Global Information Assurance Certification Paper

GIAC Certified Windows Security Administrator (GCWN)
Practical Assignment Version 5.0, Option 1

Cost Effective Intrusion Detection for Microsoft Windows

By Steve Staden
August 14th, 2004

Table of Contents

Abstract

Microsoft Windows operating systems have an inherent security issue, which has the potential to allow vulnerabilities and intrusions. This security issue is a lack of logging intrusion attempts from malicious users or attackers. After various tests a solution to the security issue was found. A remedy for this issue is using a third-party product until Microsoft incorporates a utility for this security issue. A cost effective solution is to use Snort on a Windows machine to actively monitor for system intrusions. Monitoring for system intrusions will help aid the system administrator in determining what threats may exist on critical systems.

Introduction

Internal memos detailing the company's financial resources, employee salaries, credit card and Social Security numbers are all susceptible to a breached system. There are measures to prevent this loss of information. Having a clear and correctly defined firewall rule set, appropriate patch management, proper network design and security guidelines are all effective ways to protect a system. However, once you have properly secured a system how do you remain proactive to keep it secure? The old adage goes, "An ounce of prevention is worth a pound of cure." In the case of security the pound of cure can be easily translated into long hours and expensive solutions to remedy a problem or issue. That is why it is essential to be proactive with computer security and understand what may be happening to a system. Administrators need to know if a system is being actively probed or if an attacker was able to gain access into a critical system and steal sensitive information.

There is no obvious way to determine whether a Windows system is being actively probed for insecurities or vulnerabilities. Windows 2003 does include a software firewall that will block inbound connections, but by default the logging is turned off on this feature and only blocking will occur. Where logging is enabled, the logs provided by the firewall are text files that are hard to sort and search for intrusion attempts. Clues may be given in the Windows Event Viewer such as failed logons or locked accounts. However, system administrators often overlook these clues. What makes this a serious security issue is the amount of damage caused if an intruder successfully breaks into a system unknowingly to an administrator. Windows lacks an efficient and effective way to sort logging facilities or viewers to determine intrusion attempts. Therefore an administrator needs to write custom scripts to parse through log information possibly using either VBScript or Perl. One would think being the world's most popular operating system the developers would have this security feature in mind so an administrator could easily determine intrusion attempts from log information. Unfortunately, log segmentation is understated in the current operating system, but hopefully strides will be made in future releases.

Moreover, the security issue of not being aware of attempted attacks is not due to local configuration or architectural requirements. In fact, configuring the

Local Security Policy or Group Policy in a more security minded fashion vastly improves the overall security of the system. Some examples include changing the default password policy, using NTLMv2 and many other security features that are configurable. There are no architectural requirements that the operating system is missing either. This security issue exists because the utility to detect intrusions simply does not exist. Also a sorting, parsing and alert feature for the many logs Windows produces is missing from the operating system. IIS logs may contain a lot of data, but an administrator needs to know the exact signature of an attack and be able to sort the log based on an intrusion. Unfortunately, this security issue exists in all versions of Windows and is more prevalent in older versions that do not include some of the more advanced monitoring tools. Microsoft has finally come to terms that their operating system needs to be more secure and less feature rich. This has been heralded as a major stepping block for Microsoft. Microsoft's operating system is in dire need of a better way to manipulate and sort log information that is generated as well as detect possible intrusion. Microsoft is making its first steps to securing the operating system and will hopefully move towards more security features for system administrators.

As companies grow they also purchase new systems for processing their data. These systems need to be hardened and secured as soon as they are setup. Then it is also necessary to keep these systems up-to-date with all the latest software upgrades, patches and fixes. Keeping systems updated is a step that is often overlooked by busy system administrators struggling to meet their company's requirements or service level agreement. This leads to the potential of having systems with known vulnerabilities available to attackers. There are many different methods to keep systems up-to-date with the latest releases and also the latest hotfixes available. A diligent system administrator will be proactive and subscribe to security lists to receive bulletins about possible outbreaks and bugs for their systems software.

Consequently, the implications of doing nothing and being negligent about security are devastating. The media is full of stories about financial companies crippled by the work of a diligent intruder, or the latest outbreak of a virus that impaired a corporation. In the simplest sense there exists the potential for a stealthy intruder to break into a system for a number of months and yet nobody ever notices. Once the intruder has access into the system they may have access to extremely confidential files. Although, in some cases the intruder may not have access to these files because of properly applied file permissions, it's only a matter of time before the intruder figures out a way to escalate their privileges. In any case, a number of scenarios can play out once the intruder has access to your system. Over the past couple years various worms have been written to take advantage of existing vulnerabilities in the Windows operating system. These worms have been the cause of many headaches for system administrators and lead to better security practices to avoid these outbreaks. One example of a worm that caused outages for various companies was the Code Red worm (http://www.cert.org/advisories/CA-2001-19.html), which had another variant shortly after called Code Red II (http://www.cert.org/incident_notes/IN-2001-09.html) that caused much more

chaos. Between the Code Red worm and the Nimda worm many organizations realized a need for better security and patch management.

There are a couple different solutions to help defend your system. The most obvious solution is having the latest releases of software and applying bug fixes when necessary. Administrators should always test fixes with development boxes before deploying the fix into production. Deploying a fix that breaks a current application can be as disastrous as being vulnerable. The optimal solution to prevent intrusion and similar attacks would be a somewhat lightweight intrusion detection system. This solution assumes the administrator keeps the systems updated with the latest patches and maintains correct configurations regarding the security policy. Not every company is willing to pay exuberant amounts of money for an intrusion detection system. Nor is every company an extremely large Fortune 500 company that would require a massive intrusion detection system. Cost effective solutions rule in these days of slashed technology budgets. This tool should be able to detect a variety of intrusion attempts on a given system. The tool should also be able to log the information in an organized fashion and then be able to send e-mail alerts to system administrators or provide a console interface in case of serious intrusions detected on a system. A couple other features to look for would be detailed alerts of the intrusion and custom configuration of the logging for the monitor.

