Global Information Assurance Certification Paper

Limitations of Network Intrusion Detection

Steve Schupp

December 1, 2000

The use of Network Intrusion Detection System (NIDS) is becoming more pervasive as site security moves beyond the passive firewalling that has become fairly commonplace to integrated security environments that actively monitor and possibly respond to threats. It is important to understand the limitations of NIDS in order to ensure that NIDS is used in an appropriate manner.

'If the only tool you have is a hammer, everything looks like a nail'; securing a site with an IDS alone would present you with a lot of nails! Integrating an IDS into a network that has well thought out security can greatly enhance the security of a network. This paper looks at the limitations of Network IDS, and outlines the tactics the scanning tool Whisker uses to evade detection.

Background

In general NIDS are configured with a pattern file of known attack signatures. This pattern file is analogous to a Virus Scanner's pattern file. The NIDS compares network traffic to the signatures in the pattern file. If a match is made, then the NIDS can take the appropriate action as configured by the administrator.

A policy is configured by a Security Administrator which defines the site specific configuration for the NIDS. The policy should define what is considered acceptable network traffic, and what is deemed to be unacceptable. The policy will also determine what response the NIDS should take when it encounters various attack signatures.

The response types which most NIDS employ include various forms of notification (Email, Pager, SNMP trap), the ability to reset sessions or record sessions, and the ability to integrate into other security products, for example, to dynamically update firewall rules.

There are two main forms of NIDS which are common in commercial products which are in use today. The first is the 'Raw' pattern matching NIDS which are designed to do a comparison to the packets they capture and match attacks based on the data captured. This style of NIDS can be considered a 'packet grep' NIDS, examples being Snort or Dragon. Alternatively, a 'Smart' NIDS can interpret the packet, and attempt to understand the protocol that is being captured in order to identify. ISS RealSecure is an example of a Smart NIDS.

Network IDS Limitations

Switched Network

In order to get the benefit of NIDS it is necessary for the IDS to have the greatest visibility possible. The function of a switched network is to reduce network traffic by virtually connecting 2 communicating stations. If a NIDS is placed onto a switched port of a network, it may only be able to see traffic directed to it. The NIDS is only able to match attack signatures in the traffic it is able to capture, and so obviously a switched network is not an optimal topology for maximum visibility. As such, NIDS are generally located on a network choke point, such as the same segment as a Firewall.

As modern networks are switched, various techniques have been employed to ensure the NIDS has high visibility whilst maintaining the performance and efficiency of a switched environment. Some of these techniques include:

- Embed IDS within the switch: these have limited detection capability due to limited operating environments and processing power of the host switch.
- Monitor/span port: Suffer from packet loss as the monitor port runs at a much lower speed than the ability of the switch to move packets. Packet loss reduces visibility and accuracy of NIDS attack detection.
- Tap into the cable: an NIDS may be connected directly into a cable, however there still is an issue with packet loss at high speeds.
- Host-based sensors: Host based sensors can be combined with NIDS to provide greater coverage. Host

based IDS can monitor other aspects of hosts such as log files and processes, and can guard against keyboard attacks that will not be detected by an NIDS.

Resource Limitations

Network IDS can suffer from issues with Resource limitations. In order to detect attack signatures NIDS must capture, store, and analyse large volumes of data, in near realtime. The volume of data passing over network choke points can be staggering, with the IDS having to maintain connection information for potentially thousands of machines. The following lists some of the resource limitations that may reduce the ability of a NIDS to detect attacks:

- Network traffic loads: sheer packets per seconds can reduce the NIDS ability to keep up. Smart NIDS will tend to fall back to Raw detection as link utilization increases.
- TCP connections: In order to detect a wide range of attack techniques, an NIDS must maintain state for a large number of TCP connections. This requires a large amount of memory on the NIDS host.
- Other state information: NIDS may also be required to track IP Fragments, ARP packets, and other related information.
- Long term state: In order to detect 'Slow Scans' it is necessary for an IDS to maintain state information over a long period of time. Storing more history requires more memory, and so there is a trade off between detection and performance.

Attacks against the NIDS

The NIDS may become the target of an attack itself. An attacker may utilize techniques to reduce the ability of the NIDS to detect an attack in order to allow the attacker to slip their traffic though undetected.

