Global Information Assurance Certification Paper

Jonathan Kobrick
GSEC Option 1 version 1.4b
17 March 2003

NIDS Countermeasures: What, Why, Where, When, and How

Introduction

The purpose of this document is to discuss Network Intrusion Detection System countermeasures that are currently available, review their strengths and weaknesses, and provide some practical recommendations for when and how to use them. The intention is to discuss NIDS countermeasure options from a conceptual perspective while being as vendor neutral as possible.

Background

Intrusion detection systems ("IDS") have been around for a long time. Broadly defined, IDS is "a monitoring and analysis procedure that will discover any unauthorized or unwanted activity on a computer or network." (ref. 8) An IDS monitoring a single computer is called a host-based intrusion detection system ("HIDS"), while an IDS monitoring network traffic (usually monitoring traffic from many computers) is called a network-based intrusion detection system ("NIDS"). There are many different techniques which IDS may employ to identify malicious activity. Some examples include:

- *Log file monitoring* – watching for log entries that indicate malicious activity is occurring, e.g., many logon attempts in a short time period;
- *File integrity checking* – watching important files for inappropriate changes;
- *Protocol compliance* – using known rules that dictate how a legitimate process is suppose to occur or how a data structure is suppose to be formatted, etc. to evaluate the current process or data structures for compliance;
- *Behavioral analysis* – learning what normal activity looks like (e.g., what files are accessed at what times by what users, etc.) and watching for deviations; and
- *Pattern recognition* – comparing known malicious activity or data patterns with current activity or data to identify matches (or key similarities).

It is very important to consider which of these techniques are effective and/or appropriate for a given environment and circumstance, as well as which products implement them (and how they are implemented), however, these considerations falls outside the scope of this paper. Because, regardless of the type of technique employed by an IDS, the goal is the same – to identify malicious activity. But what happens after the IDS identifies an attack? To be of

any use to an administrator, one or more responses must be performed by the IDS. A response can be as simple as logging the malicious activity to a file. However, as IDS becomes more mature, more and more sophisticated responses have been developed. This paper focuses on a specific subset of response types called countermeasures. Furthermore, while there is some limited commonality between HIDS and NIDS countermeasures, this paper focuses on NIDS countermeasures. (ref. 2)

IDS Response Types

As noted above, an IDS response may be very straightforward, such as logging malicious activity to a file. There are many different responses supported by various IDS products (as well as products designed to work with IDS to provide response functionality); however, they all fall into three categories:

- *Logging* – a response that records the malicious activity somewhere (perhaps to a file or database either locally or to a remote system);
- *Notification* – a response that attempts to notify someone (perhaps an administrator) that malicious activity has been identified; and
- *Countermeasure* – a response that attempts to 'reduce a threat, vulnerability, or an attack by eliminating or preventing it, or by minimizing the harm it can cause.' (ref. 8)

Under ideal circumstances, an IDS is going to have some combination of responses configured that include at least one response from each category, as it should be clear that each performs a distinct and important function. However, of these categories, countermeasures could arguably be the most important, assuming that each response type worked perfectly (as you'd rather prevent every attack than record or be notified of every attack!). Unfortunately, where the current state of countermeasures fall today, none are anywhere near perfect, thus a generous helping of malicious activity logging, notification, and countermeasures is the best prescription for any IDS implementation.

Why Use Countermeasures

One compelling reason for using countermeasures is to have defense in depth; an essential network security principle which is "the practice of layering defenses to provide added protection." By doing so, you make it harder for an attacker to compromise your assets. Even if you patch all your servers in a timely fashion, perform regular vulnerability scanning, use strong authentication, implement private VLANs, have good user administration procedures, apply filters to all external network connections, etc. etc., there can always be additional measures in place to further bolster your security posture. To be fair, security costs money, thus it is not possible to implement every security control in the book. However, IDS, and more specifically NIDS, is on its way to becoming as ubiquitous as the firewall. In addition, most NIDS products support at least some

countermeasures, which means if you've got NIDS already, it's very likely you may be able to take advantage of countermeasures without spending money (and there's always open source NIDS countermeasure options). But considering that time is money, it should be noted that getting NIDS countermeasures to function optimally may be time consuming (especially since much of it may depend on how well tuned your NIDS is, which is usually a time consuming process). (ref. 7)