Product Evaluation

There are two tools an administrator could use to effectively safeguard a system from the aforementioned issue. The first tool to evaluate is Snort ™ (www.snort.org), which is an open source network intrusion detection system. Snort has been around since 1998 and seems to be continuously evolving into a better tool. Snort can be used in a couple different ways including sniffing, packet logging and intrusion detection. Running Snort in intrusion detection mode helps administrators in discovering attempted attacks. Since Snort is open source it can be downloaded for free and provides a low-cost solution for small businesses. Snort can be downloaded at http://www.snort.org/dl/ along with its many plugins and extras and the Windows version is located at http://www.snort.org/dl/binaries/win32. One of the more surprising items with Snort was the amount of documentation available for this utility. Users have written documents for deploying Snort and they were quite useful, however they have become out of date. Searching the web for Snort configuration yielded many results that were quite useful in researching the product. (http://www.google.com/search?hl=en&ie=UTF-8&q=snort+configuration) There is also an older document available from SANS that details configuration for Snort located here http://www.giac.org/practical/gsec/Jeff_Richard_GSEC.pdf. On a side note the Network Monitor that comes with the Microsoft operating system was not evaluated due to the complete lack of logging capability with the tool.

The second tool to evaluate is LANguard Security Event Log Monitor available from GFI at http://www.gfi.com/languard/. LANguard S.E.L.M. is

available as a full demo or free version. Once the demo expires the product is stripped of some features and is retained as the free version. The software is available for download from http://www.gfi.com/downloads/downloads.asp?pid=6&lid=1. After looking through the online material for GFI it seems there are many features available that would be beneficial for the issue at hand. A short list of features listed on GFI's main web site includes the following:

- ✓ Detect attacks on these machines in real time
- ✓ Monitor users attempting to access secured shares and confidential files
- ✓ Create alerts for specific events and conditions occurring on these machines
- ✓ Back up and clear event logs automatically and archive event logs to a central database.

The machine used to test these two products was a Windows 2000 Service Pack 4 system with all the latest updates as of August 31st 2004. The machine has an Intel 2.8 GHz processor and 192 MB of RAM.

To install Snort you need to download the Windows binary. Snort requires the WinPcap library. This can be installed either before or after the Snort installation, but if not installed Snort will fail to work. WinPcap is available for download here http://winpcap.polito.it/install/default.htm and both 3.0 and 3.1 beta 3 were tested and only 3.0 version worked correctly at the time of writing. Next run the Snort binary to install Snort onto the system. Once Snort is installed the "snort.conf" needs to be edited for the systems specifics. This file is located in the "c:\snort\etc" directory on the Windows platform. Snort and WinPcap both need Administrator access to be installed on the Windows platform. Snort does not open up any ports on the system. Complete documentation for using Snort is available at http://www.snort.org/docs/snort_manual and supplementary documentation is available at http://www.snort.org/docs. Snort had many various options to use to run and while the documentation does a pretty good job of outlining these options there is still much information to digest at one time. To run snort in the basic IDS mode use the following command:

```
C:\Snort\bin>"snort -d -h 192.168.3.0/24 -l c:\snort\log -c c:\snort\etc\snort.conf"
```

The options used above are the "-d" option, which dumps the application layer. The "-h" option is used to specify the home network. The home network needs to be configured correctly for Snort to properly log alerts. The "-l" option is the directory to which the logs files will be written. In testing this option is still needed if logging to a database. The "-c" option is used to specify the Snort configuration to use. There are numerous other options available to use with Snort. These options and details on how to use them are available at http://www.snort.org/docs/snort_manual/. A listing of these options is provided in Appendix A and the same listing can be produced by giving snort the "-?" option.

To install LANguard run the executable, using an account that has domain administrator privileges. LANguard appears to use MS Access as its default database, but does not require it to be installed separately. It can also use

Microsoft SQL server. After LANguard is installed it will bring up the interface to run the wizard for configuration and the administrator will have to input the necessary values. The first test for both products was a basic port scan. Nmap (www.insecure.org) was used to test both products for alerts to a scan probe against the target machine. Nmap is a port scanner that is available for many different platforms and is well known for its effectiveness in port discovery. Nmap can be downloaded at http://www.insecure.org/nmap/nmap_download.html. However, users with Windows XP Service Pack 2 should be cautioned that older versions and even some newer versions would not work with Service Pack 2. Yet another port scanner is available for Windows called SuperScan that is available at http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/superscan.htm. The type of scan used was a TCP SYN scan against the host machine. The machine in question was running MOVEit® DMZ (http://www.standardnetworks.com), which is used to transfer files securely. MOVEit® DMZ is an ASP .NET application that uses IIS and MySQL for transferring files securely. Some local security policy settings were applied to the test system. These changes revolved around auditing and account policy lockout. Also various unneeded services were disabled. To learn more about MOVEit® DMZ and similar products visit http://www.standardnetworks.com/.

```
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-09-01 15:09 CDT
Interesting ports on smsdmz.xxx.stdxxx.com (192.168.3.171):
(The 1645 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
990/tcp  open  ftps
1025/tcp open  NFS-or-IIS
1033/tcp open  netinfo
2105/tcp open  eklogin
3306/tcp open  mysql
3372/tcp open  msdtc
3389/tcp open  ms-term-serv
```

The results from Snort's "c:\snort\log\alerts.ids" are as follows:

```
[**] [1:469:3] ICMP PING NMAP [**]
[Classification: Attempted Information Leak] [Priority: 2]
08/31-20:16:46.753977 192.168.0.50 -> 192.168.0.102
ICMP TTL:48 TOS:0x0 ID:39021 IpLen:20 DgmLen:28
Type:8  Code:0  ID:27084   Seq:20055  ECHO
[Xref => http://www.whitehats.com/info/IDS162]

[**] [1:1420:11] SNMP trap tcp [**]
[Classification: Attempted Information Leak] [Priority: 2]
08/31-20:16:47.136617 192.168.0.50:61684 -> 192.168.0.102:162
TCP TTL:50 TOS:0x0 ID:60537 IpLen:20 DgmLen:40
******S* Seq: 0x7C8D2A35  Ack: 0x0  Win: 0xC00  TcpLen: 20
```

