Some Representational Limitations of the Common Intrusion Specification Language

Jon Doyle

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

October 26, 1999
Revised November 5, 1999

Abstract: This note examines the suitability of the Common Intrusion Specification Language (CISL) for describing events of interest in Cyber Command and Control (CC2).

This work was sponsored by DARPA through its Cyber Command and Control (CC2) program (grant F30602-99-1-0509).

1 Introduction

The DARPA Information Assurance (IA) Cyber Command and Control (CC2) effort requires means for specifying information about computational and communications activities and events, specifically information useful in recognizing these events, interpreting their import for command and control, and generating responses to the events. Participants from the broader IA effort discussed the suitability of various languages for IA, CC2, and Strategic Intrusion Assessment (SIA) at a meeting held in Phoenix, Arizona in August 1999. Though the discussion produced no definitive conclusion, some opinions were given that the existing Common Intrusion Specification Language (CISL) suffices for these several purposes, possibly with some extension of vocabulary.

It is clear that CISL satisfies the immediate intentions of its designers, and provides a common reporting format and transport encoding for reports of concrete descriptions of actual events of the form generated by current-day intrusion detection (ID) sensors. The question addressed here is whether CISL also provides expressive mechanisms adequate to CC2 needs, the object being to determine how CC2 should proceed in adopting, adapting, or inventing languages for its own use.

This note suggests that while CISL contains much useful to CC2, it simply does not provide expressive abilities for major classes of CC2 information. A common vehicle for command and control communications must provide for expression of many things well beyond the original intent of CISL.

One might also ask whether CISL satisfies the needs of SIA or even future sensors that report richer forms of information. The following does not seek to answer those questions directly, but the limitations noted for CC2 needs certainly raise questions about the suitability of CISL for reporting information one can easily conceive of entering into the reports of future front-line sensors.

I warn the reader that this conclusion may say more about my inexperience with the expressive capacities of CISL than about fundamental limitations of the language itself, as I have not had the deep experience with it possessed by the language developers. I thank Cathy McCollum, Bruce D'Ambrosio, and Dan Schnackenberg comments on an earlier draft. I alone bear responsibility for any misreadings and oversights remaining in the comments below.

These comments refer to the document "A Common Intrusion Detection Language (CISL)", authored by Rich Feiertag, Cliff Kahn, Phil Porras, Dan Schnackenberg, Stuart Staniford-Chen, and Brian Tung (editor), draft of June 11, 1999.

2 Summary

CISL provides a reasonably rich vocabulary for conveying the structure of concrete instances of a set of events involving only networked computers. It provides essentially no vocabulary for describing classes of such events; no facilities for quantification or modal qualification; inadequate and inconvenient facilities for representing ambiguity and nonexistence or negation; no facilities for representing trends or other complex behavioral patterns; ill-specified, inexpressive, and essentially meaningless facilites for representing decision-theoretic information about probabilities and utilities; no facilities for describing signals and sensor characteristics; and no facilities for describing the purpose, intent, or operational characteristics of applications or organizational activities, and in particular, no facilities for reporting on intent, execution status of plans, or other key elements of command and control.

I expect one can take the rather general CISL syntax and expand on the content of the language to address the issues raised below, since CISL syntax is based on S-expressions, which also form the underlying syntax for numerous fairly expressive languages for representing general knowledge. This syntactic generality contrasts strongly, however, with the very narrow scope of the content of CISL, the vocabulary of which contains very little beyond the bare minimum needed to report the specific conclusions of current-day intrusion detection systems.

A more serious limitation of CISL concerns its low-level digital encoding scheme, which apparently restricts individual representations to only describing small objects and short lengths of time in the near-term future. These restrictions surely do not matter in the realm of microsecond responses to internet packets and millisecond performances of operating system primitives prior to the provision of truly secure global computer and communications infrastructure, but the restrictions do promise to limit the continuing utility of CISL in describing strategic intrusions and other long-term, large-scale activities.

3 Command and control reporting requirements

Command and control activities require reporting of information of information about the current situation. Situational information includes several broad categories of information.

The first category includes information about actors, their missions, and their progress, such as identifications of the actors present or hypothesized and their relations to one another; their intentions, motivations, and plans and the relations of these to the the doctrine, procedures, or supposed workflow of the actors; the progress of the actors in carrying out their intentions; the role of environmental objects in carrying out these intentions; the status of objects and actors as it relates to carrying out plans and intentions.

The second category includes information about signals, sensors, tasks, and mechanisms, such as identifications of sources of signals, their sampling rates, information content, the operating characteristics and placement of sensors, resource requirements of tasks, and input-output behavior of applications or systems carrying out these tasks.

The third category includes information about the structure of classes of events and criteria, both statistical and nonstatistical, for recognizing when concrete and abstract events occur in signals.

The fourth category includes decision-making information to aid the commander in interpreting, exploiting, and responding to reports, primarily the probability of events or circumstances and the utility of outcomes and actions.

