Global Information Assurance Certification Paper

Snort – Free Graphical IDS for the Windows Environment

Kenneth Rode
Version 1.2b

Introduction

The goal of this paper is not only to provide a tutorial on the use of Snort in a Windows environment but also to examine the growing need for Intrusion Detection systems independent of network size. For analysis and review of the Unix version of Snort please refer to the two other SANS papers listed in the references.

Overview

Use of any computer system entails accepting risk. The goal of security professionals is to implement products and processes which can mitigate these risks while not limiting access to essential services. Achieving this goal requires multiple security tools operating in concert. This concept is commonly referred to as defense-in-depth.

This in-depth approach to security provides multiple layers of isolation and control to a networked system. However, when these systems are subverted or bypassed, it is critical to be alerted. As noted in a Compaq white paper "Should the isolation measures fail, and an event occur that could damage the system, a prudent security policy provides the means to detect the failure and report the event to an assigned person for resolution"[1]. Unfortunately, due to a general lack of awareness or concerns about manageability and cost, the detection portion of many security policies is frequently omitted.

One of the major misconceptions about Intrusion detection is that it does not add to the general protections offered through proper configuration of systems and the isolation offered by firewalls. Robert Graham puts it best in his FAQ where he says,

> A firewall is simply a fence around your network, with a couple of well-chosen gates. A fence has no capability of detecting somebody trying to break in (such as digging a hole underneath it), nor does a fence know if somebody coming through the gate is allowed in. It simply restricts access to the designated points.
>
> In summary, a firewall is not the dynamic defensive system that users imagine it to be. In contrast, an IDS *is* much more of that dynamic system. An IDS *does* recognize attacks against the network that firewalls are unable to see.[2]

Further, an actively monitored IDS can improve the general safety of the Internet through distributed identification and elimination of unwanted activity. Since I began monitoring Internet traffic with Snort, I have sent an average of 5 emails a day informing address owners of port scans or other intrusion attempts from their networks. Responses to these emails have ranged from nothing to expressions of thanks for uncovering compromised systems within an organization. Typically, I am informed either of an exposed system that has been hardened or an Internet account that has been disconnected. Either way, an access point has been closed. If every node on the Internet examined local traffic for questionable activity and every router filtered local IP addresses by port, how many of the current "hackers" would be able to remain active?

Another use of an IDS system is monitoring and reporting on internal network traffic. From all accounts, the biggest danger to any corporate network is its own users and typically firewalls are not in

place to control a significant portion of this activity. Further, attacks originating from your network can result in legal difficulties or loss of Internet connectivity.

The last argument to address is that of cost. Historically, inexpensive or free network tools have been almost exclusively limited to Unix or Linux systems. Consequently, any cash-strapped Windows shop or home hobbyist was generally unable to readily apply layers of security. Fortunately, this is no longer the case. With the advent of WinPcap, WinDump and even a recent port of NMap to Windows NT, this is no longer a viable excuse. Further, Snort add-ons are available to provide GUI interfaces to the entire operation.

Selecting and Locating the IDS Sensor

Anyone considering the use of Snort must be cognizant of its Lightweight nature. To quote Martin Roesch "Snort is a tool for small, lightly utilized networks. Snort is useful when it is not cost efficient to deploy commercial NIDS sensors."[3]. For enterprise environments there are high-end applications such as Real Secure from ISS (http://www.realsecure.com) that would be more suitable.

The system selected to be the sensor does not need cutting-edge performance but should provide enough memory, disk space and CPU power to process the information collected without dropping packets. Further, if the sensor will be placed in an exposed location, two network cards are advisable. In that way, the "sniffer" interface can be configured with an inaccessible IP address. Since it will be set to promiscuous mode and collecting all traffic regardless of destination, any address will work. I prefer the use of a private IP address not used elsewhere within the network with a subnet mask of 255.255.255.255. As the internal card must be accessible, it is still critical to secure the system but this configuration will help hide the sensor from external eyes.

When selecting the location for a sensor, it is again important to remember the lightweight nature of Snort. Referring back to the Robert Graham FAQ,

> "IDS is most effective on the network perimeter, such as on both sides of the **firewall**, near the **dial-up** server, and on links to **partner** networks. These links tend to be low-bandwidth (T1 speeds) such that an IDS can keep up with the traffic."[2]

The final placement issue is to note that Snort will only be able to inspect and record network traffic that it can "see". Consequently, locating a sensor within a switched environment requires planning and extensive experience with your network hardware.

