some of the costs, or harm, of these incidents so they apply not just to the direct targets but also to the intermediary defenders who enable their success.

The interests of the potential third-party defenders involved in the Spamhaus denial-of-service attacks are similarly diffuse. The service providers who carry the malicious traffic, the owners of compromised machines used in these attacks, and the DNS operators who are poised to control some elements of amplification attacks are not directly harmed by the attacks and therefore have less incentive to take action to defend against them. Spamhaus, which clearly has the strongest interest in defending against these attacks, is in turn forced to hire another third-party, CloudFlare, to protect them. CloudFlare's interests, meanwhile, include fulfilling its contract with Spamhaus by effectively defending them—but also keeping themselves in business by maintaining the pressure on denial-of-service attack targets to find and hire their own defense mechanisms, rather than shifting that burden to some other defender who might not require CloudFlare's assistance. In other words, while actors who do not view defense as central to their business may be eager to avoid assuming any responsibility for it, those for whom defense is in fact their core business may be equally invested in preventing other third parties from assuming stronger defensive roles.

Given the number and variety of third-party defenders discussed in the previous cases, it is not surprising that externalities are common in computer security incidents and contribute to the incentives of defenders, or lack thereof. But understanding the interests and decisions of these potential defenders goes beyond just issues of who is—and is not—directly harmed. The externality issues are important, undoubtedly, but they're compounded by the limited visibility of individual defenders and the very different capabilities available to different actors. The lingering uncertainty around policy regimes governing these incidents and how liability will be assigned makes potential defenders even more wary to take any unilateral, voluntary steps for fear of inviting further responsibility. Furthermore, a media and legal landscape that focuses blame primarily on centralized access defense (i.e., on access defense measures that can be implemented by a single, centralized entity or organization) has enabled other potential defenders—particularly those whose capabilities correspond to harm defense, or who are too diffuse to be easily organized—to dodge defensive responsibilities without risking their own interests.

Chapter 5

Application Design as Defense

One of the most basic and fundamental assumptions underlying any attempt to defend computer systems is that malicious and legitimate uses of these systems are—at some point, in some way—different, either because of who is using them, how they are being used, or both. Accordingly, the hardest threats to defend against are the ones that most closely resemble legitimate use by legitimate users. The initial access stages of attacks are often indistinguishable from legitimate activity—consider Gonzalez connecting to the Marshalls wireless network, or the perpetrator of the DigiNotar compromise connecting to the company's public web servers, or PLA Unit 61398 sending emails to their targets, or the Spamhaus attackers sending DNS queries; all of these are activities that, in the hands of someone else, might be perfectly reasonable to allow.

In the aftermath of security breaches, there is often disagreement about who was responsible for noticing early indicators of maliciousness, or best poised to constrain attackers' ability to masquerade as legitimate, and while these arguments tend to center on the organizations (or individuals) whose systems were breached, those organizations are themselves constrained in many ways by the applications they use, especially when those applications are designed intentionally to facilitate interaction with unknown and uncredentialed parties. Application designers can therefore play an important role in contributing to the defense of these systems by making it easier for users to distinguish between malicious and legitimate activity, and harder for attackers to disguise the former as the latter.

One way to do this is to increase the amount of work required of users to acquire potentially malicious capabilities, making it a slower or more resource intensive process for attackers, essentially by forcing them to acquire some additional credentials or reputation in order to exercise the potentially harmful elements of a capability. Another model for defense is to offer clues to legitimate users, administrators, or other potential defenders, to help them detect potential malicious activity. Ideally, these two forms of defense work in tandem—with the work needed to acquire potentially malicious capabilities itself serving as a signal to legitimate users, rather than clues being offered only after those capabilities have been acquired. But for that to be possible the work and the signals must both occur within a single protected system, where the same defender can both witness it and send signals to the relevant parties—for instance, the work of decrypting stored card information in the TJX breach did not take place in the context of TJX's own computer systems, and the company therefore had no ability to see or signal that activity. By contrast, the work of figuring out how to tunnel through DigiNotar's various firewalls did happen in the context of their own system—and perhaps could have triggered some warnings, based on unusual traffic patterns or repeated failed attempts to communicate between separate network segments.

Both increasing work and signaling methods can be applied to two types of application-specific access defenses: those aimed at restricting the malicious capabilities of non-credentialed users, as well as those designed to restrict the ability of malicious users to take advantage of stolen credentials in harmful ways. Non-credentialed capabilities are often, though not always, the means by which malicious actors steal credentials, so while restricting malicious uses of credentialed and non-credentialed capabilities present rather different challenges to defenders, the two forms of defense are related. In fact, a significant component of restricting non-credentialed capabilities is trying to prevent them from being used to steal credentials—recall that in the TJX case, DigiNotar compromise, and the Unit 61398 espionage efforts, as well as hundreds of MIT security incidents, attackers leveraged capabilities that required no authentication to steal credentials that then enabled several of the most essential attack capabilities. Except for insider threats, in which people who were legitimately issued trusted credentials misuse them for malicious purposes, security incidents are often initiated by attackers who are not in possession of trusted credentials and must find some way to procure them through initial access capabilities and applications that do not require credentials. Often, this means security incidents start with applications that enable online interaction between strangers and therefore require no authentication—applications like email and the Web.

The credential space involves multiple different defenders, since both applications and organizations issue and manage credentials in several different, and sometimes overlapping contexts. So the defenses that application designers and organizational administrators can put in place to protect credentials are very much entwined. For instance, application designers have opportunities to restrict the extent to which non-credentialed capabilities permitted by their applications can be exploited to steal credentials, while managers may be able to force attackers to steal multiple different credentials in order to access credential-protected capabilities, and application designers may, in turn, make it more difficult for attackers to use the access afforded by those credentials for malicious purposes. In some cases, applications like web browsers, may be responsible for storing credentials issued by other applications or by organizations.

Further complicating this picture, many applications are built and operated by individual organizations, which both design applications and issue credentials to their users. So the proposed distinction between application designers and organizations as different types of defenders is not always clear cut, but it is important especially when dealing with defenses that address applications that are not operated by a single, centralized entity (e.g., email) and that facilitate interaction between unauthenticated parties. In these cases, where any individual organization has such minimal visibility into whom they are interacting with, they have little to rely on to distinguish between malicious and legitimate interactions other than the clues afforded them by the design of their applications.

5.1 Access Defense at the Application Layer

The central challenge of application access defense is identifying specific capabilities of particular applications that indicate or lend themselves to malicious behavior. The narrower and more habitual an application's intended function is, the easier it tends to be to distinguish between malicious and legitimate use, because legitimate activity takes on a very specific and repeated form. Indeed, the value in focusing defensive interventions at the application layer is precisely applications' specificity of function and habitual use which make it easier to pinpoint malicious indicators and ascribe intention to early access capabilities. More general applications (especially the Web) are more difficult to defend because they offer less specific templates of what allowable behavior looks like, but repeated exploitation of certain capabilities may still be adequate to warrant defensive intervention.

If the strength of application defenses lies in their designers' ability to tailor definitions of malicious and legitimate activity to individual applications' functions, their weakness stems from the wide range of applications used by individual people and machines, which leads to the access capabilities offered by any single application being easily replaceable for many threat actors. This is a problem common to access defense more generally—that restricting or blocking off one access capability does not prevent attackers from using other capabilities (or applications) to achieve the same ultimate goal—but it is especially poignant for application designers who, at best, can hope only to defend against malicious capabilities acquired through only one of many substitutable channels at that layer. Only a subset of those applications, however, facilitate interaction with unknown users, or users who do not possess vetted credentials. Since this tends to be a smaller pool of applications, and these capabilities afforded to unknown users are often the starting point for acquiring credentials that grant access to other applications or capabilities, it offers an interesting set of challenges for defenders rather different from those faced by designers of authentication-based applications.

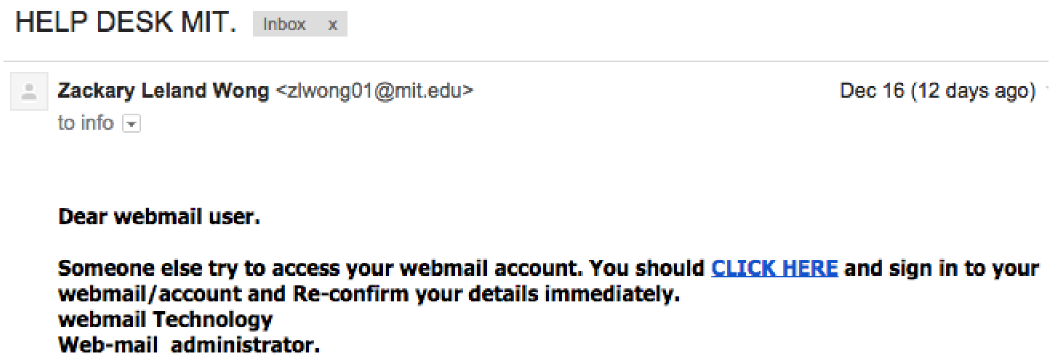
5.1.1 Restricting Non-Credentialed Capabilities

Applications often facilitate interaction between people and companies with pre-existing relationships—for instance, between customers and a known company, or between friends or colleagues—thereby requiring that malicious actors procure someone else's credentials in order to reach their victims. Applications that do not have this requirement present fewer barriers to attackers seeking to acquire useful capabilities, often providing them with initial access pathways that enable the theft of credentials or other, further malicious activity. Email and web browsers are applications that are designed to facilitate interaction between strangers and can therefore offer malicious capabilities even to those who have not managed to illicitly obtain credentials. In both cases, there are legitimate reasons to facilitate that interaction—many users want to be able to receive emails from people they do not know, or visit websites operated by strangers—but also risks that warrant constraining those interactions to limit their potential to serve malicious ends. Understanding how application-based capabilities can serve malicious ends requires being able to identify how the use of these applications by malicious and legitimate users differs, or which specific capabilities present the greatest risks in the hands of users with unknown, or no, credentials.

Email

The espionage incidents Mandiant investigated in 2013 all start the same way: with an email (or several) sent to employees at the target organizations. It's the same way that a 2012 breach of the South Carolina Department of Revenue tax records and financial data began, as well as the starting point for many of MIT's compromised accounts, which are themselves used to send spam and more phishing emails. Thanks to their ubiquity and flexibility, the access capabilities afforded by email figure in a variety of security incidents spanning different classes of harm from espionage to financial fraud, and because anyone can send an email to anyone else, it is often an early, or even first, step in these incidents—one that requires no previous capabilities or access. For malicious actors who have not managed to illicitly procure credentials, email offers opportunities to do just that—as well as to encourage recipients to initiate financial transfers or download malware. These malicious uses of email are not necessarily distinct—for instance, malware sent via email attachments may be used to capture credentials that can then be used to initiate financial transfers, as in the case of the Zeus malware, which is distributed via email and then captures users' keystrokes to collect banking credentials. Other uses of email do not require malware—for instance, many of the phishing emails targeting MIT users in 2013 and 2014, such as the one shown in Figure 5-1, contained no attachments, and instead prompted recipients to visit a website where they were asked to input account credentials, as shown in Figure 5-2. (In other cases, particularly when attachment screening defenses are in place, these models are combined, and users are prompted to visit websites which themselves deliver malware.) And email capabilities do not necessarily involve capturing credentials; for instance, the CryptoLocker malware is also distributed via email but does not target credentials, instead encrypting infected machines' files and demanding that victims pay a ransom fee using Bitcoins or pre-paid cash vouchers in order to decrypt their data.

So, as is often the case when it comes to exploiting access capabilities, there is not a single pattern for malicious use of email—a malicious email could lead in multiple different directions. But there are repeated themes and elements in these uses of non-credentialed email capabilities, and in thinking through defensive models it is helpful to map out some common email capabilities that do not require stolen credentials according to their potential for malicious and legitimate use, as shown in Figure 5-3. Existing security mechanisms for email have focused on the related concerns about impersonation and inaccurate sender information listed in the lower



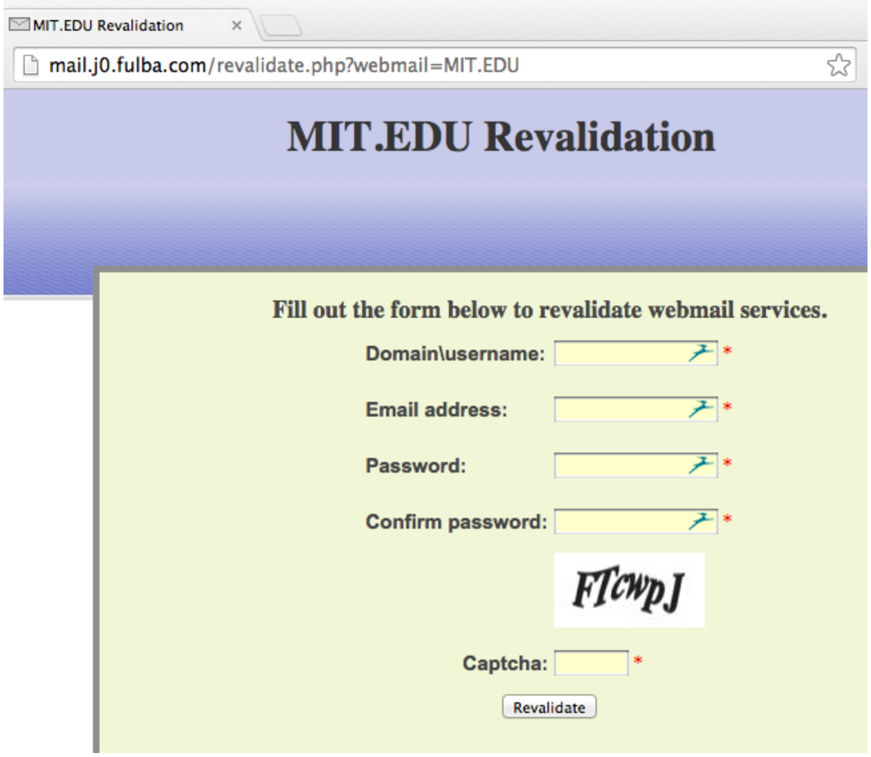
HELP DESK MIT.

Zackary Leland Wong <zlwong01@mit.edu> Dec 16 (12 days ago)

Dear webmail user.

Someone else try to access your webmail account. You should CLICK HERE and sign in to your webmail/account and Re-confirm your details immediately.
webmail Technology
Web-mail administrator.

Figure 5-1: An email sent to MIT email accounts prompting recipients to visit a website for security purposes.



mail.j0.fulba.com/revalidate.php?webmail=MIT.EDU

MIT.EDU Revalidation

Fill out the form below to revalidate webmail services.

Domain\username: *

Email address: *

Password: *

Confirm password: *

Captcha: *

Figure 5-2: The website linked to by the email in Figure 5-1 prompting visitors to enter their MIT username and passwords.

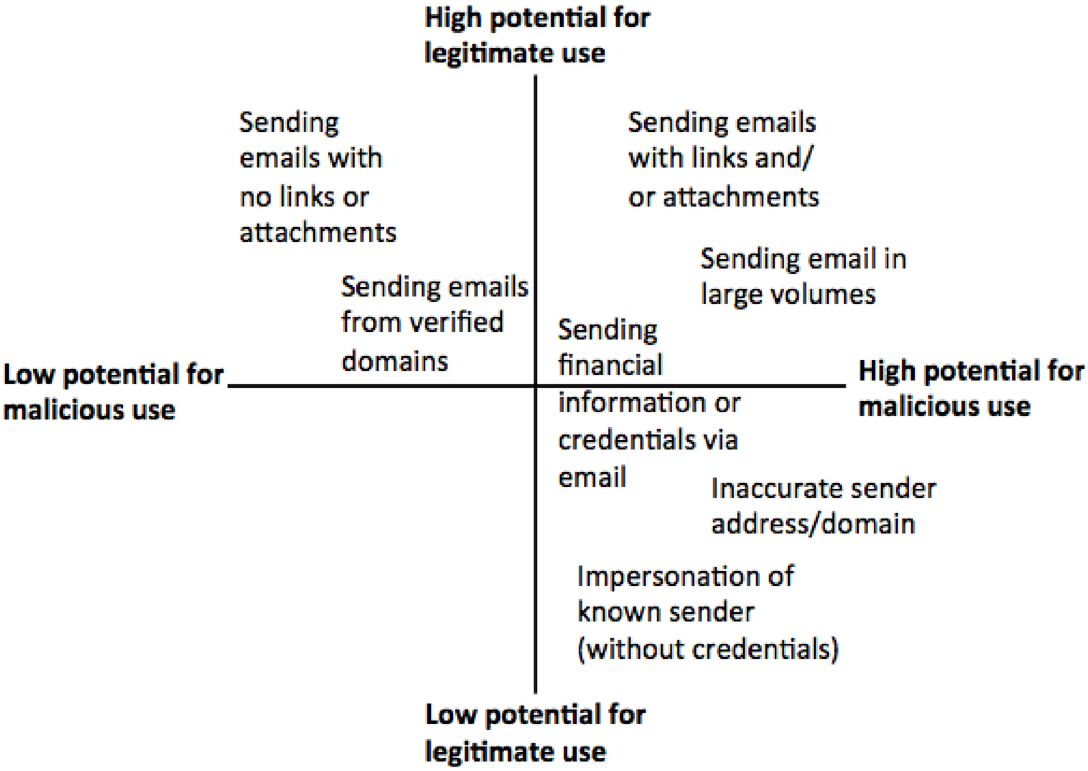

High potential for legitimate use
Sending emails with no links or attachments
Sending emails with links and/ or attachments
Sending email in large volumes
Sending emails from verified domains
Low potential for malicious use
Sending financial information or credentials via email
High potential for malicious use
Inaccurate sender address/domain
Impersonation of known sender (without credentials)
Low potential for legitimate use

Figure 5-3: Potential of different email capabilities afforded to users with unknown credentials to be used for malicious and legitimate purposes.

right quadrant of Figure 5-3. These include defenses implemented by individuals, such as personal digital signatures, as well as those that rely on domain administrators and SMTP server operators, including the Sender Policy Framework (SPF) email validation system and DomainKeys Identified Mail (DKIM) signatures.

SPF enables administrators to create a record in the DNS specifying which hosts are allowed to send mail from a given domain. Emails purporting to be sent from that domain can then be verified against the DNS record. DKIM allows for a mail transfer agent—rather than a mail user agent—to add a signature verifying the originating domain to an email's header—instead of adding it to the message body, as is done with S/MIME (Secure Multipurpose Internet Mail Extensions) personal signatures. A receiving SMTP server verifies this signature by querying the DNS to look up the sender domain's public key in the DNS. Both SPF and DKIM therefore rely on the DNS as a means of sender or signature verification, eliminating the need for a third-party certificate authority. Both also allow for email to be received from senders who do not utilize these verification procedures, since both the SPF and DKIM signature header fields are optional.

