Before Applying New Technologies

TR-2001-05

Todd Heberlein
Net Squared, Inc.
todd@NetSQ.com

Abstract

During a recent DARPA teleconference, many of the problems PACOM intrusion detection analysts face were described. Two related and critical problems are operator overload and the large number of false alarms that their sensors generate. I have heard of similar problems with analysts working with ASIM deployments, and these problems are cited by many DARPA program managers to motivate the development of new technologies. However, I believe a significant portion of the problem can be addressed with the little or no new technology. This paper makes several recommendations that should be considered, and I hope it serves as the beginning of a dialogue on these issues.

1 Introduction

During the DARPA TDM teleconference held 2 April 2001, Danny Vukelich presented a summary of what he had learned during his first week of "PI at the Front" in Hawaii. Mr. Vukelich's interactions focused primarily on the intrusion detection operations, their people, and their technology. In particular, he worked with Mike Crabtree, one of the critical "gurus" of the group, he worked with the JIDS (a.k.a. NID) network-based sensors, and he attended several meetings that were largely oriented towards management issues.

While reviewing Mr. Vukelich's slides and listening to his presentation, I kept wanting to ask, "Why are the operations folks doing things this way?" In fact, for several years now, during interactions with Rome Labs while they deployed prototypes along side ASIM, I have been asking this same question or the related question "Why are they not doing things this other way?" The "thing" is usually the design and operations of their overall sensor architecture. Answers typically range from "I don't know" to "politics" to "the inability to secure the sensors."

The question is not just an academic one. The first bullet in Mr. Vukelich's slide titled "Issues" declared "Analysts contending with large volumes of traffic" as a significant problem. At DARPA's recent Cyber Panel PI meeting, Catherine McCollum, the program manager, presented a slide titled "Cyber Panel – Problem." The second bullet declared "Operators are flooded." This theme is sounded at nearly every DARPA meeting I have attended, and it is one of the primary arguments for the development of new technologies.

While I am a DARPA contractor who appreciates the funding to solve these and other problems, I believe we are frequently putting too much emphasis on the technology and not enough on the overall process of cyber defense. Certainly DARPA is a technology provider and not a general purpose solutions provider, so re-architecting network configurations, processes,

procedures, and policies largely lie outside the scope of DARPA's mission. However, I believe DARPA must consider these issues for at least two reasons.

First, DARPA's funding to at least some degree depends on satisfied customers. During much of DARPA's history of funding intrusion detection related work, DARPA has been "customer free." That is, we had no identified end-user to satisfy. We wrote papers, briefed program managers, and performed internal experiments, but we never had a mandate to deploy or deliver to operational environments. However, over the past year DARPA's information assurance program (recently renamed Operation Experimentation (OPX)), has chosen the DOD's Pacific Command (PACOM) as its exemplar customer, so solving real-world problems has now become very important. Failing to deliver noticeable improvements for PACOM may have detrimental results on future DARPA funding levels.

Second, as creators of new technologies, we would like to see our technology deployed in an environment that shows it in the best possible light. Deploying our technology in an unprepared environment may exacerbate existing problems (e.g., contributing to information overload) and can be very embarrassing. Thus, in our own self-interest (or perhaps just to protect our professional pride), we should at least examine the overall environment and how to best prepare it before introducing DARPA funded technology.

Therefore, I begin with a simplified network architecture (see Figure 1). This design is inspired by descriptions supplied by Mr. Vukelich in Hawaii and others working with ASIM deployments around the globe. It consists of a network intrusion detection sensor (i.e., JIDS, NID, or ASIM) and a firewall. The sensor is placed in front of the firewall. That is, the sensor is positioned between the firewall and the rest of the Internet, so it sees all traffic that attempts to enter a site, even traffic blocked by the firewall. The sensor generates voluminous numbers of reports, a substantial number of which are false alarms. These reports are then manually processed by analysts.

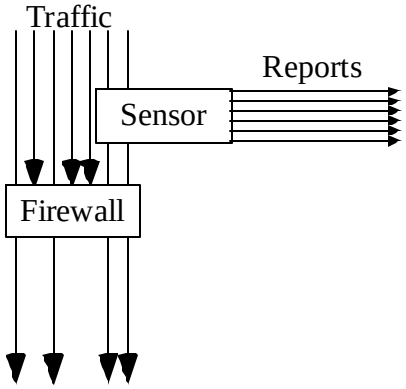

Figure 1: Today's Architecture

Over the next several sections I introduce no-tech or low-tech changes that I believe are important to (1) improving the operations at PACOM and other sites, and (2) preparing an environment that is best suited to highlight technologies being developed under DARPA. The critical changes are:

- repositioning sensors
- integrating vulnerability scanners
- establishing several feedback loops in the processing chain

2 Reposition the Sensors

The simplest and probably the most effective step to reducing the volumes of reports analysts must process is placing the sensor *behind* the firewall (see Figure 2). The firewall should be the first line of defense, not the second. An effectively configured and managed firewall (or set of firewalls) should screen out most attacks a site could potentially face. Because the sensor will never see these attacks, this should substantially reduce the number of reports an analyst must examine.

