

Heterogeneous Sensor Correlation: A Case Study of Live Traffic Analysis

Dan Andersson (daane@ce.chalmers.se), *
Martin Fong (fong@sdl.sri.com),
Alfonso Valdes (valdes@sdl.sri.com)

Abstract

As enterprises deploy multiple intrusion detection sensors at key points in their networks, the issue of correlating messages from these sensors becomes increasingly important. A correlation capability reduces alert volume, and potentially improves detection performance through sensor reinforcement or complementarity. Correlation is especially advantageous when heterogeneous sensors are employed because of the potential to aggregate different views of the same incident. Emerging standards for sensor interoperability with respect to alert reporting facilitate the function of correlation engines, but these standards are still at an early stage of development. Furthermore, it is apparent that these standards will not enforce uniformity in, for example, attack description, complicating the task of correlation. The immature state of standards and nonuniformity of reporting both argue for correlation technologies that are robust, flexible, and function with comparatively few underlying assumptions. Herein, we present a case study of correlating several sensors listening to live traffic using a probabilistic correlation approach.

Keywords: Network security, sensor correlation, alert management, adaptive systems

Acknowledgment. This research is sponsored by DARPA under contract numbers F30602-99-C-0149 and N66001-00-C-8058. The views herein are those of the author(s) and do not necessarily reflect the views of the supporting agency.

* This work was performed while Dan Andersson was an International Fellow at the System Development Laboratory, SRI International. Correspondence should be sent to Mr. Valdes.

Introduction

Intrusion detection technology has gained increasing acceptance in enterprise networks, with both commercially supported and open source components widely deployed. It can be argued that diverse sensors provide more complete coverage of the attack space. However, sensor diversity has resulted in a new difficulty, namely, making sense of the reports from numerous sensors using diverse approaches. As a result, the potential added leverage from sensor diversity is blunted, because, for example, it is difficult for the security administrator to ascertain which reports pertain to the same or to different incidents.

To facilitate interoperability of diverse sensors, the IETF/IDWG is developing a standard format, Intrusion Detection Message Exchange Format (IDMEF), allowing sensors to generate messages in an XML syntax [1]. Evolving correlation technologies will of necessity rely on such standards. The Silicon Defense XML plug-in for the open source intrusion detection component, SNORT [2, 3] is an early product on the market that produces messages in this format.

Heterogeneous sensor correlation provides a number of potential advantages in the operation of an enterprise sensor suite. The most obvious benefit is the reduction in the number of alerts that a security officer must address. When considering a single sensor, a correlation engine can reduce alert volume by organizing numerous alerts that are part of an ongoing attack, a capability we call *alert threading*. In the case of heterogeneous sensors, a correlation engine should recognize when reports from multiple sensors refer to the same incident. Correlation can enhance detection capability, and give a more complete picture of attacks that an individual sensor may observe only partially. In addition, correlation can exploit the complementary coverage from several sensors. Reports from several sensors employing diverse analytical techniques may reinforce each other and therefore enhance the confidence of the detection, although this reinforcement should not be rooted in a claim of statistical independence of the sensors.

However, the evolving IDMEF standard does not require uniformity of incident reporting, allowing instead a “vendor specific” designation and a URL for incident description. We fear this may lead to a “tower of Babel” situation where each IDS developer defines hundreds of unique attack descriptions. Needless to say, making sense of this is a severe burden on a correlation engine functioning in real time. IDMEF issues are further discussed below under Future Work.

Relatively few correlation components are presently deployed. SPADE [7] presents an approach, but is limited in applicability to portscans. The advanced research prototype from the EMERALD group at SRI [4] uses probabilistic techniques to aggregate diverse sensor reports according to feature similarity functions as described in the next section. The approach is relatively tolerant of incomplete, inconsistent, or inaccurate alert reports.