In 2008, PayPal became concerned about the large volumes of phishing emails being sent to its users, so its owner eBay it made an agreement with Yahoo! and Google that it would send only DKIM-signed email and the two email providers, in turn, agreed to discard any messages sent from the PayPal (or eBay) domain that were not signed, or that had invalid signatures. In a blog post announcing the agreement (Taylor, 2008), Gmail Spam Czar Brad Taylor wrote:

> Now any email that claims to come from "paypal.com" or "ebay.com" (and their international versions) is authenticated by Gmail and—here comes the important part—rejected if it fails to verify as actually coming from PayPal or eBay. That's right: you won't even see the phishing message in your spam folder. Gmail just won't accept it at all. Conversely, if you get an message in Gmail where the "From" says "@paypal.com" or "@ebay.com," then you'll know it actually came from PayPal or eBay. It's email the way it should be.

This notion of "email the way it should be" hints at the crucial challenge of trying to delineate malicious and legitimate application-specific behaviors, or defining the shoulds and should-nots of email use: users should be able to receive emails from PayPal, but should not receive such emails from senders who don't have any affiliation with the company; should be able to open links and attachments we want to view, but should not be able to open ones designed to steal our credentials or encrypt our files; should be able to correspond with strangers and receive bulk emails, but should not be able to correspond with users who are known to be malicious or receive their bulk emails.

Email defenses correspond to a range of these restrictions, not just those designed to combat impersonation. These include blacklists of known spammer IP addresses, domains, and open proxies (like those maintained by Spamhaus), visual indicators of attachment file types (of the sort manipulated by the PLA espionage efforts, in which executable files were disguised as PDFs), or filters that reject emails on the basis of particularly suspicious text or content. None of the potentially malicious behaviors that these techniques are designed to defend against—impersonation, sending email through open proxies, sending executable attachments—are actually harmful in and of themselves. Email capabilities may be used to further a larger, harmful goal—stealing money or secrets, for instance—but because they do not directly inflict harm, it is often possible for malicious actors to disguise these access attempts, via email or other applications, as legitimate. Application defenses, then, essentially increase how much work attackers must do to effectively execute these disguises, by honing in on the subtle discrepancies between potentially malicious and legitimate activity.

Web Browsers

Web browsers, like email, often facilitate interactions between users and unknown actors, who own and operate websites. Also like email, these capabilities are often used to impersonate trusted actors, obtain credentials or financial information, and install malicious programs, but not necessarily in service to a particular class of harm. Accordingly, many of the potentially malicious and legitimate capabilities afforded by browsers to users with unknown (or no) credentials, shown in Figure 5-4, mirror those afforded by email. In both cases, the potential for maliciousness increases

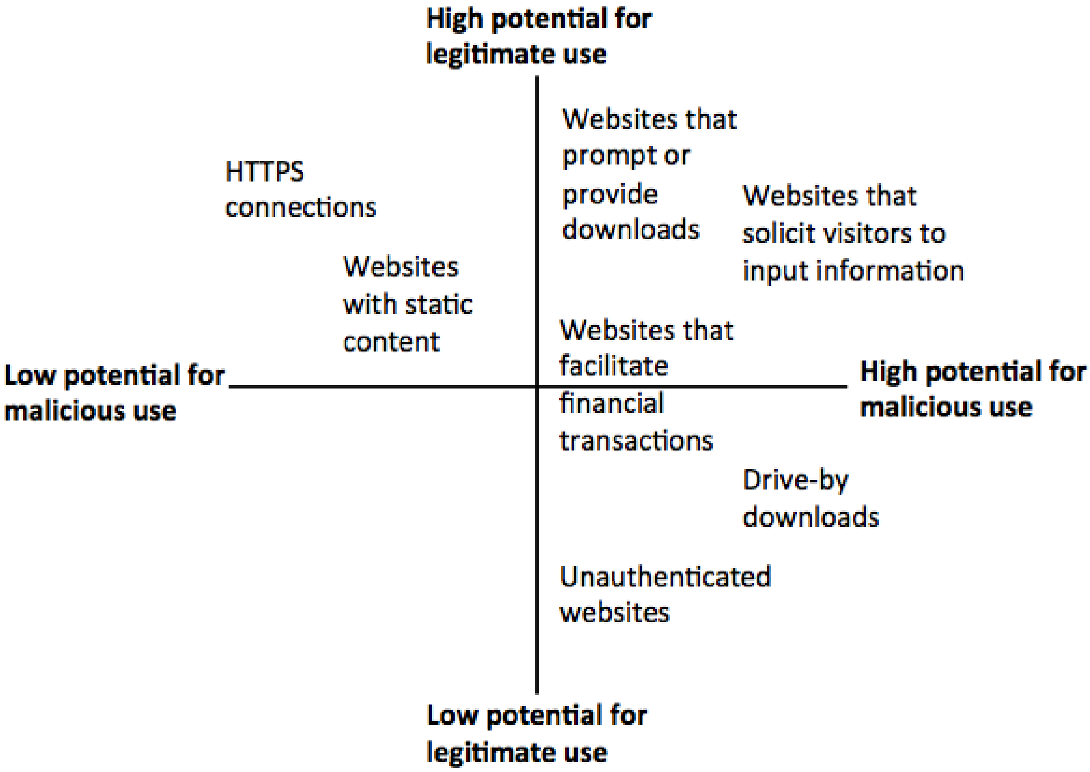

High potential for legitimate use
Websites that prompt or provide downloads
HTTPS connections
Websites that solicit visitors to input information
Websites with static content
Websites that facilitate financial transactions
Low potential for malicious use
High potential for malicious use
Drive-by downloads
Unauthenticated websites
Low potential for legitimate use

Figure 5-4: Potential of different web browser capabilities afforded to users with unknown credentials to be used for malicious and legitimate purposes.

with capabilities that allow for impersonation, downloads, and requests to users to input or provide information. Existing browser defense mechanisms address certain components of these capabilities, for instance by notifying users when they connect to sites with invalid certificates, blocking sites with known malicious certificates, or blocking JavaScript. But the variety of different ways malicious actors can use these capabilities, individually and in combination, to achieve the same goals—of stealing credentials, or being able to send outbound traffic from targeted machines, for instance—makes it difficult to pinpoint any single capability that is common to all malicious uses of browsers. In fact, perhaps the most common element of uncredentialed malicious browser capabilities is their reliance on email and other messaging applications to deliver links. By contrast, browser capabilities that make use of stolen or fraudulent credentials, as in the case of the DigiNotar breach, often enable bad actors to take advantage of URLs that their victims visit unprompted. The relationship between email and web browsers is particularly messy, since email is often used to send links to websites, while browsers are increasingly used to access email. Understanding the potential maliciousness of a capability afforded by a particular application, in other words, requires understanding how that access relates to other applications and capabilities outside an application designer's control.

The interplay between the capabilities acquired through different applications adds to the challenges of defending them individually—for instance, in many of the MIT account compromises, an uncredentialed email capability is used to draw recipients to an uncredentialed website that then collects their credentials so that attackers can exploit credentialed email capabilities (i.e., sending messages from the verified MIT domain). From a defensive standpoint, the number of steps involved in this sort of incident is both advantageous, in that it offers lots of opportunities for intervention, and problematic, in that it complicates the question of who (or what application) is responsible for identifying and preventing which element of the malicious activity. There are two, rather conflicting, defensive strategies that follow from this dilemma: one is to limit the allowed interaction between different applications—the extent to which capabilities in one can be used or useful in the context of another—for instance by restricting the circumstances under which links to websites can be sent via email.

This approach, in some sense, narrows the broader range of malicious capabilities presented by other applications that a designer must worry about. An alternative is for application designers and operators to actively broaden their scope by creating their own versions of multiple, related applications and using their increased control of and visibility into the interaction between those applications to reinforce security mechanisms across all of them. For instance, the fraudulent certificates stolen in the DigiNotar breach and used to access Gmail accounts were largely ineffective in the cases of users who used both Google's email service and its browser, because the approved credentials for the company's email site had been pinned to its browser. Had both of those applications not been operated by the same entity, it would have been much more difficult to identify the use of fraudulent certificates—but, on the other hand, having both applications tied to the same company introduces a common dependency across all of their defenses.

In the context of individual applications, the independence of defenses is necessarily narrow as compared to the defenses that can be implemented across an entire organization or multiple different entities beyond the boundaries of a computer system. Applications operate in the context of browsers or operating systems—that is, they all depend on those common interfaces and, accordingly, their individual defenses are all similarly dependent on those shared platforms. The central challenge of constructing defense in depth for individual applications therefore lies not in the independence requirement but in the overlap criteria, because attackers may not need to make use of all of the various restricted capabilities in order to achieve their goals. In other words, many application access capabilities are non-essential to attackers: they can (and sometimes do) forego forging email sender domains or using attachments, relying instead on plausible false email addresses for impersonation purposes, as well as website links or appeals for financial transfers. Still, attackers seeking to exploit application capabilities must disguise their activity as legitimate in ways that evade all of the restrictions placed on different potentially malicious capabilities, and in this sense, restricting different capabilities in the same application is a form of defense in depth so long as each restriction further narrows the range of acceptable legitimate behaviors available to attackers.

Increasing Work & Sending Signals

Increasing the work required of malicious actors to take advantage of non-credentialed application capabilities essentially means forcing them to acquire credentials in order to exercise any but the most basic and harmless application uses. This approach is predicated on the notion that there are certain, very narrow parameters under which people and organizations with no preexisting relationship should interact via computer systems, excluding any forms of interaction that have clear potential to be harmful or serve malicious ends. From a defensive perspective, this means making malicious actors work harder to be able to achieve those forms of interaction—making it harder for them to acquire the access capabilities that have a high potential for both legitimate and malicious use. In the context of email and web browsers, those capabilities are primarily tied to impersonation and malicious downloads or websites. These relate to a more general pattern across malicious uses of application capabilities that do not require authentication with known credentials: they are exploited in a variety of incidents primarily to steal credentials or deliver malware (or, occasionally, both—using one to facilitate the other). Those aims, once accomplished, can be leveraged to inflict several different kinds of harm, ranging from espionage to financial theft to denial-of-service attacks, that often occur beyond the scope of the original application through which they were acquired. Defending applications that facilitate communication, or interaction of any form, between people with no prior relationship therefore often means looking for the specific indicators and behaviors that are most closely or commonly tied to these two activities—like including attachments and links in emails—and restricting how easily users can exercise those behaviors.

These restrictions can take different forms, from only restricting the capabilities of known malicious actors—by blacklisting their IP addresses, certificates, or other specific identity indicators—to restricting the capabilities of all unknown actors, for instance by blocking JavaScript or attachments by default for all websites or emails. The primary weakness of defenses based on blocking known malicious actors is that bad actors' identity indicators are almost always replaceable. The extra work imposed by blacklists may therefore be sufficient to fend off bad actors with fewer resources—those who have neither the money to purchase an endless supply of domain names nor the expertise to write their own malware—but they are unlikely to significantly hinder well-financed adversaries. For instance, along with their report on PLA Unit 61398, Mandiant released a list of 13 X.509 encryption certificates, 40 families of malware, and more than 3,000 domain names, IP addresses, and hashes of malware, associated with the unit's activities. While those lists may help defenders identify reuse of those resources, the volume of indicators also suggests how easily the PLA can afford to replace them with new ones. Defenses that target known malicious capabilities rather than known malicious actors may be less easily circumvented. These might entail adding extra layers of scrutiny or screening to all messages with attachments or links, or interactive websites, or downloaded files, perhaps by quarantining them or requiring they be approved by or cleared through a third party.

Another model is to link how easily people can acquire potentially malicious capabilities to the likelihood that they are impersonating others, using impersonation as a proxy for malicious activity. One way to do this is to build on the mechanisms already in place to reduce impersonation for these applications, for instance, by affording potentially malicious capabilities like including attachments and links only to email senders whose accounts can be authenticated via DKIM, SPF, or personal signatures, or restricting interactive features to websites authenticated using HTTPS. Using these optional authentication mechanisms to allocate access capabilities in the context of applications where authentication is not required, or even necessarily widespread, ideally lessens the burden on legitimate, credentialed users and sites without entirely eliminating the presence of all uncredentialed users, while still restricting their ability to act maliciously. Other, less rigorous mechanisms for detecting impersonation might include whether someone has visited a site or corresponded with a particular email sender previously (though this measure is most reliable when tied to one of the authentication mechanisms), under the assumption that new senders and sites are less trustworthy, or whether the content of a site, or an email sender's displayed name closely resembles that of a commonly visited page or existing email contact (a check that is useful only if the sender has not also forged the 'from' address).

Several of the existing authentication mechanisms—including email signatures and HTTPS—function primarily to signal users about the possibility of impersonation, providing icons, warning messages, and other indications within a browser or email client of suspicious activity. Often, however, these signals go unnoticed or unheeded by users—especially when they indicate the presence (or absence) of HTTPS (Dhamija, Tygar, & Hearst, 2006; Egelman, Cranor, & Hong, 2008; Schechter, Dhamija, Ozment, & Fischer, 2007). This suggests that signaling to recipients or users when there are indicators of impersonation or malicious activity may not be a very effective defense mechanism for protecting against non-credentialed access capabilities, though Egelman et al. (2008) offer some recommendations for how such signals can be more effectively implemented, including designing the warnings to interrupt users' primary tasks, ensuring that users are forced to read them before proceeding, and providing users with clear choices about how to proceed. However, there are few opportunities to use attackers' malicious activity itself as a signal in these cases, because exploiting uncredentialed capabilities does not involve any work that interacts with the victims' systems. That is, all the work that goes into these attacks—from crafting and sending phishing emails to designing and registering fraudulent sites—occurs outside the context of systems the victims can monitor. By contrast, the work involved in using credentialed capabilities for malicious purposes—for instance, guessing passwords or exfiltrating large volumes of data—occurs within the context of a victim's system, and therefore can be flagged as a direct signal of malicious activity, as well as providing means of restricting capabilities.

So signals of malicious intent associated with unauthenticated capabilities tend to rely on highlighting the absence of that authentication—the lack of work that went into the attack, rather than the work behind it, in some sense—and do so in ways that users do not necessarily heed. Forcing outsiders to do more work that involves establishing a relationship and interacting directly with their victims in order to attain their desired capabilities may help address the ease with which attackers exploit unauthenticated application capabilities. For instance, requiring that senders have some previous correspondence with their recipients before being able to send them attachments, could create more work for attackers in a manner that also serves to signal victims of potential malicious intent (or at least new and unknown email senders and websites). However, relying on that previous low-risk correspondence to enable subsequent higher-risk interactions can backfire—Mandiant's report on PLA Unit 61398 describes an incident in which the recipient of a PLA phishing email with an attachment replied to the sender, "I'm not sure if this is legit, so I didn't open it." The sender responded: "It's legit," and the recipient then opened the attachment. In light of the challenges posed by crafting effective signals, stronger forms of these defenses occasionally go beyond signaling and are actually used to limit attackers' capabilities, as in the case of Google using DKIM verification to delete all of the unauthenticated mail sent to Gmail users from particular domains, such as PayPal. In this instance, DKIM is not being used to signal recipients of potentially malicious interactions but actually to restrict what capabilities are afforded to unauthenticated senders attempting to impersonate specific restricted domains in such a way that recipients have no interaction with those senders.

So there are several defensive options even for the relatively limited pool of unauthenticated applications—restricting the capabilities afforded to known malicious actors and restricting for everyone the capabilities that have been used repeatedly for malicious purposes both fit the model of increasing the work required of attackers. Highlighting indicators of impersonation can help signal maliciousness, and using those indicators as the basis for assigning capabilities can serve as both a signal to legitimate users and a means of increasing the work required of malicious ones. These types of defense, that specifically restrict the extent to which unauthenticated application users can exercise potentially malicious capabilities, of course do nothing to protect against the malicious capabilities afforded to people who have successfully stolen credentials. However, by forcing attackers to obtain credentials in order to acquire such capabilities, these defenses effectively increase the work those attackers must undertake. Furthermore, since phishing emails and websites are common initial vectors for stealing credentials, restricting the unauthenticated capabilities afforded by these applications might well impact how easily they can be used to obtain credentials—rendering credential theft not only essential, but also harder to accomplish.

5.1.2 Restricting Credentialed Capabilities

Access defenses for unauthenticated capabilities all essentially serve to narrow the capabilities open to those uncredentialed attackers, placing greater pressure on them to acquire credentials (and, ideally, also making it harder for them to do so). While application designers play a uniquely important role in defending against those unauthenticated capabilities, since individual users have so little visibility into the interactions afforded by them, they can also help contend with defending against attackers who have successfully acquired credentials. Defending against the malicious use of stolen credentials to acquire capabilities is distinct from defending against the theft of credentials; the former deals specifically with how suspicious use of credentials can be restricted or signaled after an adversary has successfully acquired all of the needed credentials for authentication.

Once an attacker has acquired the necessary credentials to exercise a certain capability, distinguishing between malicious and legitimate use of those capabilities becomes much trickier because the credentials themselves typically serve as the crucial indicator of legitimacy. This kind of defense is largely predicated on the idea that much application use is habitual—that users routinely go to the same sites, communicate with the same people and in fairly regular volumes, transfer money to the same places and in fairly regular volumes, use the same devices and IP addresses—and deviations from those habits may therefore serve as indicators of possible maliciousness. Such defenses are clearly most effective for applications that encourage habitual use patterns, and the role of application designers lies in understanding what particular regular habits or routines users are likely to develop in the context of their applications that might be difficult for attackers to replicate. Often, these fall into the categories of volume-based habits (e.g., number of emails sent per day), interaction identity-based habits (e.g., popular contacts or frequently visited websites), user identity-based habits (e.g., typing speed or eye movement patterns), or hardware and network-based habits (e.g., MAC addresses or IP addresses). Sometimes these behaviors signal legitimate activity not just because they are recurring but also because of the nature of malicious uses. For instance, volume-based indicators tied to how many emails (or DNS queries) a user typically sends serve a dual function in highlighting both anomalous and potentially malicious activity, since large volumes of activity may itself be a sign of a compromise.

Deviations from an application user's habits can be leveraged by application designers either to directly increase the work required of an attacker or to signal malicious activity to legitimate users. Increasing work in this context essentially involves requiring an extra set of additional credentials in the presence of certain behavioral anomalies—for instance, asking users to answer questions about themselves, or forcing them to have their unusual activity approved by a different user, or via another application. This, for instance, is the model of multi-factor authentication systems that allow users to designate exempted known devices so that multiple factors are only required to authenticate when connecting from new or unknown devices—an anomalous behavior triggers the requirement of additional credentials which would otherwise not be necessary. A more stringent (if more cumbersome) approach to defending applications based on deviations from habitual use might entail limiting specific capabilities for authenticated users exhibiting anomalous usage patterns. For instance, high-volume activity, or capabilities with particularly high potential for misuse might be restricted to known devices, or users adhering to their customary patterns, while deviations from those habits would actually make it impossible for users to exercise certain capabilities.

Defense is hardest when malicious use of authenticated application capabilities most closely resembles legitimate use. For instance, Bursztein et al. (2014) analyze the behavior of manual hijackings of Google accounts using stolen credentials and find that these incidents are difficult to defend against in part because "what manual hijackers do when interacting with Google's services is not very different from what normal users do. Normal users also search their inboxes and read emails, set up email filters, and change their password and their recovery options. Thus the rules or models derived from those behaviors are not crystal clear and are certainly not high enough confidence to act upon easily." In other words, in this context, malicious activity looks too much like legitimate activity to be reliably identified or defended against.

