Strategies to Reduce False Positives and Negatives in NIDS, Part Two

This is the second of a two-part series devoted to the discussion of false alarms on network-based intrusion detection systems. The first article offered an overview of false alarms, of false positives as they are commonly known, and false negatives. This installment will look at a few ways to reduce false alarms.

Assessing Methods of Reducing False Alarms

The primary purpose of a network-based intrusion detection system (NIDS) is to identify attackers trying to expose vulnerable network services. The NIDS device can either respond to the attack or alert proper personnel, who can then take appropriate action. This core functionality is important to recognize when deciding how to effectively reduce false alarms.

To properly analyze false alarm reduction strategies, it is necessary to quantify risk and the NIDS role in risk reduction. For demonstrational purposes we will focus on two different formulas that can be used to quantify risk. It should be stated that these formulas are not foolproof: the mathematics on which they are based are dubious; however, they are the most prevalent formulas for use in these situations, so I will include them in this discussion.

One formula assumes that risk is roughly equivalent to single loss expectancy (SLE). Single loss expectancy attempts to quantify the potential loss if a security breach were to occur. The formula for this quantification is:

SLE = (Asset Value x Exposure Factor)

In this formula, 'Asset Value' represents the monetary value of the information asset that may be lost or damaged in the case of intrusion. 'Exposure Factor' represents the percentage of the value of the asset that is likely to be lost due to the intrusion. For example, if your car is worth $20,000 and the incident we are concerned with involves colliding with an 18-wheeler. In such a situation the car would almost certainly be totaled, therefore the SLE would equal $20,000. The SLE can therefore be considered to be the result of the asset value multiplied by risk of damage that is likely be inflicted by exposure factor. In this formula a NIDS device reduces risk by dynamically reducing the exposure factor.

The second formula states that risk is equal to exposure multiplied by threat:

Risk = Exposure x Threat

In this equation, 'Threat' is determined by the type of attack that is being detected. For instance, an IDS can alert to three types of attacks:

1. Port scans , automated scans and sweeps;
2. Denial of service; and,
3. Service attacks and compromises (database ,web , ssh)

Scans and sweeps do not really result in immediate loss. They can be a predecessor to more serious incidents but in and of themselves do not cost the organization any down time or money.

As a result of this, the threat posed by scanning is minimal. On the other hand, a denial of service attack can cost the organization lost revenue; however, it is not likely to have long-term consequences for the company, so the threat factor is medium. The third type of attack, service attacks and compromises, are more likely to be damaging to the business, as a result it is considered to have a high threat factor.

The point of this discussion is that, depending on the risk formula that is chosen, a NIDS device can reduce risk by either reducing exposure factor (as in the first formula) or reducing threat (as in the second formula.) Therefore, in order to determine the effectiveness of different false positive reduction strategies, we need to examine how effective a NIDS device is in reducing the exposure or threat, which these two formulas should allow us to do.

As mentioned, of the three categories of attack, service compromises have the greatest potential for monetary loss. For the purpose of demonstration, we can use a concept that I refer to as Degree of Attack Accuracy (DAA), which is weighted to reflect the increased importance of service compromises. It should be stated that DAA has been created for this article to demonstrate relationships and to quantify the relationship between NIDS implementation models. Without this formula we have only opinions and opinions are difficult to quantify. Degree of Attack Accuracy is proportional to the amount of attempts an attacker has to expose vulnerabilities. The formula by which we calculate the Degree of Attack Accuracy is :

DAA = service compromise accuracy x (square root of (reconnaissance attempts x denial of service recognition)

For purpose of this analysis we will rate these on a 1-5 numbering system (1 being the worst and 5 being the best, therefore a 25 would be a perfect score.) DAA is weighted to include the potential for false negatives to the best of NIDS device and networks designs capabilities. A design that reflects a high DAA is preferred.

Common techniques that can be used to reduce false alarms include placing the NIDS device behind a firewall, tuning signatures to only include certain platforms and or services offered on the network, and reduction through thorough network analysis. Each of these techniques has potential benefits and drawbacks.

Placing the NIDS Behind a Firewall

Placing the NIDS behind a firewall is a common technique that is very easy to implement. This technique requires no alteration of the default configuration of the device and very little expertise. When an alarm is received there is a high percentage of certainty that is it a real threat than if the NIDS was not behind the firewall. However, there are still instances where it will trigger benign alarms, such as the web page posted by Joe sys-admin about the Code Red worm from the example in the the first article in this series. There also exists a potential for false negatives such as denial of service events. The DAA formula for this design would be as follows:

Reconnaissance attempts = 3 (host sweeps may be recognized, port scans probably not) DOS recognition = 3 (The IDS will only see DOS attacks on services allowed through the network firewall) Service compromise = 5 DAA = 15

Tuning NIDS Signatures

Another common and relatively easily deployable solution is to tune the NIDS device's signatures to only watch for services or operating system-specific conditions that apply to the network being monitored. This requires more skill and vulnerability knowledge than does placing the device behind a firewall and it is less prone to false alarms. However if we analyze the DAA formula we realize that the fruits of our labor our not really realized.