[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref => http://www.securityfocus.com/bid/4089][Xref => http://www.securityfocus.com/bid/4088]

[**] [1:1421:11] SNMP AgentX/tcp request [**]
[Classification: Attempted Information Leak] [Priority: 2]
08/31-20:16:47.141319 192.168.0.50:61684 -> 192.168.0.102:705
TCP TTL:37 TOS:0x0 ID:57809 IpLen:20 DgmLen:40
******S* Seq: 0x7C8D2A35 Ack: 0x0 Win: 0x800 TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref => http://www.securityfocus.com/bid/4089][Xref => http://www.securityfocus.com/bid/4088]

[**] [1:1418:11] SNMP request tcp [**]
[Classification: Attempted Information Leak] [Priority: 2]
08/31-20:16:47.793293 192.168.0.50:61684 -> 192.168.0.102:161
TCP TTL:52 TOS:0x0 ID:35786 IpLen:20 DgmLen:40
******S* Seq: 0x7C8D2A35 Ack: 0x0 Win: 0x400 TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref => http://www.securityfocus.com/bid/4089][Xref => http://www.securityfocus.com/bid/4088]

Snort logged four alerts during the Nmap scan against the host machine. These alerts give details about what was triggered and the alerts then provide links to appropriate resources for even further details. Each alert is given a priority based upon the severity of the alert triggered. Low severity alerts are given in a higher number, while high severity alerts are given in a lower number. For example, the port scan triggered a high priority alert. The alerts also give the exact date and time of the suspected intrusion along with the source and destination IP addresses and relevant ports. Interestingly, LANguard failed to log any activity from this basic port scan. This might be understandable if a computer has an IPSec policy defined that clearly allows only certain ports, but in this scenario this is not the case. Also, port scans are pretty common occurrences throughout the Internet and it may be viable to not trigger an alert. On the other hand, if someone is constantly probing a system for open ports then there may be cause for concern.

The next test was simulating an actual vulnerability/attack attempt. To simulate this the Retina RPC DCOM Scanner from eEye Digital Security was used. This tool is available for download at http://www.eeye.com/html/Research/Tools/RPCDCOM.html. This tool seeks to exploit a known vulnerability in the Microsoft operating system. The exploit in question will allow the intruder to take control of the system. This should be a very interesting test for both products, as the tool will simulate an intruder attempt at the system and should be something both products produce alerts on. The details about this vulnerability can be found at http://www.microsoft.com/technet/treeview/?url=/technet/security/bulletin/MS03-039.asp.

After running the vulnerability tool against the system running Snort the alert located below was logged. The priority of this alert was high and it lists quite a few references for the alert. Another feature of Snort is that it logs a classification of the alert. In this case the classification was Attempted Administrator Privilege Gain. It would be fairly interesting to see a feature like this

that logged in the Windows Event Viewer. One idea might be a separate section similar to the DNS or Directory Service segmentation in the Event Viewer. This sort of an alert should stand out to system administrators so they are aware an intrusion was attempted. However, the administrators should have the system patched so that the intruder cannot exploit the machine in question. After running the tool against the system running LANguard no alerts were logged to this activity. The machine had been patched with the proper hotfix to fix the vulnerability, however no information in the log was provided to show an attempt to exploit the machine happened. This was disheartening news because a cautious administrator would probably want to know if his machine was being attacked. Then the administrator would be able to block the source IP address using an IPSec policy or take any other action to resolve this issue.

[**] [1:2251:13] NETBIOS DCERPC Remote Activation bind attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
09/01-10:09:20.801676 192.168.3.161:10713 -> 192.168.3.171:135
TCP TTL:128 TOS:0x0 ID:60170 IpLen:20 DgmLen:112 DF
AP Seq: 0x60229D50 Ack: 0x99AD8D71 Win: 0xFFFF TcpLen: 20
[Xref => http://cgi.nessus.org/plugins/dump.php3?id=11798][Xref =>
http://cgi.nessus.org/plugins/dump.php3?id=11835][Xref =>
http://www.microsoft.com/technet/security/bulletin/MS03-039.mspx][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0715][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0605][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0528][Xref =>
http://www.securityfocus.com/bid/8458][Xref => http://www.securityfocus.com/bid/8234]

The final test against both products was an enumeration of users, shares and LSA policy information. The tool used to enumerate the share information was enum.exe, which is available at http://download.microsoft.com/download/9/3/0/930e1ecb-a6c6-445f-bd79-82fc3e66f009/enum.exe. Enumeration is a good way to see, after determining the machine is a Windows host, what might be available on the machine itself. Enumeration will allow an intruder to map the resources the machine has and list information that could be used to attack a machine. Some of the important information that can be enumerated using this method is various usernames and shares belonging to the machine. It is somewhat difficult to block enumeration attempts as listing the information is pretty standard when computers are talking between each other. Although a cautious administrator should at least be aware of these attempts and proper auditing should be utilized. To list these enumerations run the following command:

enum -U -S -L -G -d “IP address”

If the enumeration worked a listing of items should look similar to the results in Appendix B.

Another very surprising result as LANguard S.E.L.M. did not identify any security events even with all security events enabled. This seemed very odd that LANguard would not flag this as even a low level activity. LANguard did not log any alerts for a vulnerability attack in the previous test so the result wasn’t as

surprising. However, running against Snort two alerts were identified and they are listed below. One of the alerts was marked as a high priority alert with a classification of Attempted Administrator Privilege Gain while the other alert was marked lower with a classification of Generic Protocol Command Decode. This would be good information to know if a system administrator were concerned about a machine constantly being probed by an intruder. Also, it would also be useful information to alert the administrator that someone may be planning a future attack by first gathering information from the target machine.