Their analysis also suggests a slightly different approach to identifying malicious use of authenticated capabilities—looking for identifiable habits of attackers, rather than legitimate users. For instance, they find that account hijackers are likely to search for certain terms in their victims' email (e.g., wire transfer, bank, investment, password, username), connect from IP addresses in certain countries (China, Ivory Coast, Malaysia, Nigeria, and South Africa), and send messages to, on average, 630 percent more recipients per day than the legitimate users of those accounts. On the other hand, they also describe active efforts on the part of the hijackers to "blend in" with legitimate activity, noting particularly that "on average, the hijackers attempted to access only 9.6 distinct accounts from each IP, which makes their activity extremely difficult to distinguish from organic traffic." The number of outgoing emails from hijacked accounts was also only 25 percent higher, on average, than the number of legitimate messages sent from the account the day before the hijacking, though each message is sent to many more recipients, accounting for the 630 percent increase in distinct recipients (Bursztein et al., 2014).

These attempts by hijackers to disguise their activity to more closely resemble that of legitimate users make sense in the context of a defensive strategy that is dependent on distinguishing between malicious and legitimate behavior. It also highlights how different elements of malicious activity may be more or less easily disguised in this manner—IP addresses, for instance, can be easily altered to avoid suspicion, as can the volume of outbound emails, but hijackers must still reach a large number of recipients for attempts at email-based phishing or financial scams to be successful. And some legitimate users' activity may more closely resemble malicious behavior than others, so linking defenses to deviations from habitual application use may affect who attackers view as valuable targets. For instance, if an account is restricted so that the number of email recipients cannot deviate too greatly from the previous day's (or week's) activity, then the credentials of users who routinely email large numbers of people become more valuable, and more likely to be targeted, than those of users who don't.

Compromised MIT accounts also show some regular patterns of malicious use—including sending uncharacteristically large volumes of outbound email, creating filters to automatically delete all incoming mail (in order to avoid alerting legitimate users via bounce-backs or responses), and downloading unusually large numbers of academic papers from library databases (often via connections initiated from Chinese IP addresses). In some cases these malicious behaviors may generalize to other targeted institutions, though some (for instance access to library resources) may be more victim-specific. As with non-credentialed capabilities, the potential for legitimate use and degree of suspicion associated with a particular application capability may vary between user environments. Application designers can try to limit particularly suspicious access capabilities based on deviations from legitimate users' habits, or patterns of activity associated with malicious users, but these patterns are not always sufficiently distinctive or clear-cut to warrant full-fledged capability restrictions.

However, while it can be trickier to distinguish particular capabilities to restrict in the context of authenticated application capabilities as compared to unauthenticated capabilities, there may be more opportunities for signaling malicious activity to legitimate users in the former case. This is because capabilities acquired through stolen credentials are exercised in the context of the same system used by the victim, or at least an intermediate victim (i.e., the person whose credentials were stolen). Additionally, while attackers use uncredentialed capabilities primarily to catch victims' attention in some fashion and lure them into clicking on an attachment or a link, once they've successfully stolen credentials their behavior often shifts to emphasize greater discretion and they put more effort into covering their tracks so as to prolong the value of the stolen credentials. This makes signaling a more potent form of defense and provides opportunities to signal legitimate users about any unusual activity undertaken using their (or others') credentials—by highlighting when those credentials were used and from what IP address, what they were used for, and whether they are being used simultaneously in multiple instances. This information can be captured and presented to legitimate users in ways that details about uncredentialed capabilities cannot, because the targeted system has much less visibility into how uncredentialed capabilities are being used. Signaling potential malicious use of authenticated capabilities shifts some of the burden of distinguishing between malicious and legitimate activity onto legitimate users and system operators, relieving application designers of the need to be able to characterize maliciousness and legitimacy in general terms that apply to everyone.

Still, application designers can help constrain the extent to which those signaling defenses can be altered or hidden by malicious actors. For instance, people who hijack MIT email accounts routinely set a filter to delete all incoming messages, so that legitimate users will not immediately begin receiving responses or notifications of undeliverable messages alerting them to the fact that their accounts have been compromised. In doing so, the hijackers essentially remove a signal of malicious activity—though they also create another one, by deleting all of that user's legitimate inbound email, which eventually tips off many users that something is wrong (in fact, detections of compromised accounts at MIT often stems from user complaints about not receiving any email). If all clues of malicious activity can be so easily altered using the same credentials that enabled the activity, then these signals provide minimal obstacles for attackers determined to cover their tracks. So for signaling to be an effective defensive measure against malicious use of credentialed capabilities, there must be some signals that are unalterable even for authenticated users—for instance, lists of access IP addresses, login times, or even aggregated statistics on activity volume and alerts about unusual activity or settings changes.

Related to the challenge of crafting signals that malicious actors cannot alter is the risk of signaling malicious actors themselves, while they are authenticated, and thereby warning them they need to better cover their tracks. This suggests a possible role for differentiated signaling activity based on the same types of habit indicators that might be used to restrict capabilities. Just as certain capabilities might not be granted to users who exhibit anomalous behaviors or access patterns, so too, might certain signals be reserved for users whose authentication credentials are reinforced by familiar activity consistent with their usual patterns.

Application defenses for authenticated capabilities can increase the work required of attackers by forcing them to obtain additional credentials when they wish to exercise particularly dangerous or suspicious capabilities that are known to be associated with security breaches (for instance, searching for particular key terms, or deleting incoming emails). Signaling can also be a meaningful line of defense for these capabilities, by flagging anomalies or other suspicious behavior to legitimate users and system administrators. And, as before, it is also possible to combine these two models so that extra credentials and authentication measures are required of users behaving in particularly anomalous ways, or even by cutting off certain capabilities to users exhibiting those anomalies.

So application defenses for authenticated capabilities mirror those that apply to unauthenticated capabilities in that both center on finding ways to distinguish between malicious and legitimate uses of these capabilities and leveraging those differences to make the malicious uses harder to pursue and more immediately apparent. However, while defending against the use of uncredentialed capabilities hinges on determining which capabilities are intrinsically more likely to be tied to malicious use, defenses for authenticated capabilities tend to be more closely tied to deviations from users' routine activity, offering a less clear picture of what activity is specifically malicious rather than just unusual. In this, the two modes of defense echo slightly the two definitions of secure computers suggested in Chapter 1—one tied to the use of computers to cause harm, the other, more broadly, to any unexpected behavior whatsoever.

5.2 Harm Defense at the Application Layer

While applications dictate many of the crucial capabilities malicious actors seek to access on protected computer systems, they generally have comparatively little influence over the harms those actors ultimately aim to inflict. That is, an application's capabilities may enable harm, but are unlikely to inflict it directly, given their limited scope. There are some exceptions, where restricting application capabilities can actually directly restrict harm—for instance, restricting capabilities afforded by financial applications by limiting the size or frequency of transactions and flagging anomalous activity (as is already done by most major credit card companies) can impact adversaries' ability to inflict financial losses on victims. More often, however, application capabilities cannot be used to harm others in and of themselves, but instead serve as indirect channels, bringing attackers one step closer to their ultimate goals. The challenge of access defense is that there are sometimes many different ways to achieve those goals, so restricting one capability merely forces attackers to take advantage of a different one (and perhaps even a different application)—just as restricting uncredentialed capabilities pushes them to acquire credentials. There is less guesswork involved in harm defense—less uncertainty around the question of what is malicious activity because the infliction of harm is, by definition, what makes activity malicious. But for the most part that harm is inflicted in the physical world not the virtual one—fraudulent credit cards are manufactured and sold, proprietary information and intellectual property is acted upon to make political and economic decisions or develop new products, physical systems are manipulated—in ways that often occur outside the context of the computer applications whose capabilities initially enabled the end results.

Harm defenses that target these later stages of attacks, following the successful exploitation of application capabilities, are therefore not usually the domain of application designers. For digital harms, including political espionage and disruption of digital service, application access defenses may be closely linked to harm defenses because the distinction between the two classes of defense is blurred for harms that occur in a purely virtual context. Therefore, the access defenses that application designers put in place to protect against hosts being compromised as part of botnets or accounts being accessed from unrecognized devices and in unusual ways may, in some cases, actually be the last possible lines of defense against these types of harm. Another potential role for application designers in mitigating—or, in fact, pre-empting—the harm caused by security incidents lies in designing applications that delete information by default, allowing for exceptions designated by users, limiting the scope of damage that can be imposed across multiple different classes of harm, including political and economic espionage, as well as financial fraud and public humiliation.

Designing for deletion is a form of restricting access capabilities in the sense that it restricts everyone's capability to access large volumes of archived records or data, and, as such, it is dependent on those archived records not being essential for legitimate purposes. However, by allowing legitimate (authenticated) users to designate when information should be stored for longer than the default duration, it may be possible to serve legitimate needs without resorting to a design that only deletes the information specifically selected by users. As a form of harm defense, this is a fairly coarse and untargeted approach—which is perhaps unsurprising in light of applications' limited involvement in the process of inflicting harm—and it is unlike most other types of harm defense in that rather than waiting for attackers to achieve access capabilities and then interrupting their attempts to use those capabilities, it limits the usefulness of access capabilities even before they are acquired by adversaries. Still, it shares with late-stage harm defense interventions the central function of degrading attackers' ability to take advantage of computer access capabilities to cause harm.

Encryption also presents a potential mode of harm defense for application designers—one that could make it more difficult (though not impossible) for attackers to make use of stolen data, even if they are able to access it. However, encryption is a complicated harm defense because its effective implementation is dependent on access. For encryption to be effective as a line of defense, someone—whether the application designer and operator or someone else—needs to be able to protect a key, and if those access defenses protecting the encryption key fail, then so does the broader harm defense role served by the cryptography. This dependence on access defense renders encryption rather limited as a mode of harm defense—it does not serve as a wholly independent line of defense to mitigate harm should the access defenses protecting a computer system fail. Rather, its effectiveness is entirely caught up in the ability of access defenders to protect a crucial piece of information, and if those defenses cannot be relied upon, then neither can the harm defense role served by encryption. This intertwining of access and harm defense, in which a form of harm defense depends on access defense, hints at the importance of considering both framings in parallel and the risks of focusing only on one, at the expense of the other.

Chapter 6

Management as Defense

While application designers play an important role in shaping how easily malicious actors can acquire different access capabilities, security incidents are more often closely associated with—and blamed on—individual institutions whose systems are breached than the particular applications through which their adversaries gained access. The assumption implicit in the ways we label and discuss these incidents (the TJX breach, the DigiNotar compromise) is that the organization which owned and operated the compromised machines failed in its responsibilities to protect them. And yet, as the growing number of lawsuits against these organizations illustrate, it is not entirely straightforward to articulate what those responsibilities are or how they can be fulfilled. Organizations that own and operate computer systems are constrained and influenced in their defensive postures by both the design decisions of application developers and the policy decisions made by government actors. These organizations play a wide range of defensive roles, spanning both access and harm defenses depending on the incident—recall that in the TJX breach and DigiNotar compromise, the central organizations were primarily capable of implementing access defenses, while Spamhaus and the victims of the PLA espionage efforts were positioned more squarely as harm defenders.

So institutional defense is partly oriented towards furthering the access defenses put in place by application designers, but it also includes some elements of harm mitigation, especially with regards to digital harms, and essential computer-based intermediate harms, that blur the boundary between access capabilities and harm. Organizations bridge the more strictly access-oriented defensive role of application designers and the harm-mitigation efforts of law enforcement and policy-makers. While managers have less control over the distinctions between legitimate and malicious activity baked into the applications they use than do designers, and less ability to trace and regulate illicit money flows than government actors, they occupy an interesting set of in-between defensive roles that center on restricting credential theft and outbound information flows, covering some of the holes left by the other two groups' defensive capabilities. Most crucially, these two roles make use of organizations' particular visibility into isolated computer systems and their limited scope of control over their members—they do not require of organizations a more global window on incidents than they actually possess with regard to either access or harm stages that happen beyond the boundaries of their own systems.

6.1 Administrative Access Defense

Access defense at an institutional level is closely tied to the access defenses built into the applications those institutions use—this is true even for federated communication applications like email, but the intermingling of designer and organizational defensive roles is especially manifest in applications that are designed and operated by a single, centralized company which assumes responsibility for elements of both design and implementation of defenses. The defensive roles of application designers center on making it easier to distinguish between legitimate and malicious activity in the context of their applications by restricting potentially malicious capabilities to users with trusted credentials (or to no one, for capabilities that serve no sufficiently vital legitimate capability) and signaling anomalous use of those trusted credentials. Organizations and end users can bolster those access defenses by tailoring applications' definitions of malicious and anomalous activity to their own legitimate uses and threat models, to the extent permitted by the application designers. But perhaps even more critically, organizations often play a vital role in issuing and protecting the trusted credentials required to access potentially malicious capabilities. As before, access defense centered on protecting credentials can take multiple forms, including both increasing the work needed to acquire such credentials illicitly and signaling attempts to do so.

In the context of access defense, the administrative roles of tailoring application restrictions and protecting authentication credentials are intended, respectively, to restrict any potentially malicious capabilities to credentialed users and to restrict any access to trusted credentials to only that group of users, so that, ideally, in order to gain the capabilities needed to inflict harm adversaries must acquire trusted credentials, and in order to acquire those trusted credentials, they must kidnap or corrupt trusted insiders. This is the driving ambition of administrative computer defense, even though, in reality, it usually falls short of this aim. And, as is the case for application designers, this central defensive ambition is dictated primarily by what institutional actors can control. Application designers can dictate what capabilities are afforded by the applications they use under what conditions, so their defensive function centers on distinguishing between malicious and legitimate capabilities. Organizations have greater control over physical security—including access to machines on protected premises, personnel screening, and credential issuing procedures—so their defensive role centers on using that to reinforce computer security by trying to force adversaries to tackle physical security measures and personnel screening processes in order to achieve desired computer capabilities.

6.1.1 Tailoring Application Restrictions

Sometimes the distinction between legitimate and malicious capabilities varies according to context—for instance, some organizations may require regular communication with outside, unknown entities, while others may view that capability as a threat and wish to instead limit communication to authenticated users within their own organization. Similarly, certain types of authenticated activity may be viewed as more or less suspicious depending on the setting—periods of unusually high-volume email, for instance, may be routine at places that send large-scale legitimate mailings, new and unknown devices or IP addresses may not trigger any strong suspicion for users who travel frequently or regularly test out new machines. So part of implementing access defense as an organization involves building on the analysis of potentially malicious and legitimate capabilities done by applications designers to determine how well the distinctions and indicators decided on by the designers actually mirror those dictated by the organization's threat model and function.

Where there is a mismatch between the types of malicious capabilities and signal selected by application designers, organizations may either choose not to implement a given application or tailor it to their needs, depending on the extent to which designers have made it customizable. These customizations may apply at either the individual user or institutional level—for instance, many browsers allow users to choose whether or not to enable JavaScript, or manage the default list of trusted SSL certificates. Application designers may find it useful to allow users and organizations to define some of the distinctions between legitimate and malicious activity for themselves; however, enabling this kind of flexibility can give rise to additional risks by creating pathways for adversaries to enable malicious capabilities or remove useful signals under the guise of tailoring an application to fit legitimate needs.

There is a tension between allowing users to define their own security parameters and enabling attackers to change those parameters, or remove defensive signals. One approach to dealing with this is enabling users to customize applications only to restrict more capabilities and offer more signals than the designer initially did by default. Another approach is accepting that the same applications may not suit the needs of all organizations and users, and focusing on designing different applications to meet these different actors' requirements, rather than a single application that can be tailored to everyone. (This premise, however, goes against the motivation of many application designers, who, understandably, want their programs to be used as widely as possible.)

The defensive decisions made by people and organizations using applications, like those made by designers, are guided in part by the ways those users wish to interact with outsiders, or people without trusted credentials. That desired level of interaction should guide managers' choice of applications as well as, where appropriate, customization of application defenses. In the context of a defensive mission that hinges on reducing the space of computer-based risks to threat vectors governed by physical security, this essentially means assessing what capabilities an organization is comfortable granting to people whom it has never encountered in person or subjected to personnel screening or training procedures. As with uncredentialed application capabilities, it is likely that many, if not most, organizations will see fit to severely restrict the circumstances under which these interactions with unknown users may take place—but the extent to which that is the case and the precise nature of those interactions may vary.

6.1.2 Protecting Authentication Credentials

The crucial access defense role of organizations lies in protecting the authentication credentials that restrict who can access potentially malicious capabilities. For organizations that link credentials to users' real identities (e.g., employers, governments, schools) this includes ensuring that credentials are issued to the correct people, and for all organizations, it means trying to issue credentials that cannot be easily stolen, guessed, or imitated by anyone. This form of defense is again about trying to tie credentials to a particular user in a way that makes them difficult to replicate or extract without physical access to that person. Authentication also presents a relatively rare opportunity for defenders to implement multiple, completely overlapping lines of defense in the context of a computer system through multi-factor authentication. Non-credential based defenses, i.e., those that restrict users' capabilities rather than requiring them to produce credentials in order to exercise those capabilities, overlap in a slightly different fashion. That is, because each one must target a slightly different capability, they overlap only insofar as they restrict sub-capabilities of some larger, overarching malicious capability (e.g., impersonation or installing malware). Such defenses protect against slightly different classes of behavior—and often do not block classes of behavior that are essential to the attacker's aim (as appears to be the case, for instance, with MIT's password complexity requirements and one-year expiration policy defenses)—unlike multiple credentials which can all protect exactly the same set of capabilities.

This means that an adversary wishing to take advantage of any of those capabilities must compromise all of the protected credentials—in other words, there is automatic and complete overlap, in the language of defense in depth. And in the capability-based environment of access defense, where different defenses typically target slightly different types of behavior which may or may not be essential to attackers, that is an unusual and valuable feature. That does not mean multi-factor authentication systems cannot be compromised—they can, and indeed, have been. Mechanisms for compromising two-factor authentication schemes that rely on both a password known to the user and a one-time code transmitted by phone or other physical token include compromising each of the victims' credentials individually, and tricking victims into entering all of the credentials into forged or compromised authentication interfaces. FBI Special Agent Elliott Peterson (2014) describes one such scheme, as implemented using the GameOverZeus (GOZ) bot, in his declaration to support a restraining order against use of the bot's infrastructure, writing:

> GOZ . . . is sufficiently advanced that its operators can harvest both static and variable information in real time from the victims. Specifically, after the initiation of a man-in-the-middle attack, the GOZ operators will be queried by the bank for the variable portion of the victim's two factor initiation. The GOZ operators pass this query on to the victim in the form of a fictitious web injection. While the victim thinks that the information is being sent to the bank, it is instead sent directly to the GOZ operators. After stealing victims' personal information, the defendants use the stolen credentials to log into victims' bank accounts and to initiate fraudulent

> electronic funds transfers from the victims' banks.