In addition to being just another security control, countermeasures can perform a unique function, by providing automated attack mitigation that would take much longer to perform manually (which can often be too late). Since IDS has the ability to detect attacks occurring in or near real-time, the potential is created for mitigation of the attack before initial damage is done (or also possibly before additional damage can be done). (ref. 4)

NIDS vs. HIDS Countermeasures

HIDS in general has a few inherent advantages over NIDS, which can translate to more effective countermeasures. Upon a closer analysis, HIDS detection techniques are a superset of NIDS detection techniques. This is because a HIDS can monitor the computer's operating system, applications, and data, as well any network traffic going to and from the computer. A NIDS can only monitor network traffic (however, going to and from many computers, thus, it is cost effective, but at the cost of scope of monitoring capabilities). A good HIDS product should include detection techniques that take full advantage of information available at the host level (OS, apps, data) as well as the network level (communications with other hosts via the network). Another advantage afforded by a HIDS's position on the host itself is the ability to directly interact with (and manipulate), in real-time, any software running (including operating system calls, the network stack, etc.) This directly benefits its ability to launch effective countermeasures. Since NIDS sits on the network, usually passively monitoring the traffic, it is not nearly as easy to prevent an attack from being processed by the target host. However, there are some NIDS products that are able to sit inline on the network (e.g., RealSecure Guard, Hogwash, etc.), which affords them the ability to manipulate network traffic in real-time. (ref. 2)

Caution is Required When Using NIDS Countermeasures

Since most NIDS countermeasures are designed to disrupt communications (in one form or another) between a malicious host and the target, they can mean serious trouble if used on legitimate traffic. Thus, they should never be configured to trigger on every event identified by the IDS. Rather, countermeasures should only be enabled on carefully selected signatures. These select signatures usually have a few common characteristics. First and foremost, they must not be prone to false positives. As it is well known, most IDSs suffer from chronic false positive issues. However, in general, some

signatures are more prone to false positives than others. This can sometimes be by design, to ensure suspicious activity is identified and reviewed, rather than go undetected. Other signatures are significantly less prone to false positives, as they are usually looking for some very specific characteristics that would never be exhibited in legitimate traffic. In addition, good countermeasure enabled signatures must be very difficult to spoof. If an attacker is able to determine that countermeasures are being used (which may not be very difficult at all if they are paying attention), they could try to spoof legitimate sources (effectively using your NIDS as a denial of service tool against your own network!). Usually, signatures that monitor attacks requiring TCP connections are nearly impossible to spoof, as the requirement exists for the attacker to be able to receive data back from the target (at least to complete the three way handshake). While there's the risk that an attacker could gain the ability to launch an attack from a trusted source, chances are you would want to use countermeasures against the trusted source in this scenario. It's just a matter of choosing the right countermeasure and configuration (e.g., don't block a trusted source indefinitely, etc.). Finally, even if a signature is not prone to false positives and can't be easily spoofed, its significance may not warrant a countermeasure. The point is to avoid the risk of accidental countermeasures wherever possible (as this risk exists with every signature that is countermeasure enabled). However, it should be noted that some high security environments might have sufficient reason to put security over operational requirements. In this case, it may be appropriate to default to disruptive countermeasures for any potentially suspicious activity, rather than permit the activity to continue, investigate it manually, and take the chance that damage was already inflicted in the meantime.

Another important step to successfully implementing countermeasures is the proper initial NIDS policy configuration and baselining. Depending on the size of the network and the NIDS deployment, this can take several months (or more) to complete appropriately. During this baselining process, the NIDS events must watched closely to remove the false positives generated by legitimate traffic. In addition, suspicious and malicious events must be reviewed to formulate an appropriate countermeasure implementation strategy. It is not safe to enable countermeasures on signatures without a well tuned NIDS in place.