Installing and Running Snort for Windows

Obtaining and installing Snort is a fairly straightforward process. The first item required is a packet capture driver for the Windows environment. This tool, called WinPcap, is available from the Computer Network and Network Intelligence Group of Politecnico di Torino. The downloaded file is a self-extracting executable. Simply running the file will lead you through the installation process.

Once the packet-capture driver is in place, Snort for windows may be downloaded from http://www.snort.org. This file is received in Zip format and must be uncompressed for use. One reasonably priced unzip utility is Winzip from Niko Mak computing (http://www.winzip.com). Simply unzip the file to a local disk to complete the installation.

The final installation tasks are to customize the snort configuration file and create the log directory. The configuration file (snort.conf) is a simple text file and well annotated. The two key items to verify are the Home_Net and DNS_Server variables. Once these are set, create a subfolder within your snort directory called log and you are ready to begin capturing data.

Detailed explanations for each Snort command line option are available within the readme file from the distribution. However, the majority of users will simply need to use the line

snort –A full –c snort.conf –D

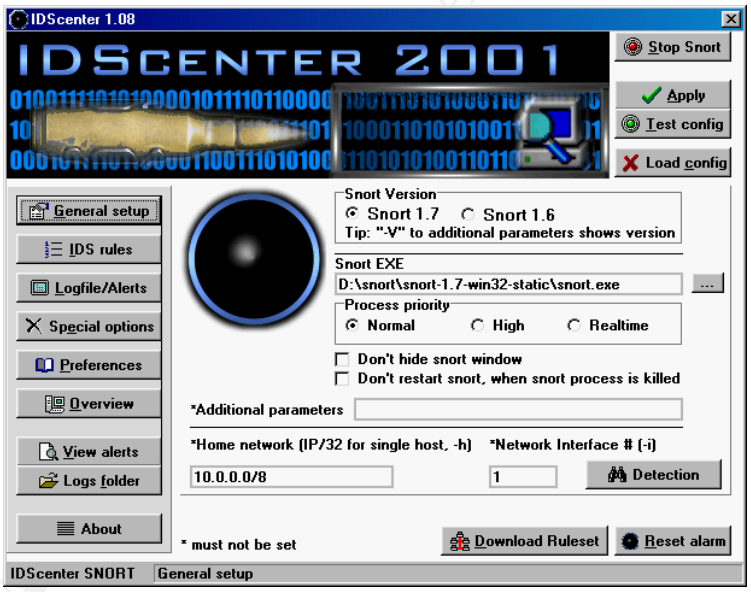
This instructs the application to provide full alerts, use the snort.conf file for configuration information and run as a Daemon. All alerts will be written to alert.ids within the log folder created previously.

Running Snort under the above scenario is very straightforward and will work reliably. Unfortunately, it does not include provisions for active notification of alerts nor easy methods of starting/stopping or reconfiguring the service. However, thanks to Ueli Kistler, and alternative is available. IDSCenter provides a graphical front-end which provides all the noted services. As noted by Whitehats.com;

> IDScenter is a panel for controlling, managing and auditing SNORT IDS for WIN32. It supports all the functions of snort.panel by XATO. New Features: IP/Interface detection, Alarm sound (WAV/Beep), Implemented log viewer, EXE-File start on alert...[4]

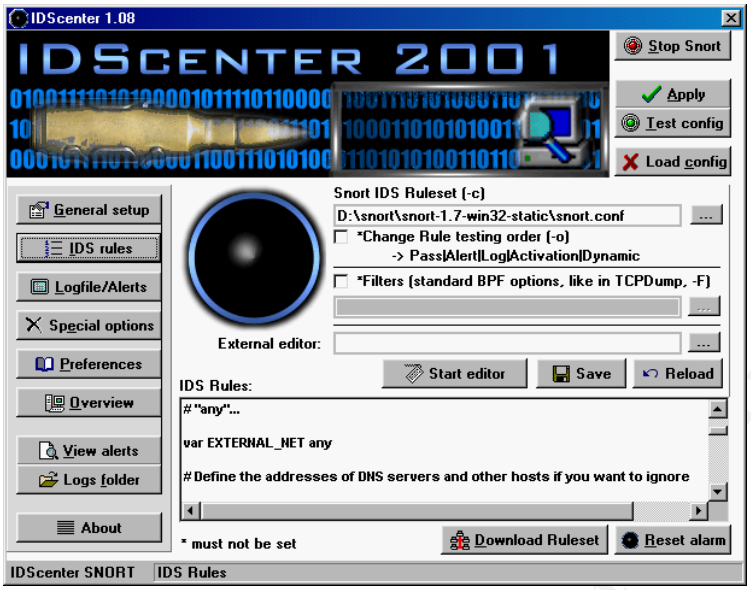
To install IDScenter, download the zipped file from http://www.eclipse.fr.fm/snort.htm (also available from http://www.whitehats.com and http://www.snort.org) unzip it and run the executable.