This approach calls into question the independence of multiple authentication credentials: if the credentials are all entered into a single interface then it may not actually be necessary for adversaries to independently compromise each credential individually—though that has also been done, using cellphone malware to intercept text messages containing authentication codes (Danchev, 2009). This implies that a key component of protecting credentials is actually the protection of the authentication interface—that is, increasing the work required to impersonate or compromise that interface, and providing signals that help users verify the legitimacy of the interface demanding their credentials.

Defending authentication credentials is complicated because, while the credentials themselves may be essential to attackers in some cases, there are a number of replaceable methods for obtaining them—and only some of those methods can be effectively controlled by the organization issuing the credentials. Those organizations can, generally, control how easily credentials can be guessed, through the implementation of complexity requirements and internal attempts at guessing as well as limits on how often users can enter incorrect guesses. But guessing is easier to protect against than imitation and interception because it provides signals of the attacker's work within the context of the protected system—that is, the very act of guessing, or entering an incorrect credential, signals potentially malicious behavior. The work required to intercept or imitate an authentication credential does not necessarily provide any such signal to the authenticating organization, even if the work involved is considerable. Consider, for instance, a German group's approach to spoofing the iPhone's TouchID fingerprint reader (Greenberg, 2013):

> First, the fingerprint of the enrolled user is photographed with 2400 dpi resolution. The resulting image is then cleaned up, inverted and laser printed with 1200 dpi onto transparent sheet with a thick toner setting. Finally, pink latex milk or white woodglue is smeared into the pattern created by the toner onto the transparent sheet. After it cures, the thin latex sheet is lifted from the sheet, breathed on to make it a tiny bit moist and then placed onto the sensor to unlock the phone.

There is a lot of effort that goes into imitating that fingerprint—just as there was presumably a lot of work that went into decrypting the PIN numbers in the TJX breach perpetrated by Gonzalez—but all of that work is invisible to the defending organization. Similarly, the work of intercepting credentials and forging authentication interfaces often relies on the use of unauthenticated application capabilities (e.g., phishing emails, web injections), and while organizations may exercise some control over those capabilities by selecting and tailoring applications, the extent to which they are signaled or restricted is often in the hands of the application designers. In other words, the behaviors that enable credential interception and imitation—as opposed to those that enable guessing—may not be behaviors that the issuing organizations are able to restrict.

MIT's struggle with compromised account credentials also illustrates this challenge: plagued by routine account compromises, IS&T implemented password complexity and expiration policies in 2013 to restrict how easily passwords could be guessed or stolen credentials could be reused. However, those measures' minimal (even counterproductive) impact on the rate of compromised accounts indicates that those capabilities were easily replaced—in this case, by email phishing and fake login websites—two access avenues IS&T has minimal control over. The challenges of defending against credential interception and imitation in an institutional context have led to a slightly different defensive approach that organizations can control: devaluing individual stolen credentials, rather than trying to make them more difficult to steal. This is essentially the model of multi-factor authentication: instead of providing individual credentials with stronger protections that make each one harder to steal—which organizations may not be poised to do—they can ratchet up the work required of adversaries by forcing them to steal multiple credentials. The perfect overlap of authentication credentials makes this an effective defense in depth construction, but the need for a centralized authentication interface may undermine that depth to some extent.

Multi-factor authentication is not just a tool for increasing adversaries' work; it can also serve as a signal to legitimate users of malicious activity even before an adversary has successfully authenticated (in contrast to the application design defenses that signal malicious activity after-the-fact). This means using the successful entry of one credential to signal to the person to whom it was issued that authentication has been initiated using their credentials—for instance, if users receive a one-time passcode via text message that message functions both as a credential and a signal that someone is attempting to login to their account. Similarly, if additional authentication credentials are delivered via email or other applications, the receipt of those messages can serve as signals in the event of attempted credential theft.

Authentication credentials provide an interesting defensive chokepoint for security breaches because they are so ubiquitous and so commonly exploited by attackers. Stolen credentials featured prominently in the TJX, DigiNotar, and PLA espionage incidents—though in the first two cases, where credential theft was an intermediate rather than initial stage of the attack, there was relatively little focus on the role of credentials in the ensuing investigations. TJX was lambasted for its poor wireless security, DigiNotar for its flawed firewalls, suggesting a more general focus on how incidents begin and the earliest possible lines of defense, rather than their common, recurring stages or defensive chokepoints. Still, some organizations' failures to adequately protect authentication credentials have been singled out for scrutiny and criticism in the aftermath of security breaches—particularly when acquiring those credentials is the attacker's first move. For instance, in 2012, attackers used phishing emails to steal the credentials of an employee at the South Carolina Department of Revenue and then used those credentials to log in to the Department's servers remotely and exfiltrate approximately 74.7 GB of data. The subsequent investigation and media coverage focused heavily on the lack of two-factor authentication at the Department of Revenue, with headlines asserting "$25K upgrade could have prevented hacking, panel told" referring to a two-factor authentication scheme requiring a code from a physical token as well as a password, and members of the South Carolina state senate investigation panel echoing the claim—that for the price of a $25,000 physical-token two-factor authentication system, the entire incident could have been avoided—in interviews (Smith, 2012). Media coverage of a 2014 breach of JPMorgan Chase followed a similar narrative, with investigators and reporters focusing on a network server at the bank that had not been upgraded to require two-factor authentication (Goldstein, Perlroth, & Corkery, 2014).

Failing to implement multi-factor authentication is not the only thing organizations are taken to task for in the wake of breaches—when several celebrities had naked photos stolen from their Apple iCloud accounts in 2014, critics blamed Apple's failure to rate limit unsuccessful log-in attempts in order to prevent adversaries from guessing passwords by brute force (Fung, 2014). These critics are not wrong to point out that Apple (as well as the South Carolina Department of Revenue and JPMorgan Chase) could have more effectively protected authentication credentials, but the tenor of their criticism implies that these stronger protections would have prevented the attacks, rather than simply rerouting the attackers through different pathways, and that, amongst the myriad different ways in which an attack like the one perpetrated against South Carolina might have been defended against—phishing protections, remote access restrictions, limits on data exfiltration—it was the absence of multi-factor authentication that most clearly indicated negligence and inadequate security. This tendency to single out institutions' earliest failures in the sequence of attacks, whether those include adequately protecting authentication credentials or encrypting wireless networks, is part of a larger theme of these postmortems, in which blame is most often laid on the particular company or organization whose resources were compromised, and attention is focused most strongly on the access capabilities and defenses in place, rather than later-stage harm mitigation efforts. When these institutions function as harm defenders, as in the case of Spamhaus or the targets of PLA Unit 61398, we appear much more likely to view them as victims, rather than negligent defenders. In this context, where mental models of defense and defenders are limited largely to access and organizations, it makes sense that the access defenses implemented by those organizations—namely multi-factor authentication and mechanisms to prevent guessing credentials—come in for particular scrutiny. However, organizations are only one set of actors involved in access defense—and access defense is only one of the defensive roles they are equipped to fill.

6.2 Defending Against Intermediate Harms

While capabilities such as sending phishing emails or connecting to unprotected wireless networks are not absolutely necessary for inflicting any broad class of harm, some computer capabilities are, in fact, crucial to the infliction of certain types of harm. This is particularly true for digital harms, or harms that are inflicted solely through the manipulation of computer data and services. Defending against these crucial capabilities, or essential intermediate harms, that straddle the access-harm divide is primarily a job for organizations that own and operate computer networks both because of those organizations' visibility into—and ability to restrict—network traffic. That visibility is shaped, in turn, by what an institution views as its threats and what it chooses to look for and classify under the heading of security problems.

6.2.1 What Security Looks Like to an Institution

From 2004 to 2014, MIT's records suggest a gradual evolution in both the types of threats facing the university and the notion of what constitutes a security incident. The early years of the security ticket queue are littered with reports of students printing too much or eating in the campus computer labs, complaints about email and online harassment, and complaints about online content hosted by MIT users. The constant themes of the MIT security queue over the past decade—and the strong focus of the incidents reported to the university in recent years—are compromised user accounts and compromised hosts.

The compromise vector, or access capabilities exploited to achieve the compromise, may vary across the years and incidents, but malicious actors' reliance on using organizations' trusted accounts and machines has only become more pronounced with time. More specifically, attackers' ability to send outbound traffic from these accounts and machines has played a central role in many of the incidents MIT sees, and it is this particular capability that is so closely tied—and essential—to many classes of harm that it functions as a crucial intermediate harm. This is true not just at MIT, but also in several of the other cases described in Chapter 4. The TJX breach depends on Gonzalez' team being able to exfiltrate large volumes of payment card information from the company's servers; the PLA Unit 61398 espionage efforts are similarly focused on retrieving sensitive information from the targeted systems; and the Spamhaus denial-of-service attacks required control of a bot, or the ability to send outbound traffic from thousands of machines.

Financial fraud, espionage, and digital service disruption attacks all generally share this dependency on the capability to send outbound traffic from protected machines. (Denial-of-service attacks could conceivably be perpetrated by an adversary who actually owned and operated thousands of computers himself, instead of using a bot comprised of other people's machines, but in practice this rare—and it drives up the cost of initiating such an attack considerably.) In some cases, this is a capability that application designers may be able to restrict—for instance, by flagging unusually high volumes of outgoing email messages—but in others it may be more effectively restricted by monitoring network connections and traffic rather than individual applications, implying a strong defensive role for the organizations that operate these networks.

6.2.2 Outbound Traffic

If the ability to send outbound traffic to other machines and accounts that are not trusted by, or known to, an organization is, in some sense, what defines an account or host as being compromised, then defending against these intermediate harms means focusing specifically on this capability. These defenses lie somewhere between access and harm defenses to the extent that they are both focused on restricting computer capabilities, in the spirit of access defense, but are also intrinsically tied to the prevention of harm in many cases (particularly, espionage and denial-of-service attacks, in which the successful sending of outbound traffic is essentially the direct cause of the intended harm). So, restrictions on outbound traffic have in common with harm defenses the fact that, if they are effective, they can protect against entire classes of harm and it does not matter what other capabilities an adversary has acquired in the context of a computer system—stolen credentials, or successfully delivered malware are worthless if they do not enable the attacker to initiate outbound information flows. However, protecting against malicious outbound traffic also exhibits some features of access defense—namely the challenges of distinguishing between malicious and legitimate outbound activity.

Distinguishing between malicious and legitimate outbound activity can be aided by a variety of factors, that may also serve as signals to legitimate users, including the volume of traffic, its destination, and the repetition or regular patterns and timing with which it is sent. Restricting the volume of data that can be sent, or the ease with which it can be sent to unknown recipients, may increase the work required of adversaries to exfiltrate information. This can also be achieved by requiring extra layers of independent approval (or credentials) to send outbound traffic from a protected network—particularly in large volumes or to new recipients—separate from the credentials and restrictions placed on other capabilities. Dedicated monitoring and approval processes devoted solely to exfiltration help shore up the independence of these defenses because their capabilities can be severely restricted to serve only a single function, providing fewer opportunities for them to be compromised through other capabilities serving additional purposes. Returning to the broader institutional goal of reducing computer access capabilities to physical access, compromising these dedicated layers of approval or additional credentials would ideally require an adversary to gain physical access to a monitoring machine or person.

Defending against outbound traffic flows may also involve signaling legitimate users about unusual traffic patterns and destinations, or requiring them to verify that they intentionally initiated certain outbound connections, to help them identify and address potential malicious activity. The restrictions placed on outbound traffic may force attackers to do extra work that can, itself, serve as an additional set of signals to organizations. For instance, if organizations use firewalls to restrict outbound connections to only certain servers, or allow outbound data to be sent in limited volumes, then activity that involves moving or copying information to those servers and compressing it or sending it at a steady, gradual rate may indicate malicious exfiltration. Forcing attackers to stage information in this manner before it can be successfully sent to an outside destination may provide organizations with useful signals of intended exfiltration even before it actually occurs. In this light, the role of defenses like firewalls is not just to prevent some forms of infiltration and exfiltration but also to channel those behaviors through specific and identifiable paths that can then be monitored for signals of malicious activity.

In contrast to the misguided defense narrative around not letting attackers "in" to protected machines, this defensive approach actually focuses on not letting them "out"—or, rather, not letting them acquire the capability to send outbound traffic from protected systems. In fact, it suggests a more radical reframing of the notion of access defense against certain types of harm, one in which getting access to a computer or a system is actually defined by being able to send information out of it. This ability to link an access capability (sending outbound traffic) to particular classes of harm (espionage, disruption of digital service) is what defines a crucial intermediate harm—more than a helpful computer capability that gets an adversary one step closer to his end goal, these serve as essential, irreplaceable platforms for reaching that goal.

Outbound traffic is a platform not just for harming the organizations that have failed to restrict it but also others. MIT, for instance, regularly receives complaints from other universities and organizations that they are experiencing denial-of-service attacks from MIT IP addresses, or receiving spam and phishing emails from MIT's domain. The same is true, though to a lesser extent, of espionage incidents, which often rely on exfiltrating sensitive information through an organization that is trusted by the victim but has less stringent outbound traffic restrictions, and from there sending it on to servers controlled by the adversary, which might otherwise trigger suspicion due to location or ownership. PLA Unit 61398, for example, used American universities as an intermediary for data exfiltration from targeted U.S. companies. In defending against unintended outbound traffic, organizations therefore protect not just themselves but also others from some classes of harm. This makes it all the more crucial as a line of defense in a societal context—but, at the same time, may also make it is less easily justifiable as a form of direct protection within the context of an individual institution.

6.3 Institutional Harm Defense

Outbound traffic can directly cause certain types of harms—digital harms—but for other classes of harm, including financial and physical, the actual damage is incurred outside the context of a computer network, and sometimes outside the scope of the breached organization, as well. In these cases, harm defense means intervening after attackers have successfully acquired some access capabilities to ensure that those capabilities cannot be used for financial gain or physical disruption. In this regard, harm defense is the class of computer system defense that most strongly relies on a sense of sequential attack phases—that there is some final step, or set of last steps, attackers must successfully undertake outside the confines of a computer network to inflict certain kinds of harm (e.g., manufacturing fraudulent credit cards, or manipulating the physical operations of a critical system). This idea returns to the notion that the attackers' options narrow as they get closer to their end goal—that there may be many ways to initiate an attack motivated by a particular aim, but only one way to finish it, or one final goal. So access defense efforts are hindered by the myriad capabilities open to adversaries and the ease with which those adversaries can substitute those capabilities when they encounter defenses, making it difficult to ascribe specific sequences of steps to the access stages of attacks—and, accordingly, difficult to dictate the order in which adversaries will encounter certain defenses, or even to ensure that they will encounter a particular defense at all. Harm defenses, by contrast, are intended to protect against the essential, unsubstitutable final stages of attacks—the stages that necessarily occur in a certain sequence after adversaries have already acquired certain capabilities or information.

6.3.1 Overlooking Harm Defense

Though harm stages of attacks are necessarily the most consistent and sequential, they are often glossed over or left out of stage-based analyses of security breaches, an omission which reflects a more general over-emphasis on access defense and disinterest in defenses that are not within the control of targeted organizations. For instance, Skoudis and Liston (2006) split attacks into five general stages—reconnaissance, scanning, gaining access, maintaining access, and covering tracks—and Hutchins et al. (2011) propose a "kill chain" model, which divides attacks into seven stages: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. Mandiant's APT 1 report on PLA Unite 61398 identifies seven slightly different steps in a more specific set of espionage attempts perpetrated by the Chinese government: reconnaissance, initial intrusion into the network, establish a backdoor into the network, obtain user credentials, install various utilities, privilege escalation/lateral movement/data exfiltration, and maintain persistence. Lowry and Maughan (2003), by contrast, estimate what percent of an attacker's time is spent on each of five different elements of the attack process: intelligence/logistics (40%), live/system discovery (5%), detailed preparations (30%), testing and practice (20%), and attack execution (5%).

Lining up these different stage-based divisions side by side, as shown in Table 6.1, reveals considerable divergence, especially in the later stages following the initial access. For instance, the two stages that Skoudis and Liston (2006) identify after "gaining access" are maintaining access and covering tracks, while the Hutchins et al. (2011) kill chain includes four fairly different stages—exploitation, installation, command and control, and actions on objectives—following the delivery phase, and the Mandiant report states that after the initial intrusion attackers take five other steps: establish a backdoor, obtain credentials, install utilities, data exfiltration, and maintain persistence. In the vocabulary of Lowry and Maughan (2003), by contrast, all of these stages (including the initial access) are included in the final stage—attack execution. One consequence of this uncertainty and divergence around defining the later stages of attacks may be an emphasis on defenses that operate at the earlier stages, where there is greater consensus. Many of these analyses lead their creators to conclude that it is preferable to stop attacks sooner rather than later. "Defenders must be able to move their detection and analysis up the kill chain," Hutchins et al. (2011) write, though they add that it is important to maintain protections at all stages. Lowry and Maughan (2003) feel even more strongly, writing that by the time an attacker reaches their final stage (attack execution) it is often too late to stop it. "Our experience has been that if the adversary is allowed to reach this stage unhindered, then the attack will succeed," they note. While these findings

Table 6.1: Comparison of different divisions of attacks into stages.

Skoudis and Liston (2006)	**Hutchins et al. (2011)**	**Mandiant (2013)**	**Lowry and Maughan (2003)**
Reconnaissance	Reconnaissance	Reconnaissance	Intelligence/logistics
Scanning	Weaponization	Initial intrusion	Live/system discovery
Gaining access	Delivery	Establish backdoor	Detailed preparations
Maintaining access	Exploitation	Obtain credentials	Testing and practice
Covering tracks	Installation	Install utilities	Attack execution
	Command and control	Privilege escalation/lateral movement/data exfiltration	
	Actions on objectives	Maintain persistence	

do not necessarily apply to all attackers—the researchers focus specifically on more persistent sets of intruders with specific targets and significant resources—they do reinforce the idea that there is little value in trying to bolster defenses towards later stages of the attack.

One reason these examples of attack stage analysis may lead to this conclusion that defenses should be leveraged at earlier stages is that the later stages of attacks are more varied, and therefore more difficult to define and identify defenses for using a generalized model. Another possible reason is that defenses at some of these later stages—after attackers get their hands on sensitive information—may be less technical in nature and depend more on policy-based and law enforcement interventions beyond the control of a defending organization. A defense strategy limited to or focused primarily on institutions and technical controls may therefore be better suited to protecting access pathways to computer system information and resources than they are to devaluing those assets after they are accessed, or preventing them from being used successfully by attackers. Technical tools, including encryption and intrusion detection systems, certainly play a role in hindering attackers at these later stages, but crackdowns on fraudulent credit card manufacturers, or policies limiting data storage and collection, may be equally—if not more—important, in some cases. At the later stages of attacks, much more than at the earlier stages, the possibilities for defense depend a great deal on what the attackers ultimately want to do, and this requires going beyond a single model of the attack chain.

