Global Information Assurance Certification Paper

Anti-IDS Tools and Tactics

By Steve Martin

SANS Security Essentials GSEC Practical Assignment V1.2e

1 Introduction

Over the past twenty years the World Wide Web has grown from a network used purely for the exchange of academic information to the mainstream medium we now use for communication, education, business and a plethora of other uses.

This growing reliance on the Internet has forced us to look closely at the lack of security surrounding the interaction we have with the web, and as exposure and risks increase organizations constantly seek to improve their security stance. The latest addition to the range of security technology that can be deployed is the Intrusion Detection System (IDS).

IDSs can be installed on a host to monitor system level activity, on a dedicated PC to monitor network traffic, or there is a hybrid version that combines the host based with network traffic analysis to provide greater intelligence. Regardless of the variant chosen, it is a valuable tool in the armour of the discerning Security Manager wishing to add further depth to his defence strategy.

This paper focuses purely on Network ID Systems, and discusses the technical detail behind techniques that can be employed to counteract the utility of these systems and identifies tools that actually use the techniques described.

2 The Technology

The market for Network ID Systems has grown rapidly over the last few years, but despite a range of products from different vendors there are still only two main types of Network ID Systems.

2.1 Raw Analysis Systems

Raw Analysis Systems capture frames of data from the network and compare these to strings from a database of attack signatures looking for a match. This process is known as 'signature analysis'.

These systems do not attempt to perform any processing on the data captured, they simply scan the raw data looking for sequences of characters that signal a potential attack. Due to the limited amount of work these systems commit themselves to, they scale reasonably well in network segments with high bandwidth utilization.

2.2 Pseudo Intelligent Systems

Pseudo intelligent systems also capture the raw frames of network traffic, but they also understand the protocols seen and the rules that govern their operation. When they capture traffic from the wire they try to emulate the host and application based on the protocol traffic flow they detect.

This modus operandi provides an engine that reduces the number of false positives reported and enables the detection of more complex attacks. However, pseudo intelligent systems may not scale to meet the demands of network segments with high bandwidth utilization due to the higher level of processing required.

3 Problems with Network ID Systems

All Network ID Systems, regardless of the detection technique(s) they employ have a number of weaknesses:

3.1 False Positives

Pattern matching within network traffic payloads is not a precise science. The same exploit can take many slight variations in form, and this means the pattern-matching algorithm has to be flexible and able to cater for these variations. The side effect of this flexibility or over cautious approach is the false alarm or 'false positive' alert. False positives are the bane of security administrators accounting for the vast majority of alerts witnessed.

3.2 False Negatives

A 'false negative' situation is the opposite of the false positive and occurs when the pattern-matching algorithm fails to detect an attack because the string being looked for is too specific, or, in the case of Raw Analysis systems, the signature database does not include the latest exploit.

3.3 Scalability

Raw analysis systems are significantly more efficient in dealing with high volumes of traffic, but still have difficulty scaling to cope with 100Mbps full duplex networks, and fall far short of being able to scale up to Gigabit Ethernet speeds.

4 The Ploys

There are a number of techniques that can be used to reduce the effectiveness of a Network ID Systems.

I have categorized the ploys under the broad headings of evasion, insertion and denial of service. Terms first used by Thomas H Ptacek and Timothy N Newsham in their paper Insertion, Evasion, and Denial of Service, Eluding Network Intrusion Detection. Jan 1998.

Additionally, I have also described each ploy using, where possible, titles first used by Rain Forest Puppy in his landmark paper, a look at Whisker's anti-IDS tactics.

4.1 Denial of Service

Denial of service techniques are designed to do one of two things:

1. Render the Network ID System ineffective by giving it too much work to do.

   If you bombard a Network ID System with false activity you can potentially

slip a genuine attack past unnoticed. The flooding traffic can be either directed at the Network ID System itself, or at a genuine host from source addresses from every possible, and impossible, address on the Internet.

If directed at the host the packet would need to be crafted such that the host would reject the packet once validated, but the Network ID System would still process it. An example of such a crafted packet would be an invalid IP header checksum.

Imagine trying to analyze 60,000 attacks, which one if any is genuine? How many people are available to assess all the alerts? This denial of service attack is more against the human processes that support intrusion detection systems than the systems themselves.

2. Preventing the Network ID System from performing any analysis.

   Network ID Systems are either installed on standard operating systems such as NT, Linux or on 'appliance devices', which run customized operating systems, tuned to solely run the application. Despite these systems normally being 'hardened' against vulnerabilities, no system is 100% secure and it may be possible to crash the system by sending invalid data to the TCP/IP stack rendering it inoperative.