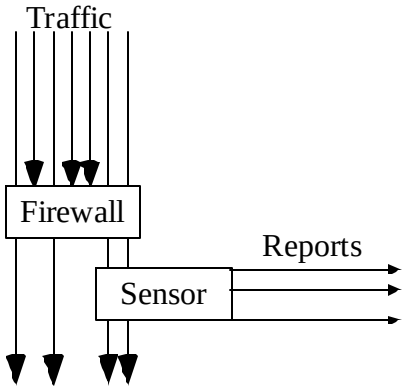

Figure 2: Moving the Sensor Behind the Firewall

The two most common answers to my question of why a site has placed the sensor in front of the firewall are: (1) politics and (2) security concerns about opening a hole in the firewall to access the sensor. In the first case there appears to be an ownership division line centered at the firewall. Security providers such as ASIM analysts can put equipment on the outside of the firewall, but individual sites are very protective about letting anything inside the firewall into their domain. Part of this protection is attributed to the second problem. The sites are hesitant to "punch a hole" into their firewall in order to allow external organization to reach in and manipulate extremely sensitive data. Technically, we (supposed security experts) should be able to reasonably address the technical issues behind the secure operation of a limited security device.

Another answer that I have received in the past is that the security operators want to know about all attacks, even those that are blocked by a firewall. For this I have two responses. Analysts should probably not complain about data overload when they also want reports of attacks that are routinely blocked. Second, most firewalls can generate voluminous logs of blocked traffic and attacks. In other words, the data is available. It probably should not be process by humans, but the data could potentially be used for tracking trends.

Firewalls should be a first line of defense. Most attacks should be stopped by a firewall. Security concerns can be addressed within reasonable levels of risk. Most attacks stopped by the firewall can still be recorded by the firewall itself. Therefore, placing a sensor behind the firewall makes sense from a technological perspective, and it can dramatically reduce the number of reports generated by an intrusion detection sensor.

3 Integrate Vulnerability Information

A second important step is to tightly integrate the results from an intrusion detection sensor and a vulnerability scanner. Assuming an analyst can only examine a limited number of reports generated by the sensor, an organization should use site vulnerability information provided by a scanner to identify and filter out (or at least downgrade) reports of attacks that will

certainly fail. The goal is to identify a subset of attack reports that should be of the greatest concern, namely reports of attacks that may have succeeded (see Figure 3[1]).

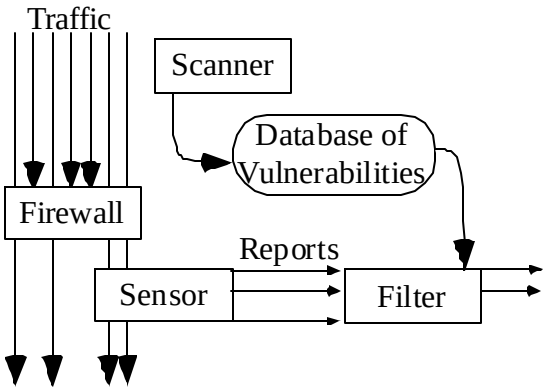

Figure 3: Integrating Vulnerability Information

Technically the greatest challenge is to clearly relate vulnerabilities identified by the scanner to reports generated by the sensor. Ideally both systems would use a common identification system. For example, when a sensor detects a particular DNS worm, ideally the report would also include an ID (e.g., a CVE identifier) identifying the vulnerability that the attack exploits. Many different attacks might exploit the same vulnerability, one may propagate a worm while another may fork a shell, so it is important to distinguish between vulnerabilities and attack signatures. With this capability, a filter could look into the database to determine if the system targeted in the attack was tested for that vulnerability, and if it was tested, whether the system was vulnerable. If the system was tested for the vulnerability and was secure, then the report does not have to be submitted directly to the analyst. On the other hand, if the vulnerability was tested and the system was vulnerable, then the attack report should be elevated to a higher level. Attacks for which success or failure cannot be determined should also be forwarded to the analyst.