The differences in the proposed attack sequences already seem to suggest that after attackers gather information about their targets and access it in some manner, there are several different paths they may choose depending on who they are, what their goals are, and the nature of their targets. The kill chain comes closest to acknowledging the challenges of defining a single set of stages that encompasses all varieties of attacks with its final stage. "Only now, after progressing through the first six phases, can intruders take actions to achieve their original objectives," Hutchins et al. (2011) write of the seventh kill chain phase, actions on objectives. They note that although data exfiltration is the most common such objective, "violations of data integrity or availability are potential objectives as well. Alternatively, the intruders may only desire access to the initial victim box for use as a hop point to compromise additional systems and move laterally inside the network." Depending on what their objectives are, some attackers may take pains to cover their tracks, per the Skoudis and Liston (2006) model, others may take the Mandiant route of trying to obtain credentials—others may do neither. Different objectives impose different sequences of stages on attackers. In other words, there is not a single pattern that all attacks conform to, there are several. These patterns can be organized around the overall objectives of the attackers, or the class of harm that they ultimately aim to inflict. Attempting to define a set of attack phases that is sufficiently general as to encompass all of these objectives yields stages so vague as to offer little guidance to defenders, particularly when it comes to harm defense.

The harm infliction stages of attacks and corresponding defense opportunities that are often overlooked in attempts to formulate a single, all-encompassing pattern for attacks do not always lend themselves to defensive interventions by targeted institutions and organizations. Often, restricting the scope of these harms that go beyond these organizations' computer systems also requires defenders beyond the bounds of those organizations. Still, there are classes of non-digital harm—particularly physical harm—where organizations can play an important role in defending against damage, as well as access capabilities

6.3.2 Disruption of Physical Service

Organizations that operate physical systems can monitor changes in those physical services and restrict dangerous or harmful adjustments through physical checks and monitors. The emphasis of this defensive strategy comes back to requiring attackers to physically access these systems in order to harm them, transferring the burden of protection to physical security mechanisms and defenses. Of course, those physical defenses can still be breached—but by forcing adversaries to bypass physical barriers as well as digital defenses, organizations may increase the work required of them. The Stuxnet malware, which was used to increase the frequency of centrifuges at an Iranian uranium enrichment plant in Natanz from 1,064 Hz to 1,410 Hz, is one of very few known examples of physical damage intentionally inflicted solely through the use of computer capabilities. The speed fluctuations appeared to induce excessive vibrations that caused the rotors of roughly 1,000 IR-1 centrifuges at the Natanz Fuel Enrichment Plant (FEP) to break.

While a great deal remains unknown about exactly what defenses were in place at FEP, it seems clear that Stuxnet required physical delivery to the plant (by means of an infected USB drive) and also bypassed the frequency converters that controlled the centrifuges' motor speed to "shut off the frequency converters' warning and safety controls aimed at alerting operators of the speed up or slow down" (Albright, Brannan, & Walrond, 2010). Whether there were other defensive mechanisms in place to limit the extent of the damage, beyond the scope of Stuxnet's control, remains unclear. For instance, Albright et al. (2010) speculate:

> Other control systems may inhibit Stuxnet from destroying centrifuges during an attack sequence. For example, if a centrifuge rotor assembly were to run down with the uranium hexafluoride inside, the rotor could become unbalanced and "crash," or break. As a result, in the event of a malfunction, the safety systems are designed to quickly empty the centrifuges of uranium hexafluoride. . . . For example, each IR-1 centrifuge has a vibration sensor, and any vibration over a certain tolerance can cause the control system to isolate the centrifuge and transfer the uranium hexafluoride gas in the cascade to a dump tank. The reason is that the IR-1 is vulnerable to vibrations that if left unchecked can destroy the centrifuge. The shock wave from a crashing centrifuge can destroy other centrifuges in the cascade. In addition, the control system may automatically reduce the centrifuge's speed in a controlled manner. If this command originated in another control system, would Stuxnet override this command? Symantec stated . . . that its researchers found no code in Stuxnet that would block the dumping of uranium hexafluoride from the centrifuges. Thus, it remains unclear whether safety systems independent of the control system targeted by Stuxnet would intervene to save the centrifuges or reduce the number destroyed.

In other words, there may have been other safeguards built into the centrifuges that Stuxnet did not anticipate or override and that served to contain the damage. Too little is known about the FEP infrastructure to say definitively whether this was the case—but the overriding lesson of the incident is that more lines of dedicated, independently operating defenses that do not communicate with each other create more work for adversaries wishing to impact physical systems. In fact, just overcoming the existing defenses to target the FEP centrifuges appears to have required considerable work and resources—leading to speculation early on that Stuxnet's sophistication made it a "far cry from common computer malware" and the likely work of government actors (Sanger, 2010). The challenges of circumventing physical safety controls are reinforced by the apparent rarity of physical harm resulting from computer-based attacks. In January 2015, *Wired* reported that attackers had caused "massive" unspecified damage at a German steel mill by manipulating a blast furnace through access to the plant's business network (obtained via phishing emails); the piece appeared under the headline "A Cyberattack Has Caused Confirmed Physical Damage for the Second Time Ever" (Zetter, 2015).

In some sense, defending against physical system disruption recalls some of the same challenges as defending against malicious application capabilities: both require being able to identify specific types of behavior—whether of a blast furnace or an email user—that ought to be restricted. In the case of physical services, however, it is often easier to define unambiguously malicious or harmful behaviors, and defenders can operate within a much broader sphere of independence when implementing defenses for physical services because defenses need not rely on the same computer, operating system, network—or even, necessarily, on computers at all. That is not to say protecting against physical service disruptions is easier, exactly, than defending against access capabilities, but it does offer organizations a much clearer picture of precisely what outcomes need to be avoided and serve no potential legitimate purpose—as well as a much more independent and isolated set of potential dedicated defenses.

6.3.3 Disruption of Digital Service

In contrast to defending against disruptions of physical services, there is relatively little an organization can do, left to its own devices, to defend against disruptions of digital service and denial-of-service attacks. Defending against outbound traffic and restricting recursive access to DNS servers can help an organization protect its resources from being used in such attacks, but does not actually protect itself from being targeted by them. For the most part, harm defense for digital service disruption requires organizations to enlist third parties—as Spamhaus did with CloudFlare, to filter incoming traffic, and MIT did with Akamai, to provide caches of the content hosted at mit.edu to off-campus users. In both cases, the third-party organizations provide an extra intermediary step for adversaries in achieving their aims. For Spamhaus, this extra step entailed the traffic filtering that CloudFlare performed prior to that traffic actually reaching Spamhaus' servers, so attackers could no longer directly flood Spamhaus and instead had to go through an intermediary screening process. MIT's contract with Akamai could also potentially mitigate denial-of-service attacks, by offloading heavy volumes of traffic to Akamai's significantly larger server capacity, but it also addresses digital defacement, of the type MIT encountered following Aaron Swartz's suicide, by introducing some delay in how quickly such changes to the university's web presence are propagated. In both cases, the targeted organizations responded to disruptions in digital service by finding third parties that could create additional intermediate stages for attacks that would otherwise have had no final harm-infliction stage their victims could control once adversaries had gained the necessary access capabilities.

Organizations can also defend against the harms imposed by denial-of-service attacks themselves, without the assistance of third parties, for instance by blocking incoming traffic from certain IP addresses or high-volume, repetitive activity. But this can interfere with non-malicious behavior when there is a legitimate reason for high-volume activity. In June 2009, following the death of Michael Jackson, Google misinterpreted the spike in searches for the singer as an attack and briefly blocked them, displaying the message: "We're sorry, but your query looks similar to automated requests from a computer virus or spyware application. To protect our users, we can't process your request right now" (Krazit, 2009). But in situations where legitimate high-volume activity is sufficiently rare or inessential, the benefits of rate limiting defenses may still outweigh any potential side effects.

Security incidents that have no physical consequences for the victims and offer no financial benefits to the perpetrators encounter a much less independent set of defenses because, by definition, all of the applicable defenses will have to exist within the context of a computer system. Furthermore, access to that system, in the form of whatever capabilities the perpetrators require to achieve their aims, is the only goal of such incidents, so there are no later stages at which they can be stopped and their intended harms prevented or mitigated—no equivalent of inserting manual centrifuge vibration sensors or tracing deliveries of cash back to Gonzalez. This is why inserting new, independent actors like CloudFlare, as well as new intermediate stages that occur before harm is inflicted on the targets, such as the redirection of Spamhaus traffic through CloudFlare's data centers or MIT.edu visitors through Akamai's content caches, can have useful defensive effects: it helps approximate the defensive opportunities of attacks whose impact extends beyond the context of computer systems, both in terms of degree of independence of defenses and the number of possible stages at which those defenses can operate.

6.3.4 Espionage

Espionage, like disruption of digital service, offers relatively few late-stage opportunities for harm prevention. Just as application designers can build in some pre-emptive harm defense by designing for routine deletion, organizations may also be able to pre-empt the harm that can be done by espionage by limiting how much information they store and for how long. Encrypting sensitive information may also add to the amount of work required of attackers to make use of stolen data, though, as in the case of the TJX breach, the work of decryption can generally be done outside the context of the targeted organization's systems, making it more difficult for victims to detect or interrupt. (Alternatively, encryption can serve as a form of access defense by forcing adversaries to acquire the necessary credentials, or keys, that enable decryption.) One challenge to using encryption as a defense against espionage is that it can hinder attempts to monitor and restrict outbound traffic by making it more difficult for organizations to recognize attempts to exfiltrate sensitive information and easier for adversaries to disguise such attempts.

Restricting outbound traffic is essentially organizations' last opportunity to prevent the harms inflicted by espionage, unless the attackers' ultimate aim is to share the stolen information publicly—either for purposes of humiliation or to devalue the victims' intellectual property—in which case organizations can sometimes try to stem the spread of that information even outside the context of their own systems, though this usually depends on third-party cooperation or support. For instance, in 2014 when large volumes of sensitive information including email archives, financial information, scripts, and movies were stolen from Sony Pictures Entertainment and released publicly, the studio reportedly initiated denial-of-service attacks directed at the sites that were hosting its stolen data and planted fake torrent files so that users who believed they were downloading the stolen information instead spent hours downloading empty files (Chmielewski & Hesseldahl, 2014). The studio also attempted a less technical, law-oriented means of stemming information flows, sending letters to news organizations demanding that they delete the data and cease to report on its content. “If you don’t comply with this request,” Sony lawyer David Boies wrote in the three-page letter, Sony “will have no choice but to hold you responsible for any damage or loss arising from such use or dissemination by you” (Cieply & Barnes, 2014). Attempts by organizations to quell the further distribution of information after it has left the confines of its own systems rely heavily on the goodwill of others (e.g., journalists) or the support of government actors in criminalizing the indirect propagation of stolen information. In a country like the United States, with strong policy protections for journalistic freedom, this is therefore a relatively ineffective mode of espionage harm defense. In countries which place greater restrictions on speech and the press, however, it might prove significantly more useful to organizations defending themselves against public-facing espionage harms.

6.3.5 Financial Loss

If organizations are not well equipped to defend against harm by stemming information flows without the support of third parties and government actors, they are even less able to address financial harms and money flows. As with espionage harms, organizations may be able to preempt financial harm by minimizing the amount of sensitive financial data they store and carefully monitoring outbound traffic for suspicious exfiltration attempts, but that is essentially where their unilateral ability to prevent financial harm ends. That harm may take several different forms—it may be inflicted directly on the targeted organization, through the theft of valuable intellectual property, or instead be directed at that organization’s customers or clients whose data is stored on the targeted systems. Either way, there is relatively little organizations can do themselves to prevent these losses following the successful theft of targeted information on their systems.

Payment processors and law enforcement officials, on the other hand, may be well poised to address these later stages of financially motivated attacks—and targeted organizations may be able to assist, or at the very least initiate, those efforts to some extent by reporting relevant information about the nature of such attacks and who might be affected by them. Preventing malicious actors from profiting off their actions in particular, and harm defense more generally, is not easily implemented by individual organizations. At an institutional level, therefore, harm defense primarily entails cooperating with (or hiring) the requisite third parties who are able to address particular classes of harm that go beyond the reach of a single company or organization.

From an individual organization’s perspective there is a significant difference between harm defense that means defending itself against harm versus defending others (perhaps customers or partners, but also potentially total strangers) by restricting capabilities in the context of its own system. For some classes of harm—particularly disruption of digital service, but also potentially espionage and financial fraud—organizations may actually be better poised to achieve the latter goal and forced to rely on third parties for the former. This misalignment of incentives and capabilities is part of what makes harm defense challenging—and why it often requires the involvement of policy-makers and governments. In contrast to access defense, where individual actors—such as application designers and managers—have significant autonomous control over defenses in a particular context but may be faced with an enormous range of potentially malicious capabilities to defend against in that context, harm defense implies a much narrower, more specific and clearly malicious set of attack stages for defenders to hone in on, but requires the coordination of a wider range of different actors in order to do so effectively.

Chapter 7

Policy as Defense

Application designers wield significant defensive power in determining how much work adversaries must do to attain certain capabilities, and managers can tailor and augment those lines of access defense by trying to tie authentication credentials to physical people and resources and restricting inbound and outbound network traffic, but both of these classes of defenders are limited in their ability to enact harm defenses on the final stages of certain types of attacks that extend beyond the context of their particular applications and networks. Policy-makers and law enforcement actors are better suited to some elements of harm defense, particularly those related to addressing harm externalities and stemming criminals' ability to profit off their activity. Policy-makers are particularly well poised to undertake harm defense because, unlike application designers and organizations, they are able to restrict and target not just the behavior of attackers but also that of defenders.

7.1 Targeting Attackers Versus Targeting Defenders

Cybersecurity policies can either target attackers—the criminals or malicious actors who are responsible for developing and driving threats, or defenders—the mix of organizations and individuals with the ability to help prevent those bad actors from achieving their ultimate aims. Laws like the Computer Fraud and Abuse Act (CFAA) in the United States, or the Computer Misuse Act in the United Kingdom, are examples of policies that target attackers: they criminalize certain activities (e.g., unauthorized access to computers), enabling law enforcement authorities to prosecute and punish offenders directly. Notably, these policies that directly target attackers—rather than going through defender intermediaries—tend to focus primarily on criminalizing access, leading to considerable controversy in the legal world around what constitutes "unauthorized access" or "access in excess of authorization" in the context of a computer system. (The CFAA, for instance, defines the latter thus: "to access a computer with authorization and to use such access to obtain or alter information in the computer that the accesser is not entitled so to obtain or alter," implying a fairly broad set of capabilities—centered on data exfiltration and editing—that constitute access.)

It makes sense that policies designed to directly target and punish malicious actors would focus on the access stages of attacks, rather than the harm, since those harms are, for the most part, already covered by existing laws—with the possible exception of disruption of digital service. Denial-of-service attacks present a particular challenge for policies that target attackers because the harm they impose is not clearly covered by existing laws—but neither is it clear what element of the access that enables them is covered by laws like the CFAA or Computer Misuse Act. For instance, at least two people allegedly involved in orchestrating the attacks on Spamhaus have since been arrested and charged under the Computer Misuse Act—but under that statute it is unclear whether their crime was in flooding Spamhaus' web servers (harmful, but not unauthorized access) or using compromised hosts to do so (access to those hosts to send outbound traffic was unauthorized, but not directly harmful to the hosts themselves). Similarly, in 2013, when thirteen people associated with Anonymous were indicted in the United States under the CFAA for launching a series of distributed denial-of-service attacks directed at the copyright industry, the indictment charged that the individuals had conspired to "intentionally cause damage, and attempt to cause damage, without authorization, to a protected computer . . . causing loss to victims resulting from the course of conduct affecting protected computers aggregating at least $5,000 in value" (*Indictment, United States of America v. Dennis Owen Collins, Jeremy Leroy Heller, Zhiwei Chen, Joshua S. Phy, Ryan Russell Gubele, Robert Audubon Whitfield, Anthony Tadros, Geoffrey Kenneth Commander, Phillip Garrett Simpson, Austen L. Stamm, Timothy Robert McClain, Wade Carl Williams, and Thomas J. Bell*, 2013). But the targeted servers that actually incurred those losses had not been accessed in an unauthorized manner, and the computers that had been used without authorization, to bombard the targeted servers, incurred no such financial losses. So attempts to use policy to target attackers, especially in the case of digital harms that are less clearly governed by existing laws, can sometimes conflate the elements of attackers' behavior that policy-makers actually wish to punish—the access capabilities or the ultimate harm.

While arresting and punishing attackers directly is an important role—and one that governments are uniquely suited to—there are several reasons policy-makers may wish to extend their reach beyond criminal-focused regulations to influence defenders. The challenges of attribution and international investigation make it difficult to identify and prosecute responsible parties in many cybercrime cases. Furthermore, policies that focus on punishing crimes that have already been successfully committed have minimal direct defensive impact, other than future deterrence. A government that wants to prevent or mitigate threats must look to policies that govern intermediary actors and the security measures they have in place. Perhaps the strongest incentive for governments to develop security policies that target intermediaries is simply that those actors are identifiable and, to some extent, cooperative and governable within a national context. A government cannot reach every attacker who targets its citizens, but it can reach every company that carries traffic, stores data, and provides services within its borders.

That does not mean national policies have no role to play in punishing international bad actors. In fact, recent policy efforts in the United States have focused on trying to cut off money flows to criminals outside the country's borders. The Deter

Cyber Theft Act introduced in the U.S. Senate in June 2014, for instance, proposed to block imports of products manufactured overseas by companies that were determined by the U.S. Government to have benefited from cyber espionage. Measures in the controversial U.S. Stop Online Piracy Act (SOPA), introduced in the House of Representatives in 2011 but never passed, also aimed at cutting off income sources for perpetrators of online crimes. Since SOPA targeted infringing websites, rather than imports, however, it took a slightly different approach than the Deter Cyber Theft Act. Its provisions could have prevented online advertising and payment companies from conducting businesses with infringing websites, required U.S. service providers to block access to those sites, and forbidden search engines from linking to them.

SOPA, and the closely related PROTECT IP (PIPA) bill proposed in the Senate, had the same ultimate goal as the Deter Cyber Theft Act: rendering cybercrime less profitable, foiling criminals by cutting off their cashflows, and mitigating the economic harm inflicted on U.S. industry by international actors. But the means by which SOPA and PIPA proposed to accomplish these goals required extensive involvement of Internet intermediaries—the advertising companies, payment processors, service providers, and search engines who would have carried out the policy measures in accordance with court orders. The measures proposed in the Deter Cyber Theft Act target physical imports rather than online content and are therefore more direct, requiring the involvement of fewer intermediaries. Though none of these bills have been passed by Congress, they offer interesting examples of how governments can attempt to mitigate financial harms through manipulation of both international trade and domestic markets.

Strengthening international cooperation around law enforcement efforts and the development of norms governing cyber threats is clearly crucial for the future landscape of computer security policy. National efforts on their own are unlikely to provide sufficient protection or consistency for firms and individuals operating in a global context. Furthering these international efforts, especially when it comes to defining malicious behavior and holding bad actors responsible for that behavior, should be a central focus of every government looking to engage with computer security issues. Long term, the most promising policy outcomes for computer security will likely derive from strong, detailed, and comprehensive international partnerships. But as nations struggle to find common ground on these issues and even, in some cases, indicate their mistrust of each other's motives and activities, the short-term future of cybersecurity policy will likely involve national governments taking more unilateral action and implementing policies within their own borders aimed at defenders.