After installation, starting the program will bring you to the general setup window shown below. Basic configuration instructions follow.

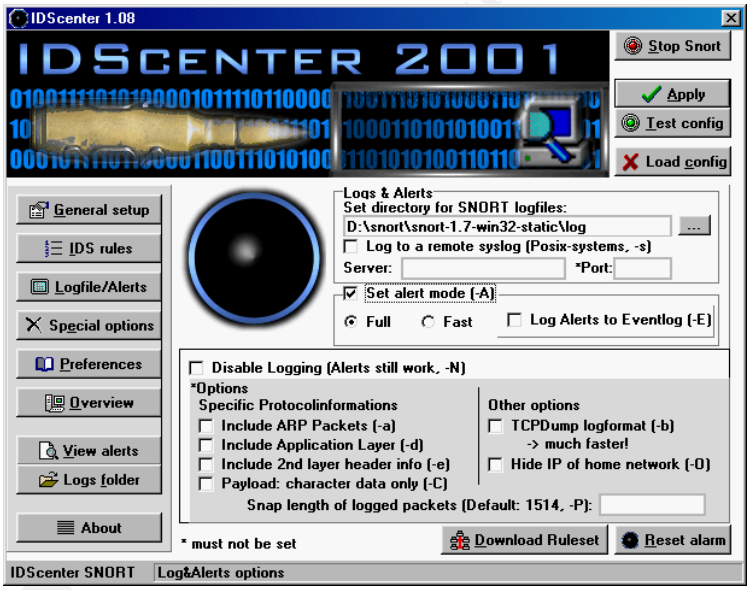


1. Select the snort version you are using (Probably 1.7)
2. Enter the location of the Snort executable file (D:\snort\snort-1.7-win32-static\snort.exe)
3. Enter the network addresses that you are protecting (10.0.0.0/8)

Once General setup is complete, select IDS rules as detailed below.



1. Enter the location of your snort configuration file (D:\snort\snort-1.7-win32-static\snort.conf)
2. Review the IDS rules and enter any desired changes.
3. Select Logfile/Alerts and follow the steps below



1. Enter the location of your snort log directory (D:\snort\snort-1.7-win32-static\log)
2. Click Apply to create the script.
3. Click Start Snort (shown as Stop in the graphic) to activate the service.

Examining the Snort Logs

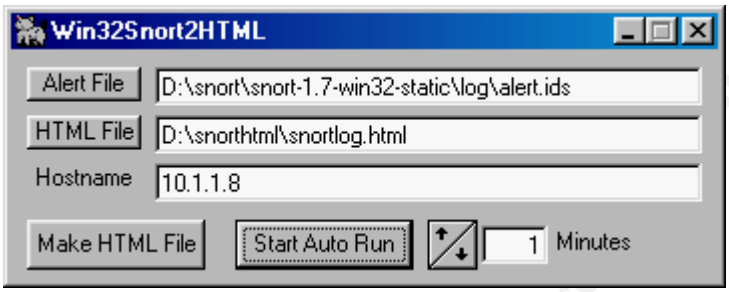
Snort log files are text and may be viewed using the IDScenter interface. The sample log file in Appendix A was generated by an NMap Port scan.

The log file is detailed and informative but not exactly easy to skim through and evaluate. However, once again the diligent efforts of others come to the rescue through applications to display, sort and examine log files within a Graphical environment.

The first of these applications is Win32Snort2HTML by Chris Koutras. Using this application, the log file shown in Appendix A can be transformed into the web page shown in Appendix B. Not only does

this application present the information in a more readable and pleasing format but also, links to further information on the exploit discovered, the source address and the ports accessed are a click away.

To install this application, download the zip file and expand it to a temporary folder. Run the setup.exe and start the program. From the configuration window (see below) enter the location of the alert file (D:\snort\snort-1.7-win32-static\log\alert.ids) the name and location of the HTML file to be generated (D:\snorthtml\snortlog.html) and a hostname for the system. Selecting "Make HTML File" will immediate generate an output from the current Alert File while enabling Auto Run will update the file at whatever schedule is specified.