[**] [1:2383:13] NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
09/02-11:47:39.583100 192.168.3.161:13442 -> 192.168.3.171:445
TCP TTL:128 TOS:0x0 ID:8751 IpLen:20 DgmLen:304 DF
AP Seq: 0x2430B99 Ack: 0x84660781 Win: 0xFE65 TcpLen: 20
[Xref => http://cgi.nessus.org/plugins/dump.php3?id=12065][Xref =>
http://cgi.nessus.org/plugins/dump.php3?id=12052][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0818][Xref => http://www.securityfocus.com/bid/9635][Xref =>
http://www.securityfocus.com/bid/9633]

[**] [1:2466:4] NETBIOS SMB-DS IPC$ share unicode access [**]
[Classification: Generic Protocol Command Decode] [Priority: 3]
09/02-11:47:39.591391 192.168.3.161:13442 -> 192.168.3.171:445
TCP TTL:128 TOS:0x0 ID:8752 IpLen:20 DgmLen:136 DF
AP Seq: 0x2430CA1 Ack: 0x846607FA Win: 0xFDEC TcpLen: 20

After conducting these tests the results favor Snort as the product to help monitor a system for active probes and intrusion attempts. Snort's alerts are very detailed and give very detailed information on what has happened to a machine. A couple other benefits to Snort are the customization that can be used along with the logging capabilities that can be configured. GFI's LANguard was easier to install than Snort, but was not able to log the information a proactive administrator would be looking for on a designated system. This could lead to an intrusion unbeknownst to a system administrator. LANguard's more effective use would probably be a central system monitoring many systems for changes in various event logs across multiple machines. Snort monitors the activity that should caution administrators to various attacks or attempts to gain access to a system. While it lacks the e-mail notification that LANguard was able to supply, it has very detailed logging information and is a flexible application itself.

There are a couple ideas to keep in mind when implementing Snort. First, Snort's IDS mode can run either as a service or through the command prompt. Once the command prompt is stopped summary information is displayed. The logs for Snort are kept in a folder off of the Snort root directory. These logs contain the alert log and other logging information specified by the configuration file. The nice feature about Snort is the ability to fine tune logging via the configuration file. Logging information can be stored in many ways on the system and Snort has the ability to utilize databases for logging information. Some of the supported databases include Oracle, Microsoft SQL, PostgreSQL and MySQL. Snort's rule sets are also customizable which helps administrators cut down on false positives and allows custom rules sets. Snort's configuration and operation is not nearly as fine tuned as other IDS solutions available. This is

probably because Snort is available for free download. Snort seems to be geared toward highly flexible solutions and seems to be more of a piece of the puzzle rather than the puzzle itself. One feature that is not available is the ability to e-mail administrators of triggered alerts. With this feature not available administrators will have to manually monitor the log, or setup a central logging computer to compile the alerts.

However, there are a good amount of tools available to utilize Snort to its full potential. If a company can afford a feature rich Intrusion Detection System then DeMarc's (http://www.demarc.com) Sentarus might be the application to use as it uses Snort and MySQL to produce a nice interface for intrusion detection. This product uses a lot of the same utilities discussed in the implementation guide further in this document. The online demo seems to speak for itself, as the interface is very appealing. The demo can be viewed at http://www.demarc.com/products/sentarus/software/screenshots. Also, another useful GUI application is available IDScenter (http://www.engagesecurity.com/products/idscenter/). IDScenter appears to be a very nice utility for configuration, report creation and even alert notification. However, after downloading and installing version 1.1 RC4 problems persisted and IDScenter was not able to function correctly and locked up the operating system. Looking through the console though there were many nice features that it was able to provide for the administrator. The final choice was to use the Analysis Console for Intrusion Databases (ACID) with MySQL. This requires another server for use to maintain all the database entries and provides a web interface for the Snort monitor. ACID is one of the more simplistic interfaces to use with Snort and MySQL and is fairly easy to install.

Implementation Guide

Hardware requirements - 1 central server monitor for monitoring Snort alerts. (E.g. A PIII 600+, 256mb RAM, 20GB hard drive, 10/100 network card and either Linux or Windows operating system). Depending on the budget a new or older machine may be used. The monitor server should be a fairly significant server as it will store a MySQL database and have a web server for internal use. An extra step of security can be given by obtaining a SSL certificate from a company such as Verisign (http://www.verisign.com/) or Thawte (http://www.thawte.com/) for the ACID monitor. However, creating a temporary or test certificate with OpenSSL (http://www.openssl.org) can be done for testing purposes.

Software Requirements for Snort nodes -
__ Snort - http://www.snort.org/dl/binaries/win32/snort-2_2_0.exe
__ WinPcap -http://winpcap.polito.it/install/bin/WinPcap_3_0.exe

Step 1 Install ACID and MySQL on the central server

ACID installation requires various packages and configurations and there are a couple documents already available on the web that do a good job of explaining

this part of the installation. Documentation for this step is available at http://www.andrew.cmu.edu/user/rdanyliw/snort/acid_config.html. The author used Linux (Debian) for easier compatibility, but installation will work under a Windows platform. Another source for installing ACID and MySQL are also available from a previous practical, which is available at http://www.giac.org/practical/gsec/Jeff_Richard_GSEC.pdf. It should be noted that the document is fairly out dated and most packages have been updated to reflect new releases and fixes as well as features. MySQL if installed on the Windows platform will be installed as a service and automatically run at each startup.

Step 2 Install Snort and WinPcap on a system node.

Select the default options for both software programs. For this installation WinPcap was installed before Snort was installed. However, either can be installed first as the user sees fit. Snort will give an error message if it does not find the WinPcap library. Also, as noted earlier the WinPcap 3.1beta3 version did not work with Snort 2.2.0. It is recommended to use the stable WinPcap version 3.0.

Step 3 Configure Snort for ACID and MySQL access on a system node.

There are four sections to the Snort configuration file and they are

1) Set the network variables for your network.
2) Configure preprocessors.
3) Configure output plugins.
4) Customize your rule set.

We are interested in sections 1 and 3 for our purposes. Once you become more familiar with Snort, you may configure sections 2 and 4 with your specific purposes in mind, but for now, the defaults are acceptable. Section 1 requires you to define your particular network setup. It is essential to set this up correctly or there will be numerous false positives as a result. For Section 3 the most important configuration is the following line:

```
output database: log, mysql, user=snort password=huskies dbname=snortdb host=somehost
```

This is the configuration for using the database as the back-end for logging information. The options here are pretty self-explanatory and you will need to substitute your information for the user, password, dbname and host fields.

Step 4 Install Snort as a service on a system node.

Snort can be used from the command prompt given numerous command line options. To determine these options view the online documentation mentioned previously or use “snort –?” at the command line. To install Snort as a service type in the following command

"c:\snort\bin\>snort /service /install –c c:\snort\etc\snort.conf –l c:\snort\log"

If this was successful you will see a message similar to the following.

```
[SNORT_SERVICE] Attempting to install the Snort service.

[SNORT_SERVICE] The full path to the Snort binary appears to be:
   C:\Snort\bin\snort /SERVICE

[SNORT_SERVICE] Successfully added registry keys to:
   \HKEY_LOCAL_MACHINE\SOFTWARE\Snort\

[SNORT_SERVICE] Successfully added the Snort service to the Services database.
```

Snort will be installed and you may view the service in the Services menu from Administrative Tools. Snort by default will be set as manual and will not be started. You may need to test Snort and to make sure Snort is working correctly start the service and try producing some intrusion type activity at the host machine. Then take a look at either your ACID monitor or the "alert.ids" log on the machine for recent activity. If new alerts exist than Snort is running correctly and you may want to set the service as Automatic. If new alerts do not exist make sure you have the correct settings and retry using the command line. Also, a good test to produce alerts in Snort is a TCP port scan against the host machine.

Step 5 Create a snort user account on a system node.

Create a user with Administrator privileges to use for the Snort service. Replace the local system account with the user you created to log on to the system. After this is done you should be able to start the Snort service and Snort will begin monitoring. One item to note is that Snort is running as an Administrator and should be closely watched. Snort will not run as a regular user or power user or with the act as operating system option. All these users were tested with various configurations and Snort would not start unless the service is run under Administrator credentials. There are many implications and possibility of running Snort as an administrator. It seems to be a requirement though and through careful and precise configuration and proper auditing this should alleviate concern. You may want to setup auditing on this user account and keep a close watch on it. Also, be sure to use a strong password when creating the user account.

Step 6 Check and monitor for intrusion activity on the ACID monitor.

This should be done on a basis that is suitable for company needs. Another effective step may be to dedicate a machine to ACID so the console may run all the time. This would allow administrators and others to see the activity that is targeting their critical machines and also allow them to take necessary action against it. This also allows other administrators to view results from other

locations and keep close watch on critical systems. Screenshots from the ACID monitor are listed in Appendix C.

The Snort service will normally take up around 32MB of RAM when running. Depending on the activity of the server that Snort is monitoring this may be less or more. It is essential to have enough available physical memory for Snort. Also, if the machine is constantly being probed or attacked having Snort log all the activity may decrease the performance of the machine. Therefore it is necessary to either reduce logging or fine-tune the necessary logging information to gain critical data. If any errors are produced while Snort is running in service mode the errors will be logged to the Event Viewer in Windows. The most common errors deal with incorrect configurations with the snort configuration file "snort.conf".

In conclusion, there are a wide variety of IDS solutions available to administrators to keep proactive with security. These solutions range from many thousands of dollars to simple time spent configuring and setting up the proposed solution. Each of these solutions has different benefits and drawbacks that need to be taken into consideration. Snort is a very low-cost solution to help smaller companies and administrators monitor their critical systems for breaches and vulnerabilities from intruders. Snort is a flexible tool that if used correctly will assist system administrators maintain secure systems.

The implementation guide featured previously is a basic setup for setting up a basic IDS solution to help prevent intrusion from attackers. An IDS is just one piece of the puzzle, and a diligent administrator needs to keep abreast of the latest information throughout the security community. It is essential to have systems with the latest patches available to fend off attacks. In today's ever-changing world it is critical to be proactive about security and certain that each system has been updated to avoid outages and loss of sensitive information. Companies and people are relying and putting more trust into computers than ever before. Administrators need to take steps to make sure that trust is not broken.

References

'Code Red II:' Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. 6 Aug. 2001. 10 Aug. 2004. Carnegie Mellon University <http://www.cert.org/incident_notes/IN-2001-09.html>.

ACID: Installation and Configuration. 9 Oct. 2002. 10 Aug. 2004 <http://www.andrew.cmu.edu/user/rdanyliw/snort/acid_config.html>.

Caswell, Brian, and Marty Roesch. http://www.snort.org/docs/. 2002. 01 Sept. 2004

Caswell, Brian, and Marty Roesch. http://www.snort.org/dl/binaries/win32. 2002. 01 Sept. 2004

CERT® Advisory CA-2001-19 'Code Red' Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. 17 Jul. 2001. Carnegie Mellon University. 10 Aug. 2004 <http://www.cert.org/advisories/CA-2001-19.html>.

Configuring Snort, MySQL and ACID on WindowsNT. 2000. 10 Aug. 2004 <http://www.giac.org/practical/gsec/Jeff_Richard_GSEC.pdf>.

Demarc Security. 2004. Demarc. 01 Sept. 2004 <http://www.microsoft.com/technet/security/bulletin/MS03-039.mspx>.

eEye Digital Security - Vulnerability Management Solutions http://www.eeye.com/html/Research/Tools/RPCDCOM.html. 10 Sept. 2003. eEye Digital Security. 09 Aug. 2004

IDScenter. 2003. 10 Aug. 2004 <- http://www.engagesecurity.com/products/idscenter/>.

LANguard GFI, . http://www.gfi.com/languard/. 2004. GFI Limited. 08 Aug. 2004

Microsoft Security Bulletin MS03-039 http://www.microsoft.com/technet/security/bulletin/MS03-039.mspx. 10 Sept. 2003. Microsoft. 10 Aug. 2004

Nmap Doownload http://www.insecure.org/nmap/nmap_download.html. 10 Aug. 2004

Snort's Place in a Windows 2000 Environment. 15 Apr. 2002. 10 Aug. 2004 <http://www.snort.org/docs/snort-win2k.htm#5>.

Snort.org Caswell, Brian, and Marty Roesch. http://www.snort.org/docs/. 2002. 01 Sept. 2004

Windows Packet Capture Library http://winpcap.polito.it/install/default.htm. 08 July 2004. 23 Aug. 2004

Appendix A

```
USAGE: snort [-options] <filter options>
Options:
       -A        Set alert mode: fast, full, console, or none  (alert file ale
rts only)
                 "unsock" enables UNIX socket logging (experimental).
       -b        Log packets in tcpdump format (much faster!)
       -c <rules> Use Rules File <rules>
       -C        Print out payloads with character data only (no hex)
       -d        Dump the Application Layer
       -D        Run Snort in background (daemon) mode
       -e        Display the second layer header info
       -f        Turn off fflush() calls after binary log writes
       -F <bpf>   Read BPF filters from file <bpf>
       -g <gname> Run snort gid as <gname> group (or gid) after initialization
       -h <hn>    Home network = <hn>
       -i <if>    Listen on interface <if>
       -I        Add Interface name to alert output
       -k <mode>  Checksum mode (all,noip,notcp,noudp,noicmp,none)
       -l <ld>    Log to directory <ld>
       -L <file>  Log to this tcpdump file
       -m <umask> Set umask = <umask>
       -n <cnt>   Exit after receiving <cnt> packets
       -N        Turn off logging (alerts still work)
       -o        Change the rule testing order to Pass|Alert|Log
       -O        Obfuscate the logged IP addresses
       -p        Disable promiscuous mode sniffing
       -P <snap>  Set explicit snaplen of packet (default: 1514)
       -q        Quiet. Don't show banner and status report
       -r <tf>    Read and process tcpdump file <tf>
       -R <id>    Include 'id' in snort_intf<id>.pid file name
       -s        Log alert messages to syslog
       -S <n=v>   Set rules file variable n equal to value v
       -t <dir>   Chroots process to <dir> after initialization
       -T        Test and report on the current Snort configuration
       -u <uname> Run snort uid as <uname> user (or uid) after initialization
       -U        Use UTC for timestamps
       -v        Be verbose
       -V        Show version number
       -w        Dump 802.11 management and control frames
       -X        Dump the raw packet data starting at the link layer
       -y        Include year in timestamp in the alert and log files
       -z        Set assurance mode, match on established sesions (for TCP)
       -?        Show this information
<Filter Options> are standard BPF options, as seen in TCPDump
```

Appendix B

```
server: 192.168.3.xxx
setting up session... success.
opening lsa policy... success.
server role: 3 [primary (unknown)]
names:
  netbios: SMSDMZ
  domain: WORKGROUP
quota:
  paged pool limit: 33554432
  non paged pool limit: 1048576
  min work set size: 65536
  max work set size: 251658240
  pagefile limit: 0
  time limit: 0
trusted domains:
  indeterminate
netlogon done by a PDC server
getting user list (pass 1, index 0)... success, got 7.
  ASPNET (Account used for running the ASP.NET worker process (aspnet_wp.exe))
  attributes: no_passwd
  GFI_MONITOR_USR (Built-in account for GFI LANguard SELM Monitor)
  attributes:
  Guest (Built-in account for guest access to the computer/domain)
  attributes: disabled no_passwd
  IUSR_AGP-VM-WIN2K (Built-in account for anonymous access to Internet Information Services)
  attributes: no_passwd
  IWAM_AGP-VM-WIN2K (Built-in account for Internet Information Services to start out of process
applications)
  attributes: no_passwd
  toortoor (Built-in account for administering the computer/domain)
  attributes:
  TsInternetUser (This user account is used by Terminal Services.)
  attributes: no_passwd
enumerating shares (pass 1)... got 5 shares, 0 left:
  ipc: IPC$ (Remote IPC)
  fs: D$ (Default share)
  fs: scratch ()
  fs: ADMIN$ (Remote Admin)
  fs: C$ (Default share)
Group: Administrators
SMSDMZ\toortoor
SMSDMZ\GFI_MONITOR_USR
Group: Backup Operators
Group: Guests
SMSDMZ\Guest
SMSDMZ\TsInternetUser
SMSDMZ\IUSR_AGP-VM-WIN2K
SMSDMZ\IWAM_AGP-VM-WIN2K
Group: Power Users
Group: Replicator
Group: Users
NT AUTHORITY\INTERACTIVE
NT AUTHORITY\Authenticated Users
SMSDMZ\ASPNET
Group: Web Anonymous Users
SMSDMZ\IUSR_AGP-VM-WIN2K
Group: Web Applications
SMSDMZ\IWAM_AGP-VM-WIN2K
cleaning up... success.
```