7.2 Defender-Oriented Policy Levers

Security responsibilities that policy-makers can impose on defenders include ex-ante safety regulation, ex-post liability, and information reporting or disclosure (Romanosky & Acquisti, 2009). Actions are specific steps and measures defenders must take to mitigate threats. Encrypting sensitive data, implementing a firewall, mandating password length and complexity, and organizing security efforts according to a set process are all examples of security actions, or ex-ante safety measures. Liability regimes, by contrast, focus on security outcomes, or the responsibility of a defender to prevent certain types of damage—e.g., host infection, data theft, financial fraud—but leave the specific means by which those outcomes will be avoided to the discretion of the defenders. Finally, reporting responsibilities—which have been the focus of much existing and proposed security policy to date—have to do with what security-related information defenders must share and with whom. These responsibilities are interrelated; security actions are only useful insofar as they help defenders achieve positive outcomes, outcomes can only be measured by means of robust reporting, and reported information is only relevant if it can be used to inform the actions defenders should take and assess the associated outcomes.

The ex-ante and ex-post policy categories correspond loosely to notions of access and harm defense, in that the former is primarily concerned with dictating how defenders should limit access to attackers and the latter places greater emphasis on forcing those defenders to protect against harmful outcomes by any means necessary, as well as creating avenues for harmed parties to recoup their losses. The comparison to access and harm defense models also resonates with the longevity of these policy options—just as access capabilities span a wide variety of different options for attackers and evolve over time, so, too, ex-ante safety regulations are likely to require regular updating. Liability policies, on the other hand, cover a more stable set of harms and may therefore require less frequent revision. Reporting policies, meanwhile, can support both access and harm defense initiatives, depending on what information defenders are required to report and for what purpose.

Each of these categories of responsibilities—focused on actions, outcomes, and reporting—may be applied to defenders with different degrees of pressure. Policy-makers may choose to present any of these responsibilities as mandatory, incentivized, or voluntary for defenders. These three levels of pressure correspond to different types of policy: rules, which mandate certain behavior; inducements, which seek to encourage those behaviors through pressure or rewards; and facts, or policies that aim to influence behavior solely through the provision of information. Policies that target defenders have tended toward the voluntary end of this spectrum, with several governments providing suggested guidelines for security controls (including the NIST 800-53 catalog discussed in Chapter 2), educational materials for organizations and end users, voluntary codes of conduct for industry actors, but relatively few concrete incentives or rules around these issues. Unsurprisingly, policies aimed at bad actors almost exclusively fall into the category of rules—either those designed to punish offenders or to cut off their profits—since there is little expectation that these actors would be likely to cooperate with less stringent policies.

This presents yet another dimension in which the policy defense space is much more nuanced and expansive when it comes to dealing with defenders than in dealing with criminals. Policy-makers not only have a much wider range of actors and greater variety of activities to consider when crafting these policies, but also a broader spectrum of pressure levels to choose amongst. For defenders, this in turn presents an opportunity to shift responsibilities towards the voluntary end of the spectrum by being cooperative and eager adopters of suggested responsibilities in order to pre-empt more stringent measures. In a few cases, regulators themselves have even seized on this tendency, urging industry to strengthen security measures and wielding policy interventions as a thinly-veiled threat, or punishment, should private firms fail in their efforts (Wheeler, 2014).

Uncertainty around what the most effective measures are and reluctance to issue inflexible legislation around these issues has led policy-makers to explore other options. In particular, governments dissatisfied with the success of purely voluntary regimes have focused significant attention on the role of incentives in driving cybersecurity policy, discussing options that range from tax incentives and liability limitations to government funding and technical assistance. Still, there are relatively few government cybersecurity policies with formal incentives built into them. Instead, the primary incentive—to date—for adoption of security practices has been the avoidance of further regulation.

Different levels of pressure and forms of policy may also be better suited for certain types of security responsibilities. Security actions, which evolve rapidly with threats and can be relatively specific to a defender, may not be well suited to rules or inducements. Outcomes, which are likely to be both more stable and generalizable than specific countermeasures (though more difficult to guarantee or certify), may warrant forms of policy that exert more pressure on defenders, such as incentives. Finally, security reporting regimes are most useful and informative with broad and consistent participation, and may therefore call for even stricter, mandatory policies. The type of responsibility is not the only factor that may influence the appropriate policy lever and degree of pressure, however. The nature and capabilities of the defenders to whom a policy applies are also important considerations.

7.2.1 Tailoring Policies to Different Defenders

Different defenders have particular perspectives and sets of capabilities that can be leveraged to protect not just themselves but also their customers, coworkers, and even complete strangers. The value of policy-makers in enacting harm defense lies largely in recognizing the particular roles of individual types of defender and ensuring that those actors take defensive measures that protect not just themselves but also others. But in order to be effective policies must be tailored to the specific of capabilities and roles of the intermediaries they affect. Policies should be targeted at the class, or classes, of defenders best situated to have the ultimate desired impact, and in some cases it may not be immediately clear either which sector of defenders, or which individual actors within a given sector, are in that position. While no defender can be said to be "responsible" for security breaches in the same manner that perpetrators are, some may routinely forego commonly accepted security measures and practices, thereby enabling threat actors (or even profiting from them, for instance by marketing themselves as hosts with few scruples and a shady clientele). Other defenders may not be weak links but instead offer convenient chokepoints for cutting off certain types of threats in an efficient and convenient manner. Identifying both weak link and chokepoint defenders in the security ecosystem and designing policies aimed at them are two means of trying to implement harm defense through policy measures.

For these reasons, defining the appropriate classes of defenders to target with individual policies requires careful consideration of both a policy's intended impact and the specific actors best positioned to affect that change. Furthermore, policymaking should reflect the complex interactions between the wide variety of actors involved in defending against these threats and the limitations of any individual group's reach and capabilities in that context. There is no exhaustive list or uniform categorization of those groups, especially given that the appropriate categories and granularity may vary from policy to policy. Several examples of different, potentially policy-relevant classes of defenders are suggested in Table 7.1.

7.3 Policy for Access Defense

Computer security actions are specific measures and processes that defenders can use to block or mitigate threats. Examples of policy efforts centered on actions include government-developed lists of security practices, such as the NIST 800-53 catalog of defenses, government-driven codes of conduct like Australia's iCode initiative, and the security standards required by governments for their own systems and services, such as the Federal Risk and Authorization Management Program (FedRAMP). These examples and others suggest how governments can influence defenders' security actions in a variety of ways, from providing catalogues, to pressuring industry actors to form action-based initiatives, to requiring government vendors to fulfill certain guidelines.

Policies directed at ex-ante security precautions are likely to be most useful for classes of defenders with diffuse and numerous actors and relatively low levels of resources to devote to security efforts. Individuals, small organizations, and starting developers or entrepreneurs are unlikely to undertake extensive security audits and initiatives based on outcomes or expensive international standards. For these actors, governments can play a vital role in providing a contained set of security practices that can be easily understood and adopted and whose effectiveness is backed by evidence. Existing lists of security practices, including NIST 800-53 and FedRAMPS, tend to be long and offer little data to support their utility or indicate which—if any—of the actions they propose impact security, and how. Answering those questions relates closely to defenders' reporting responsibilities and could be a key function of policy-makers in this area. Particularly if such actions are to be voluntary, as in the case of private organizations choosing to adopt NIST 800-53 practices or its more recent Cybersecurity Framework, they are likely to be more widely adopted if there is evidence supporting their effectiveness. More importantly, however, they are likely to be more effective if there is data indicating their impact.

This is particularly significant given that many policies governing security actions are voluntary for private actors. Applying mandatory, or even incentivized, security actions to broad classes of diffuse defenders presents considerable enforcement problems—a government cannot easily enforce a policy requiring every Internet user or organization within its borders to adopt a standard set of security practices. Moreover, such a policy would ignore the unique risks, processes, and preferences of each, and also require constant updating to keep up with new threats. Therefore, specific

Table 7.1: Different classes of defenders and the scope of their control within the security ecosystem.

Class of Defender	**Scope of Control**	**Number and Size of Entities**	**Ease of Regulatory Intervention**
Hardware manufacturers	Device supply chain counterfeit	Relatively contained group, primarily for-profit firms	Subject to domestic commerce and import/export regulations
Software developers	Exploitable coding errors	Numerous entities, ranging in size from large companies to individuals	Difficult to regulate, both domestically and internationally, due to number of developers and ease with which code crosses national borders
Service providers	Malicious traffic	Relatively few major companies within an individual country	Fairly straightforward to regulate, especially since many major service providers are already subject to existing telecom regulatory regimes
Content providers & hosts	Malicious content	Large number of disparate entities, ranging from major firms to individuals	Difficult to regulate both due to sheer number and diffuse nature of entities
DNS operators	Fraudulent records	Many thousands of DNS servers are operated across the world, mainly by organizations	Difficult to regulate because of how many DNS operators there are and how loosely they are tracked
Merchants & payment processors	Fraudulent transactions	Large number of merchants, ranging from large firms to individuals; fewer payment processors	Policy-makers may find it difficult to regulate merchants directly, but can more easily regulate credit card associations—which can, in turn, influence merchants through private contracts
System administrators	Compromised machines, breaches of sensitive data	Very numerous with immense range in size and scale of systems	Difficult to regulate as a group, but can be subdivided and regulated according to those who possess certain types of data, operate at a certain threshold size, perform certain functions, etc.

security actions are often implemented through voluntary policies or private contracts, as in the case of the Payment Card Industry data security standards which individual merchants agree to in order to do business with credit card companies. Voluntary adoption and private agreements of this nature may be more quickly updated than policies and can spread through a large population of defenders even in the absence of major government enforcement efforts. On the other hand, allowing the payment card industry to dictate the requirements of responsibilities of access defenders in the context of incidents where they themselves serve as central harm defenders introduces considerable bias. Balancing the responsibilities of access and harm defenders is perhaps not best accomplished by allowing one party to decide when the other should be held liable.

For some classes of defenders, notably Internet service providers, governments have also looked at codifying specific security practices through voluntary codes of conduct. While still ostensibly voluntary, these codes—including the Australian iCode, developed in conjunction with the Australian Department of Communications, and the United States Anti-Bot Code of Conduct driven by the Federal Communications Commission (FCC) Communications Security, Reliability and Interoperability Council (CSRIC)—are generally spurred by government actors aggressively encouraging industry participation and adoption. The resulting codes focus on broad classes of actions service providers may engage in to help detect and remediate malicious traffic and infected machines on their networks, also touching on reporting and information sharing responsibilities. Ultimately, the codes of conduct still leave individual service providers considerable leeway to choose and implement their own set of security actions.

This freedom, and the lack of transparency surrounding who implements the recommended practices and how effective they are, can be frustrating for the government actors encouraging such activity. In July 2014, the FCC issued a request for comments on the implementation of the CSRIC voluntary recommendations for service providers. The request asked service providers and "other members of the Internet community" to comment on questions including: What progress have stakeholders made in implementing the recommendations? What significant success stories or breakthroughs have been achieved in implementing the recommendations? What are stakeholders' views and/or plans for full implementation of the recommendations? How effective are the recommendations at mitigating cyber risk when they have been implemented? In other words, the FCC was trying to determine whether anyone had adopted (or planned to adopt) the recommended, voluntary security practices and, if so, what impact those measures had—questions the FCC had no answers to because the Anti-Bot Code made no concrete demands on any defenders to implement specific measures, or even to disclose which measures they adopted.

The area where industry defenders enjoy the least flexibility in terms of security actions is in the security requirements for services and systems sold to the government itself. In these cases, the prospect of receiving a government's business may serve as a strong incentive for companies to alter their security practices, but such policies are limited in their applicability and the range of defenders they can impact. Another model for how governments might aim to incentivize specific security actions is the

U.K. Cyber Essentials Scheme, in which businesses can be awarded cybersecurity "badges" by the government for implementing a set list of security controls. This labeling scheme certifies that badge recipients have implemented five types of security controls—boundary firewalls and internet gateways, secure configuration, user access control, and malware protection.

There is no lack of information about what different cybersecurity actions are out there—either from government or private actors. There is, however, a lack of information about which of these actions have an impact and what those impacts are. That is an area of access defense where governments and policy-makers may be able to use reporting policies to fill some much needed holes and, in doing so, drive the adoption of effective security actions across a range of different defenders and give meaning to the notion of "best practices."

7.4 Policy Harm Defense

Security actions are important only insofar as they help defenders achieve positive security outcomes, just as access defense is only useful insofar as it reduces attackers' ability to inflict harm. Implementing a firewall or an encryption scheme is not an achievement in itself—these measures only matter if they help thwart threats and drive down the success of attempted security breaches. Similarly, those breaches only matter if they can be leveraged to inflict some form of harm. The ultimate aim of policy-makers should be mitigating these harms and it may therefore be useful to frame some policies in terms of the desired outcomes, rather than the specific actions intended to achieve those outcomes.

For instance, a policy-maker concerned about the problem of botnets is not worried about the existence of bots but rather the potential of those bots to be used to inflict economic harm by means of financial fraud, denial-of-service attacks, or other avenues. That policy-maker's ultimate goal is not to reduce rates of malware infection, or even to reduce the size and number or active bots, but rather to reduce the amount of economic damage that is inflicted with those bots. Crafting policies around anti-malware protections and anti-bot practices for service providers are ways of trying to move closer to that end goal, but they may not be the only—or even the most effective—means of doing so. An alternative strategy might center on cutting off payments to the operators of bots, thereby preventing the economic profit even while the technical threat (the bot) remains active. This is also partly the rationale behind policies like the Deter Cyber Theft Act which focus on thwarting perpetrators' ultimate aims (profit) rather than trying to go after the technical means by which they steal intellectual property. Of course, not all bad actors have financial motives, but focusing on the ultimate threat of harm posed by a digital compromise, rather than the technical details that enabled it, may in some cases guide policy-makers towards more targeted and creative solutions.

Policies aimed at security outcomes need not be exclusively focused on an attacker's end goal. Rather, they may center on the end goal of the defenders to which they apply. For instance, previous discussion highlights the elements of threats that different defenders are well positioned to address. Policy-makers may seek to drive those efforts forward either by specifying how those actors should address threats (i.e., which security actions should be taken) or by articulating what those actors should aim to achieve when tackling those threats (i.e., which outcomes to strive for). A policy for hardware manufacturers might therefore give specific guidelines for combating supply chain counterfeit or instead detail metrics for assessing the extent of such counterfeit and benchmarks for driving down those cases, leaving it to the affected firms' discretion how they want to meet those benchmarks. Similarly, policies aimed at end users might specify actions (e.g., installing security updates) or designate outcomes that those users are responsible for avoiding, such as the participation of their machines in bots.

Holding actors responsible for the security outcomes that result from their actions, rather than the specific actions they must take, allows them more freedom in designing their own security strategies and tailoring those strategies to their business and needs. In some cases, where defenders have limited security resources and expertise, they may not want that freedom; they may prefer to have clear and concrete security actions laid out for them and be held accountable only for whether they have implemented those regardless of whatever else may go wrong. In other instances, however, especially for defenders with significant resources and expertise in this area, it may be more effective to allow the industry actors who encounter these threats first-hand to craft their own, customized set of security measures. This has the advantage of enabling rapid updating and development of security actions, while also keeping policies more focused on the attackers' end goals rather than individual defensive maneuvers.

Security outcomes are significantly more static than actions. While the specific, technical means by which threats propagate are constantly evolving, the ultimate aims—and even the intermediate aims—of malicious actors have remained fairly consistent over time. Financial gain, political espionage, and system disruption or degradation continue to motivate the bad actors in cyberspace, and they continue to use a fairly stable set of tools, including malware, botnets, and denial-of-service attacks, to achieve those aims. The primary drawback to designing policies around outcomes rather than actions is that the former are often challenging to measure or verify. Accordingly, there has been relatively little policy-making focused on these outcomes, as it is nearly impossible to implement an effective outcomes-based regime in the absence of comprehensive reporting responsibilities.

In the final report on the U.S. Anti-Bot Code for service providers, for instance, the authors highlight the difficulties associated with trying to apply metrics to botnet remediation efforts, noting that, "Without consistent objective agreed upon industry based measurements, ISPs may find it difficult or impossible to tell the extent of the bot problem on their network, and if so, whether efforts to correct it will have, or have had, any material effect." In particular, the authors note, participants in the voluntary code of conduct may not witness any reduction in bots on their networks since bots do not operate in a closed system. They therefore advocate for a combination of outcome- and activity-based metrics to assess participant progress—that is, looking at how many people a participant notifies of infections (activity), not just the changes in rate of bot infections (outcome). A commitment made by the same group to develop such metrics to assess the code was later dropped when the participants could not agree even on a definition for what constituted a bot—eventually spurring the FCC to explicitly solicit comments on what impact service providers had witnessed when implementing the suggested measures (FCC, 2014).

One example of government intervention driven partially by outcomes rather than specific actions comes from the U.S. Federal Trade Commission (FTC) suit against Wyndham Worldwide Corporation, the owner of a chain of hotels that suffered three data breaches in a period of two years. The FTC's complaint alleges that Wyndham failed to "maintain reasonable and appropriate data security for consumers' sensitive personal information," emphasizing that the company's security failures "led to fraudulent charges on consumers' accounts, more than $10.6 million in fraud loss, and the export of hundreds of thousands of consumers' payment card account information to a domain registered in Russia." Much of the complaint focuses on listing the security practices that Wyndham did not employ to protect its customers' data in order to establish the company's negligence, but none of those measures are required by law. What triggered the complaint was not any individual security action that Wyndham did not undertake, but rather the corporation's ultimate failure to protect customers' data coupled with the assertion on its website that it takes "commercially reasonable efforts" to do so. Those reasonable efforts were completely up to the company's own discretion, until they suffered a bad security outcome. It's a striking tension in policy-based security interventions: policy-makers want to hold someone—often access defenders—accountable for the harms the result from security breaches, but don't want to be too involved in dictating access defense responsibilities.

Similar to the legal proceedings following the TJX breach, it is not clear from the FTC's suit against Wyndham which specific measures would, in fact, have constituted reasonable efforts. Rather than pinpointing particular failures, the FTC describes a series of "insufficient" data security measures that, in combination, it views as negligent. According to the FTC, these include (Salas, 2014):

> failing to employ firewalls; permitting "storage of payment card information in clear readable text"; failing to make sure Wyndham-branded hotels "implemented adequate information security policies and procedures prior to connecting their local computer networks to Hotels and Resorts' computer network"; permitting Wyndham-branded hotels "to connect insecure servers to Hotels and Resorts' networks, including servers using outdated operating systems that could not receive security updates or patches to address known security vulnerabilities"; permitting servers on Hotels and Resorts' networks with commonly-known default user IDs and passwords; failing to "employ commonly-used methods to require user IDs and passwords that are difficult for hackers to guess"; failing to "adequately inventory computers connected to Hotels and Resorts' network" to manage devices on its network; failing to "monitor Hotels and Resorts' computer network for malware used in a previous intrusion"; and failing to restrict third-party access "such as by restricting connections to specified IP addresses or granting temporary, limited access, as necessary."