For native Windows applications, these are the main players as of today. However, through the use of Perl script, further sorting and reporting of the log data is possible. The scripts available from www.snort.org include[5],

1. **snort_stat.pl** v1.3 Perl script that provides a statistical analysis of syslog alerts produced by Snort. Yen-Ming Chen
2. **snort-sort.pl** v0.02 This script produces a sorted list of snort alerts from a snort alert file Andrew Baker
3. **Getcontact** v1 Perl Script to pull contact information out of snort_portscan.log files. The program looks up the source IP numbers in the various registry databases and outputs the email address with the relevant portion of the log. Robin Stubbs

Of the scripts listed, only Getcontact is specifically designed for Windows Systems. Any other scripts may require modifications to work properly. However, for any of them to work at all, a Perl interpreter for Windows is required. Fortunately, ActivePerl Binaries from ActiveState Corp. are available on the web and free.

Further descriptions of these applications and tools can be found using the links provided throughout this document and in the references.

Responding to Alerts

The first thing to keep in mind after your sensor is activated is **Don't Panic**. While there are people out to get you, they were there before you had an IDS sensor doing exactly what you are now logging. Before cutting off Internet access or calling the FBI, it is critical to be certain that the alert you see is not normal traffic or simply an errant connection.

Initially you will spend a lot of time deciding whether the event was an overt act or simply a false alarm. Unless you have the luxury of a freshly installed network, the question will remain whether a system was compromised before you began monitoring. Unfortunately, there is no easy answer to this question. However, over time, as you investigate the alerts and modify the rules you will develop an in-depth knowledge of your environment and find it easier to recognize true alarms.

One alert that is rarely a false alarm is the port scan. There is absolutely no entirely innocent reason for a rapid series of connection attempts to a single port over a range of IP addresses. When you record this type of alert, the research phase may be omitted. Other alerts will typically require some additional investigation as detailed below.

1.) Research the Alert

Unless you are well versed in the details of Intrusion detection it is highly probable that a header such as `[**] IDS162 - PING Nmap2.36BETA [**]` will not mean anything to you. However, if you enter http://www.whitehats.com/info/IDS162 into your web browser, you will find that this is potentially an NMap probe into your network. Another possible clue is a CVE identifier. Alerts that begin with either CVE or CAN may be investigated by searching the database at http://cve.mitre.org.

Unfortunately, not all alerts will provide IDS or CVE identifications. These will require a bit more detective work to determine their cause. The next piece of information to evaluate is the ports recorded. An excellent resource for investigating ports is http://www.snort.org/Database/portsearch.asp. From this detail you may be able to better determine the nature of the alert.

While this detective work may seem daunting, Win32Snort2HTML provides direct links to IDE, CVE and Port information from the web page. Also, as you become more experienced, the details provided by Snort will become sufficient to classify the nature of the alert.

2.) Determine where it came from

The physical source of the packet is noted by the source IP address recorded in the alert. With the address in hand, we can begin to research the source at http://www.arin.net/whois. Entering the IP address will return details on the owner and typically includes an email address for the primary contact. Frequently, the IP address you are investigating is assigned to one of the foreign registries. In that case, a link to either RIPE (http://www.ripe.net/db/whois.html) or APNIC (http://www.apnic.net) will be provided by ARIN. Simply follow the link and reenter the numbers to search that registry.

Once you know the domain the event originated from, you can better determine whether the source may be a business partner, client, employee or stranger. Coupling this information with knowledge of your systems, historical traffic patterns and the nature of the event should provide a fairly clear idea of whether this was a probe and attack or a false alarm.

3.) Decide how to proceed

This question has to be the most difficult to answer. However, if you have determined that your system has been or will be compromised, you need to act. To ensure you have the support of your superiors and everyone is aware of the process, it is critical that Policies and Procedures are in place. Assistance in developing these documents is freely available from http://www.sans.org.

Either way, any response to information uncovered must be reasonable and focused upon protecting your assets. Under no circumstances should a counter attack, network probe or other intrusive action be used against another system. Such a response is potentially illegal and may be directed at an innocent party. For a discussion of innocent source addresses, please refer to Network Intrusion Detection of Third Party Effects" by Richard Bejtlich[6]

My standard response to Intrusion attempts begins with the use of Portsentry (http://www.psionic.com). This tool allows me to automatically block questionable sources from my firewall. In this way, I am able to actively defend my resources while investigating the incident. Once the intrusion is identified as

malicious, emails to abuse@*TheCompany*, any authority figures I can locate through their web site and abuse@*TheirISP* if appropriate are sent. As mentioned in the overview, not all of these emails result in positive responses but any that do, eliminate an access point for these types of attacks.

Conclusion

Network Intrusion Detection systems are no longer a luxury. With the advent of inexpensive and free tools for the Windows environment, the old excuses of needing UNIX knowledge or thousands of dollars are eliminated. Further, without concerted and distributed efforts to stem the rise in malicious traffic, the risks of global connectivity may eventually outweigh the benefits.

References

"Deploying Snort, a Lightweight Network Intrusion Detection System" by David G. Sullivan (http://www.sans.orf/infosecFAQ/intrusion/snort.htm)

"An Analysis of the Snort Intrusion Detection System" by Mark D. Tollison (http://www.sans.orf/infosecFAQ/intrusion/snort2.htm)

WinPcap is available at http://netgroup-serv.politico.it/winpcap

IDSCenter is available from http://www.eclipse.fr.fm/snort.htm

Win32Snort2HTML is available at http://www.snort.org/Files/ws2htmlv.zip

EndNotes

1) "Intrusion Detection in the Enterprise Network: Managing Hacker-Related Risk" by Internet Solutions Business Unit, Compaq Computer Corporation. March 1998 (ftp://ftp.compaq.com/pub/supportinformation/papers/ecg0410298.pdf)

2) "FAQ: Network Intrusion Detection Systems" by Robert Graham March 21,2000 (http://www.robertgraham.com/pubs/network-intrusion-detection.html)

3) "Snort - Lightweight Intrusion Detection for Networks" by Martin Roesch (http://www.snort.org/lisapaper.txt)

4) http://www.whitehats.com/cgi/tools/BrowseTree?field=Category&separator=:&recurse=1&value=Intrusion%20Detection%3aNIDS

5) Text copied from the download page at http://www.snort.org/snort-files.htm

6) "Network Intrusion Detection of Third Party Effects" by Richard Bejtlich August 26, 2000 (http://securityfocus.com/data/library/nid_3pe_v1.pdf)

Appendix A
Alert.ids file from an NMap portscan

[**] IDS162 - PING Nmap2.36BETA [**]
04/06-14:28:04.601990 10.0.0.253 -> 10.0.0.118
ICMP TTL:56 TOS:0x0 ID:54947 IpLen:20 DgmLen:28
Type:8 Code:0 ID:23914 Seq:0 ECHO

[**] spp_portscan: PORTSCAN DETECTED from 10.0.0.253 (THRESHOLD 4 connections exceeded in 0 seconds) [**]
04/06-13:28:04.960000
[**] MISC-WinGate-8080-Attempt [**]
04/06-14:28:05.333107 10.0.0.253:56937 -> 10.0.0.118:8080
TCP TTL:40 TOS:0x0 ID:7674 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization [**]
04/06-14:28:06.126081 10.0.0.253:56937 -> 10.0.0.118:1031
TCP TTL:40 TOS:0x0 ID:30218 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] BIND Shell [**]
04/06-14:28:06.862348 10.0.0.253:56937 -> 10.0.0.118:31337
TCP TTL:40 TOS:0x0 ID:58972 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] default Backdoor access! [**]
04/06-14:28:07.244775 10.0.0.253:56937 -> 10.0.0.118:1524
TCP TTL:40 TOS:0x0 ID:7422 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] Netbus/GabanBus [**]
04/06-14:28:07.246914 10.0.0.253:56937 -> 10.0.0.118:12346
TCP TTL:40 TOS:0x0 ID:22100 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] MISC-WinGate-1080-Attempt [**]
04/06-14:28:07.258065 10.0.0.253:56937 -> 10.0.0.118:1080
TCP TTL:40 TOS:0x0 ID:54246 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] MISC-Attempted Sun RPC high port access [**]
04/06-14:28:07.284085 10.0.0.253:56937 -> 10.0.0.118:32771
TCP TTL:40 TOS:0x0 ID:37660 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] spp_portscan: portscan status from 10.0.0.253: 1457 connections across 1 hosts: TCP(1457), UDP(0) [**]
04/06-13:28:08.590000
[**] Netbus/GabanBus [**]
04/06-14:28:08.008268 10.0.0.253:56937 -> 10.0.0.118:12345
TCP TTL:40 TOS:0x0 ID:17693 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization [**]
04/06-14:28:08.008572 10.0.0.253:56937 -> 10.0.0.118:1032
TCP TTL:40 TOS:0x0 ID:44859 IpLen:20 DgmLen:40
******S* Seq: 0xCCA07B31 Ack: 0x0 Win: 0x400 TcpLen: 20

[**] Possible NMAP Fingerprint attempt [**]
04/06-14:28:08.394217 10.0.0.253:56946 -> 10.0.0.118:139
TCP TTL:40 TOS:0x0 ID:37374 IpLen:20 DgmLen:60
**U*P*SF Seq: 0xA2ED7978 Ack: 0x0 Win: 0x400 TcpLen: 40 UrgPtr: 0x0

TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

[**] NMAP TCP ping! [**]
04/06-14:28:08.394366 10.0.0.253:56947 -> 10.0.0.118:139
TCP TTL:40 TOS:0x0 ID:19083 IpLen:20 DgmLen:60
A* Seq: 0xA2ED7978 Ack: 0x0 Win: 0x400 TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

[**] NMAP TCP ping! [**]
04/06-14:28:08.394577 10.0.0.253:56949 -> 10.0.0.118:1
TCP TTL:40 TOS:0x0 ID:19349 IpLen:20 DgmLen:60
A* Seq: 0xA2ED7978 Ack: 0x0 Win: 0x400 TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

[**] NMAP XMAS scan [**]
04/06-14:28:08.394669 10.0.0.253:56950 -> 10.0.0.118:1
TCP TTL:40 TOS:0x0 ID:6751 IpLen:20 DgmLen:60
U*PF Seq: 0xA2ED7978 Ack: 0x0 Win: 0x400 TcpLen: 40 UrgPtr: 0x0
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

Appendix B
WinSnort2HTML page generated from Alert.ids

Snort log for 10.1.1.8

Time Legend: 00:01 to 06:00 Hrs -- 06:01 to 18:00 Hrs -- 18:01 - 00:00 Hrs

Port Legend: 0 - 1023 -> Well Known Ports -- 1024 - 49151 -> Registered Ports -- 49152 - 65535 -> Dynamic Ports

Num.	Date	Time	Attack	Source IP Addr.	Source Port	Target IP Addr.	Target Port
1	04/06	14:28:04.601990	IDS162 - PING Nmap2.36BETA	10.0.0.253	56950	10.0.0.118	1
2	04/06	14:28:05.333107	spp_portscan: PORTSCAN DETECTED from 10.0.0.253 (THRESHOLD 4 connections exceeded in 0 seconds) [**] 04/06-13:28:04.960000 [**] MISC-WinGate-8080-Attempt	10.0.0.253	56937	10.0.0.118	8080
3	04/06	14:28:06.126081	IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization	10.0.0.253	56937	10.0.0.118	1031
4	04/06	14:28:06.862348	BIND Shell	10.0.0.253	56937	10.0.0.118	31337
5	04/06	14:28:07.244775	default Backdoor access!	10.0.0.253	56937	10.0.0.118	1524
6	04/06	14:28:07.246914	Netbus/GabanBus	10.0.0.253	56937	10.0.0.118	12346
7	04/06	14:28:07.258065	MISC-WinGate-1080-Attempt	10.0.0.253	56937	10.0.0.118	1080
8	04/06	14:28:07.284085	MISC-Attempted Sun RPC high port access	10.0.0.253	56937	10.0.0.118	32771
9	04/06	14:28:08.008268	spp_portscan: portscan status from 10.0.0.253: 1457 connections across 1 hosts: TCP(1457), UDP(0) [**] 04/06-13:28:08.590000 [**] Netbus/GabanBus	10.0.0.253	56937	10.0.0.118	12345
10	04/06	14:28:08.008572	IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization	10.0.0.253	56937	10.0.0.118	1032
11	04/06	14:28:08.394217	Possible NMAP Fingerprint attempt	10.0.0.253	56946	10.0.0.118	139
12	04/06	14:28:08.394366	NMAP TCP ping!	10.0.0.253	56947	10.0.0.118	139
13	04/06	14:28:08.394577	NMAP TCP ping!	10.0.0.253	56949	10.0.0.118	1
14	04/06	14:28:08.394669	NMAP XMAS scan	10.0.0.253	56950	10.0.0.118	1

This page generated from snort logs on 04-06-2001 at 14:29:45 using WinSnort2Html by Chris Koutras. Based on the Perl script snort2html by Dan Swan.