Until recently, the above scenarios were theoretical weaknesses that could potentially be exploited. Recently however, tools have been developed that can be used to create both of the above conditions.
These tools have been tested by vendors for effectiveness on their commonly deployed Network ID Systems and have been seen to regularly cause the systems to fail within seconds of an attack being launched.

4.2 Insertion

A packet can be carefully constructed so that an IDS will accept it for processing, but ignored by the target host.

4.2.1 URL Encoding

This is the name given to the technique of substituting the characters within a URL with their hexadecimal equivalent.
The following example shows a normal request string and the encoded alternative:

```
/cgi-bin/test.cgi
```

becomes…

```
/cg%69-b%69n/t%65st.cg%69
```

The Pseudo Intelligent Network ID Systems will not be fooled by this ploy as they will parse the request before attempting a string match. In theory a raw analysis system is likely to be susceptible to the ploy though as no parsing is performed, and therefore a string match will not be successful.

This technique is well documented and implemented in most tools, and as such is unlikely to fool either IDS type as vendors have implemented string parsing as a minimum.

4.2.2 Reverse Traversal

Another ploy is to attempt to confuse the Network ID system by complicating the request with additional directory references. The request will still resolve correctly on the target host but the Network ID system will discard the packet. For example:

```
/cgi-bin/test-cgi
```

becomes…

```
/cgi-bin/redherring/../test.cgi
```

As this ploy is quite old both Pseudo Intelligent and Raw Analysis systems confidently detect the unusual '../' element of the string and will alert accordingly.

4.2.3 Self-Referencing Directories

A similar technique to reverse traversal is that of self-referencing directories.

Inserting './' into any request will have no affect as it means 'current directory'. E.g.

```
/cgi-bin/test-cgi
```

becomes…

```
/./cgi-bin/./test.cgi
```

Although a newer ploy than reverse traversal it has been used for long enough for the Network ID system manufacturers to recognize the tactic and alert accordingly.

4.2.4 Parameter Hiding

A request can contain additional information that is used to build dynamic page content, this additional information is known as parameters.
Parameters are typically used when search requests or selections are made and take this form:

```
/anypage.php?attack=paramhiding&evasion=blackhat&success...
```

The parameters are specified after the '?', and a Pseudo Intelligent ID System will probably ignore any data after the ? to improve processing performance. However, the parameter indicator can be used to potentially mask further relevant data.

```
GET /index.htm%3fparam=/../cgi-bin/test.cgi
```

becomes…

```
GET /index.htm?param=/../cgi-bin/test.cgi
```

4.2.5 Long URLs

Even Raw Analysis Network ID Systems have sampling techniques that are designed to improve performance. One such technique is to limit the amount of data sampled from each frame.

Obviously a frame that is over a particular length may only have a portion of it read and analysed. If a packet is crafted with padding to this length then any malicious content will get through unchecked.

4.2.6 Multiple slashes

It is possible to send a request to a web server that substitutes single slashes with multiple slashes and for the web server will still interpret the request correctly. The hope for the attacker is that the Network ID System will fail to successfully match the attack string.

```
/cgi-bin/test.cgi
```

becomes…

```
//cgi-bin//test.cgi
```
or…
```
///cgi-bin///test.cgi
```
etc.

Pseudo Intelligent systems would never be fooled by this tactic due to their parsing of the request before attempting a string match. Early Raw Analysis systems were fooled, but software developers have now responded by always combining slashes when they detect multiple instances.

4.3 Evasion

Carefully constructed packets can be accepted by the end system, but ignored by a Network ID System.

4.3.1 Slow Scans

Network ID Systems detect network-scanning activity by monitoring the frequency of traffic from a given IP address. If a scanning tool can artificially spread the scanning activity over a prolonged period of time the Network ID System may not detect the activity.

4.3.2 Method Matching

It is quite legitimate within the HTTP RFC to send an alternative method to GET which was originally the only method available.

Alternative methods are of use to an attacker as they allow for the detection of the presence on a web server of useful CGI scripts or files that have known weaknesses or vulnerabilities.

The alternative methods are:

```
HEAD
POST
PUT
```

```
DELETE
PATCH
PROPFIND
PROPPATCH
MKCOL
COPY
MOVE
LOCK
UNLOCK
```

4.3.3 Premature Request Ending

A technique designed to fool Pseudo Intelligent Network ID Systems is to insert an ending request prior to the genuine end of the request and more importantly prior to malicious data.