In a motion denying Wyndham's request to dismiss the suit, Judge Esther Salas notes that "the FTC does not plead the particularized data-security rules or regulations that Hotels and Resorts' procedures allegedly failed to comply with" but maintains that "this cannot preclude the FTC's enforcement action."

For policy-makers, a crucial part of defining security liability regimes is being able to distinguish between breaches where defenders were negligent and others where they were just unlucky. This, again, speaks to the interplay between the different types of policy measures—ex-post liability is determined in part by ex-ante precautions, which comes back to the question, best addressed through reporting policies, of which of those precautions are actually important and effective.

7.5 Security Reporting Policies

Policies that govern actions and outcomes are closely tied to the question of what information the defenders responsible for those actions or outcomes must report back in order to ensure compliance and measure progress. Reporting requirements represent the area of security policy where governments have been most active, but depending on their purpose, these reporting policies can vary greatly with regard to what information defenders are expected to report, and to whom. These policies may aim to accomplish several different goals, including protecting people whose information has been breached, helping others to defend against threats that have been previously experienced or identified by others, and contributing to a better understanding of the types of threats observed and effectiveness of various countermeasures. Each of these three goals has very different implications for security reporting regimes, as described in Table 7.2, and may pose different challenges for both defenders and regulators.

Perhaps the most common template for computer security reporting policies is the data security breach notification law, an early example of which was enacted in California in 2002. That law, SB 1386, requires everyone who conducts business in California to notify "any resident of California whose unencrypted personal information was, or is reasonably believed to have been, acquired by an unauthorized person." Since then, many other states in the U.S. have adopted similar laws requiring that data breaches be disclosed shortly following their discovery to the affected parties. The European Union, in its proposed Network and Information Security (NIS) Directive, is also weighing a set of related policy measures which, in their current form, would require a range of companies to report "incidents having a significant impact on the security of core services provided by market operators and public administrations" to their national governments.

The European and American reporting models differ both in terms of which defenders they apply to and who those entities are supposed to report their security breaches to—the E.U. directive specifies that reports be issued to a national government authority, while many of the U.S. laws focus instead on direct notification of those whose data was breached. Furthermore, SB 1386 applies to every "state agency, or . . . person or business that conducts business in California, that owns or licenses computerized data that includes personal information" while the NIS direc-

Table 7.2: Different purposes of security incident reporting policies.

Purpose of reporting	**Examples**	**What is reported?**	**When is it reported?**	**Whom is it reported to?**
Consumer protection	Data breach notification laws (California SB 1386)	Who was affected by a data breach and what personal information was revealed	Shortly after a breach is detected	Affected parties (i.e., those whose information was accessed)
Real-time threat mitigation	Information sharing laws (CISA, CISPA)	Signature and detection information, countermeasures	Immediately following detection	Other parties in a position to mitigate the identified threat
Analysis of root causes & countermeasure impact	Industry reports (Microsoft SIR, Verizon DBIR)	Type of threat, why it was successful, what defenses were in place, what damage it caused	Following a (potentially lengthy) internal investigation	A party in a position to aggregate it with other incident reports

tive specifically designates "internet companies" as being responsible for the reporting requirements it outlines "because it is absurd to work to protect critical internet infrastructure without obliging such companies to take responsibility for their wider role in this ecosystem" but explicitly exempts hardware manufacturers and software developers. The California law applies to a broader class of actors but a potentially narrower set of incidents, while the European directive targets a specific set of defenders but is less specific about which types of incidents or information must be reported.

One explanation for the differences in these policies may be that the regulators behind them are motivated by different aims. Where the state data breach notification laws are clearly aimed at protecting consumers whose information has been leaked from suffering any adverse consequences, the NIS Directive's stated objective is establishing an "effective mechanism at EU level for effective cooperation and collaboration and for trusted information sharing on NIS incidents and risks among the Member States." Cybersecurity information sharing generally refers to reporting policies aimed not at the notification of users affected by security breaches but rather at the other entities that may be able to learn from or defend against those breaches themselves. Two proposed policies in the United States—the Cyber Intelligence Sharing and Protection Act (CISPA), introduced in Congress in 2012, and the Cybersecurity Information Sharing Act (CISA) of 2014, introduced in the Senate, focus on this second potential function of security reporting by shielding companies from being held liable for sharing threat information that may aid defense efforts. However, where the European Union has proposed a mandatory sharing regime, the U.S. proposals merely seek to encourage voluntary sharing by eliminating potential legal obstacles; they do not mandate any reporting or sharing. Furthermore, where the NIS Directive focuses on reporting duties related to incidents that have already had a serious impact, the U.S. proposals are more broadly aimed at sharing information on all threats—both successful and unsuccessful—and especially dispersing that knowledge in advance of serious impacts. In other words, CISA and CISPA take a more preventative approach to information sharing, while the NIS Directive adopts a more reactive stance.

Existing security reporting policies tend to fall into one of these two categories: notification policies intended to protect consumers or information sharing policies intended to spread defenders' knowledge and experience to aid real-time threat detection and remediation. The European NIS Directive blurs the line between the two categories to some extent, incorporating elements of data breach notification and information sharing into a single policy. Another important role of reporting responsibilities that is less commonly fostered by policy-makers but gets directly at questions of access and harm defense is data collection on the types of threats defenders face and the effectiveness of different strategies at defending against those threats. This third function of reporting policies is not about helping individuals or companies protect themselves against current threats and risks in the short term, but rather focuses on what can be learned from threat trends and defensive strategies over time and across a large number of incidents and actors.

Policy-makers have different roles to play in promoting each of these three goals. All three may be challenging for private actors to address adequately in the absence of government intervention, but for different reasons. For instance, industry actors may be reluctant or unwilling to notify customers of security breaches for fear of damaging their reputations or incurring legal action. Policy-makers who want to ensure that individuals are aware of any breaches of their personal information and the consequent risks can overcome these obstacles by mandating notification procedures—voluntary, and even incentivized, policies are unlikely to be widely effective given the potential drawbacks of notification to the breached parties.

Real-time threat information sharing between defenders may be hindered by some of the same fears, particularly concerns about litigation and liability that may arise from sharing sensitive information or publicizing security breaches, as well as logistical and competitive considerations. Logistically, private actors may not always have easy avenues for spreading threat information to all of the other defenders who could benefit from it. Moreover, given that some of these defenders are competitors, firms may not want to share that information with everyone else who could benefit from it. The varied security resources and expertise of industry actors also mean that a small number of firms, with the largest investments in security, are likely to glean most of the threat information and their smaller peers are likely to have relatively little novel intelligence to offer in exchange, creating a heavily imbalanced information sharing ecosystem in which the defenders who have the most valuable information to share have little incentive to do so with those who would have the most to gain from receiving it.

Policy-makers wishing to encourage information sharing between defenders to combat real-time threats have several options to try to lessen these barriers, many of which have been proposed in pending policies in the United States. Absolving the sharers of liability for providing that information to others is one possible role for policy. Creating channels for government organizations to provide industry with real-time threat intelligence is another. CISA includes versions of both of these measures in efforts to facilitate both more private-to-private sharing and government-industry sharing. Coordinating information sharing efforts through a centralized government authority—as proposed in the NIS Directive—is another potential function for policy and may help address the logistic and competitive barriers to sharing. But a government-centric model also presents drawbacks. In particular, information sharing done for the purpose of real-time threat mitigation requires very rapid turnaround that may be hindered by the insertion of a government body between private defenders.

Another significant difference between the proposed policies in this area is the extent to which they require, or merely try to encourage, industry information sharing. The U.S. proposals, which attempt to lower barriers to information sharing but still rely on voluntary sharing by private actors, are premised on the idea that the benefits to private actors of such sharing will outweigh the potential costs, especially if their antitrust and liability concerns can be reduced through policy. The European Directive, by contrast, mandates reporting of sufficiently severe incidents under the assumption that in the absence of such a requirement important information will go unreported.

Finally, for the third function of security reporting—collecting data on threat trends and countermeasure efficacy—government actors can help industry actors overcome a similar set of obstacles related to the challenges of collective action. While such data would potentially be valuable to nearly all defenders, individually, firms are reluctant to champion such an effort in the absence of participation by their peers. No company wants to be the first to release that data about the threats they see and the impact of their countermeasures, for fear of alarming customers by drawing attention to their security incidents and thereby harming their reputation and business. Furthermore, no company stands to gain anything by unilaterally releasing this information since they benefit only by what they learn from others and the creation of a broader data set beyond what they already know internally. And mandating that companies report information on all of their breaches could even be counterproductive, in encouraging defenders not to actively look for such incidents for fear of having to report them.

These three distinct potential goals of security reporting policies rely on very different types of information, and in some cases this may even put them at odds with each other. Breach notification generally involves reporting what information was breached and for whom. This often translates into the sorts of media attention—focused on the magnitude of breaches and the number of breached records—that firms most wish to avoid and may discourage them from engaging in further sharing. Information sharing, by contrast, revolves around sharing specific threat signature information, or the ways that other defenders can identify current threats and remediate them. Finally, reporting intended to contribute to longer term data on threats and countermeasures would require detailed description of the root causes of security incidents and the defensive measures that were (and were not) in place when they occurred. This information, collected en masse over time, would enable analysis of both the broader engineering design decisions that might combat the most common threats and the effectiveness of different existing countermeasures against these threats. There may be legitimate reasons to share each of these types of information with different actors at certain points, but, where possible, restricting the fields of necessary data may help encourage participation from defenders. For instance, not requiring firms to report data about the magnitude of breaches may help assuage their concerns about the reputational damage that may be incurred by releasing that information.

This ability of policy-makers to aggregate security-related information from a range of different defenders ties into both access defense—in assessing the impact of specific access restrictions—and harm defense—in indicating what types of harm are being incurred. Beyond restricting illicit money flows and coordinating harm defenses that require the cooperation of third parties, reporting policies therefore offer a pathway by which governments can offer defenders a clearer picture of what they should be trying to defend against and how.

Chapter 8

Conclusion

The structure of computer system defenses dictates the structure of attacks. The invention of a defense is part of what categorizes the attacks it interrupts—and those it doesn't—and contributes to the meaning of ideas like overlap and independence. So identifying classes of defense for computer systems is ultimately about understanding how the defensive landscape—the technical tools, the managerial and government policies—organizes the ways we think about attacks along multiple different axes. At a purely practical level, classes of defense are useful if they help defenders identify holes, or vulnerabilities, in their defensive strategies; on a more theoretical level, they can be used to think about what it means for there to be a "hole" in a set of computer defenses, and how the strengths and vulnerabilities of one defense line up with those of others. An important piece of trying to think about defenses in aggregate is understanding that that is not always possible—that two defenses may restrict behaviors in ways that are completely unrelated and cannot be mapped on to each other because they are defining two completely different conceptions of attack classes. Part of defining a set of defense classes is therefore understanding where these disconnects occur and the fundamentally different perspectives on defense that give rise to them.

8.1 Two Frameworks for Defense

This thesis presents two frameworks for classifying computer system defenses, one oriented around the access capabilities that attackers take advantage of in the context of computer systems and another focused on the types of harm that those attackers inflict through the use of those capabilities. These are two totally different lenses through which to view computer security incidents—and computer security, more generally. Looking at security from the perspective of restricting access capabilities means classifying attacks according to the behaviors and activities that attackers exploit in the context of computer systems even if those behaviors are not, themselves, directly harmful. It means restricting those capabilities so that adversaries have to do more work to acquire them, and legitimate users have more signals to indicate when that work is being done. It defines computer security incidents not by how they end but how they begin and attackers not by what they want to achieve but how they achieve it. The access framing is inherently suspicious of new or unknown capabilities and their potential to be used for malicious purposes in ways that defenders may not have thought of. Access defense allows for few, if any, presumptions about what path an attacker will take through a system or what order he will encounter defenses in, and it fails when adversaries are able to do anything in the context of a computer system that they were not explicitly intended to be able to do. It is a fundamentally general notion of defense—one with a broad notion of what it is trying to defend against and a vague charge to prevent the unwanted and unknown.

Harm defense offers a more specific picture of what defenders are trying to protect against by organizing attacks according to their ultimate aim, rather than their particular implementation. Rather than trying to restrict the unknown or unexpected, harm defenses are intended to block only the direct infliction of damage—and just as the access framing disregards the attackers' intent, so, too, the harm framework disregards the question of how attackers may have exploited systems and instead focuses only on what they ultimately want and how to prevent them from achieving that result. Harm defenses are not concerned with the possibility of unanticipated or new behaviors, so long as those behaviors are not directly harmful, and they fail only when the class of harm they are intended to prevent is actually inflicted on a victim.

For the most part, access capabilities do not map onto particular classes of harm, leaving a disconnect between the defenses that are designed to limit damage and those that restrict potentially malicious capabilities. Each set can be considered and understood in relation to the others within its own framework, but not in relation to those that have a fundamentally different goal. There is, however, some overlap where computer capabilities are used to directly inflict harm—these digital harms that never go beyond the targeted computer system and rely on specific capabilities blur the line between access and harm, suggesting a class of "intermediate harms" that exhibit a hybrid of the features of the access and harm framings.

These access and harm framings help reconcile some of the conflicting general wisdom around computer defense, particularly the tension around whether attackers or defenders have the advantage in the context of computer systems. R. Anderson (2001) argues that "Attack is simply easier than defense. Defending a modern information system could also be likened to defending a large, thinly-populated territory like the nineteenth century Wild West: the men in black hats can strike anywhere, while the men in white hats have to defend everywhere." This is a viewpoint that echoes the challenges of access defense—defenders have to find and protect every possible access pathway to be effective, attackers only need to find one because these pathways are so easily replaceable, or substituted for each other. In proposing their attack kill chain model, Hutchins et al. (2011) contend just the opposite, namely that "the adversary must progress successfully through each stage of the chain before it can achieve its desired objective; just one mitigation disrupts the chain and the adversary . . . the defender can achieve an advantage over the aggressor." This speaks to the harm defense framing, in which individual harm infliction stages, or even intermediate harms, are so essential—so irreplaceable—to an attacker's end goal that a harm defender who can interrupt one of those stages can pose a much greater obstacle than an access defender, requiring much more additional work on the part of the attacker to overcome. Understanding where defenders may have some advantages over attackers is not just about recognizing the difference between more and less replaceable stages of attacks, or between access and harm capabilities—it also relates to the question of *which* defenders may be most advantageously poised to interrupt certain types of attacks.

8.2 Roles of Different Defenders

The access and harm framings of defense suggest different roles for various types of defenders who have insight and control over particular elements of computer security incidents. The three classes of defenders discussed in this analysis—application designers, organizations, and policy-makers—are able to influence and intervene at very different stages of security breaches. Application designers can contribute to access defense efforts by distinguishing between malicious and legitimate capabilities and placing more cumbersome restrictions on the former. Organizations are well poised to address the digital harms that lie between access and harm defense, particularly through careful monitoring and restriction of outbound network traffic. Policy makers have a more limited ability than organizations to restrict information flows to attackers, but can do much more to cut off illicit money flows and interrupt financially motivated security incidents, as well as collect the needed information to understand how these incidents occur and who is best poised to interrupt them. None of these classes of defenders are entirely confined to just one mode of defense, however, and part of the value in having multiple defense frameworks lies in enabling defenders to think about what they can do both with regard to capabilities and harms.

The access and harm framings also help illuminate some of the limitations of any individual defender, or class of defenders, and the extent to which each relies on the defense implemented by others. Harm defense, in particular, tends to rely on the coordination and cooperation of multiple different parties beyond the one, or ones, being directly harmed. To the extent that these frameworks can help make sense of how different defense mechanisms relate to and interact with each other, they also offer some insight into how different defensive actors can bolster each others' efforts, as well. These coordination problems also imply further roles for policy-makers, not just in determining how responsibilities should be divided among access and harm defenders but also among the different actors that fulfill each of those functions for any individual incident.

Crucial to understanding the roles of these different defenders, and making decisions around who is responsible for what—when access defenders have failed in their duties, when harm defenders have abandoned their posts reinforcing access defense—is an appreciation for the limited visibility and scope of control belonging to any individual defender. When we talk about (and report on, and litigate) successful security incidents, our inclination is too often to latch on to the first or the most easily understood point of access—the phishing email, the dictionary attack, the unprotected wireless network—and harp on the simple line defense that seems like it would have made all the difference—two-factor authentication, or rate limiting logins, or WPA encryption. But that perspective oversimplifies the much more complicated narrative of the gradual, escalating capabilities acquired by perpetrators, as well the much more limited and challenging environment that individual defenders operate in, constrained both by the extent to which they can see and control access attempts and by their ability to witness and mitigate the ultimate harms.

The optimistic takeaway from security breaches like those that targeted TJX, DigiNotar, the PLA victims, Spamhaus, and MIT is that there are lots of opportunities for defending against computer security incidents and lots of defenders who can help contribute to those efforts. The more pessimistic interpretation has to do with how little impact many of those defensive measures would likely have—forcing only a slight readjustment of attackers' plans, rather than a radical re-engineering of the attack or dramatic increase in work—and how reticent many of those defenders are to play an active role or assume any responsibility. It's a mindset that is aggravated by the externalities of many computer security breaches and the extent to which those breaches often involve multiple intermediary parties, but perhaps also by the challenges associated with catching and prosecuting the perpetrators of these attacks.

The attribution challenges posed by computer networks do not make computer incidents impossible to defend against—none of the defensive interventions proposed for the case study incidents rely on stronger global attribution—but they do put greater pressure on non-law enforcement defenders to provide protections, and they often mean those defenders are left with no one to blame but each other. There are rare exceptions, where we are more willing to view defenders as victims and focus blame and repercussions on the perpetrators, but this is usually only the case when news of an incident is coupled with an immediate and forceful accusation about whom is responsible—as happened with the report on PLA Unit 61398 and the Sony breach which was widely reported as the work of North Korea. In the absence of someone to hold responsible, we end up focusing blame on the defenders, and those defenders, in turn, devote their time and energy to trying to shift defensive responsibilities onto each other, and we rapidly lose sight of how limited their individual windows into security incidents really are, and how blindly they are often forced to make decisions and exercise control without knowing exactly what they are defending against.

The proposed access and harm framings shed light on the ways in which we—in our media coverage of security incidents, in the ensuing litigation, and even in the proposed policy interventions—tend to over-emphasize the role of access defense, and particularly the responsibilities and capabilities of individual centralized access defenders. Applying these framings to case studies of actual incidents and institutions also suggests that the externality problems that are often used to explain why we do such a poor job of defending computer systems are, in many cases, compounded by poor visibility into incidents and weak coordination among the different actors who play roles—or could play roles—in defense. Both of these contributions, which focus on the need to pay greater attention to harms inflicted through computer security incidents when considering defensive interventions and the related need to expand visibility into those incidents so it is more possible to trace the full chain of events leading up to them and the associated actual harms, point to a broader lesson centered on making computer security less about computers and more about how they impact people.