Types of NIDS Countermeasures

A number of NIDS countermeasures types exist. Some may be self-contained within the NIDS product, while others may be dependant on additional products performing actions at the request of the NIDS. Available NIDS countermeasure types include:

- *Shunning* – filtering a suspicious/malicious host at a network gateway. It involves the NIDS talking to a filtering device on the network (such as a router or a firewall) to prevent a malicious host from communicating with the target host or network;

- *Session Sniping* – interruption of a suspicious/malicious host's communication session with a target. This involves the NIDS sending packets (such as TCP resets or certain ICMP types) at the target or both the attacker and target;
- *Non-blocking* – the prevention or minimizing of an attack impact by triggering specific customized actions other than attempting to block the attack), e.g., triggering a script to scrub a worm from a targeted host;
- *Re-direction* – changing the route and/or target of a suspicious/malicious host's communication. This would be used to apply additional filtering or security checks, or to divert an attacker to a controlled environment (sometimes called "fish-bowling") (ref. 5);
- *Counterattacks* – where the NIDS or another device directed by NIDS launches a counterattack on the suspicious/malicious host, usually to attempt to neutralize their ability to complete an exploit or perform any additional damage; and
- *Filtering* – removing specific network traffic on the wire in real-time that is suspicious/malicious. NIDS offering this type of countermeasure have started to take on the name Network Intrusion Prevention Systems ("NIPS").

When considering these various countermeasure options for a specific environment, there are many important factors that must be understood in order to be able to make a sound decision on what countermeasures to choose, as well as how to implement them effectively. Let's look at some of these in detail.

Shunning

As described above, shunning, or host blocking, involves a NIDS management station (or NIDS sensor, as the case may be) requesting a dynamic access control list or policy update to a network filtering device. The filtering device is usually either a firewall or a router, but it could be any network gateway device (with filtering functionality and the ability to somehow accept an automated policy update request from the NIDS). This countermeasure usually has a limited degree of usefulness, mostly because of the fact that an attack may have already been successful by the time the filtering device's policy has been updated to block the attacker. Thus, even when shunning is optimally configured, it can be assumed that many types of attacks may not be blocked. However, there are a number of distinct benefits to using this countermeasure, as well as ways to increase its effectiveness. Firstly, while an attacker may have successfully executed an exploit on a target and compromised the machine, preventing the attacker from having any additional access can be very helpful. The attacker may not be able to make use of the compromised machine, which provides the administrator time to address the breach and prevent the attacker from doing anymore damage. However, it is important to realize that shunning may not provide this benefit in all circumstances. For instance, if an attacker has been able to compromise the target with an exploit that has the target initiate its own

connection back to the attacker, then the attacker may still be able to control the target (unless the outbound firewall policy restricts the target, which may or may not be the case). Ideally, a shunning configuration should include filtering to prevent both the attacker from accessing the network and the target from being able to communicate with the attacker. In addition to preventing an attacker from being able to communicate with the compromised host, there is also an increased chance the initial attack was blocked to begin with. Even when a single packet attack is used, which negates the possibility that blocking just the attacker from accessing the target network would block the initial attack, the target could be prevented from sending any reply to the attacker. Of course, this requires that the NIDS get the filtering policy updated quickly enough, which may not be possible if the filtering device must reload or is delayed for some other reason. The speed of policy update must be greater than the time that it takes a target to receive and process the attack and sent the reply back through the filtering device. NIDS running on a filtering device, if the circumstance permits, is the most optimal from a performance perspective. (ref. 1) Many exceptions and nuances exist, such as:

- Single packet denial of service attacks or worms, or any other single packet attack where no reply is needed makes shunning useless;
- If any attacker has more than one source IP address to use (which is often the case), they may try to get a target host to communicate with a different IP address, to mitigate their risk that their attack IP has been shunned;
- If an attacker is on the internal network, there may not be a filtering device available to block them from the target; and
- Sometimes applying an access control list or policy update requires a reload of the filtering device (as mentioned above), interrupting current sessions – depending on operational requirements and the frequency of events triggering shunning, this may be unacceptable. (ref. 9)

There's also the duration of a block to consider. In some circumstances, it might be safer to block attackers for ten minutes, rather than ten days. The optimal duration setting will depend on the specific environment.

It may also be valuable to use "white lists," which include networks and/or hosts that can never be shunned. Common entries may be business units, partners, vendors, root DNS servers, etc. However, consider these entries carefully, as you must be sure that you want to completely bypass your control. A white list may be useful when you are first implementing shunning, or when you know that your NIDS is not under control, but are okay with occasionally blocking legitimate traffic, which the exception of key networks and/or hosts.