- Blind the sensor: by taking advantage of the fact that a NIDS will drop packets on heavily loaded links, an attacker may attempt to flood a link whilst trying to attack a particular asset.
- Blind the event storage (snow blind): By simulating a large number of simultaneous attacks, the Administrator will have a hard job when attempting to determine which attack was 'the real one'. Tools such as Nmap have options to run Decoy scans to ensure it is difficult to determine the real source of a scan.
- DoS (Denial of Service): Due to the nature of NIDS systems, and the need for them to analyse protocols as they are captured, NIDS systems can be susceptible to same protocol based attacks that network hosts may be vulnerable. Invalid data and TCP/IP stack attacks may cause an NIDS to crash.

Evasion

Attackers are experimenting with and finding ways to evade NIDS. By evading NIDS the attacker can slip 'under the radar' to probe systems which may be monitored by a NIDS without detection.

There are a number of techniques which attackers are using, the following are considered 'simple' measures which can be taken to evade NIDS:

- Fragmentation: by sending fragmented packets, the attacker may be able to evade the NIDS by breaking up the payload into smaller chunks and possibly reducing the ability for the NIDS to detect the attack signature.
- Avoiding defaults: The TCP port utilised by a protocol does not always provide an indication to the protocol which is being transported. For example, an IDS may expect to detect NetBus (a trojan similar to BackOriface) on port 12345. If an attacker had reconfigured NetBus to use a different port, maybe 53, the IDS may not be able to detect the presence of NetBus. Attackers can use non-default ports to evade IDS.
- Slow scans: by scanning a network slowly, say one port per hour, the attacker may be able to evade the IDS by ensuring the scan is beyond the 'memory' of the IDS, so the scan is not recognised.
- Coordinated, low-bandwidth attacks: coordinating a scan among numerous attackers (or agents) and allocating different ports or hosts to different attackers makes it difficult for the NIDS to correlate the captured packets and deduce that a network scan is in progress.
- Address spoofing/proxying: attackers can increase the difficulty of the ability of Security Administrators to determine the source of the attack by using poorly secured or incorrectly configured proxy servers to bounce an attack. This way the source of the attack appears to be an unsuspecting victim instead of the

attackers' source.
- Pattern change evasion: NIDS generally rely on 'pattern matching' to detect an attack. By changing the data used in the attack slightly, it may be possible to evade detection. For example, an IMAP server may be vulnerable to a buffer overflow, and an NIDS is able to detect the attack signature of 10 common attack tools. By modifying the payload sent by the tool, so that it does not resemble the data that the NIDS expects, it may be possible to evade detection.

Complex evasion techniques concentrate on exploiting TCP/IP attacks. One attack method documented by Thomas H. Ptacek and Timothy N. Newsham utilised the following technique:

1. Establish a TCP/IP connection with the target host
2. Send packets to close the connection, ensure the NIDS sees the TCP/IP close, but ensure the target host does not receive the close. This can be achieved by using a DOS attack to a router beyond the NIDS, but before the target host or by modifying packet TTL so the close does not reach the target host.
3. At this stage the NIDS will remove the session from its state table.
4. The target host will maintain an open session for a long period of time, and so it is possible for the attacker to resume the connection and continue an attack hours after the session was opened, and possibly evade the detection of the NIDS.

There is other Complex Evasion techniques that involve modification of the TCP session synchronization and are aimed at confusing the NIDS so that it cannot successfully track the state of the session.

Whisker: A tool which implements Evasion Techniques

Rain Forrest Puppy wrote a tool called Whisker whose primary focus is Web and CGI vulnerability scanning. Whisker evades IDS systems by slightly modifying the http request. As most IDS systems are expecting to pattern match particular requests to indicate an attack, modifying the request may allow the attacker to scan without detection.