A typical genuine request would look like this:

```
GET /some.file HTTP/1.0\r\n
Header: blah \r\n
Header: blah \r\n
Header: blah \r\n
Header: blah \r\n
\r\n
```

A Pseudo Intelligent ID System will not generally scan the headers in a request as there is little point. Therefore the IDS can stop looking after the "HTTP/1.0\r\n". However, the following example shows the danger in this method:

```
GET /%20HTTP/1.0%0d%0aHeader:%20/../../cgi-bin/test.cgi HTTP/1.0\r\n\r\n
```

becomes…

```
GET / HTTP/1.0\r\nHeader: /../../cgi-bin/test.cgi HTTP/1.0\r\n\r\n
```

The above is a valid request to a web server but will be missed by a Pseudo Intelligent ID System that parses the request.

4.3.4 HTTP Mis-Formating

Although there is a clearly defined structure to any HTTP request, many web servers will accept a request that does not conform exactly to this specification. A request that conforms exactly to the rfc takes the form of:

```
Method <space> URI <space> HTTP/ Version CRLF CRLF
```

Some web servers will allow an alternative separator to be used, for example:

```
Method <tab> URI <tab> HTTP/ Version CRLF CRLF
```

Any IDS analysis dependant on the 'assumed' RFC format of a request will now fail.

4.3.5 DOS directory syntax

When Bill Gates wrote DOS he decided to buck the Unix trend and use the '\' character as the directory separator rather than the '/'. As a result, DOS based web

servers have to transparently translate the forward slashes in requests into back slashes.

The following is an example request and the subsequent translation by a DOS based webserver:

```
/pages/login/password.lst
```

becomes…

```
/pages\login\password.lst
```

Notice that the initial slash is still a forward slash. This is in order to comply with the HTTP RFC.

4.3.6 Case sensitivity

Another difference between DOS and Unix systems is their interpretation of case.

In Unix the following are all different files:

```
password
PASSWORD
PassWord
```

A DOS based system would interpret each of the above strings as the same file.

By forcing a request into all upper case the request will still be interpreted correctly by a DOS based webserver, but may be missed by a Unix based Network ID system attempting a pattern match. Pseudo Intelligent ID systems may still be vulnerable to this tactic.

4.3.7 Fragmentation

Fragmentation is the method by which the TCP/IP protocol handles the problem of traversing Wide Area connections of varying bandwidth or Maximum Transmissible Unit (MTU) capability. TCP being connection oriented allows for various scenarios for the delivery of data, such as packets arriving out-of-sequence or duplicates within a data stream.

Network ID Systems, both Raw Analysis and Pseudo Intelligent, do not implement fragment reassembly due to the severe overheads this approach would create. Therefore, any packet containing malicious content that has been fragmented will pass unnoticed.

Not implementing fragment reassembly means that Network ID Systems are also totally oblivious to the more complex techniques that are available through the use of easily obtainable tools, such as forward overlap in IP fragments that will successfully by-pass a firewall. i.e. a stream of fragments contains the string HEAD, a forward-overlapping fragment can then be sent to rewrite the HEAD string back to GET on the target host.

4.1.8 Session Splicing

Session splicing is different to fragmentation as described above as it concerns sending just the HTTP payload of the data in chunks with the sole purpose of preventing a Raw Analysis Network ID System from successfully detecting a string match.

Reassembly is possible, potentially being implemented by a Pseudo Intelligent ID System, but is unlikely due to the severe processing overhead that would be incurred.

4.1.9 NULL Method Processing

This technique relies on the fact that C string libraries use the NULL character to denote the end of a string.

Pseudo Intelligent ID Systems will parse a request and as such will interpret the request incorrectly potentially ignoring the malicious payload.

```
GET%00 /cgi-bin/test.cgi HTTP/1.0
```

5 Combining The Ploys

Generally most of the tools listed in section 6 allow the user to combine multiple tactics together to create hybrid ploys, for instance URL encoding, Self Referencing directories and Parameter hiding can all be combined.

For example:

```
/cgi-bin/test.cgi
```

becomes…

```
/index.html%3Fparam=/.././cg%69-b%69n/./t%65st.cg%69
```

6 The Tools

Name:	Stick (not yet released)
Author:	Coretez Giovanni, coretez@8thport.com
Relevant URL:	http://www.eurocompton.net/stick
Capability:	IDS DOS Utility that uses the Snort rule set and produces C code that when compiled is capable of triggering that rule from a spoofed IP range. The tool can produce around 250 alarms per second.

Name:	Mendax v.0.7.1
Author:	Min G Kang
Relevant URL:	http://www.securityfocus.com/tools/2096
Capability:	TCP de-synchronizer that can be used to inject overlapping segments in a random order. It can inject an attack signature or single typed lines.

Name:	Snot 0.91
Author:	Sniph sniph00@yahoo.com
Relevant URL:	ftp://ftp.st.ryukoku.ac.jp/pub/security/tool/snot
Capability:	IDS DOS Utility that generates floods of packets using Snort rules as the basis for its packet information.

Name:	Sidestep
Author:	Robert Graham
Relevant URL:	http://www.robertgraham.com/tmp/sidestep.exe
Capability:	Windows based (CLI) scanner that incorporates anti-ids

Name:	TWWWscan v1.2
Author:	pilot
Relevant URL:	http://search.iland.co.kr
Capability:	A Windows based www vulnerability scanner which features anti-IDS url encoding and passive mode scan.

Name:	Babelweb v1.0
Author:	Stephane Aubert Stephane.Aubert@hsc.fr
Relevant URL:	http://www.hsc-labs.com/tools/babelweb/
Capability:	Automated tester of HTTP servers that incorporates anti-IDS tactics

Name:	fragrouter
Author:	Unknown
Relevant URL:	http://www.monkey.org/~dugsong/
Capability:	Fragrouter is a program for fragmenting network traffic in such a way as to elude most network intrusion detection systems, and implements all the fragmentation attacks as outlined in the Ptacek and Newsham paper (see references)

Name:	fscan 1.
Author:	f0bic
Relevant URL:	www.low-level.net/code.php
Capability:	An anti-IDS CGI scanner using uri encoding, obfuscation, and various other scanning methods.

Name:	infinity-t-3
Author:	Azrael tempazrael@hotmail.com
Relevant URL:	http://infinityproject.cjb.net
Capability:	Perl based scanner that incorporates anti-ids hex http querys

Name:	whisker 1.4
Author:	Rain Forest Puppy rfp@wiretrip.net
Relevant URL:	http://www.wiretrip.net
Capability:	CGI scanner with anti-IDS capability built in at the design stage. rfp also publishes separately the perl module used by

whisker under the name libwhisker

Name:	Mutate 2
Author:	Efrain 'E' Torres et@cyberspace.org
Relevant URL:	http://www.securityfocus.com/archive/96/189704
Capability:	CGI scanner with enhanced anti-IDS tactics that build upon the original ploys within rfp's whisker tool.

Name:	Malice 6.1
Author:	Unknown
Relevant URL:	http://nullbyte.extremenetworking.net
Capability:	CGI scanner that implements some anti-IDS tactics as per whisker.

Name:	CUM Security Toolkit [CST] v1.3
Author:	toxic ocean toxic@blackhat.be
Relevant URL:	http://www.securityfocus.com/tools/1799
Capability:	CGI Scanner with anti-IDS

Not a tool as such, but Xtremist has written a paper on the writing of anti-IDS shell code, which demonstrates the interest within the hacker community on tools with these capabilities.

Author:	xtremist@2xs.co.il
Relevant URL:	http://hackersclub.com/km/library/hack2001/stealthcode.txt

7 Conclusion

Network IDS are no silver bullet solving the security dilemma; Network IDSs in fact have some significant design flaws that drastically reduce their utility.

There is no doubt that Network ID Systems should be incorporated into a security infrastructure, however Security Managers should continue to add layers to their defence strategy and not place too much reliance on this technology.

A large amount of research is still being undertaken to develop the next-generation of Network ID Systems, with the goal being a system that can effectively flag an attack without crashing under the weight of its own logs, operate relatively maintenance-free, and respond appropriately to benign anomalous events without raising too many false alarms.

8 References

Hakan Kvarnstrom, A Survey of commercial tools for intrusion detection
http://www.ce.chalmers.se/staff/hkv/IDS_Survey-99.pdf

Coretez Giovanni, Passive Mapping: An offensive use of IDS
http://www.eurocompton.net/stick/papers/OffensiveUseofIDS.pdf

Fred Cohen, 50 ways to defeat your Intrusion Detection system
http://all.net/

Rain Forest Puppy, "A look at whisker's anti-IDS tactics"
http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html (28/11/00)

T. Ptacek and T. Newsham, Secure Networks Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection, January 1998.
http://citeseer.nj.nec.com/ptacek98insertion.html