In this paper, we present a case study of the deployment of several sensors listening to live network traffic, and the application of correlation technology in a real network environment. We identified several issues with the reporting standards, as well as the implementation of the SNORT XML plug-in, that impact the quality of the correlation result achievable. With modifications to the plug-in, we hope to improve our results.

The rest of this paper is organized as follows. In the next section, we give an overview of the sensor correlation component used in this analysis. We follow this with a description of the sensor landscape. We then examine the sensor and correlation results over a period of observation. This period includes a series of vulnerability probes launched by our central IT office against various administrative domains in our company, as well as whatever attacks (such as residual Code Red probes) happened in the wild at the time.

Sensor Correlation

The EMERALD probabilistic correlation engine employed in this live traffic analysis is the system described in [4], and the interested reader should consult that report for a more in-depth treatment of the details of the approach. Briefly, the system considers features in alert reports and evaluates the similarity between two alerts (or a new alert and a meta alert) as reported in a standard alert template. For two alerts (typically a new alert and a meta alert), we begin by identifying features they have in common (feature overlap). Such features include the source of the attack, the target (hosts and ports), the class of the attack, and time information. For each overlapping feature, we have defined an appropriate similarity function that returns a value between 0 and 1. New alerts are compared to a list of existing meta alerts. We assign an expected similarity and a minimum similarity to each feature that depends on the situation as well as user preference. For example, to correlate incidents, the user may require a tight match on time, source, and target, while relaxing the match requirement on the reporting sensor.

The overall similarity is the weighted average of the feature similarities, with expectation of similarity as the normalizing weights. If the minimum match criterion fails for any feature, the match is rejected regardless of the overall similarity. The new alert is correlated with the most similar meta alert, assuming the similarity is above a specified minimum match threshold. Otherwise, the new alert starts a new meta alert thread. Each meta alert may thus represent a single alert or may be composed of several alerts, potentially from heterogeneous sensors.

As stated above, we have a similarity function that returns a number between 0 and 1, with 1 corresponding to a perfect match for each feature. Similarity is a feature-specific function that considers such issues as

- How well do two lists overlap (for example, lists of targeted ports)?
- Is one observed value contained in the other (for example, is the target port of a denial-of-service (DOS) attack one of the ports that was the target of a recent probe)?
- If two source addresses are different, are they likely to be from the same subnet?

In order to address the nonuniformity of attack descriptions, we are forced to examine the alert messages from each system and assign these to a much smaller number of classes. We prefer to consider attack classes rather than attack signatures, which are much more specific and numerous but may be reported inconsistently across sensors. We implement attack classes and signatures in the EMERALD incident handling knowledge base (IHKB) which is shared by all EMERALD sensor and correlation components.

Currently, 13 IHKB classes encompass hundreds of signatures from a variety of sensors. This approach, with defined fallback classes for INVALID and ACTION LOGGED,

tolerates a degree of mismatch across sensor reports for the same incident. For attack class similarity, we maintain a matrix of similarity between attack classes, with values of unity along the diagonal and off-diagonal values that heuristically express similarity between the corresponding attack classes. The similarity matrix attempts to express, on a 0 to 1 scale, how plausible it is that an attack step of class B follows one of class A.

Not all sensors produce all possible identifying features. For example, a host sensor provides process ID, while a network sensor does not. Features not common to both alerts are not considered for the overall similarity match.

The meta alert itself supports the alert threading concept described previously, so we can visualize composing meta alerts from meta alerts.

By appropriate settings of similarity expectation and minimum similarity, the correlation component achieves the following hierarchy of correlation. The system is composable in that we can deploy multiple instances to obtain correlation at different stages in the hierarchy. For example, we can infer threads (within sensor correlation) and then correlate threaded alerts from heterogeneous sensors into security incidents.

Synthetic Threads: For sensors that do not employ the thread concept, the correlation synthesizes threads by enforcing high minimum expectation similarity on the sensor itself (the thread must come from a single sensor) and the attack class, as well as source and target (IP and ports). We have wrapped the alert messages from a leading commercial sensor and observed that this facility reliably reconstructs threads and significantly reduces alert volume in simulated attack environments [4].

Security Incidents: By suppressing minimum expectation of similarity on the sensor identifier, and relaxing expectation of similarity for this feature, we can fuse reports of the same incident from several heterogeneous sensors into a single incident report. In this case, we enforce a moderately high expectation of similarity on the attack class. This is not unity, because different sensors may report a different attack class for the same attack. We construct a similarity table between attack classes that expresses which ones are acceptably close. For security incident correlation, we enforce minimum expectations on the source and target of the attack. Using this technique, we have been able to fuse alert reports from commercial and EMERALD sensors into security incident reports.

Correlated Attack Reports: By relaxing the minimum expectation of similarity on the attack class, we are able to reconstruct various steps in a multistage attack. Each stage in an attack may itself be a correlated security incident as described above. In this fashion, it is possible to recognize a staged attack composed of, for example, a probe followed by an exploit to gain access to an internal machine, and then using that machine to launch an attack against a more critical asset.

Sensors Suite

To evaluate the utility of correlation, we considered a diverse set of intrusion detection monitors. By “diverse,” we mean diversity of analytical technique. The following sensors were deployed:

- EMERALD eXpert-Net. This rule-based system [5] keeps a moderate amount of session state, and has signatures for a number of attacks against TCP, http, FTP, SMTP, ICMP, and UDP protocols.
- EMERALD eBayes TCP. This system detects attacks visible in TCP header data using Bayes inference [6]. It is particularly effective against general probe attacks.
- SNORT 1.8 [2] is a widely deployed open source network IDS, which currently contains more than 1200 rules. We ran SNORT with the IDMEF XML plug-in from Silicon Defense [3]. We implemented a translator to convert the XML alerts to EMERALD binary messages.

The EMERALD sensors are installed on an IDS network appliance, which is a dual-processor platform with a passive interface to the network gateway and a private interface for alert reporting and administrator functions. Thus it is not visible on the monitored network. We have been developing this appliance under various EMERALD programs for some months, and have extensive experience using it to examine live traffic. In particular, we are pleased with its stability; the appliance runs for weeks without intervention.

SNORT is installed on an Intel i686 machine listening to the same interface. In a day of monitoring, SNORT observed 7.8 million packets. According to SNORT diagnostics, no packets were dropped. Due to the Code Red activity at the time of monitoring, we do not know if this volume is higher than average for the network in question. At any rate, the SNORT process was never more than a few percent of the total CPU load on the machine. However, we have observed that, while SNORT CPU use remains minimal, the memory usage appears to grow. For example, after eight hours of operation, the SNORT process grows from 9 to 39 MB. Figure 1 below illustrates SNORT process growth over time. We do not know the reason for this memory growth.

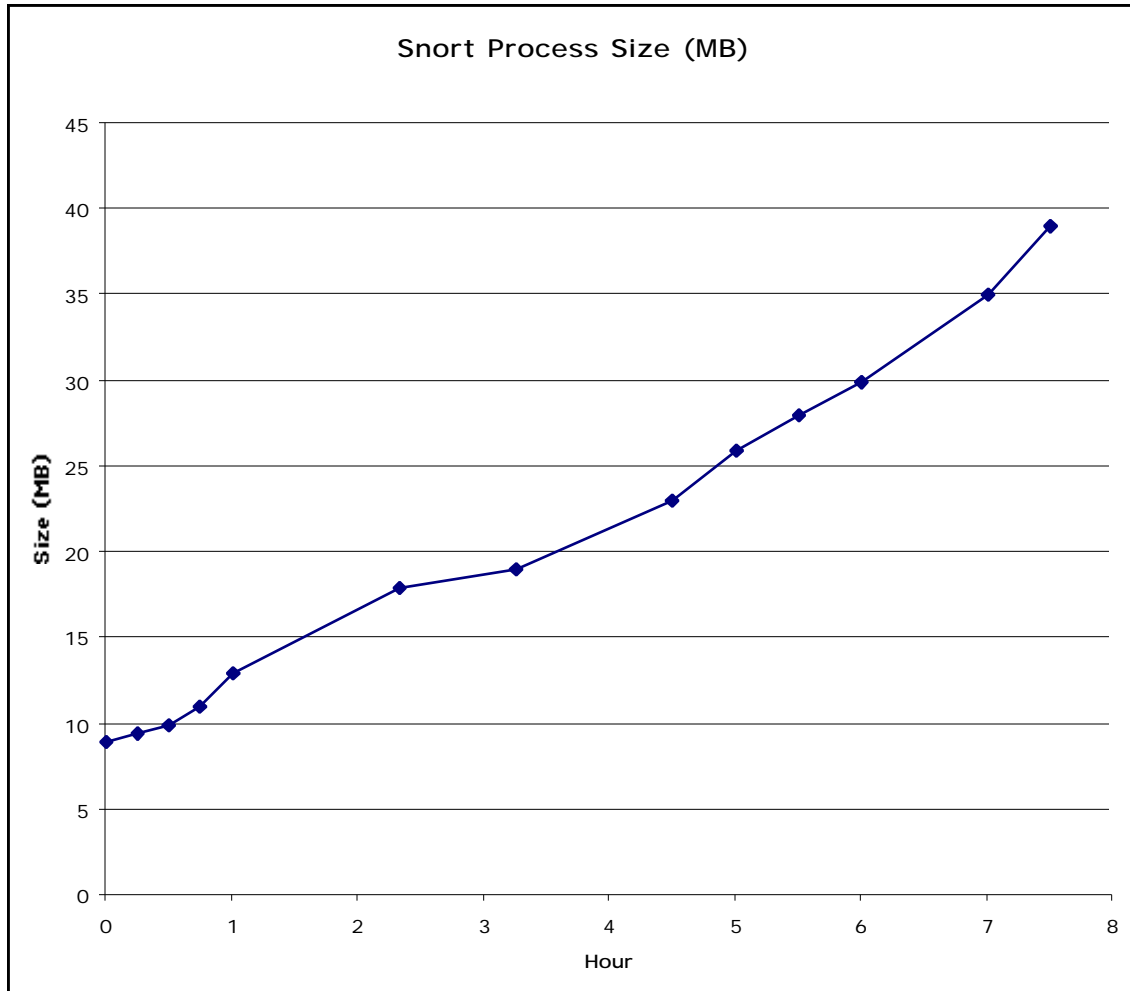


Figure 1. SNORT Memory Usage

The EMERALD components implement the concept of *alert threading* by the system maintaining a degree of session state and is thus potentially consolidating a large number of messages from a single attack into a smaller number of messages and updates. Systems such as SNORT generate a very large number of messages for certain attacks (e.g., portsweeps), presenting a difficult management problem for the security officer, as well as possibly burying more important alert messages. SPADE mitigates this difficulty to a degree, but early on we identified the need to infer synthetic threads as an important capability of a correlation engine.

The next section describes our recent experience with correlation of alerts from these components listening to our internet gateway.

Live Traffic Analysis

Evaluation studies of this type are faced with a quandary, trading off the controlled conditions of a test environment for which ground truth is known versus the realism of actual internet traffic. We have (perhaps impetuously) opted for the latter, because we wish to consider the “value added” of sensor correlation, and we need realistic alert

volumes and false positive rates. We have considerable experience with the EMERALD sensors in continuous operation for weeks or months against live traffic. Although we do not have ground truth, we have a sense of what these systems report. Also, the monitoring period included a set of probes launched by our central IT staff, which gives us some known attack data. The experimental period also encompassed a period of high activity of the Code Red and Code Red V2 worms.

The probabilistic correlation engine considers attack class as one of the features in its similarity matching algorithms, as described in [4]. For the first look of this analysis, we were not able to populate the mapping of SNORT alerts to EMERALD incident classes, so SNORT alerts are assigned to a fallback “ACTION LOGGED” class. An advantage of probabilistic techniques is that this approach produces slightly lower fidelity results, but the technique is sufficiently robust to tolerate this as a minor deficiency.

Because of the overall architecture of EMERALD, we are able to deploy a correlation capability at one or more points in a monitoring network, and can in fact correlate correlated alerts. We chose to separately correlate the SNORT alerts, the EMERALD alerts, and the entire set. The first function of correlation, as presented in the introduction, is to reduce the raw number of alert reports that a security administrator must examine. The following reflect totals for a one-day collection period in our laboratory, starting at 10 am PDT, August 6-7, 2001.

Sensor	Raw Alerts	Correlated Alerts
SNORT	4816	487
EMERALD	1586	523
Composite	6402	869

As described above, it is possible that a raw alert will fail to correlate with any other alerts. In this case, the corresponding correlated alert will consist solely of the contents of the single contributing raw alert. Therefore, the set of correlated alerts contains information for all of the raw alerts. We observe that correlation achieves about a 10 to 1 reduction in SNORT alerts, and about 3 to 1 for EMERALD alerts. This occurs because the EMERALD sensors attempt to thread alerts, as we have previously discussed.

Sensor Complementarity

The most frequent SNORT alert for which there is no corresponding EMERALD alert is the DNS named version attempt. The full SNORT set contains 815 alerts of this class, which reduce to 224 correlated alerts, or 47% of the total SNORT correlated alerts. To the degree that we can ascertain, we believe most, if not all, of the sessions in question reflect innocuous activity, and these are most likely false alarms. Of these, one alert correlated with an alert from the Bayes sensor, which considered the session’s combination of ports highly unusual. In addition to the DNS request, the same session accessed ssh. This detection was from the Bayes sensor’s anomaly detection capability and implies that the port combination was extremely unusual but not necessarily a

probabilistically good match to any of the misuse models in the Bayes sensor's knowledge base.

The Bayes sensor also detected port sweeps such as sessions that probe for NETBUS and SUB7. The approach used is probabilistic, not based on a rule such as "so many ports within a given interval of time" nor on a rule such as a match to a pattern of ports corresponding to a known probing script such as MSCAN. The Bayes sensor was the only sensor in the mix to report this apparent probe.

One likely DNS sweep detected by the EMERALD Bayes and eXpert-TCP sensors probed 255 IP addresses on port 53. This was not reported by SNORT.

The correlator assembled the following apparent multistep attack from several reports from SNORT and the EMERALD Bayes sensor.

- An attempt to connect to port 1243, detected by the Bayes sensor.
- A near-simultaneous DNS request from a different IP address in the same subnet as above, detected by SNORT.
- Nine hours later, a connection attempt to port 3128, detected by both sensors, from the same subnet.
- One hour later, a connection attempt to port 3150, detected by the Bayes sensor, from the same subnet.

The composite alert is an apparent probe distributed in time and source for ports 53, 1243, 3128, and 3150. Formation of this composite alert highlights the utility of the techniques we are proposing. The Bayes and SNORT sensors have complementary coverage, and the ability of the former to detect probes with no prior signature is essential to seeing parts of this attack. In live operation, the initial alert is reported as it is generated, and future alerts are considered updates to this one. The ability to correlate over time and across sensors is the final critical capability needed to form a more complete picture.

SNORT reports a number of ICMP-related alerts, such as "ICMP Ping," "ICMP Source unreachable," and "ICMP Source quench," for which there is no EMERALD counterpart. For these, the reports in the correlated set come only from SNORT. We have not yet investigated these to see which, if any, are valid alerts. The originating address of the "Source unreachable" alerts is our network switch, which leads us to believe that the traffic is legitimate.

