

Analyzing IDS Data

by *Chris Jordan*

last updated Tuesday, May 30, 2000

Relevant Links

[CIDF
Common
Intrusion
Detection
Framework](#)

[IDEF
Intrusion
Detection
Exchange
Format](#)

Introduction

Intrusion Detection services are experiencing a hot bed of activity on the venture capital list. It seems that everyone involved with networking or security of some type is trying to roll out and sell these services. The problem for these companies is that the maturity of Intrusion Detection Systems (IDS) alone is insufficient to create a level of quality. The number of persons that have experience with large scale IDS is small in comparison to the number that are going to be hired by these firms. So, how are large-scale IDS networks successful with low experience personnel and an immature technology? The answer is that the solution architectures are surrounded by processes.

This paper was written after a discussion with a fellow security expert from a military research lab. As this research organization grew, it needed to address the growing pains tied to large scale versus small-scale intrusion detection. We talked about how much work there is surrounding the operations of IDS services and how management was not aware of the amount of effort involved. Management had been influenced by marketing that demonstrated that intrusion detection was as simple as installing a real-time detector to the network and respond *before* any real damage occurred. This paper is the beginning of outlining the processes that surround the IDS framework. It is not a paper to discuss weaknesses in IDS technology, but the weaknesses of IDS implementation.

Defining the Analysis Process

This paper addresses the need for further analysis of IDS data. To do so, one must understand the process of handling the data the IDS network produces. In the military, the process is defined simply as protect, detect and respond. Here, we are examining the detection process.

IDS Network

The smallest element of intrusion detection data is referred to as an event. An event is an auditable occurrence on the network. Events can be generated from IDSs, firewalls, telephone calls, e-mails, and faxes.

It is important to note that IDS is a technology of risk mitigation and not avoidance. Risk avoidance is the addition of security element to eliminate a threat or weakness from the system and therefore avoiding a risk. Risk mitigation, not as popular in the United States, is the addition of a security element to reduce the damage or chance of damage from a threat or weakness. In general, firewalls perform the function of risk mitigation while IDS perform the function of risk mitigation to the network.

As IDS develop automated response to network traffic they begin to perform the task of firewalls. The issue is often confused in evaluation of products. IDS differ in intrusion analysis in that a firewall produces "security events" while IDS produces both events and alarms. This is because the firewall has already avoided the risk and so the reaction has been made. "Alarm" terminology is used when a security event has occurred that needs an action.

Although in small groups the amount of event data is also small, larger organizations must deal with a substantial collection of information that may be overwhelming. For example, OC-12 connections can generate about 850 megabytes of event data in an hour. A significant issue in developing a large-scale IDS infrastructure is how to collect and organize the incoming data.

Some commercial IDS networks have collection stations. But the stations are limited to their product suite. There is the Common Intrusion Detection Framework/Language (CIDF/CIDL) for formatting and standardizing intrusion data, however there is no commercial implementation of this standard. If CIDL was implemented, interoperability in data collection could be possible. However, from operational standpoint it is essential not to belabor the direction of IDS design but instead to determine how to solve the problem here and now.

In approaching data analysis, you will need to consider how the data is analyzed. The most common technique is referred to as "real-time detection" or "first level analysis". In higher security areas a technique called "second level analysis" is performed.

1st Level Analysis

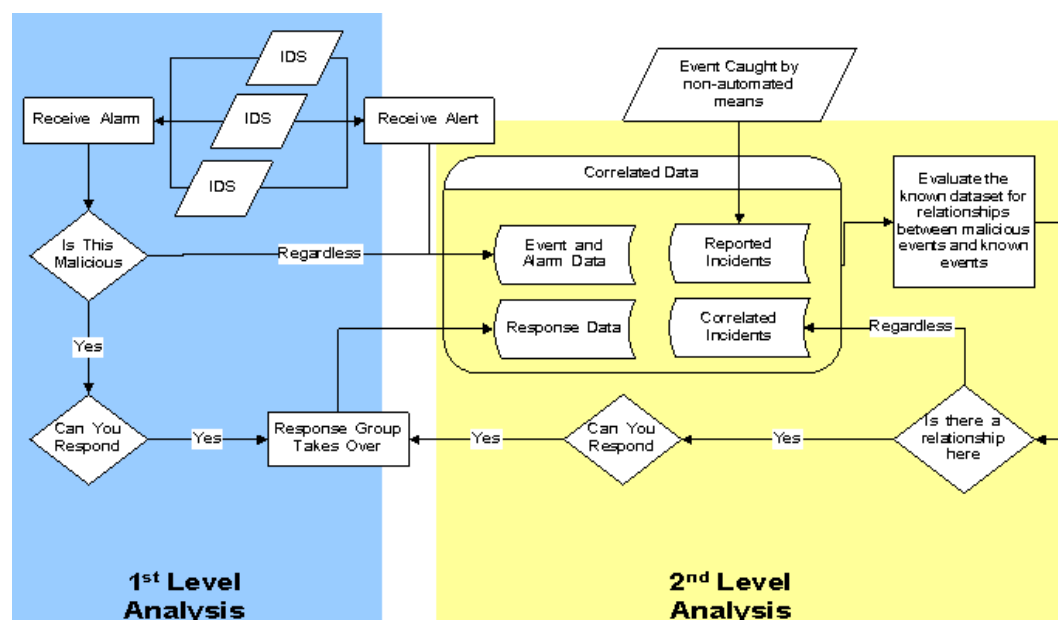


Figure 1.

Those organizations that solve the collection element of intrusion detection will find that the work has just begun. Once information is collected there needs to be a process that implements the data. To do so, an organization needs to analyze the data to *detect* a malicious event.

The most common form of IDS analysis is referred to in some circles as "1st Level Analysis" (shown in light blue on Figure 1). This is simply receiving an alarm and determining if it is malicious. Because 1st level analysis is not context sensitive then the event, analysis and response do not need to consider other events and therefore can be handled in real-time.

Real time detection systems fake context sensitive analysis by using thresholds. Thresholds alarms are a counting mechanism. If so many connections occur in a given time it is called a SYN flood. If so many different ports are visited in a given time

it is called a port scan. Details of each event are not recorded. Instead, the occurrence is temporarily logged until the threshold is met or the time window is over. This technique is vulnerable to definition of time and the leaves little information to determine reaction to the threat. Threshold analysis is how people often use firewalls to perform IDS functions.

Outside of threshold alarms, 1st level deals with the intrusion alarm as a single *uncoupled* event. What that means to the processes is that the analyst will look at the quality of the alarm trigger to determine if it passes two tests.

1. Is the alarm a malicious event?
2. Is there a response to this alarm?

Only if both answers are "yes" does the security organization respond. Responding to the real-time alarms is how most IDS monitoring organizations work. The process is simple and apparently responds directly to the intrusion problem. However, this process assumes that the IDS real-time alarms catch all the intrusions that can possibly be detected.

2nd Level Analysis





2nd Level Analysis is the correlation of events (alarm events and logged events) that show malicious intent. Correlation occurs when events demonstrate a greater pattern. This is the difference between solving a single robbery versus catching a thief committing multiple robberies. Both if successful catch a robber, but each chases the robber differently.

2nd Level Analysis is aimed at correlating events across the network, time and sources allowing the ability to detect attack whose signature is expressed with the irregular grammar and context sensitive lexicons. It is difficult to perform malicious trend analysis over an uncharacteristic datasets. Consider attempting to read a book where there is no punctuation or spaces. Reading is slow and meaning sometimes takes an understanding of previous or later information to divide the words and sentences.

 SSH CONNECTION FROM Y

Therefore, analysts will first take one event that is known as threatening and use that to determine if there are any other events related to it. This can be done with a honey pot or thorough analysis of events. For example, a secure shell (ssh) might trigger an alarm because it comes from an unknown source.

The analyst would then backtrack the source to assess if other events were detected. In this case there are connections from an unknown source that occurred previously, but they do not contain a malicious string that the IDS recognizes:

 HEAD /%62%69%67%63%6e%6e%66%2e%63%67%69 HTTP/1.0 FROM Y
 HEAD /%63%67%69%2d%6c%6e%63%61%6c HTTP/1.0 FROM Y
 GET /%62%69%67%63%6e%6e%66%2e%63%67%69 HTTP/1.0 FROM Y
 HEAD /%71%75%69%6b%73%74%6e%72%65%2e%63%66%67 HTTP/1.0
 HEAD /%63%67%69%62%69%6e HTTP/1.0 FROM Y

Here a good analyst would realize that the strings are awkward for HTTP calls even if he does not understand the impact. The result is that the analyst would mark the pattern as a highly probable attack (probe: based on number of connections) from a new form of attack/scan. This pattern happens to be that of "Whisker" tool, and it is using an application layer protocol to hide the attack (the URL formation language).

Now the response team has something more to work with. The new information can be used to establish if these patterns (Whisker and Whisker followed by SSH) exist elsewhere in the dataset. More than likely, where this new pattern will appear it will highlight new attackers and attacks that were missed.

2nd level analysis is constantly looking for connections between malicious events and non-malicious events. This occurs anytime problems are found. Historical data becomes critical to this style of analysis because it constantly queries past data to determine if an intrusion was missed.

Respond

When developing an intrusion detection capability an organization must consider why. Regardless of what responses used by an organization, they need to understand the supporting processes to enhance the response capability. An intrusion capability needs to position the security team to respond to the detection of a weakness or a suspected violation.

1. Define what responses the organization is capable of using - Before the development of an intrusion detection capability, the organization needs to plan how to handle an external and internal events.
2. Determine for each response what data is needed - If an organization decides to handle internal attacks by revoking a user's key to the network, then security needs to have software that can determine what user is responsible for the malicious activity.
3. Determine where can that data be collected - System and network data provide different data and different levels of accuracy. It is possible to spoof network information down to the machine address. Therefore, for higher assurance levels, host based information is usually more accurate. Accuracy and cost considerations will be the primary discriminators in determining detection elements.

On a side note, it is not advised to use the IDS to actively collect data (i.e., ping an attacker or lookup DNS information). Active information gathering allows the attacker to determine if counter-measures (like IDS) are in place. Also, it can allow an intruder to determine (by lack of defensive response) what thresholds do not trigger the IDS.

If the security group does not respond to the data, then there is no need for security.
This means:

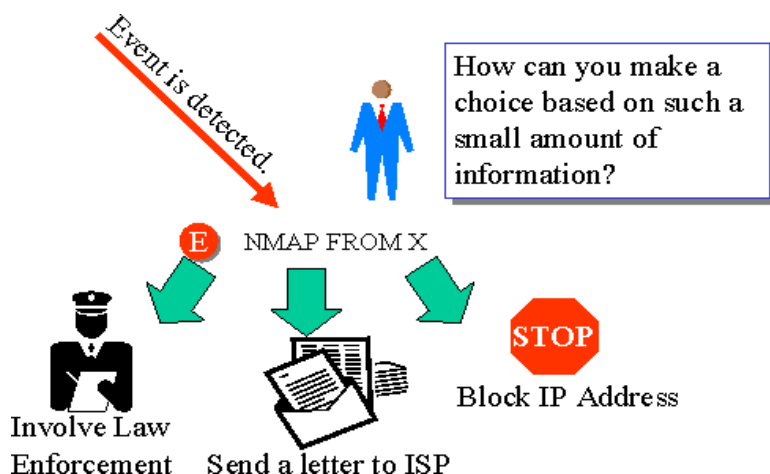
1. An alarm that has no response does not need to be reviewed.
2. An event (or a characteristic of the event) that is not responded to or used for analysis does not need to be recorded.

These two points are difficult for new organizations to handle. There is an attitude that all data should be collected and one day a data mining technique can use it to determine "unseen" problems. From an operational point of view, and not a research one, this is not worth the effort. There are other issues that personnel and management should deal with (like the improvement of the detect and response process) than the implantation of an infrastructure that has no known use yet.

In summary, information collected should support three concerns: management oversight, detection of malicious events, and response to an event.

Difficulties in 1st Level Analysis

The first difficulty of first-level analysis is a loss of context. A common situation is that an analyst sits behind a "real-time" alarm console. The console has some form of screen that queues the alarms as they come in. Some IDS management systems look like an SNMP trap screen; others let each alarm have its own small windows that imposes itself on older windows. Regardless, the analyst examines the alarm considering its IP address, alarm criteria, and some surrounding keystrokes if lucky. The analyst makes a decision on whether a response is needed. If there is a response, it is recorded and a trouble ticket is created for tracking.



To understand this weakness, consider the example of a NMAP scan being detected. The analyst receives a message that an event has occurred. The source IP address appears to be outside the network. What should they do? The normal defensive actions that can be made are to either block the IP address, send an email, or alert law enforcement. Going through the processes, one must question whether this attack is malicious. Yes, it appears so. Can you respond to this attack? It should be obvious that you cannot make a simple decision. Is the IP address real or spoofed (NMAP has a spoof option)? Are there any other events that make you believe that this is a more serious attack? Is it likely that such an attack would come from the same IP address? 1st level analysis does not give the security team the capability to determine a trend of a single IP address source or target. This is a challenge of operating an organization with a single level analysis model.

Many organizations rely on this as the only IDS process. At the end of a week, or a month, the statistics are collected to create elaborate charts that are presented to the management (or customer). On the rare case that a noteworthy event occurs, another group of experts are put into action to investigate. Regardless of how extensive the IDS recording capability is, the data these investigators receive is no better than the same alarm data the analyst had to use. This is because most IDS systems are designed to detect and not to react.

Another weakness of this analysis technique is not so easy to see. The alarms only trigger off known events. This means that there is no mechanism to catch or respond to new attacks. In fact, most IDSs only alarm, not on the attack, but on the scanning signature or upon the connection traffic after the intrusion occurred (noticing BO2K and Netbus traffic on default ports).

Threshold alarms (most notable are scan detectors) can be easily tricked into missing the purpose of the scan. For example, BO2K scans often are hidden inside what appears to be random ports scan. The alarm of a system or port scan hides critical data from the response team that is the purpose of the scan. If the response team

goes into action they may not actually check their network for this threat.

Consider the following "events" that are recorded. These events do not set off any commercial alarm because they are not known malicious strings. A common technique is to lower the threshold of alarm to a point where the majority of data is not threatening. When alarming off a simple action, one is really only recording an event (event alarm).

```

E HEAD /%62%69%67%63%66%6e%66%62%e%63%67%69 HTTP/1.0 FROM Y
E HEAD /%63%67%69%2d%6c%66%63%61%6c HTTP/1.0 FROM Y
E GET /%62%69%67%63%66%6e%66%62%e%63%67%69 HTTP/1.0 FROM Y
E HEAD /%71%75%69%6b%73%74%66%72%65%2e%63%66%67 HTTP/1.0
E HEAD /%63%67%69%62%69%6e HTTP/1.0 FROM Y

```

This is a subset of alarms where the IDS is set to record any web transaction, and seen before are from Whisker. On large-scale networks, these types of alarms are ignored. The events above are not considered hostile by any commercial network IDS on the market in March 2000, but this is an example of a malicious scan.

Problems with 2nd Level Analysis

The problem with 2nd level analysis is determining what information needs to be recorded. Although it is possible to control what alarms produce data, most IDS do not allow the user to determine what characteristics are recorded.

A worst-case example of high data capture rates is Shadow. Shadow is a popular freeware tool, but at high usage points, it produces loads of 850 megabytes of data per hour on OC-12 lines.

As the sensors increase the amount of events that are recorded, the more difficult it is to maintain an infrastructure to support the analysis. The increased number of events effect bandwidth, database storage, and the speed of searching and scanning the dataset. This is the inverse relationship between the amount of data recorded and the ability to determine malicious trends.

Last year I spoke with a security professional for a major web hosting service. His problem was centered on the amount of data versus the ability to capture and review it. As security operation centers are created, they are centered on the act of collecting data and alarms. When the alarm information is pushed (or pulled) to a central site for analysis, the security organization must deal with the infrastructure issue of handling all this data. Bandwidth usage for security, data collection, sensor management, data storage, backup and recovery: all become issues that are not really security but infrastructure management.

When information is reviewed and quickly tossed or tallied these problems are not severe. But when information is collected and stored for continual processing and analysis, as needed for 2nd level analysis, the infrastructure needs to be more tightly designed and managed to reduce the effects of exponential growth.

Also, remember that 2nd level analysis is inherently slow because of the type of pattern recognition that is occurring. That means that a system will use both processing speed and primary memory to crunch through the data. In turn, this pattern analysis will make it difficult to perform 2nd level operations in a real-time mode. The numbers of complex patterns being looked for are then limited. The first signs of IDS stress in performing these operations are dropped packet warnings. This occurs when the analysis engine cannot keep up with the packet capturing side of the system. Playing around with Network Flight Recorder's N-code demonstrates this relationship.

Conclusion of Analysis

This paper's intention was to lay the groundwork to better define what is needed from Intrusion Detection Systems. Currently IDS are weak in determining attacks against a network and supporting the response to those that are detected. An organization must create processes around the IDS to better implement them. 1st Level analysis is a successful technique in detecting known events with high probability of correctness. 2nd level analysis increases an organizations capability to detect attacks and to find other related activity that 1st level missed. 2nd level analysis however requires planning and foresight to implement successfully.

Chris Jordan currently develops, deploys and manages Security Network Operation Centers for Computer Sciences Corporation commercial and federal clients. He is an undergraduate of Virginia and obtained a Masters from George Mason University - both in Computer Science. Mr. Jordan was an original member of the Army CERT. He is a regular guest speaker at the National Defense University and System Administrations, Networking and Security (SANS) conferences on information assurance. He has received Government recognition for his work and has recently received CSC's technical excellence award.

copyright
Interested in advertising with us?