The filtering update request channel is another consideration, for both security and performance reasons. An out-of-band connection between the NIDS and filtering device may increase performance and prevent the possibility of a disruption of the filtering request. If this request must go over the monitored

network, it is important to consider a sound authentication and encryption scheme between the NIDS and the filtering device, or you may find that someone could easily make unauthorized policy updates by spoofing a NIDS request. But be mindful of the fact these controls may impact the performance of the update. (ref. 9)

Session Sniping

This countermeasure works by sending packets to the target and/or the attacker in an attempt to terminate the session, preventing the attacker from completing the attack. Like shunning, session sniping (or sniping for short) must be enabled with care to prevent the interruption of legitimate sessions. However, since sniping only targets a specific session, the host may still be able to re-establish a session with the target, provided the same signature is not re-triggered on the NIDS. Another benefit to sniping, in contrast to shunning, is that there is no reliance on another device; the NIDS product does not have to support communications with a filtering device, as it can provide a sniping countermeasure on its own. However, these benefits are often overshadowed by limited effectiveness, as is discussed below.

The primary sniping technique that a NIDS might employ is TCP connection resets. Less common sniping techniques include ICMP messaging as well as higher layer protocol messaging.

TCP connection resets are the most common sniping technique by far, and involve the NIDS sending TCP reset packets to either the target or both target and attacker to attempt to close the TCP connection. Obviously, TCP resets can only be used with TCP based attacks. If either of the two communicating parties accept and process the connection reset request, the session will be terminated. However, a NIDS must forge an appropriate packet with the TCP RST bit set, which includes the same source and destination information, sequence numbers, acknowledgement numbers, etc. that would be sent by the attacker or target. (ref. 4) In addition, this packet must be in received in sequence with the traffic to work. This opens the possibility for an attacker to completely circumvent this type of sniping. An attacker can evade TCP resets if they can do the following:

1. Calculate how many attack packets will trigger the NIDS signature;
2. Generate an extra packet that goes after the attack packets in sequence; and
3. Either send the extra packet before the last attack packet or send the extra packet after the last attack packet very quickly.

As the NIDS detects the attack, it generates a TCP reset with the appropriate sequence information relative to the last attack packet that triggered its signature. However, the TCP reset will not succeed. The target ignores it because it is no

longer in sequence, as the attacker's extra packet has moved the sequence forward.

Another method by which an attacker may be able mitigate the TCP reset countermeasure is by using the TCP push bit, which instructs the target to process that packet right away, rather than wait for enough packets to fill its buffer before sending the packets up the stack. Though a successful TCP reset from the NIDS would prevent the target from sending any replies to the attacker, as well as clear the buffer (which occurs as the connection is terminated), the attacker may have already gotten the target to process enough attack packets to do damage. (ref. 6)

What about non-TCP based attacks? While TCP resets are useless, ICMP messaging could be considered. With ICMP messaging, the attacker is sent various ICMP types; usually type 3 (destination unreachable) messages that indicate the target is not available. The aim is to have the attacking host process the message and stop sending any more packets to the host. Some ICMP type 3 codes that might be used are Net Unreachable (code 0), Host Unreachable (code 1), and Port Unreachable (code 3) (ref. 1). These countermeasures are largely ineffective, because an attacker can easily filter out ingress ICMP messages. And even if the messages are not filtered, chances are attacker's software is going to ignore the ICMP messages. Higher level messaging could involve a NIDS forging application layer payloads with instructions to terminate any additional processing or to cease an additional interaction with the attacker. While there may be some merit to this approach, it appears as though very little has been done to date in this area. One possible explanation may be that forging higher layer payloads is impractical for a NIDS to do effectively. (ref. 4)

Having NIDS send sniping packets to an attacker in addition to a target can be risky. There may be an increased likelihood of preventing the attack, however, an attacker may be able to capture and review this sniping packet. It is quite possible they will determine which NIDS product sent the packet. This information can be very valuable to an attacker as it can greatly assist in planning for NIDS evasion during future attacks. (ref. 1)

Non-blocking, Re-direction, and Counterattacks

These countermeasure types are the least conventional, often being used for very specific purposes. Some examples include using re-direction to send attackers to a honeypot or using non-blocking to execute a logging script on a target. Counterattacks are unlikely to be appropriate for most circumstances. In addition, it is probable that this type of countermeasure is not legal in most places.