Sensor Reinforcement

In addition to the cases cited above, the sensors agreed on a large number of apparent attacks. As mentioned previously, the monitoring period contains a set of scripted vulnerability probes launched by the central IT office at SRI, intending to uncover vulnerabilities that may be unknown to individual laboratory domains. All the sensors in the suite generate numerous alarms from these probes.

Even though sensors may alert in response to the same incident, we found (as expected) that the calls are often different. For example, signature systems may have an explicit

signature for BACK ORIFICE, whereas the Bayes sensor detects this as an extremely unusual access attempt. To permit correlation of these messages, the correlation engine must explicitly allow imperfect matches, and should work at the incident class rather than signature level. Our probabilistic engine satisfies this requirement.

Code Red activity over the period in question was sufficiently significant to warrant further discussion. Both SNORT and eXpert-TCP were used with signatures for this worm which were very new at the time of our analysis. The Bayes sensor potentially detects Code Red activity if several target IP addresses are probed; in this case, the alert is a TCP ADDRESS SWEEP which increases in confidence as more addresses are probed. SNORT detects Code Red and Code Red II with the rule “WEB-IIS .ida attempt.”

To the best of our ability to analyze the results ex post facto, all Code Red attempts that trigger the SNORT “WEB-IIS .ida attempt” are also detected by one or both of the EMERALD sensors. The correlation facility identified repeated re-infection attempts from the same source or source subnet. We observed Code Red traffic from several sources within the same subnet in numerous correlated Code Red alerts. The correlation engine effectively aggregates these reports across time, sensor, and source subnet. As we have no IIS servers in our facility, all of these attempts failed.

Code Red II often triggers the SNORT “WEB-IIS cmd.exe” rule in addition to the “WEB-IIS .ida attempt” rule, although we observed some instances of “cmd.exe” with no “WEB-IIS .ida attempt” alert. We observe that the payload field for these reports looks like that of the alerts related to Code Red, so we are not certain why the traffic did not trigger the prior rule as well. Since EMERALD did not alert for these sessions, correlation sheds no more light on this issue.

Future Work

For the immediate future, we intend to populate the EMERALD incident handling knowledge base to comprehend the SNORT signatures. We will extend the SNORT XML utility to provide the SNORT rule and version corresponding to each alert, which will then enable us to obtain an accurate incident class for SNORT alerts. This will enable a higher-fidelity analysis of heterogeneous sensor alerts.

We have found that transforming SNORT signatures into the EMERALD attack class incident handling knowledge base (IHKB) is not straightforward. Within the IDMEF specification, the optional “Analyzer” attribute “analyzerid” does not adequately describe the IDS sensor/analysis component. Thus, there is no automatic way to interpret the Alert Classification through, for example, predefined tables. Exacerbating this problem, the Classification attribute “origin” can have only one of four values: “unknown”, “cve”, “bugtraqid”, and “vendor-specific”. Unfortunately, because neither the cve nor bugtraqid classifications provide a complete and consistent alert taxonomy (for example, alerts are not vulnerabilities, per se), the only usable origin is “vendor-specific”. However, this attribute value employs a global namespace for the Classification name, making the task of simultaneously consuming data from multiple vendor-provided sensors nontrivial.

To address these deficiencies in the IDMEF specification (and consequently, conformant applications and libraries), we will add the following named Classification attributes:

- vendor (e.g., “snort.org”),
- product (e.g., “snort”), and
- version (e.g., “1.8”),

These three attributes, whose values are managed by their respective vendors, effectively segment the Alert Classification namespace. Because the Classification name is occasionally used as an alert description, we also will also modify the SNORT IDMEF module to support the addition of these snort-specific Classification attributes:

- sid_rev (viz., snort ID ‘:’ module revision number),
- class_priority (viz., class priority).

As far as other IDMEF specification mechanisms are concerned, we think that it is extremely unlikely that the Classification URL can or will be used by realtime analysis components to resolve Classification names. For example, with data sets we’ve received from third parties, some of the URLs resolve into Adobe PDF files while others are references to expired http links.

Once we extend our IHKB and modify libidmef accordingly, we will incorporate the EMERALD Mission-Based Correlation functionality [8] to further leverage the utility obtained from a heterogeneous system of the type described here. Mission-Based Correlation extends the utility of what we have introduced here with the ability for multiple subscribers to specify different preference profiles. These profiles express criticality of system resources, concern about particular classes of attacks, and an adaptive capability to provide to each subscriber a customized list of ranked, prioritized correlated alerts. We would be able, for example, to assign a lower priority to the DNS alerts from SNORT without taking the more drastic approach of disabling the rules entirely.

Moreover, it is not clear that the proposed IDMEF standard is adequate for reporting correlated alerts. IDMEF specifies that a correlated alert is essentially a series of data base keys to the individual contributing alerts. Such a report does not permit inferences that are possible only with some collective knowledge of the alerts, such as what information or capabilities an attacker has acquired at some step in the attack. For example, the system in [4] is aware of incident classes that correspond to host compromise. If there is a reliable indication of successful outcome for an attack step from this class, the victim machine is considered a potential asset for the attacker for future attack steps (that is, it is both a target and a potential source). The rationale for correlating the attack sequence may not be apparent, and reassembly of the entire sequence may not be possible, unless the system keeps more state for the correlated alert than simple data base pointers.

Summary

We have presented a case study of practical IDS alert correlation using heterogeneous sensors, test-driving the IDMEF standard as far as its ability to provide adequate content

for such a study. This was carried out at a time when the Internet as a whole was subject to a fairly high attack frequency due to variants of the Code Red worm. We sought to explore the ability of heterogeneous sensors and sensor correlation to provide alert volume reduction, complementary coverage, and sensor reinforcement. Considering SNORT as a widely deployed sensor that does not thread alerts, we are able to reduce alert volume by an order of magnitude. The alert volume reduction for the EMERALD sensors is less, in part because these sensors implement alert threading. We may argue that 869 alerts a day is an unacceptably high number of reports for an administrator to comprehend. However, the 224 SNORT DNS alerts can be assigned lower priority, and we must recognize that many of the rest are very likely valid because of the Code Red activity.

We were able to identify likely attacks where sensor heterogeneity and correlation brought out a more complete picture of what actually happened. We have identified enhancements we would like to make to the IDMEF standard to extend the functionality of our correlation engine. However, even at this stage of the game, the probabilistic approach of our engine has proven itself sufficiently robust to be extremely useful to a security administrator.

References

1. Curry, D. and Debar, H. "Intrusion Detection Message Exchange Format", <http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-03.txt>
2. Roesch, M. <http://snort.sourcefire.com/>
3. Silicon Defense. <http://www.silicondefense.com/idwg/snort-idmef/>
4. Valdes, A. and Skinner, K. "Probabilistic Alert Correlation", to appear in the proceedings of Recent Advances in Intrusion Detection (RAID) 2001, Springer-Verlag Lecture Notes in Computer Science.
5. Lindqvist, U. and Porras, P. "Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)", Proceedings of the 1999 IEEE Symposium on Security and Privacy IEEE Computer Society Press, Oakland, California. May, 1999.
6. Valdes, A. and Skinner, K. "Adaptive, Model-Based Monitoring for Cyber Attack Detection", Recent Advances in Intrusion Detection (RAID) 2000, Springer-Verlag Lecture Notes in Computer Science, October 2000.
7. Hoagland, J., McAlemy, J., and Stanniford, S. "Practical Automated Detection of Stealthy Portscans", <http://www.securityfocus.com/library/3019>
8. Porras, P. <http://www.sdl.sri.com/projects/M-Correlation>