The following sections provide examples of limitations of CISL relating to each of these categories. The main content of these sections consists of example statements a CC2 system or its components might transmit or employ. We phrase most of these schematically, using English terms without precise definitions, in order to convey the sense of the needs without proposing specifics of a solution. In all cases, one could generate many more examples without much difficulty, but hopefully the ones presented illustrate the apparent limitations of CISL.

4 Actors, mission, and situation

CISL has plenty of term concerning computers, networks, packets, and the like, but essentially nothing on actors (other than user accounts and OS processes), their intentions and plans, their workflow or doctrine, or aspects of the world situation (politics, conflicts, physical attacks) related to these intentions. CC2 systems must be able to report tentative hypotheses about actor identifications, actor intentions, and progress in carrying out intentions.

CISL does provide the "ByMeansOf" conjunction, but this does not distinguish causal abstraction from intention. An autoclave sterilizes a knife by means of high temperatures; the murder's intention was to destroy DNA samples. I cannot tell if CISL provides for hypothetical events, or only just concrete ones. One cannot talk about failed intentions without the former (he intended to destroy the DNA by means of the autoclave, but forgot to plug it in). ByMeansOf also apparently does not allow branching, where two simultaneous conjoined events constitute the more abstract event. Perhaps tellingly, the Adverb SIDs currently include only When and Outcome, not Why or How.

Even in the computational realm, existing CISL terms may prove too limiting for description of the CC2 status. The "ProcessStatus" term may provide a case in point. This term allows only 256 possible values, which apparently refer to common ways that operating systems classify the runability status of processes. While there certainly is a role for such OS classifications, and while such OS classifications may well prove small in number, descriptions of the CC2 situation probably need a much richer notion of process status, starting with abstract categories and possibly bottoming out in provision of the actual image in memory of the running process.

Examples:

- The intent of this denial of service attack is to cripple the FBI's intranet.

- The intent of this denial of service attack is to distract us from an ongoing probe of NYSE systems.
- This attack was intended to install a trojan horse in the Serbian embassy's computers but wiped out the Chinese embassy's computers instead due to outdated Internic records.
- Attacks on our command resources are increasing, but their success rate is dropping.
- These attacks pose no threat to the Whitehouse computers because the attacker does not seem to be competent. It appears they are really trying to hit the pornographic site Whitehouse.com.
- The Russo-Columbian mafia is trying to take out all three branches of the US government, but so far they've only managed to flood the Department of Agriculture, and our defenses seem to be tying up their attacks in Congress.

5 Signals and sensors

The first need here is to be able to describe where signals come from and what properties they exhibit, such as sampling rates, delay, and jitter. The second need is to be able to describe the operating characteristics (selectivity and sensitivity) of sensors. The third need is to be able to describe trends, waveforms, and statistics characterizing the observed signals.

One also needs the ability to characterize the effects of different operations and systems. Specifications of points on ROC curves provide some of this information for sensors, but more generally damage assessments must be able to convey how systems are failing to meet their normal input-output relations.

The CISL document early on (section 3.1) speaks of components receiving an input stream and producing an output stream. It isn't clear to me whether the language is intended to cover multiple input and output streams for a single component. In any event, the language does not provide a way of describing these input and output streams at all.

Perhaps a symptom of this is that CISL provides a "Message" term but no corresponding description of message types. Indeed, the "Message" terms describe only the notion of messages at a very low level of abstraction, such as Ethernet and IP packets, and would require extension to cover higher-level message types.

Examples:

- Sensor X samples packets at an average rate between 1 and 1.2 per millisecond, and cannot track 100MB ethernet traffic.

- Sensor X reported an oscillating level of congestion on the network, with a frequency starting at once per hour, but then increasing over the span of a day to a frequency of six times per hour.
- The observed success rate of attacks has been decreasing over the past two days.
- The probability of being attacked has been increasing steadily for the past hour.
- The traffic volume through node X has been increasing while the traffic volume through Y has been decreasing, all during a period in which the frequency of port scans has dropped precipitously.
- Sensor X is operating at selectivity Y and sensitivity Z.

6 Events classes and properties

The main need here is to be able to describe classes of events and criteria for recognizing their occurrence in signals and sensor outputs. These descriptions must include abstract entities such as landmark time points or boundaries, comprehension rules, and ambiguous or absent events. CISL currently provides the logical operation of conjunction and terms which represent specific quantified or negated concepts, but no general means for expressing constants of indeterminate value, disjunction, negation, implication, universal or existential quantification, or modal qualification (possibility, necessity, belief, obligation, etc.). It provides means for expressing concrete times, but not temporal intervals or indefinite time points, or temporal intervals with indefinite endpoints.

Examples:

- A software piracy transshipment event consists of an interval of increasing levels of seemingly unsuccessful attacks, immediately followed by an interval of normal operations of duration between one and twenty-four hours, followed by an interval of increasing load average and FTP operations.
- An mixed-mode communications attack event consists of a computational attack on the network during which occurs a series of bombing and line severing attacks against physical network facilities.
- A lull event consists of an interval of normal operations of an enclave.
- An false-lull event consists of lull during which all traffic from sister enclaves has disappeared.
- A serial gang attack event consists of an attack by an attacker X and an attack by an attacker Y, where the attack by X precedes but overlaps the attack by Y, and where the overlap of the attacks lasts at least one hour.

7 Decision-making information

CISL provides ways of specifying numeric measures of certainty and severity which may have been intended to help convey information needed to make command decisions. Unfortunately, these measures are essentially meaningless.

Take certainty measures first, which we may interpret as kin to probability information, and which engendered a protracted discussion at the 1999 DARPA BAA 98-34 kickoff meeting. The CISL specification does not distinguish between certainty measures that represent conditional probabilities, which relate to the reliability of sensors and correlation mechanisms, and posterior probabilities, which relate to all-things-considered judgments. This distinction is crucial, for CC2 must combine information from multiple sensors and sources. Such combination requires conditional probability information, for these can be combined using standard rules of probabilistic inference. Posterior probabilities, in contrast, cannot be combined in any sensible way without internal information about the various factors that went into their construction, which if one is lucky one can use to remove these additional factors and recover the underlying conditional probabilities.

While CISL provides a "HelpedCause" construct, it provides no way of expressing conditional probability relations between combinations of these causal factors and the causal outcome. In particular, it has no way of expressing "noisy-or" or other constructs of common utility in probabilistic modeling.

Now consider severity measures. The CISL specification suggests we may interpret these as kin to normalized expected utility judgments. These measures seem even more problematic than the certainty measures, for they combine posterior probability judgments with utility judgments from the perspective of the system making the statement, without any indication of what factors went into these embedded utility judgments. The same compromise can have very different utilities to different agents, and to the same agent at different times or during different activities. The severity measure specification provides no way of expressing such distinctions.

Examples:

- The probability of an attack has been increasing for the past hour.
- Attacks are increasing on the more important targets in spite of attempts to conceal their locations.
- The danger posed by successful attacks is increasing because system degradations are increasing the concentration of key resources.
- It is becoming more difficult to detect attacks; the probability of detecting attacks has been dropping the past hour.
- The number of attacks and success rate has been constant over the last week, but the disutility of the compromises has been increasing steadily.

8 CISL encoding limitations

CISL employs a digital encoding scheme that prevents representation, or at least convenient representation, of large objects and extended temporal intervals. It also ignores the lessons of the Y2K difficulties and builds in temporal representations that work only for less that four decades. Finally, it limits the number of events it is possible to communicate succinctly.

The digital encodings of CISL statements start by stating their length in octets, 8-bit bytes. This length must fit in 32 bits, so the maximum length of a CISL statement is approximately 1.1 terabytes. This length limitation does not matter in current CISL applications, and may not matter for some time to come. One may conceive, however, of national-level CC2 systems requiring the ability to report large files or data sets, or perhaps sets of files and change histories, say to permit forensic analysis by the FBI or to facilitate offline data recovery. In such cirucumstances, in which the reporting need may be conveying the image of a hard disk many times over, one might well exceed the terabyte capacity of CISL. Given the rapid escalation in the size of storage devices via Moore's law, prudence might call for enlarging this length restriction or avoiding it altogether.

A more immediate limitation of CISL stems from the limitation of its digital encoding to representing microsecond durations of events by 32 bit numbers. This restricts durations to about a million seconds, or under 13 days (about the length of an IFE). CC2 events routinely exceed such durations, as red teams probe for weaknesses over the course of months or years.

The digital encoding of CISL uses a timestamp system based on the 32 bit Unix epoch 1970. This forbids representation of times beyond sometime in 2038, less than four decades hence. Times beyond that point surely do not enter into reports of generated today, but one expects a good communication language to stay in use quite some time, and it seems foolish to acquiesce to such a near-term limitation in designing a program-independent event specification language. More immediately, this encoding means that CISL messages cannot refer to events occurring prior to 1970, as they might need to do to convey the manufacture date of some computer, missle system, or historical anniversary (Guy Fawkes Day?) motivating terrorist groups.

The digital encoding uses fixed-size encodings for event classes and types, where these sizes seem small compared with the sizes of actual and contemplated knowledge bases. The Class ID header field limits the number of categories of reports to about 64K. Since a basic English dictionary has about 50K concepts, this seems woefully small. It seems smaller still since big blocks are allocated to specific existing classes. Perhaps the intent is that the somewhat larger fields for representing AttackIDs carry the burden here, but not all events of interest constitute attacks.

The IP address component of the Originator ID encoding provides only four octets. Are future expansions of IP likely to retain this small address space size?

I do not understand the referent SID mechanisms or their restrictions, in particular the proscription against ever reusing a referent in the same thread. I don't understand what threads are. Does this proscription mean no component can ever reuse a referent for anything different, no matter when? What happens when referents are passed on from one component to the next? Do these components have to avoid reuse as well? If so, what happens when independent components use the same referent, and pass these distinct uses to the same receiver?

9 Conclusion

However well it satisfies its design intentions, CISL does not serve the needs of CC2. It may be possible to extend it to serve these broader needs, but this constitutes a major expansion of the language requiring substantial effort by numerous designers.