Filtering

Filtering, or as many call it these days, network intrusion prevention systems ("NIPS"), might be considered the best countermeasure type because of their potential to be so effective in mitigating an attack. Its level of effectiveness stems from the fact that a malicious packet stream (or specific packets) can be prevented from ever reaching the target. With NIPS, attacks are detected and removed on the wire in real-time before ever passing the NIPS host. To accomplish this, a NIPS host must sit inline on the network (as any filtering device must). This is an ideal countermeasure when operating correctly, as it completely removes the attack traffic before ever passing the perimeter (or whichever network gateway the NIPS is positioned). In some respects, NIPS can be thought of as an enhancement to shunning, in that both are blocking countermeasures designed to filter attack traffic. However, since NIPS can filter more granularly than shunning, they provide less risk with false positives (e.g., a legitimate source may not be completely blocked, just the packets triggering the false positive). (ref. 6) However, there are different requirements for NIPS implementation as opposed to other countermeasure types. The inline requirement translates to a single point of failure, therefore redundancy must be considered. In addition, real-time detection and filtering requires significant performance, depending on the amount of monitored traffic. These performance and redundancy requirements mean that this countermeasure will initially be more expensive as compared with other countermeasure types. (ref. 3)

NIPS have recently been given a significant amount of attention as many security technology vendors having been investing time and money in the development (and marketing) of products with this functionality. However, this technology has been around for quite some time in products such as RealSecure Guard. (ref. 10)

Conclusion

NIDS countermeasures must be used carefully and sparingly. While they can offer enhanced security, there are many risks to employing their use improperly. Of all the countermeasures, filtering has the potential to be the most effective countermeasure.

While countermeasures can prevent compromises, they can't be counted on to work all the time. Thus, NIDS events that trigger countermeasures still need to be investigated as if no countermeasure was used. The bottom line is they supplement a good security posture, and when properly configured, should be deployed whenever possible.

References

1 Chuvakin, Anton. "Intrusion Detection Response." Linuxsecurity.com. URL: http://www.linuxsecurity.com/feature_stories/ids-active-response-printer.html (22 April 2002)

2 Convery, Sean and Bernie Trudel. "SAFE: A Security Blueprint for Enterprise Networks." Cisco Systems. URL: http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safe_wp.htm (2000)

3 Cummings, Joanne. "From Intrusion Detection to Intrusion Prevention." Network World Fusion. URL: http://www.nwfusion.com/buzz/2002/intruder.html (23 September 2002)

4 Hawrylkiw, Dan. "Network Intrusion and use of automated responses." SANS. URL: http://www.sans.org/resources/idfaq/auto_res.php

5 Hayes, Keith. "Active Security Monitoring and Containment with Cross Technology Correlation: The Next Generation in Computer Security Technology." SecurityFocus. URL: http://www.securityfocus.com/guest/10414 (11 February 2002)

6 Larsen, Jason, and Jed Haile. "Understanding IDS Active Response Mechanisms." SecurityFocus. URL: http://www.securityfocus.com/infocus/1540 (29 January 2002)

7 Paul, Brooke. "Building an In-Depth Defense." Network Computing. URL: http://www.networkcomputing.com/1214/1214ws1.html (9 July 2001)

8 Securitywatch.com. "Securitywatch Glossary." URL: http://www.securitywatch.com/EDU/glossary.html (2001)

9 Turner, Aaron and Chris Winter. "IDS: IPS and IDS (was RE: Changes in IDS Companies?)." IDS Focus Security List. URL: http://lists.insecure.org/lists/focus-ids/2002/Oct/0076.html (29 October 2002)

10 Walder, Bob. "Gigabit IDS." SC Magazine. URL: http://www.scmagazine.com/scmagazine/2003_01/test_nss/index.html (January 2003)

Last Updated: July 24th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor