Why Anomaly Detection Sucks

Version 1.0

Todd Heberlein
Net Squared, Inc.

8 Feb 2005

1 Introduction

The techniques used by intrusion detection systems are often described as signature-based or anomaly-based. In this overly simplified view, signature-based techniques detect specific and known attacks or attacks against known vulnerabilities, and anomaly-based techniques detect unexpected activity (presumably an intrusion would be unusual). Anomaly-based detection generally has a longer history and has had more extensive government sponsored funding than signature-based detection. Anomaly-based techniques also hold the promise of detecting a wider range of misuse than signature-based techniques, including misuse by insiders that do not exploit any vulnerabilities and previously unknown attacks against unknown vulnerabilities.

However, despite these apparent advantages that anomaly-based techniques have over signature-based techniques, signature-based techniques have enjoyed considerably more operational success than anomaly techniques.

Why haven't we seen more success in anomaly-based techniques? Because anomaly detection sucks for users. Anomaly detection tends to produce non-actionable reports, requires the user to devote hours to understand the underlying cause of the report, and ultimately may leave the user with no resolution but plenty of angst.

A couple of years ago we wrote a proposal (which wasn't funded :^) to address these shortcomings of anomaly detection. Part of that proposal contrasted hypothetical reports from signature detectors and anomaly detectors. Ironically (or not), this past week I have been experiencing a security report that is remarkably similar to the hypothetical anomaly report from that old proposal, and as expected, I am mostly filled with frustration and angst.

This report revisits the question of why anomaly detection sucks. Section 2 reproduces the original text in the old proposal. It contrasts the potential of a signature-based technique to produce actionable information and the inability of anomaly-detection technique to produce similar actionable information. Section 3 presents my recent real-world example of a report of an anomalous or unexpected event and how frustrating this type of report can be to a user. After several days of investigation, we have not identified the root cause of the report. Section 4 describes some paths for future analysis for this report, and Section 5 summarizes this document.

2 Signature Systems Versus Anomaly Systems

[The text and figure in this section are from an old proposal.]

Anomaly detection systems can produce rather vague results when compared to the more commercially successful signature-based systems. When an analyst creates a new signature for an attack, he is usually aware of (1) the attack that the signature should

detect, and (2) the vulnerability the attack exploits. In systems like Snort it is very easy to include both the attack name and an ID for the vulnerability it exploits (e.g., a CVE identifier) with each report of a detected attack. Likewise, when a security scanner looks for a vulnerability, it should be able to provide a well known ID (e.g., the CVE number) for each vulnerability it finds. By combining the attack report, analysis from the vulnerability scanner, and a service such as ICAT that links a CVE ID to a set of patches and links for additional details, an intrusion detection system can easily generate a report such as the one in Figure 1, column A.

	A		B
Target:	128.131.7.2 : 161	**Target:**	128.131.7.2 : 161
Attacker:	128.120.56.31 : 5611	**Attacker:**	128.120.56.31 : 5611
Attack Name:	xdr_router_crash	**Attack Name:**	unknown
Vulnerability ID:	CVE-2002-0391	**Vulnerability ID:**	unknown
Vulnerable:	Yes	**Vulnerable:**	unknown
Damage:	Crashes Cisco routers	**Damage:**	unknown
Link to Patch:	Cisco_patch	**Link to Patch:**	none
Details:	Security Focus CERT CC	**Details:**	none

Figure 1: Hypothetical Reports From Signature & Anomaly Systems

The report in column A can be considered actionable information: it tells you (1) what the attack was, (2) whether you were vulnerable and need to do something, (3) where to get a patch to secure the system, and (4) where you can go for additional details.

An anomaly-based intrusion detection system, on the other hand, is more likely to generate a report that resembles Figure 1, column B. It might detect something suspicious, but it cannot give you a name for the attack, it cannot tell you about a specific vulnerability that needs to be addressed, and it cannot tell you what you need to do about it. This is definitely not actionable information.

The end result is that while an anomaly-based intrusion detection system may be more effective at detecting new attacks than a signature-based system, the approach requires the operator to diagnose the cause of the anomaly himself. As a commercial approach, this does not sell well.

3 Real-World Example

The other day I ran into a real example of this – an alert of an unexpected event but without any useful information for me to act on. This section describes this event.

The system of interest has a fair amount of security protection. It is the only Windows XP system on the network, and it protected by the following features: department firewall, Windows XP firewall, Norton firewall, Norton AntiVirus software, and automatic software update turned only. But despite all this security, we had a disconcerting unexpected/anomalous event, or actually series of events.

Figure 2 shows the unexpected/anomalous event (after clicking on the Details button) that first occurred on January 31st. Apparently “Windows Subsystem” wanted to connect to the Internet, which Norton recommended we should allow. No big deal? The problem is that port is 445 is used by Microsoft’s Server Message Block (SMB), and there are a

number of vulnerabilities in services on port 445 as well as exploits and worms to take advantage of those vulnerabilities!

Figure 2: Alert Window of Unexpected Event

Had we been hacked? Is our system trying to spread the attack to another host? What should we do about it? As we discovered, and as we predicted in the earlier proposal, these questions were not easy to answer.

Additional details about the incident:

- The alert comes up infrequently (roughly once every 1-2 days), so if we do have an automated attack, it moves very slowly.
- The system always tries to connect to the same IP address (204.255.139.8), so if it is an attack and it chooses targets at random, they have a problem in their random number generator (e.g., always seeding it with the same starting value).
- A reverse name lookup on the target address (204.255.139.8) shows it to be www.navysbir.com, a site for promoting the Navy's Small Business Innovation Research (SBIR) and Small Business Technology Transfer (STTR) programs.
- We had a Navy SBIR contract at one point.
- Web and network administrators for the target system could think of no legitimate reason for our system to be contacting their system via port 445.
- A full scan by Norton's showed no malicious content on the disk.
- Clicking on additional information about "Windows Subsystem", the software reportedly making the outbound connection to the navysbir.com site on port 445, only described it as the generic windowing system allowing multiple programs to run. There was no information about the process/application that was using the Windowing Subsystem.

- The task bar and task manager showed no applications were running. A number of processes were running, but their names were relatively short and meaningless.
- Windows XP Event Viewer showed no useful information.
- Norton's Internet Security log file provided no additional details.
- Clicking for more information about the alert shows that Norton Internet Security is clueless as to the cause of the threat (See Figure 3).

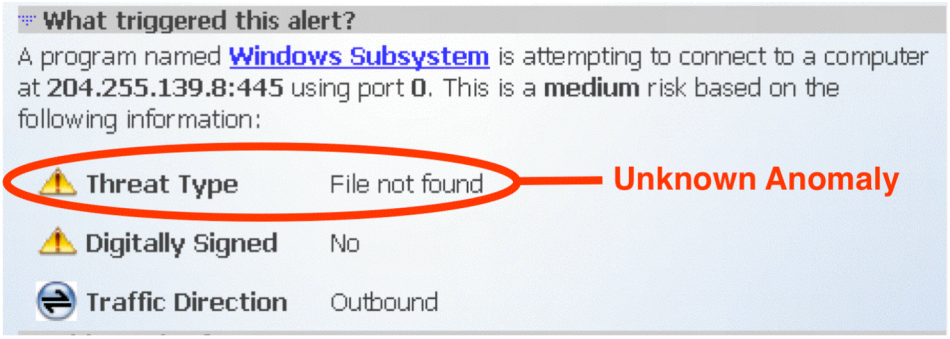

Figure 3: No Clue as to the Cause of the Event

So, like the hypothetical example in Section 2, our security system alerted us to unexpected activity, but the system can provide no details as to the underlying cause of the threat, whether we should be genuinely concerned about the activity, how we could find more information about the event, or what actions we should take.

4 Future Directions

Norton Internet Security flagged an unexpected event, but the system could not provide any useful information to resolve the potential threat. Fairly extensive review of the system (including reviewing multiple logs and performing virus scans) and questions to the target's administrator could find no underlying cause of the event. The underlying behavior (slow event activity repeatedly targeting only a single system) did not match any malicious activity with which I was familiar (although, I do not regularly track active malware). However, given that the target is port 445, a target of numerous exploits and worms, I shall continue my investigations.

Part of that continued investigation is to determine how to reliably produce or predict the generation of the report. Anyone tasked with debugging intermittent errors knows how frustrating but important this step can be. We face a least two challenges on this front. First, this is a production machine, so the user must be allowed to use it. This will produce some changes to the system and limits the experiments we can conduct. Second, the machine is set up to continually update security elements (e.g., antivirus software and signatures) and patches (Microsoft released a dozen updates this Tuesday), and any change to the machine may disrupt our efforts to reproduce the alert message.

In addition to being able to reproduce or predict the alert report, we need to collect evidence that may be useful in our research to diagnose the underlying cause of the report. Some of these activities may include:

- Continue to review information on the Internet. Unfortunately this has been a laborious process so far. The additional data I hope to collect may help me refine

the search, but the lack of semantic information on the web makes simple keyword searches difficult.

- Post requests to various discussion groups and experts to see if they are familiar with the underlying cause.
- Allow the connection attempt to continue, capture the traffic, and use network forensics tools to identify what the system is attempting to do. However, allowing our system to possibly attack a potential future sponsor's system does not seem like a wise idea. Furthermore, if the target network is blocking port 445 (which they claim to be doing), the TCP connection attempt will fail, and we will learn no additional details.
- Allow the connection attempt to continue, but change part of our network infrastructure to reroute the offending packets to a honeypot of our own. Then we can apply forensics tools to capture and analyze the activity. This will entail a fair amount of work, but may provide the most definitive answer.

Of course, given that our system is already fully patched (with the exception of the patches released in the last 24 hours) and wrapped with lots of existing security features, there may not be a lot I can do even if I do find the underlying cause of the reported event.

Stay tuned...

5 Conclusions

Anomaly detection has been a strategy for detecting malicious or intrusive activity since the field of intrusion detection started in the 1980s. The government has spent millions of dollars sponsoring years of research and development into anomaly-based detection. Anomaly-based detection, unlike most forms of signature-based detection, has the potential to detect previously unknown attacks against unknown vulnerabilities.

However, despite its long history, resources applied, and potential to detect unknown attacks, anomaly detection has not enjoyed the level of commercial and deployment success that signature-based detection has.

Several years ago we posited that while signature-based detection can produce very clear and actionable information, anomaly-based detection would be more likely to produce irritatingly vague and non-actionable information. The users would need to invest considerable time trying to understand the cause of the alert. The cause of the alert may never be found (especially if the report is transient). Even if the cause of the problem is found, there may not exist a readily available solution to address the problem. In the end the user may lose lots of time, have no resolution to the problem, and be left with lots of angst.

This past week I had the opportunity to experience this pleasure first hand.

Section 2 of this report, borrowing from an old proposal, contrasted the potential of reports from signature and anomaly-based detectors. Section 3 illustrated our recent real-world experience with the problem, and Section 4 looked at future ways we can devote more hours to resolving the underlying cause of the alert report.