An Approach to Sensor Correlation

Alfonso Valdes, Keith Skinner
SRI International
Valdes@sdl.sri.com

Abstract

We present an approach to intrusion detection (ID) sensor correlation that considers the problem in three phases: event aggregation, sensor coupling, and meta alert fusion. The approach is well suited to probabilistically based sensors such as EMERALD eBayes. We demonstrate the efficacy of the EMERALD alert thread mechanism, the sensor coupling in eBayes, and a prototype alert fusion capability towards achieving significant functionality in the field of ID sensor correlation.

Keywords: sensor correlation

Introduction

Sensor correlation consists of three key functions. First, a sensor or suite of sensors must correlate (or more correctly, aggregate) large numbers of low-level events corresponding to an extended attack; otherwise, the overall system potentially floods the security officer to a point that it becomes essentially useless. Second, a correlation utility should ideally comprehend the results of various sensors, both for improved sensitivity and for false alarm suppression. Finally, a sophisticated correlation engine should be able to recognize a scenario attack. These correlation functions are listed in order of increasing difficulty. The EMERALD system as a whole [1, 2] is well advanced with respect to the first function, which we address with the concept of alert threads. The Adaptive, Model-Based Monitoring effort (eBayes [3]) is, to the best of our knowledge, unique in achieving the second functionality to a useful degree. We believe that the third functionality is effectively addressed by intelligent fusion of content from multiple sources into "meta alerts". Content fusion is done with some awareness of which types of alerts will temporally precede others in a scenario and which features are expected to match.

The reset of this note is organized as follows. We first briefly describe a leading alternative approach to sensor correlation, presented at the July 2000 DARPA ISO

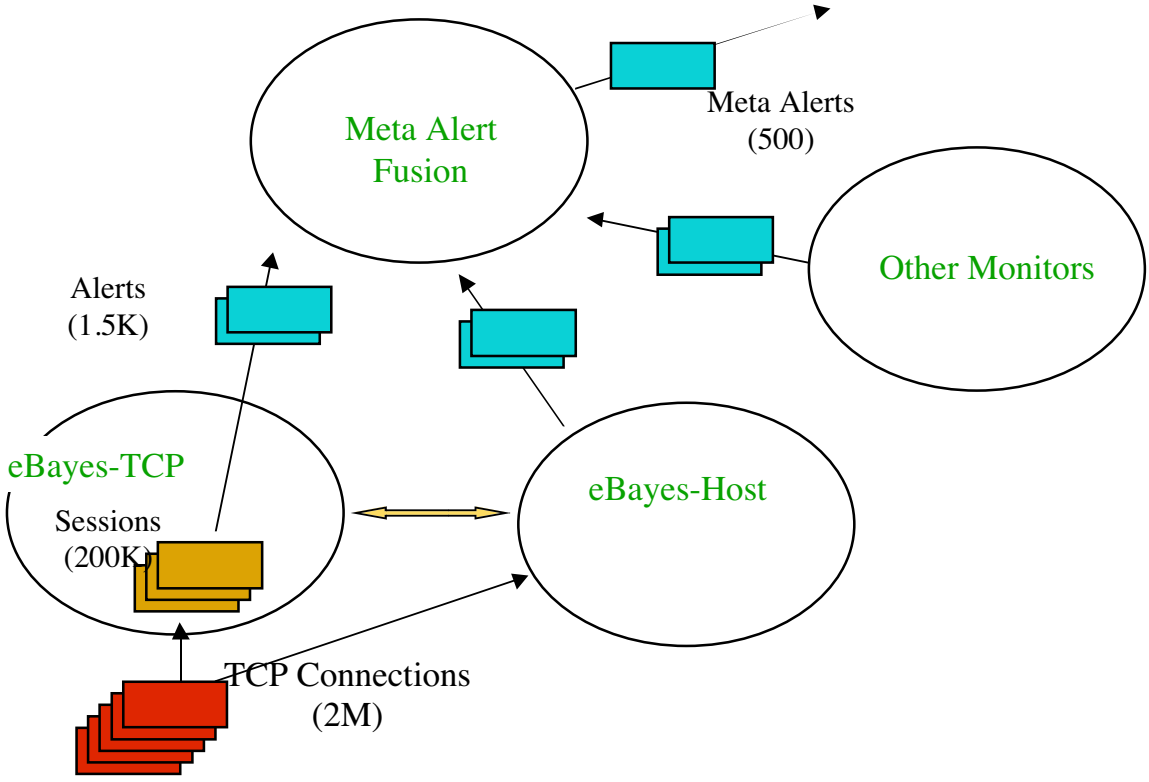
Principal Investigator Meeting. We then present our approaches to correlation: the EMERALD alert thread mechanism, followed by our approach to comprehending multiple sensors to enhance sensitivity and suppress false alarms. We conclude with a brief discussion of the EMERALD alert template, which extends earlier efforts (such as IDIP) with respect to providing the richer content we feel is essentialfor addressing the correlation problem. We have circulated this template to UC Santa Barbara, Honeywell, ISI, and the IETF for their comments.

Alternate Approach to Correlation

An alternative approach to sensor correlation, based on machine learning techniques, was proposed at the most recent DARPA Principal Investigator Meeting by Honeywell Technology Center. The underlying data are reports from various sensors for the Lincoln Laboratory 1999 evaluation data, coupled with knowledge of ground truth with respect to the attacks. The result is a sort of decision tree: for attack type X, take the result of sensor A, or if A does not report, take the "AND" of sensors B and C. As the authors pointed out, this is hampered because most of the sensors (and all the effective sensors) in this study were signature based. As such, reports were all or nothing. A learning technique such as they propose appears more useful if sensors reported a continuous-valued confidence factor. In that case, if one sensor generates an alert and another is silent, there is a chance to query the latter and find out if it "just missed" crossing a threshold, or truly did consider the event sequence benign. In this respect, we can dial back the reporting threshold, and simultaneously address the second correlation functions discussed above. However, given the nature of the sensor reports, this has not been feasible to date. For future evaluations, eBayes appears competitively effective with the better signature engines while giving a continuous output, so it is for them a potentially very useful sensor.

The Three Levels of Correlation

As noted in the introduction, we view correlation in three levels. At the first, we must aggregate large numbers of low-level events (TCP connections, audit records, and so forth) into a manageable number of alert reports. This is achieved by the EMERALD alert thread mechanism. At the next level, it is desirable for the sensors to be aware of each other's state, and adjust their own state accordingly. Finally, we would like to fuse alerts from heterogeneous sensors, as well as auxiliary information, to provide an operator with intelligently derived meta alerts. The following figure illustrates this hierarchy; the event counts reflect actual counts from a recent 45-day run of eBayes on our own TCP gateway.



Alert Threads

EMERALD sensors all have the concept of an alert thread, which we believe is sufficiently general that it should be part of a standard alert-reporting template (see below). Sensors may update reports on an existing attack based on transitions to more serious states, significant increases in the attack confidence level, or changes in what the

sensor believes the attack to be, among other factors. For extended attacks, these updated reports may flood the analyst's console. By assigning all these reports the same thread ID, a correlation or reporting engine that comprehends threads (such as EMERALD eFunnel) presents the analyst a single console report, but with the capability to easily drill down if desired.

Maintenance of alert threads is sensor specific. In eBayes-TCP, we incorporate a session model that deinterleaves traffic according to source IP, and partitions temporal bursts into distinct sessions. The definition of thread is then straightforward: all alerts for a given session share the same thread identifier. Alert reports are potentially generated periodically (by clock time as well as event count), in response to significant increases in the alert confidence, and in response to a change in the most likely attack hypothesis. The effectiveness of our approach is practically demonstrated in the EMERALD demo, where a Neptune attack spanning thousands of events is reported as a single alert on the EMERALD GUI.

Other sensors do not support the thread concept. Among other services, EMERALD's eXpert-IDIP provides this threading feature for non threaded alerts.

Comprehending Multiple Sensors

The system we informally refer to as "eBayes" is actually two sensors: eBayes-TCP, which monitors TCP sessions, and eBayes-Host, which adapts to and monitors valid hosts and services within a protected network. These are coupled in that the former queries the state of the latter, and adjusts accordingly. This adjustment is mathematically straightforward in Bayes formalisms, which start with a prior model of their world and update this model based on the state of numerous features (directly observed or derived) linked to a set of unobservable hypotheses by conditional probability relationships.

The Bayes-Availability monitor replies to eBayes-TCP whether a service requested by a particular session has ever seen a valid connection. Based on this, eBayes-TCP adjusts its internal belief that the port(s) requested by a session are valid and not part of a stealth

sweep. With this coupling, eBayes-TCP effectively detects the stealthy portsweeps in the Lincoln Laboratory data, as well as the mscan probe in the EMERALD demo. While the small number of ports in these stealth attacks is indicative of a specific probe for which a signature can be defined, eBayes has taken an approach that we feel is more general. The result is that we can detect a variety of other port sweeps (such as nmap and strobe) with no modification of our existing models. Thus, knowledge of the state of one sensor achieves a practical and significant improvement in the sensitivity of another.

The availability monitor also informs eBayes-TCP when a valid service is in a degraded state (again, this is a probabilistic call). Based on this, eBayes-TCP alters its prior expectation of the world, adjusting the prior expectation of anomalous values for particular features. In the EMERALD demo, we have a large number of simulated normal clients accessing a network that is under attack. When the attacker achieves a successful DOS, these clients suddenly appear anomalous with respect to eBayes-TCP, were it not for the fact that eBayes-TCP comprehends the state of eBayes-Host. Essentially, the internal models dynamically come to tolerate values for some features that would normally be indicative of certain attacks. There remains the potential to generate alerts from these sessions, but the evidence must now come from other features. Without his capability, a successful DOS attack might cause hundreds of alerts for the otherwise normal traffic (what we termed "collateral damage" in eStat). The large number of alerts would overwhelm the analyst, and potentially bury the one valid alert due to the attacker. The approach taken generates one alert for the attack session and one alert that the service is in a degraded state.

We therefore see that our approach, where one sensor comprehends the state of another and adjusts accordingly, both raises sensitivity (evidenced by the ability to detect stealth port scans) and suppresses false (or spurious) alarms. The use of a Bayes formalism makes this coupling mathematically convenient, but is not a requirement. We believe that at present we are unique in achieving these two important functions of sensor correlation.

Meta Alerts and the EMERALD Alert Template

We are in the process of defining an enhanced alert template that can be thought of as enriching standards such as IDIP with respect to content while avoiding issues of language specifics. The template has been forwarded to UC Santa Barbara, Honeywell, ISI, and the IETF for comment. Without going into full details, this template extends the state of the art in several important areas. First, we include the concept of alert thread, which is unique to EMERALD at this point but we believe is essential for the reasons presented above. We include an "anomaly" field in addition to the "confidence" field used in IDIP. We envision these fields being used to condition the reports of some sensors. We also include arrays to describe in greater detail the target(s) of an attack (for example, the specific ports scanned). Finally, we include fields describing the sensor type and placement.

Although this template has not been used outside EMERALD, we must point out that within EMERALD we have a diversity of sensors and use both signature and probabilistic techniques. Our early experiments indicate that diverse sensors will be able to fill this template with content that will be more useful to correlation engines than is currently available.

Our plan for the near future is to use this template and the output of multiple EMERALD sensors to build useful "meta alerts". This involves intelligent fusion of the content with respect to shared features. We are defining similarity matches for patterns of potentially different lengths, comprehending the number of features that match, the quality of the match, the number that could have matched, and the expectation that the underlying feature would match. Other issues to be considered include temporal relationships (concurrency or precedence).

Our approach to alert fusion is as follows. We maintain a list of "meta alerts" that are possibly composed of several alerts, potentially from heterogeneous sensors. For two alerts (typically a new alert and a meta alert), we begin by identifying features they have in common. Such features include the source of the attack, the target (hosts and ports),

the type of the attack, time information, and so forth. With each feature, we have a similarity function that returns a number between 0 and 1, with 1 corresponding to a perfect match. Similarity considers such issues as:

- How well do two lists overlap (for example, list of targeted ports)
- Is one observed value contained in the other (for example, is the target port of a 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?

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.

An important innovation we introduce is expectation of similarity. This is also between 0 and 1, and expresses our prior expectations that the feature should match if the two alerts are related, considering the specifics of each. For example, two probes from the same target might scan the same set of ports on different parts of our subnet (so expectation of matching target IP address is low). Also, some attacks such as SYN FLOOD spoof the source address, so we would allow a match with an earlier probe of the same target even if the source does not match (expectation of match for source IP is low).

We then compose overall alert similarity from feature similarity values normalized by expectation of similarity. The approach is valid regardless of the number of features that overlap, and is thus well suited for use with multiple heterogeneous sensors. The new alert is fused with the meta alert that is most similar, if the similarity is good enough. If no existing meta alerts are a sufficiently good match, the new alert begins a new meta alert thread. Fusion consists of combining features so that the new meta alert is a superset of the previous meta alert and the new observation. Lists are merged, comprehending “hit counts” to each list element. The merge functionality includes “trim functions” to prevent arbitrary list growth.

An important additional feature is that the meta alert stores the EMERALD thread identifiers of all the component alerts, so that the operator is always able to examine in detail the alerts that contribute to the meta alert report. There is potential for consolidation in a useful sense, but with no loss of information since the component alerts are always available for examination.

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

Demonstration of Technique

In a recent run of eBayes, we monitored traffic at the CSL TCP gateway in real time for five days. We saw several alerts similar to the following:

```
similarity to thread 4860  1
portsweep   1.000 2000-08-10 06:50:28 from         193.230.37.2 ports
1268 to 32434 dt=   0.321
  count    164 max age count   0.16 code    3 svc    1 max-err  3.83 -
opn    0 -oip    0 -oport    0
  30 dest IPs: 130.107.1.1 130.107.3.1 130.107.4.1 130.107.5.1
130.107.6.1 130.107.7.1 130.107.8.1 130.107.9.1 130.107.10.1 130\
.107.11.1 130.107.12.1 130.107.13.1 130.107.14.1 130.107.15.1
130.107.16.1 130.107.17.1 130.107.18.1 130.107.19.1 130.107.20.1 \
130.107.21.1 130.107.22.1 130.107.23.1 130.107.24.1 130.107.25.1
130.107.26.1 130.107.27.1 130.107.28.1 130.107.29.1 130.107.30.1
130.107.31.1
  6 dest ports: 635{30} 110{29} 143{29} 53{27} 21{27} 109{22}
  BEL         0.000    0.000    0.000    0.000    0.000    0.000
0.000    0.000    0.000    0.994    0.004    0.002    0.000
  PCODE       0.000    0.000    0.000    1.000    0.000    0.000
0.000
  SVC DIST    0.417    0.000    0.194    0.000    0.389    0.000

Non-sys alloc ports 6  port_anom 0.990528 code_anom 0.987018
Invalid hosts 30  Invalid ports 6  Neval 24 eval_count 0

LL-LIST 2000-08-10 06:50:28  130.107.1.1 to 130.107.31.1  portsweep
1.000
```

Similar attacks with different target IP addresses appeared over several days. The meta alert fusion utility produced the following meta alert:

```
Meta_alert thread 22
Source IPs 193.230.37.2

Target IPs 130.107.1.1 130.107.3.1 130.107.4.1 130.107.5.1 130.107.6.1
130.107.7.1 130.107.8.1 130.107.9.1 130.107.10.1 130.107.11.1
```

```
130.107.12.1 130.107.13.1 130.107.14.1 130.107.15.1 130.107.16.1
130.107.17.1 130.107.18.1 130.107.19.1 130.107.20.1 130.\
107.21.1 130.107.22.1 130.107.23.1 130.107.24.1 130.107.25.1
130.107.26.1 130.107.27.1 130.107.28.1 130.107.29.1 130.107.30.1
130.107.31.1 130.107.1.2 130.107.3.2 130.107.4.2 130.107.5.2
130.107.6.2 130.107.7.2 130.107.8.2 130.107.9.2 130.107.10.2 130.10\
7.11.2 130.107.12.2 130.107.13.2 130.107.14.2 130.107.15.2 130.107.16.2
130.107.17.2 130.107.18.2 130.107.19.2 130.107.20.2 130.107.21.2

From  2000-08-10 06:50:28 to  2000-08-14 01:47:18

Ports
Sum Hits 708.999  dot product 0.166775
Index 635   Prob 0.170455
Index 110   Prob 0.168534
Index 143   Prob 0.167341
Index 53   Prob 0.165906
Index 21   Prob 0.169958
Index 109   Prob 0.157806

Threads :4860 5564 6314 6980 7650 8223 8767 9287 9778 10350 10964 11577
12188 12634 13033 13427 13802 14166 14525 14887 15298 15677 16015 16357
16689 17040 17321 17584 17907 18238 18558 18861 19185 19607 19980
```

This sequence of attacks was confirmed as a virtually certain attack by our system administration staff. We consider the above a demonstration of probabilistic alert fusion technology on a real-world data stream, and significantly ahead of the state of the art as exemplified in the July 2000 DARPA meetings.

Benefits of the approach

Alert aggregation

The most obvious and immediate benefit is a reduction in the number of alert reports an operator must address. Considering eBayes alerts only, we believe a reduction in the number of alerts by one-half to two-thirds is possible. eBayes already achieves considerable alert reduction through the threading mechanism, but we expect further aggregation when heterogeneous sensors are considered.

Scenario reconstruction

The correlator can concatenate the attack steps and present the analyst with a scenario. For example, an attack may start with probes for common vulnerabilities, proceed to data

theft (stealing account and password information, for example), and then carry out a root compromise.

Analytical capability

The similarity threshold provides the analyst with a single quantity that he or she can control to observe how alert groupings break down. At a similarity threshold of 0.0, all alerts match; conversely, at 1.0 only alerts that are identical with respect to overlapping features are aggregated. By ranging the threshold between these values, it is possible to construct an alert tree, where meta alerts branch as the matching requirements become more stringent.

Adaptation

Yet another mode in which this technique is useful is in the adaptation of the "expectation of similarity" values to reproduce an expert's aggregation of a group of alerts. This can be accomplished via a supervised learning algorithm that adjusts these values up or down according to the features on which an analyst's alert groups match or do not. An iterative approach through a batch file of alerts is used to adjust these expectation values. This has much in common with, for example, back propagation in neural networks.

Acknowledgments

This research was sponsored by DARPA under contract number F30602-99-C-1049. The views herein are those of the author(s) and do not necessarily reflect the views of the supporting agency.

References

1. Porras, P. and Neumann, P. "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Distrurbances", National Information Security Conference, 1997. http://www.sdl.sri.com/emerald/emerald-niss97.html

2. P.A. Porras and A. Valdes. Live traffic analysis of TCP/IP gateways. In Proceedings of the Symposium on Network and Distributed System Security. Internet Society, March 1998.

3. Valdes, A. and Skinner, K. "Adaptive, Model-based Monitoring for Cyber Attack Detection", to appear in the proceedings of "Recent Advances in Intrusion Detection (RAID 2000)", Springer-Verlag