Reconnaissance attempt = 5 (host sweeps and port scans recognized) DOS recognition = 5 (device will see all DOS attacks on network) Service Compromise = 3 (will allow an attack until the attacker triggers an alarm that actually effects the network being monitored) DAA = 15

Network Analysis

The most laborious solution is to reduce false alarms through network analysis. This requires considerable networking knowledge, expertise, analysis and some imagination. In this design, the NIDS device is placed outside the firewall with a full complement of signatures. Alarms are analyzed and then tuned very specifically to eliminate common false alarms. The engineer must be careful not to introduce false negative conditions through alarm filtering. Done correctly, this method is very effective in mitigating risk. Most of this analysis is done within the first few weeks of implementation, however some ongoing analysis is required. The DAA for this method reflects its effectiveness.

Reconnaissance attempt = 5 (host sweeps and port scans recognized) DOS recognition = 5 (will see all DOS attacks on network) Service Compromise = 5 (will alert on any attack at a service) DAA = 25

Network Analysis Tools

Reducing false positives through network analysis is the most secure way to manage NIDS technology. Proper analysis of network traffic often requires the use of tools such as Snoop or Tcpdump for gathering network and session data not available with some NIDS logging facilities. A full tutorial on Snoop and TCPDump is beyond the scope of this article however some commonly used switches are

```
 # snoop -h (will show help menu) -d = device  -v = verbose  -x = hex dump  -
r = don't do name lookups
 # tcpdump -h (will show help menu) -i interface -v verbose  -x hex  -S
absolute syn-ack values -n do not look up domain names
 These quantifiers can be used with both Snoop and Tcpdump Host  Src Dst Net
TCP,UDP,ICMP  Port
```

Here is a simple example:

```
 tcpdump -n -x dst host 192.168.1.10 and udp dst port 53
```

This will catch anything destined for host 192.168.1.10 with a destination port of 53 and the protocol field set to UDP. This will also not perform name lookups and will capture hex data as well. To see how this is useful, we will debug an alarm that was captured with Ethereal. We will need to debug the hex to find the actual cause of the alarm and correctly classify it. In this next example the IPs have been changed so checksums will be incorrect. This alarm triggered an ICMP flood alarm on a NIDS device. These alarms are generally threshold / time counters.

Using Hex to Debug Alarms

```
 Frame 1 (56 on wire, 56 captured)     Arrival Time: Jul 13, 2001
12:04:08.0000     Time delta from previous packet: 0.000000 seconds     Time
relative to first packet: 0.000000 seconds     Frame Number: 1     Packet
Length: 56 bytes     Capture Length: 56 bytes Raw packet data     No link
information available Internet Protocol     Version: 4     Header length: 20
bytes     Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)         ....
..0. = ECN-Capable Transport (ECT): 0         .... ...0 = ECN-CE: 0     Total
Length: 56     Identification: 0x0000     Flags: 0x00         .0.. = Don't
fragment: Not set         ..0. = More fragments: Not set     Fragment offset:
0     time to live: 241     Protocol: ICMP (0x01)     Header checksum: 0x4f50
(correct)     Source: router1.destination.net (172.16.1.100)     Destination:
192.168.1.10 (192.168.1.10) Internet Control Message Protocol     Type: 11
(Time-to-live exceeded)     Code: 0 (TTL equals 0 during transit)
Checksum: 0xfca1 (correct)     Data (28 bytes) 4500 0038 0000 0000 f101 4f50
ac10 0164 c0a8 01a0 0b00 fca1 0000 0000 4500 00c4  9748 0000 0106  6e7b c0a8
01a0 ac10 02c8  0911 0050
```

The data we currently know is

- 192.168.1.10 received a ICMP Time to live exceeded message from 172.16.1.100
- the hex data of the original packet

This hex data is important for us to be able to determine the root cause of the alarm. One important but often overlooked design built into ICMP is that the error messages quote some of the original packet that caused the alarm. For more information the ICMP RFC can be found at RFC0792. Another excellent resource on ICMP is ICMP Usage in scanning by Orif Arkin available at http://sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf. Here is the hex from the original packet that caused 172.16.1.100 to send a time to live exceeded message.

```
 4500 00c4 9748 0000 0106 6e7b c0a8 01a0 ac10 02c8 0911 0050
```