Appendix C

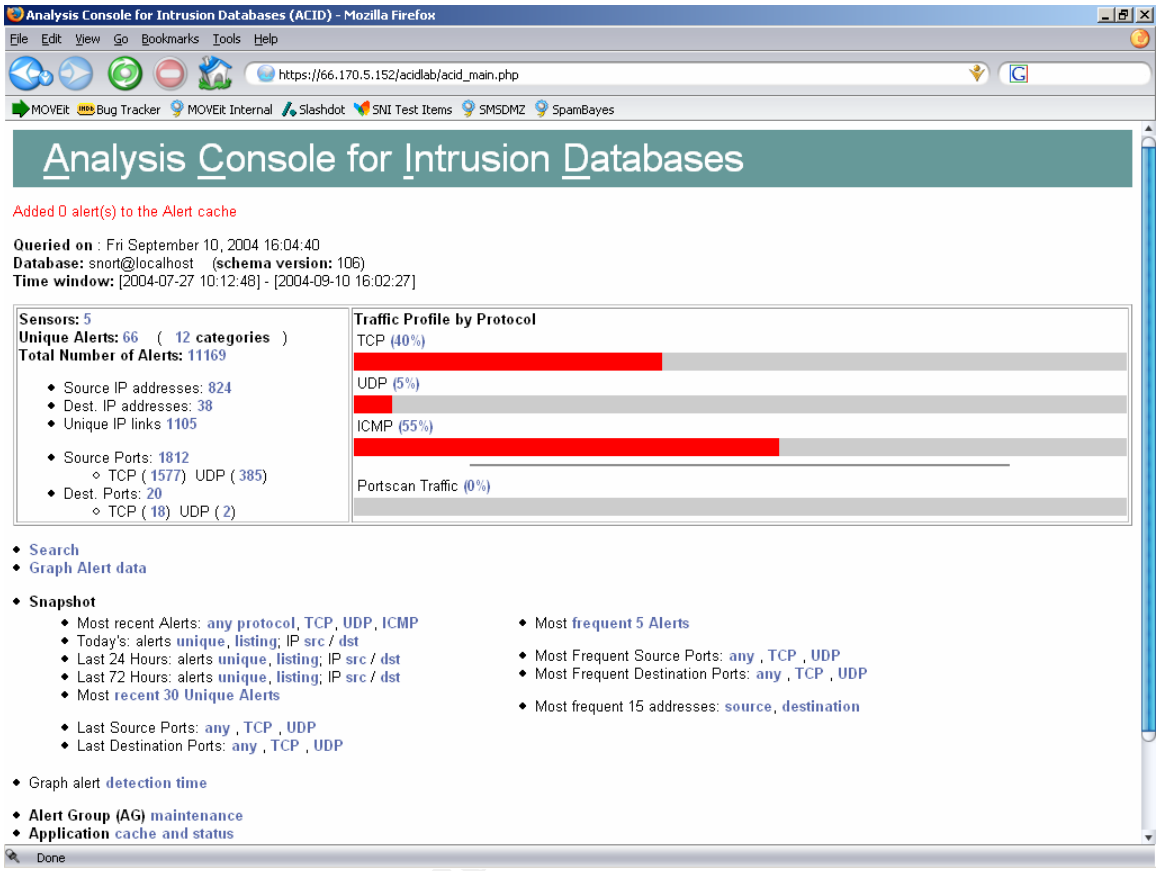

Figure 1 ACID Main page

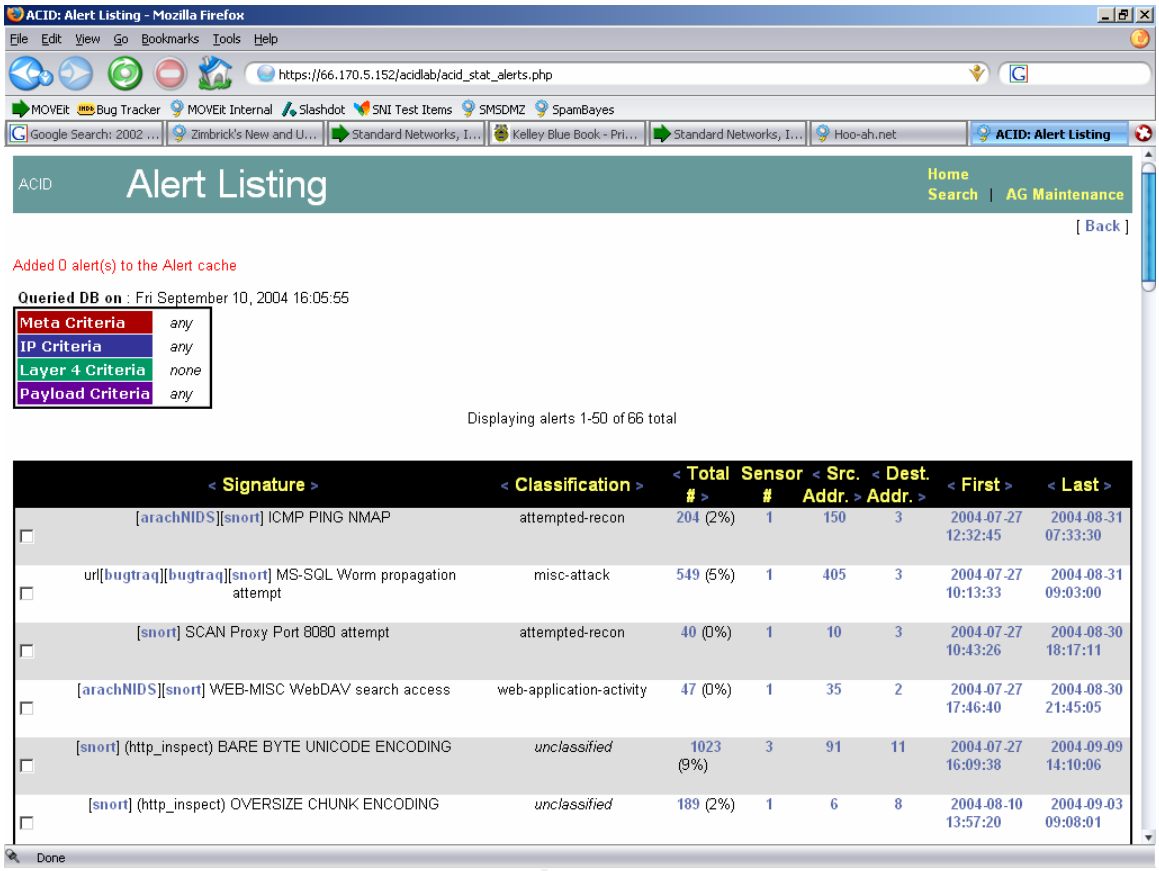

Figure 2 An alert listing from ACID