Whisker utilises the following techniques to evade detection:

- Method Matching: Whisker does not use the http GET method, but instead the http HEAD method. The HEAD method allows the attacker to determine the existence, for example, of a vulnerable CGI script or file on the Web Server. If an IDS does not detect the HEAD method, the scan may go unnoticed.
- URL Encoding: Early versions of Whisker substituted components a URL with their hexadecimal equivalent. Most recent NIDS implementation will decode the hex and will detect attacks that are obfuscated by this method. However 'Raw' NIDS systems may not detect this.
- Reverse Traversal: this involves inserting a directory in the path of a URL, so that the request for "/cgi-bin/phf.cgi" is changed to "/cgi-bin/subdir/../phf.cgi". Once again, Smart NIDS should detect this attack, and Raw NIDS should at least detect the "/../" component in the URL.
- Self-referencing Directories: a similar tactic to the above directory modification, which uses the current directory instead of a subdirectory. The path will be modified to '/cgi-bin/./././.phf.cgi'. Once again, modern NIDS systems should detect this style of attack.
- Premature Request Ending: by encoding an end of request, and following with another request in the same transaction, it may be possible to evade the NIDS. The first request will be a valid request, followed by the attack request. In order to maintain performance, some IDS systems will only decode the first request, and so it may be possible for an attacker to evade detection.
- Parameter Hiding: Most NIDS will stop processing a URL when a '?' is reached, as the IDS is not interested in scanning the parameters provided to a script. In a similar method to Premature Request Ending, it is possible to encode a request which will evade the NIDS. The URL '*GET /index.htm%3fparam=/../cgi-bin/some.cgi HTTP/1.0'* where '%3' = '?', is a valid request, and the attempt to access the some.cgi script may not be detected by an IDS.
- HTTP Mis-Formatting: Some web servers will allow mis-formatted HTTP requests. If an NIDS is basing its attack pattern matching on a standard RFC HTTP request, an attack that utilises a mis-formatted HTTP request may go undetected.
- Long URLs: in order to gain performance, an NIDS may only capture a portion of each packet. If a request has a large amount of padding the NIDS may not detect the attack simply because it does not capture the relevant part of the packet.
- DOS/Win directory syntax: MicroSoft operating systems use the backslash '\' to separate directories.

This means that MS based web servers such as IIS must translate the http '/' seperator to the Windows '\'. IIS however, will also accept a '\', and so an NIDS may not correctly match an attack which utilised '\' instead of the '/'.

- Case sensitivity: By mixing the case in the URL request, it may be possible to avoid detection by obscuring the attack pattern matching. Also, as some operating systems are case sensitive, a request for '/CGI-BIN/THIS.CGI' may be interpreted differently to '/cgi-bin/this.cgi' at the operating system, and may be interpreted differently by the IDS.
- Session splicing: This is similar to the Fragment Attack discussed earlier, but instead of fragmenting the packet the payload is spread across multiple individual packets. If and NIDS only detects an attack pattern in the current packet, and does not monitor the protocol, the attack may go undetected.

Conclusion

Network IDS should be incorporated into a security infrastructure. It can be seen from this report that NIDS is by no means a silver bullet to detecting attacks. Network managers should not be complacent where NIDS are installed. As the use of NIDS increases, attackers will become more sophisticated in the methods and tools that they use to evade or disable the NIDS.

NIDS should be considered an important backup system to an existing, solid, 'defense in depth' network security architecture. NIDS can provide the proactive monitoring to ensure that, for example, in the event of misconfiguration of a firewall, there is a system which may detect the event and alert a security administrator to allow the appropriate correction or investigation to take place.

References

Robert Graham, "Network Intrusion Detection System Frequently Asked Questions (FAQ)". Version 0.8.3, March 21, 2000 http://www.robertgraham.com/pubs/network-intrusion-detection.html (28 Nov. 2000)

Rain Forrest Puppy, "A look at whisker's anti-IDS tactics" 1.3 (12/24/99). http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html (28 Nov. 2000)

Internet Security Systems, "Network vs Host-Based Intrusion Detection" (2 Oct. 1998) http://documents.iss.net/whitepapers/nvh_ids.pdf (28 Nov. 2000)

Greg Shipley, "Intrusion Detection – Take Two" (15 Nov. 1999) http://www.networkcomputing.com/1023/1023f1.html (28 Nov. 2000)

Coretez Giovanni , "Passive Mapping: An Offensive Use of IDS" (11, Apr. 2000) http://www.securityfocus.com/templates/forum_message.html?forum=2&head=3818&id=1313 (28 Nov. 2000)