Conclusions made of the original packet by decoding Hex

- IP Version 4
- Header Length 20
- Length 196
- IP ID 38728

  TTL 1

- Protocol 06
- Src IP 192.168.1.10
- Dst IP 172.16.2.200
- Src Port 2321
- Dst Port 80

Here is what we are able to determine from the original packet. There was TCP traffic destined from 192.168.1.10 port 2321 to 172.16.2.200 port 80, this is seemingly normal Web traffic. The TTL expired at 172.16.1.100 causing the router to send back several ICMP time-to-live exceeded messages. This went above the threshold determined by the NIDS device and triggered an ICMP flood alarm. The important thing to point out is that this is not a false positive. Instead it could be classified as a false alarm that was most likely caused by a router configuration problem on the

Internet. This alarm can also give insight into network and provider issues and aid in troubleshooting, so the alarm provides value to the network even if it is not security related.

Now suppose we see this happen frequently (maybe it's a trustworthy partner with an inexpert network engineer) and we wish to not be alerted to this type of activity. You can write exclusionary rules so we are not alerted to this. It is important when writing exclusionary rules is to be as specific as possible to reduce the chance of creating false negative conditions. Here are some example exclusionary rules for Cisco IDS, Snort and Dragon.

```
 Cisco IDS RecordOfExcludedPattern 2152 11 OUT IN 172.16.1.100
255.255.255.255    Snort pass icmp 172.16.1.100 any <> any any (msg:"ICMP
Time Exceeded"; itype:11;) Dragon ICMP I 172.16.1.100/32       11 255 ;
Ignore TTL expired messages from this host
```

These are very specific exclusionary rules that should preclude any alarming from this host for time to live exceeded messages only.

NIDS Role in Mitigating Risk

One of the key ways in which NIDS devices can help mitigate risk is by detecting attacks. To reduce threat, NIDS devices can alert personnel when an attack is in its early stages and/or automatically respond by sending TCP Reset packets or changing access on access control device such as a router or firewall. It is important to recognize that threat reduction is time-dependent. Therefore, the greatest threat reduction benefit is realized when the time between an attack occurring and removal of the source of the attack from the network is minimized. This can build a strong case for automated response. However, many system and security administrators are uncomfortable with automated response and not willing to accept the possibility of denying legitimate network traffic. Since most attacks only take a few seconds, the chance of alerting a real person and having them manually mitigate the risk successfully before the attack is complete is small. Whether it is acceptable to program the NIDS device for automatic response is a business decision. Before deciding what actions are appropriate a few questions that should be asked are:

1. If the choice is made to deny access based on NIDS rule triggers should the session be stopped by sending TCP resets or by implementing changes that will prohibit connectivity with access control devices? Resets are safer. However, access changes are more effective in mitigating risk due to the fact that the offending IP is blocked and, for all intents and purposes, the attacked network appears to the attacker to be down.
2. Is there a way to limit the time period or the number of hosts that are denied at any single time in order to prevent potential mass denial of service? Most products have this functionality or scripts can be built to provide it.
3. Which alarms are being considered for automated responses? Alarms that are not easily spoofed, are relatively accurate, and potentially high risk are strong candidates.
4. What is the percentage of "false alarms" for signatures being considered for automated response?
5. What is the single loss expectancy if an event occurs?
6. What is the accumulated loss expectancy from denying access to legitimate customers?

If it can be reasonably demonstrated that no mass denial of service condition exists, that the degree of alarm accuracy is high, and the risk associated with a particular alarm is high, then a good business case can be made for automating the response of the IDS on specific alarms. For example, assume the Web server uses an older custom application and the Web server itself is vulnerable and cannot be upgraded or patched because the application will cease to work. A customer database is on this server and if it is exploited, the potential loss could be $100,000. However, if the server is compromised and database is not exploited the loss from each successful compromise could be as little as $1000 plus lost revenue that occurred while the server was being reinstalled. The average customer who visits the site graciously spends $300. Proper precautions have been taken to ensure that no possible denial of service conditions exist. The degree of accuracy on the specified alarm is 80%. In this scenario, one successful attack could cost between $1,000 and $100,000 and the chances are 4 to 1 that any occurrence of this event will be a legitimate attack. If the choice is made to respond manually, there is a high probability that the attack will be successful, thus leaving the IDS to be used for forensic data instead of mitigating risk. In the above example a strong argument could be built for automated response vs. manual response to specified events.

NIDS primary function is to help mitigate risk through a reduction of the exposure variable. False positives and false negatives severely impact the technology's ability to effectively mitigate risk. Through well thought-out implementations, proper communication and device management as well as a thorough understanding of the technology these factors can be appreciably reduced to allow for effective NIDS implementations.

Kevin Timm is a Network Security Engineer at Netsolve Incorporated in Austin TX.

last updated September 27, 2001