Commercial vendors such as ISS and Cisco sell both intrusion detection sensors and vulnerability scanners. The DOD should carefully examine how well these products are integrated. If the individual products perform well by themselves and also integrate well, the DOD should consider using these product combinations or at least modeling their own technology developments on them.

4 Close the Loops

Engineers often describe systems as "open-loop" or "closed-loop." In an open-loop system control instructions are issued and carried out regardless of changing conditions. An example of an open-loop system is an automatic lawn sprinkler – it waters the lawn at the prescribed time regardless of the weather conditions. In a closed-loop system a sensor measures the difference between a desired result and the actual result, and the results from that sensor are used to modify control instructions. An example of a closed system is a thermostat-controlled air conditioner.

As described to me by users of JIDS and ASIM, many aspects of the intrusion detection sensor and operations structure, from system creators to users to operations, appear to operate as an open-loop system. Closing some of the loops, that is, creating appropriate feedback systems,

[1] In Figure 3, the "Database of Vulnerabilities" is a list of vulnerabilities that exist within a particular site. It is not a global list of vulnerabilities.

can potentially remove much of the data loads analysts must contend with. In sections 4.1 and 4.2 I discuss feedback approaches that can decrease the number of false positive reports. In section 4.3 I discuss a feedback approach to reduce the number of attacks that can effect a site and therefore the number of reports analysts must process.

4.1 Loops Between Signature Creators and Operations

ASIM and JIDS (as well as virtually all network-based sensors) are primarily signature-based systems. A criticism I hear constantly from operators of ASIM and JIDS particularly, but also from some commercial sensors, is that they produce enormous amounts of false alarms. That is, the sensor flags activity that is ultimately determined to be benign. Feedback loops between events flagged during operations and signature creators (see Figure 4), as well as signature creators and signature *engine* creators (not shown), should reduce many of the false positives. This in turn should reduce analyst overload.

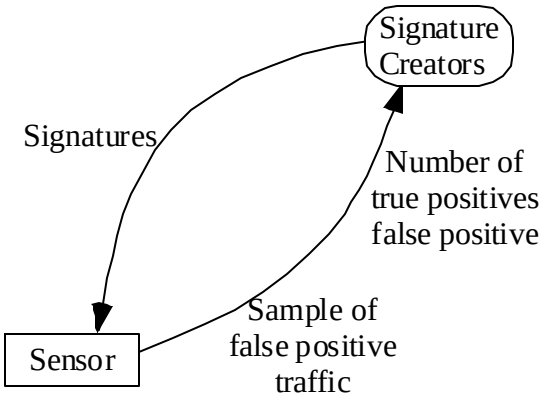

Figure 4: Global Improvement of Signatures

Analysts working with fielded systems should collect statistics on which signatures were associated with actual attacks (true positives) and which signatures were associated with benign activities (false positives) and return this data back to those responsible for creating signatures. Whenever possible, samples of data that triggered false positives should also be submitted. Signatures creators should then refine the signatures to produce a better true positive to false positive ratio. In some cases, the signature might simply need to be retired. In other cases, the existing signature engine may not be powerful enough to substantially refine the signature, so work may need to be devoted to developing a more expressive signatures engine. Below are two examples that illustrate these issues.

4.1.1 Retire the "debug" Signature

About two years ago I was working with a group fielding a major commercial sensor, and one of the signatures it supported was the "debug" signature. ASIM or JIDS sensors may also support a similar signature. The signature was designed to detect an attack against sendmail servers. In the attack, an attacker would enter the command "debug" in a connection to a sendmail server which would create an interactive shell. This "feature" was added by the sendmail developers to support testing and debugging of the sendmail software, and it was left in when the product shipped. This vulnerability was exploited by one of the biggest attacks of all time.

Unfortunately, the attack, the original Internet worm, occurred on November 2, 1988, over a dozen years ago, and by the beginning of 1989 most sites had patched this vulnerability. As most systems were patched, fewer and fewer attackers tried to exploit this vulnerability. While the signature may have been effective in the latter part of 1988, over a decade later the signature only generates false positives. Therefore, the signature should be retired.

This is a general model for signatures focused on specific attacks or vulnerability exploits. When a signature is created for a recently discovered attack or general exploit of a vulnerability, that signature may have a good true positive to false positive ratio as attackers try to take advantage of the many open doors. As the vulnerability is patched and time passes, attacks against the vulnerability will fail more frequently, and as a result, the attack will be used less often. The true positive rate will drop off, eventually reaching zero, while the false positive rate will remain near its original level. Using such a signature at this point will only generate false positives (or "noise"), and it should not be distributed as part of a standard signature suite.

A feedback system between fielded sensors and the signature creators should help signature creators decide when to relegate a signature to the "retired" file.

4.1.2 History of the "daemon" Signature

The second network-based signature ever created, the string "daemon", was added to the Network Security Monitor (NSM) around 1990. The purpose of the string was to detect someone opening, copying, transferring, or otherwise manipulating a password file, because at that time weak passwords were the primary method for penetrating systems. The string "daemon" was in almost every UNIX-based password file. The pattern was not designed to detect a specific attack but to detect an indicator of possible misuse (accessing a password file). The first day the string was used we detected an attack using the signature.

Unfortunately, the string "daemon" occurred in many sources not tied to the password file. For example, when a mail message bounced, the error message accompanying the bounced message often reportedly came from the mail "daemon," so the string generated many false positives.

Fortunately being both the operator of the sensor and the creator of the signatures, I was able to easily compare three pieces of information: (1) the signature pattern, (2) the activity I was trying to detect (the content of a password file), and (3) samples of data that were creating false positives. From this I made a simple extension to the pattern, appending a colon, ':', to the end of the signature. "daemon**:**" had a much lower false positive reporting level than "daemon," but it still easily detected someone opening or transferring a password file.

However, when I started monitoring network traffic again in the late 1990s, I found "daemon:" was generating more false positives than I as an operator wanted to deal with. To address this I took the same steps as before: examining the pattern, the activity I wanted to detect, and the samples of false positives. This time I decided the regular expression pattern

```
daemon:.*:.*:.*:
```

would virtually eliminate the false positives while still detecting access to password files.

The problem was that the pattern matching engine I was using, Knuth-Morris-Pratt (KMP), which was also used in NSM, ASIM, and JIDS, was not powerful enough to detect such regular expressing patterns[2]. To address this issue, I integrated the GNU regular expression

[2] The Boyer-Moore algorithm used by Snort is also unable to detect such patterns.

library, rx, into Network Radar, so I was able to effectively eliminate any false positives when trying to detect access to password files.

In this case, the signature *engine* developer, the signature creator, and the operator were one person, so the loop was easy to close. In the first instance, a simple modification to the signature improved results (a loop between operator and signature creator). In the second instance, the signature creator determined that the signature engine needed to be improved.

This informal closed-loop process managed by a single person needs to be extended to the entire sensor creation and operation process within the DOD. Careful examination to determine why signatures are generating so many false positives, followed by refinement of the signature system, should lower false positive rates and the workload they create.

4.2 Loops Within Operations

In the previous example we highlighted the need for and potential positive results created by closing a loop between analysts in the field, the signature creators, and the developers of signature engines. Unfortunately this might not always be a viable solution. The sensor developers may be unresponsive and may choose not to close the loop as described in the previous section. The site where the sensor is deployed might not want to provide samples of data that created false positives because of privacy or security concerns. A site might also have unique characteristics that result in high false positive rates for some signatures, and the sensor creators might not want to modify a system to satisfy a single unique user. In these cases, a local organization may want to create their own feedback loops (see Figure 5).

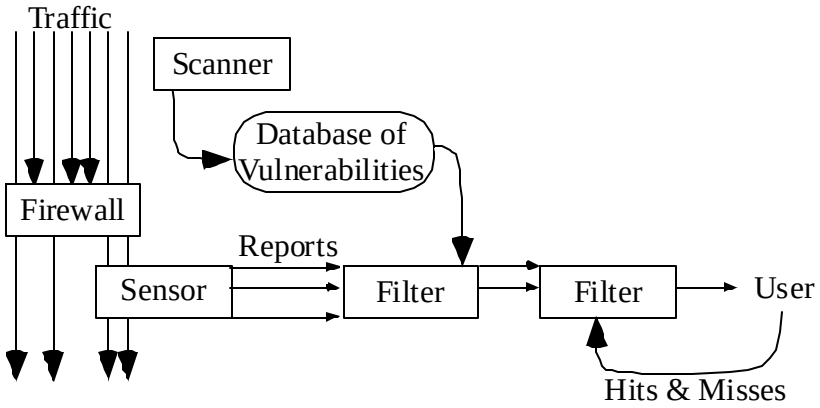

Figure 5: Local Improvement in the Use of Signatures

In this model, the analysts inserts an additional filter between the sensor and himself. As the user processes each report, he registers whether it was a true positive or a false positive. A simple feedback filter can quickly determine which signatures tend to create many false positives and can automatically downgrade those reports. These downgraded reports could be reviewed later during a lull in activity, or depending on the analysts burden, dropped altogether.