8.3 Revisiting Defense in Depth

Recall the careless amalgamation and conflation of different notions of defense in depth—drawn from other fields and applied haphazardly to computers and information—that led to the term losing much of its meaning in the field of computer security. Perhaps part of the trouble in interpreting that concept for computer security lies in the sense that there are too many different kinds of "depth" to be considered in the context of computer systems, too many different spheres of independence and conceptions of overlap to allow for a single, consistent meaning. Unlike the soldiers of the Roman Empire, defenders of computer systems do not have a clear geographic perimeter or central, physical capital to focus their protection efforts on, nor a well-defined set of sequential signals of escalating harm, like the protectors of nuclear plants. They have elements of all these things, in the context of particular incidents, or even particular stages of incidents, but there is no single consistent metaphor that can be applied across the range of security threats that involve computers. Instead, we end up with lots of metaphors, mixed and applied haphazardly, and too little sense of what we can actually learn from historical endeavors—and which versions of history we have conveniently invented to explain and reinforce our own ideas.

Combining defenses for computer systems means looking across a number of different dimensions to understand what is being defended, what is being defended against, and how those determinations channel and influence the space of relevant threats. Combinations of access and harm defenses, combinations of application and organizational and policy defenses, combinations of access defenses that target authenticated capabilities and unauthenticated capabilities, combinations of harm defenses that target financial and physical and digital harms—all of these are useful and relevant ways of understanding what it means to implement multiple lines of defense in concert in the context of computer systems. They are not the only such models for considering how different types of defense can serve to reinforce each other and plug different kinds of vulnerabilities. Nor are they defense in depth, exactly; rather, they are examples of how different framings of defense shape the landscape of security and our understandings of whose responsibility it is to address certain risks and which elements of those risks constitute the real threat.

References

Albright, D., Brannan, P., & Walrond, C. (2010, December 22). Did Stuxnet Take Out 1,000 Centrifuges at the Natanz Enrichment Plant? *Institute for Science and International Security (ISIS) Report*. Available from http://www.isis-online.org/uploads/isis-reports/documents/stuxnet_FEP_22Dec2010.pdf

Anderson, P. (2001). *Deception: A healthy part of any defense-in-depth strategy* (Tech. Rep.). SANS Institute. Available from https://www.sans.org/reading_room/whitepapers/policyissues/deception-healthy-defense-in-depth-strategy_506

Anderson, R. (2001). Why information security is hard-an economic perspective. In *Proceedings of the 17th annual computer security applications conference* (pp. 358–). Washington, DC, USA: IEEE Computer Society. Available from http://dl.acm.org/citation.cfm?id=872016.872155

Beautement, A., & Pym, D. (2010). Structured systems economics for security management. In *Proceeedings of the Ninth Workshop on the Economics of Information Security.* Cambridge, MA. Available from http://www.abdn.ac.uk/~csc335/ontology.pdf

Beeler, J. H. (1956, October). Castles and Strategy in Norman and Early Angevin England. *Speculum*, *31*, 581–601. Available from http://journals.cambridge.org/article_S0038713400201887

Bejtlich, R. (2013). *The practice of network security monitoring: Understanding incident detection and response.* San Francisco, CA: No Starch Press.

Bergadano, F., Gunetti, D., & Picardi, C. (2002, November). User authentication through keystroke dynamics. *ACM Trans. Inf. Syst. Secur.*, *5*(4), 367–397. Available from http://doi.acm.org/10.1145/581271.581272

Bursztein, E., Benko, B., Margolis, D., Pietraszek, T., Archer, A., Aquino, A., et al. (2014). Handcrafted fraud and extortion: Manual account hijacking in the wild. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (pp. 347–358). New York, NY, USA: ACM. Available from http://doi.acm.org/10.1145/2663716.2663749

Charney, S. (2009). *Rethinking the cyber threat: A framework and path forward* (Tech. Rep.). Microsoft. Available from http://download.microsoft.com/download/F/1/3/F139E667-8922-48C0-8F6A-B3632FF86CFA/rethinking-cyber-threat.pdf

Chmielewski, D., & Hesseldahl, A. (2014, December 10). Sony pictures tries to disrupt downloads of its stolen files. *Re/Code*. Available from http://recode.net/2014/12/10/sony-pictures-tries-to-disrupt-downloads-of-its-stolen-files/

Cieply, M., & Barnes, B. (2014, December 14). Sony pictures demands that news agencies delete 'stolen' data. *The New York Times*. Available from http://www.nytimes.com/2014/12/15/business/sony-pictures-demands-that-news-organizations-delete-stolen-data.html

Danchev, D. (2009, September 23). Modern banker malware undermines two-factor authentication modern banker malware undermines two-factor authentication modern banker malware undermines two-factor authentication. *ZDnet*. Available from http://www.zdnet.com/article/modern-banker-malware-undermines-two-factor-authentication/

Defence in depth in nuclear safety, insag-10 (Tech. Rep.). (1996). International Nuclear Safety Advisory Group. Available from http://www-pub.iaea.org/MTCD/publications/PDF/Pub1013e_web.pdf

Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 581–590). New York, NY, USA: ACM Press.

Drew, C., & Sengupta, S. (2013, June 23). N.S.A. Leak Puts Focus on System Administrators. *The New York Times*. Available from http://www.nytimes.com/2013/06/24/technology/nsa-leak-puts-focus-on-system-administrators.html

Egelman, S., Cranor, L. F., & Hong, J. (2008). You've been warned: An empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1065–1074). New York, NY, USA: ACM.

FCC. (2014, July 25). *FCC's Public Safety and Homeland Security Bureau Requests Comment on Implementation of CSRIC III Cybersecurity Best Practices.* Available from http://transition.fcc.gov/Daily_Releases/Daily_Business/2014/db0725/DA-14-1066A1.pdf

Fluhrer, S. R., Mantin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. In *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography* (pp. 1–24). London, UK, UK: Springer-Verlag. Available from http://dl.acm.org/citation.cfm?id=646557.694759

Fung, B. (2014, September 2). Apple's basically blaming hack victims for not securing their own icloud accounts. *Washington Post*. Available from http://www.washingtonpost.com/blogs/the-switch/wp/2014/09/02/apples-basically-blaming-hack-victims-for-not-securing-their-own-icloud-accounts/

Garfinkel, S., Spafford, G., & Schwartz, A. (2003). *Practical Unix & Internet Security, 3rd Edition.* O'Reilly Media, Inc.

Geer, D. (2004, June). Just how secure are security products? *Computer*, *37*(6), 14–16. Available from http://dx.doi.org/10.1109/MC.2004.28

Goldstein, M., Perlroth, N., & Corkery, M. (2014, December 22). Neglected Server Provided Entry for JPMorgan Hackers. *The New York Times*.

Available from http://dealbook.nytimes.com/2014/12/22/entry-point-of-jpmorgan-data-breach-is-identified/

Gordon, L. A., Loeb, M., & Lucyshyn, W. (2003). Information security expenditures and real options: A wait-and-see approach. *Computer Security Journal*, *29*(2).

Greenberg, A. (2013, September 22). German Hacker Group Says It's Broken the iPhone's TouchID Fingerprint Reader. *Forbes*. Available from http://www.forbes.com/sites/andygreenberg/2013/09/22/german-hackers-say-theyve-broken-the-iphones-touchid-fingerprint-reader/

Hansman, S., & Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers & Security*, *24*(1), 31–43. Available from http://cyberunited.com/wp-content/uploads/2013/03/A-taxonomy-of-network-and-computer-attacks.pdf

Harrison, K., & White, G. (2011). A taxonomy of cyber events affecting communities. In *Proceedings of the 2011 44th Hawaii International Conference on System Sciences (HICSS)* (pp. 1–9).

Hoogstraaten, H., Prins, R., Niggebrugge, D., Heppener, D., Groenewegen, F., Wettinck, J., et al. (2012, August 13). *Black Tulip: Report of the investigation into the DigiNotar Certificate Authority breach* (Tech. Rep.). Fox-IT BV. Available from http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/rapporten/2012/08/13/black-tulip-update/black-tulip-update.pdf

Hutchins, E. M., Cloppert, M. J., & Armin, R. M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare and Security Research*.

Indictment, United States of America v. Dennis Owen Collins, Jeremy Leroy Heller, Zhiwei Chen, Joshua S. Phy, Ryan Russell Gubele, Robert Audubon Whitfield, Anthony Tadros, Geoffrey Kenneth Commander, Phillip Garrett Simpson, Austen L. Stamm, Timothy Robert McClain, Wade Carl Williams, and Thomas J. Bell. (2013, October). United States District Court for the Eastern District of Virginia, Criminal No. 1:13-cr-383.

In the Matter of The TJX Companies, Inc., a corporation, Docket No. C-072-3055. (2008, March 27). United States of America Federal Trade Commission.

Kaminsky, D. (2011). *Dnssec amplification is not a dnssec bug, but an already existing dns, udp, and ip bug.* Available from http://dankaminsky.com/2011/01/05/djb-ccc/#dnsamp

Kao, J. (2013, January 22). MIT hacked again, URLs redirected. *The Tech*. Available from http://techblogs.mit.edu/news/2013/01/mit-hacked-again-urls-redirected/

Kesan, J. P., & Hayes, C. M. (2011). Mitigative counterstriking: Self-defense and deterrence in cyberspace. *Harvard Journal of Law & Technology*, *25*, 429.

Kewley, D. L., & Lowry, J. (2001, June). Observations on the effects of defense in depth on adversary behavior in cyber warfare. In *Proceedings of the IEEE Workshop on Information Assurance and Security.* United States Military Academy, West Point, NY. Available from http://www.bbn.com/resources/pdf/USMA_IEEE02.pdf

Kjaerland, M. (2006). A taxonomy and comparison of computer security incidents from the commercial and government sectors. *Computers & Security*, *25*(7), 522–538.

Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christin, N., et al. (2011). Of passwords and people: Measuring the effect of password-composition policies. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 2595–2604). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/1978942.1979321`

Komogortsev, O. V., Karpov, A., & Holland, C. D. (2012, May 1). CUE: Counterfeit-resistant usable eye movement-based authentication via oculomotor plant characteristics and complex eye movement patterns. In *Proc. SPIE 8371, Sensing Technologies for Global Health, Military Medicine, Disaster Response, and Environmental Monitoring II; and Biometric Technology for Human Identification IX.*

Krazit, T. (2009, June 26). Google thought Michael Jackson traffic was attack. *Cnet*. Available from `http://www.cnet.com/news/google-thought-michael-jackson-traffic-was-attack/`

Krebs, B. (2014a, June 9). Backstage with the Gameover Botnet Hijackers. *Krebs on Security*. Available from `http://krebsonsecurity.com/2014/06/backstage-with-the-gameover-botnet-hijackers/#more-26346`

Krebs, B. (2014b, December 14). SpamHaus, CloudFlare Attacker Pleads Guilty. *Krebs on Security*. Available from `http://krebsonsecurity.com/2014/12/spamhaus-cloudflare-attacker-pleads-guilty-to-computer-abuse-child-porn-charges/`

Landwehr, C., Boneh, D., Mitchell, J., Bellovin, S., Landau, S., & Lesk, M. (2012, May). Privacy and cybersecurity: The next 100 years. *Proceedings of the IEEE*, *100*(Special Centennial Issue), 1659-1673.

Layton, E. T. (1976, October). Ideologies of science and engineering. *Technology and Culture*, *17*(4), 688–701.

Lemos, R. (2009, September 18). Real-time hackers foil two-factor security. *Technology Review*. Available from `http://www.technologyreview.com/news/415371/real-time-hackers-foil-two-factor-security/`

Levchenko, K., Pitsillidis, A., Chachra, N., Enright, B., Felegyhazi, M., Grier, C., et al. (2011, May). Click trajectories: End-to-end analysis of the spam value chain. In *2011 ieee symposium on security and privacy* (p. 431-446).

Lindqvist, U., & Jonsson, E. (1997). How to systematically classify computer security intrusions. In *Proceedings of 1997 IEEE Symposium on Security and Privacy* (pp. 154–163).

Lindsay, G. (2012). *Dnssec and dns amplification attacks.*

Lowry, J. H., & Maughan, D. (2003). DARPA OPX information assurance: technologies and lessons learned. In *Proc. SPIE* (Vol. 5071, p. 52-62).

Luttwak, E. (1976). *The Grand Strategy of the Roman Empire: From the First Century A.D. to the Third.* Baltimore, MD: Johns Hopkins University Press.

Lynn, W. (2010, September/October). Defending a New Domain: The Pentagon's Cyberstrategy. *Foreign Affairs*, 97–108. Available from http://www.foreignaffairs.com/articles/66552/william-j-lynn-iii/defending-a-new-domain

Lyons, B. (2011, July). *Applying a holistic defense-in-depth approach to the cloud.* Available from http://www.niksun.com/presentations/day1/NIKSUN_WWSMC_July25_BarryLyons.pdf

Mandiant. (2013). *APT1: Exposing One of China's Cyber Espionage Units* (Tech. Rep.). Available from http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf

Markoff, J., & Perlroth, N. (2013, March 26). Firm is accused of sending spam, and fight jams internet. *The New York Times*. Available from http://www.nytimes.com/2013/03/27/technology/internet/online-dispute-becomes-internet-snarling-attack.html

Markowsky, G., & Markowsky, L. (2011, July). Using the castle metaphor to communicate basic concepts in cybersecurity education. In *2011 international conference on security and management (sam '11)* (pp. 507–511). Las Vegas, NV.

Merton, R. K. (1936). The unanticipated consequences of purposive social action. *American Sociological Review*, *1*(6), 894–904.

Mitnick, K., & Simon, W. L. (2002). *The art of deception: Controlling the human element of security.* New York, NY: Wiley.

Parker, D. B. (1998). *Fighting computer crime: A new framework for protecting information.* John Wiley & Sons, Inc.

Perlroth, N. (2014, December 4). Banks' Lawsuits Against Target for Losses Related to Hacking Can Continue. *New York Times*. Available from http://bits.blogs.nytimes.com/2014/12/04/banks-lawsuits-against-target-for-losses-related-to-hacking-can-continue/

Peterson, E. (2014, May 27). *Declaration of Special Agent Elliott Peterson in Support of Application for an Emergency Temporary Restraining Order and Order to Show Cause Re Preliminary Injunction.* In the United States District Court for the Western District of Pennsylvania, United States of America v Evgeniy Mikhailovich Bogachev, et al.

Prince, M. (2013a, March 27). *The DDoS That Almost Broke the Internet.* Available from http://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/

Prince, M. (2013b, March 20). *The DDoS That Knocked Spamhaus Offline (And How We Mitigated It).* Available from http://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/

Romanosky, S., & Acquisti, A. (2009). Privacy costs and personal data protection: Economic and legal perspectives. *Berkeley Technology Law Journal*, *24*(3), 1060-1100.

Rowling, J. K. (2005). *Harry potter and the half-blood prince* (1st American ed. ed.). New York NY: Arthur A. Levine Books.

Ruiz, I. (2013, April 2). *Strengthening Campus Security: Letter to the MIT community.* Available from http://web.mit.edu/itgc/letters/security-memo.html#details

Salas, E. (2014, April 7). *Order Denying Wyndham Hotels and Resorts LLC's Motion* *to Dismiss.* Available from http://www.ftc.gov/system/files/documents/cases/140407wyndhamopinion.pdf

Sanger, D. (2010, September 25). Iran fights malware attacking computers. *The New York Times*. Available from http://www.nytimes.com/2010/09/26/world/middleeast/26iran.html

Saydjari, O. S. (2004). Cyber defense: Art to science. *Communications of the ACM*, *47*(3), 52–57.

Schechter, S. E., Dhamija, R., Ozment, A., & Fischer, I. (2007). The emperor's new security indicators. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy* (pp. 51–65). Washington, DC, USA: IEEE Computer Society. Available from http://dx.doi.org/10.1109/SP.2007.35

Schneider, F. B. (2007). *Draft introduction for a textbook on cybersecurity.* Available from http://www.cs.cornell.edu/fbs/publications/chptr.Intro.pdf

Schneier, B. (1999, December). Attack trees. *Dr. Dobb's Journal*. Available from https://www.schneier.com/paper-attacktrees-ddj-ft.html

Schneier, B. (2006, February). *Security in the cloud.* Available from https://www.schneier.com/blog/archives/2006/02/security_in_the.html

Skoudis, E., & Liston, T. (2006). *Counter hack reloaded.* Boston, MA: Pearson Education.

Smith, T. (2012, November 29). $25K upgrade could have prevented hacking, panel told. *Greenville Online*. Available from http://www.greenvilleonline.com/article/20121129/NEWS09/311290024/-25K-upgrade-could-prevented-hacking-panel-told

Stolfo, S. J., Bellovin, S. M., & Evans, D. (2011). Measuring security. *IEEE Security & Privacy*, *9*(3), 88.

Stytz, M. (2004, Jan). Considering defense in depth for software applications. *IEEE Security & Privacy*, *2*(1), 72-75.

Taylor, B. (2008, July 8). Fighting phishing with eBay and PayPal. *Official Gmail Blog*. Available from http://gmailblog.blogspot.com/2008/07/fighting-phishing-with-ebay-and-paypal.html

Tirenin, W., & Faatz, D. (1999). A concept for strategic cyber defense. *IEEE Military Communications Conference Proceedings (MILCOM) 1999*, *1*, 458–463.

The TJX Companies Inc. Form 10-K. (2007, March 28). SEC filing.

USA v. ALBERT GONZALEZ, Governor's Sentencing Memorandum. (2008). Related cases 09-CR-10262-PBS, 09-CR-10382-DPW.

USA v. ALBERT GONZALEZ, Indictment. (2008). US Dist. Court, Dist. of Massachusetts, Case No.08-CR-10223-PBS.

Verini, J. (2010, November 10). The great cyberheist. *New York Times*. Available from http://www.nytimes.com/2010/11/14/magazine/14Hacker-t.html

Verizon Data Breach Investigations Report (Tech. Rep.). (2013). Verizon. Available from http://www.verizonenterprise.com/resources/reports/es_data-breach-investigations-report-2013_en_xg.pdf

Vincenti, W. G. (1990). *What engineers know and how they know it: Analytical studies from aeronautical history.* Baltimore: Johns Hopkins University Press.

Ware, W. H. (1979). *Security Controls for Computer Systems: Report of Defense*

Science Board Task Force on Computer Security (Tech. Rep. No. R-609-1). RAND. Available from http://www.rand.org/pubs/reports/R609-1/index2.html

Wheeler, T. (2014, June 12). *Remarks of FCC Chairman Tom Wheeler, American Enterprise Institute.* Available from http://www.fcc.gov/document/chairman-wheeler-american-enterprise-institute-washington-dc

Woodward, J. L. (2000, February). *Information assurance through defense in depth* (Tech. Rep.). U.S. Department of Defense. Available from http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA377569

Wool, A. (2004, June). A quantitative study of firewall configuration errors. *Computer*, *37*(6), 62–67. Available from http://dx.doi.org/10.1109/MC.2004.2

Yahoo! Inct v. zheng youjun, claim number: FA1109001409001. (2011, October 31). National Arbitration Forum Decision. Available from http://domains.adrforum.com/domains/decisions/1409001.htm

Zetter, K. (2015, January 8). A cyberattack has caused confirmed physical damage for the second time ever. *Wired*. Available from http://www.wired.com/2015/01/german-steel-mill-hack-destruction/