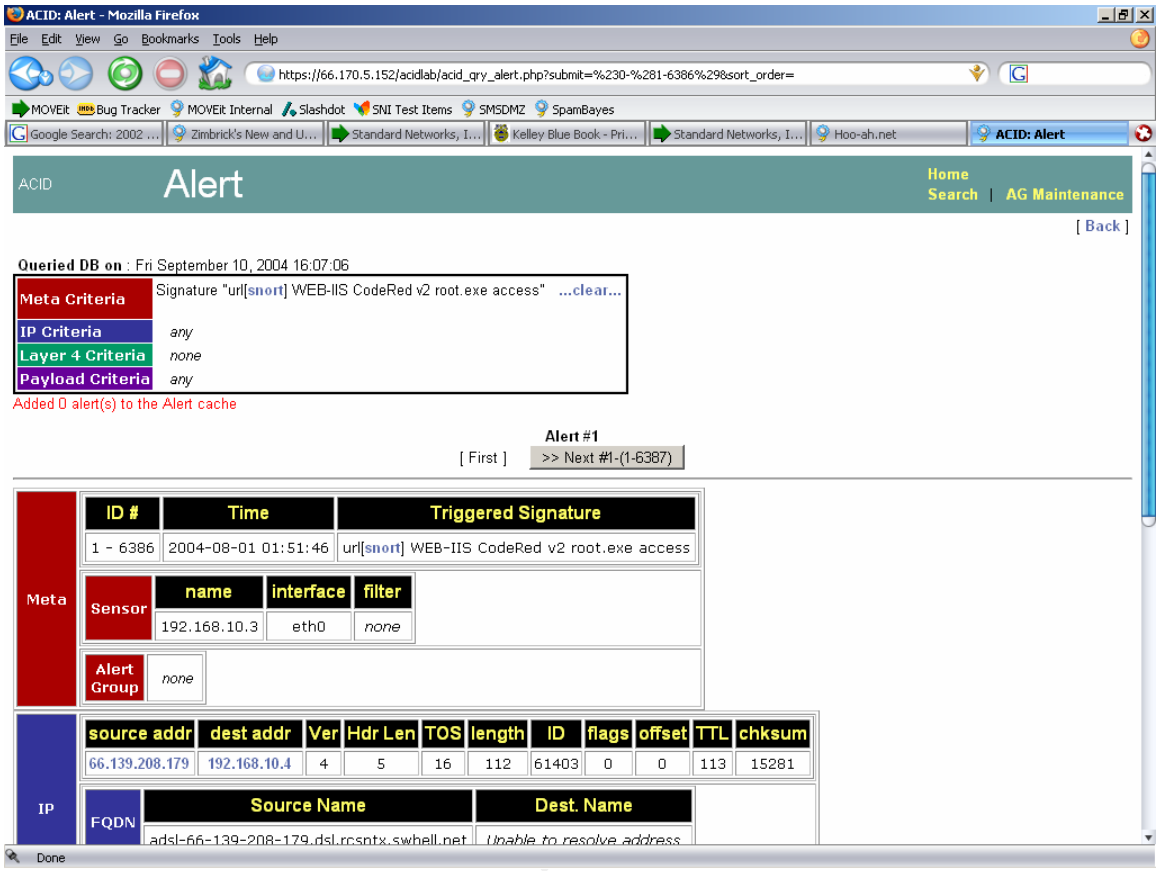

Figure 3 Detailed view of alert description

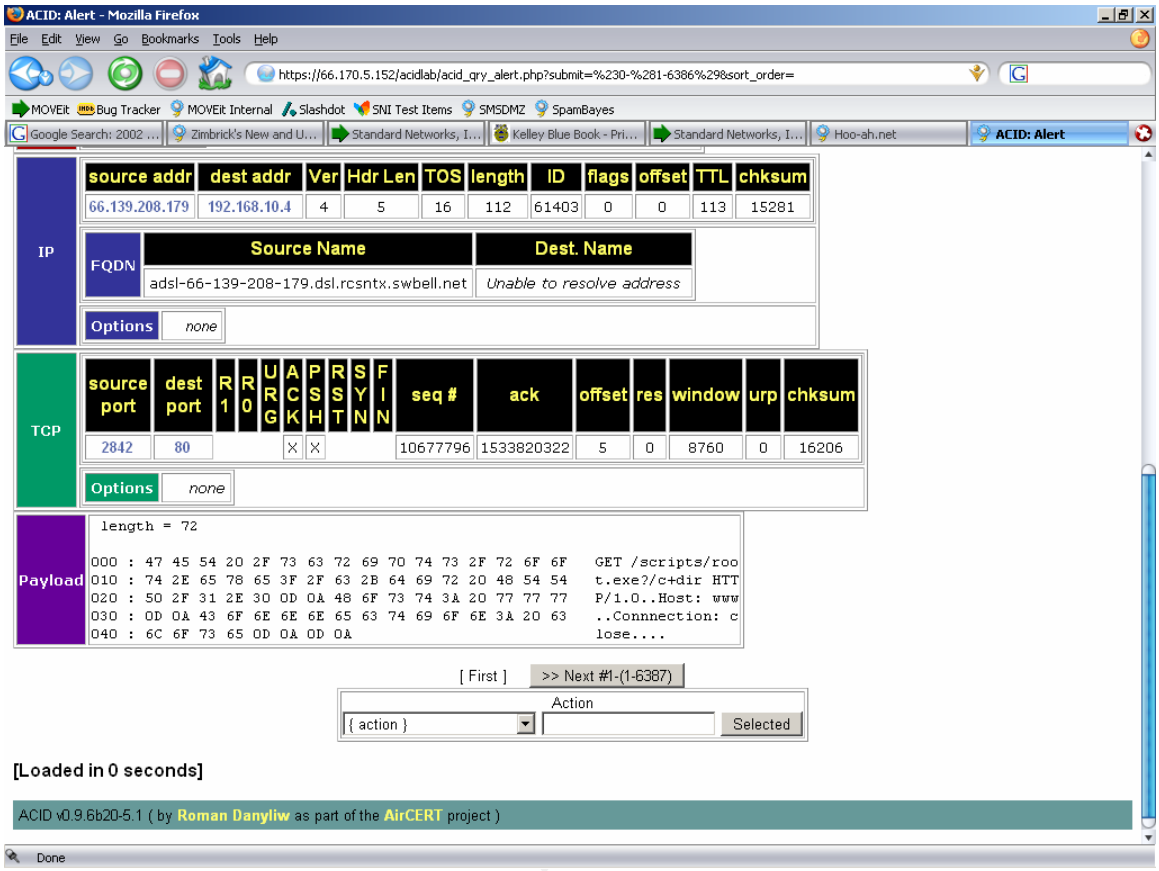

Figure 4 Detailed view of alert description (continued).