4.3 Loops With Sensors, Scanners, and Firewall

In the previous two sections we discussed approaches to reduce the number of false positives. In this section we discuss reducing the number of true positives by reducing the number of attacks that can succeed against a site.

Two additional loops can be closed to reduce the number of attacks the analyst must process. The goal is to use information collected from vulnerability scanners and attacks observed by a sensor to fine-tune a firewall. A more effectively configured firewall will reduce the number of attacks that get through. This in turn will reduce the number of attacks detected by the sensor and the number of reports that an analyst will need to process (see Figure 6).

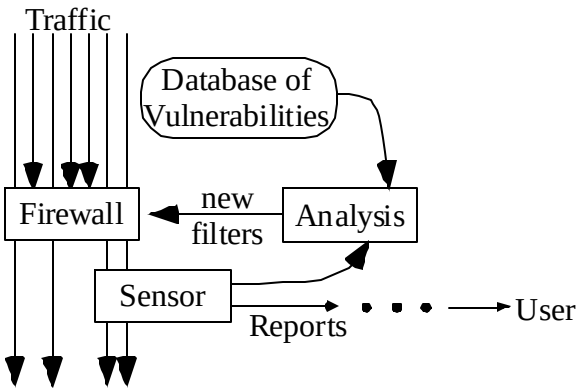

Figure 6: Integration of Sensors, Scanners, and Firewalls

The loops can be closed in a number of ways. The steps described here should be determined by a site's security policy and the existing threat posture. In the simplest case a scanner may detect that a particular server is vulnerable to a known attack, so analysts would configure the firewall to block all external access to that server until it is patched. Operators and users of the server may find such an action onerous, so such preemptive blocking by firewalls before any detected attacks should be carefully coordinated with a site's policy.

A refinement to this process integrates reports from a sensor. For example, if a sensor detects a number of attacks into a site that exploit a specific vulnerability, analysts may want to consult the database of known vulnerabilities at their site, and then block external access to systems that are vulnerable to that attack. In some instances access to servers will be blocked before they are actually attacked, but evidence of active attacks at the site may motivate analysts to take this step.

Another refinement may include integrating sensor information from other sites. For example, while a particular Air Force site may not have experienced a particular attack yet, if other Air Force sites have seen the attack analysts may still want to take preemptive steps to install firewall filters to block attacks against systems known to be vulnerable.

Please note that the approach described above is not the automatic configuration of firewalls by a sensor. This approach has been known to cause inadvertent denial of service attacks. Instead, this is a measured approach performed by analysts using knowledge about their own site's vulnerabilities to specific attacks and potentially information about active attacks as detected by their or others' sensors.

The goal in these steps is to establish feedback loops between vulnerability scanners, intrusion detection sensors, and firewalls. The result of this loop is a finely tuned firewall that prevents future attacks from reaching known vulnerable system. This in turn reduces the number of reports analysts must manually process. And perhaps more importantly, these steps reduce the incredibly time consuming work that must be performed to clean up a successfully penetrated system.

5 Conclusions

DARPA has identified "operator overload," including large numbers of false positive reports that must be processed, as a primary motivation behind the DARPA's I&A research and development agenda. Danny Vukelich experienced these problems first hand during his recent "PI at the Front" period in Hawaii. However, when I have asked about the overall architecture of security equipment as well as operations, the system is described in a way that does not make sense to me. A common design has the network sensor between a site's firewall and the rest of the Internet. This is the design described by Mr. Vukelich in Hawaii as well as Rome Labs personnel describing various ASIM deployments. When I have asked why the signature-based systems are generating so many false positives (i.e., why is so much benign activity is identified as attacks), I have never received an answer.

Based on both problems and current operations as described to me, I have created a set of recommended changes to architectures, processes, and policies that I believe can dramatically reduce the overload experienced by analysts. The ultimate goal is to reduce to the greatest extent possible the number of reports an analyst must process that are (1) false positive, (2) true positive reports of attacks that will not succeed, and (3) attacks that in hindsight should have been easily prevented. These changes do not introduce any new technologies and are probably outside the scope of DARPA's primary mission. However, now that DARPA has decided to solve real-world customers' problems (starting with PACOM), it should begin investigating these issues. Beyond the altruistic reasons of improving a customer's problems even if it does not directly involve DARPA technology, a well prepared environment will most likely show DARPA technology in its best light.