Network Security Analysis

GIAC Certified Intrusion
Analyst (GCIA)
Practical Assignment
Version 4.0

November 16th 2004

Intrusion report

Steven A. Wimmer
SANS Rocky Mountain
Intrusion In Depth
Denver, Colorado
June 5th – 10th

Abstract

This paper is part of the SANS GIAC certification process. It includes an executive summary detailing some of the challenges Universities face in securing their IT infrastructure. There is an inferred network topology provided. There are three network detects analyzed – Backdoor Q, Port 0 traffic, and scan of webroot directory traversal events. There are graphs provided for network statistics. Finally, the paper wraps up with the analysis process used in this paper.

Document Conventions

The document conventions used in this paper are as follows.

`Computer`	Operating system commands and computer output are represented in this font.
filenames	File names are represented in this font.
URL’s	Weblinks are represented in this format.

Executive Summary

The task of securing the IT infastructure of an enterprise is a enormous challenge. This task is made even greater when that enterprise is a university. These challenges include a very large number of end users, including students, administrative staff and educational staff.

Students

- Students will make up the largest body of end users. These users can be uncooperative at best, and openly hostile at the worst.
- Students may be in an IT related degree program and eager to explore, and use knowledge that they have learned.
- Students that are not in an IT related degree program and have no interest in computers other than minimal use need to complete their assignments. These users will most likely have no interest/understanding of computer security.

Administrative Staff

- Administrative staff will possibly be the most cooperative of the end users. They are there for the day to day functioning of the University and should have a vested interest in the IT infrastructure operating properly in order to carry out their tasks.

Educational staff

- Educational staff will possibly be opposed to restrictions placed on their usage of resources in the name of academic freedom.
- Educational staff will possibly place a lower priority to IT security due commitments in their academic/teaching schedule.

Resources

- Universities will have high bandwidth connections to their ISP's. This is a resource that malicious users desire for multiple reasons.
- High amounts of traffic in which a malicious user will want to hide their tracks.
- The anonymity of using computer labs available to large numbers of users.
- Universities may have trusted connections to government/military networks for research purposes, which a malicious user may want to exploit.

In this assignment I analyzed network traffic on the dates 2002.10.12 and 2002.10.13. There were a total of 1138 alerts generated from these 2 files covering both days which contained 7348 packets. I chose three sets of detects which I considered critical to look at further, per the assignment. The alerts were generated using Snort© a cutting edge Intrusion Detection System (IDS), with the latest stable rule set installed.

There was no detailed information provided on the network layout on which the traffic was captured. An inferred network layout has been provided on which the analysis is based.

The three sets of detects include a possible backdoor/Trojan, non-standard network traffic that indicates packets crafted specifically for malicious purposes, and scanning for a well known exploit left after an infection of the Code Red virus.

Following the analysis of the three detects is a summary of the traffic observed on your network over the two days analyzed. Included are graphs to help the reader understand the relationship between the hosts and the traffic on the network.

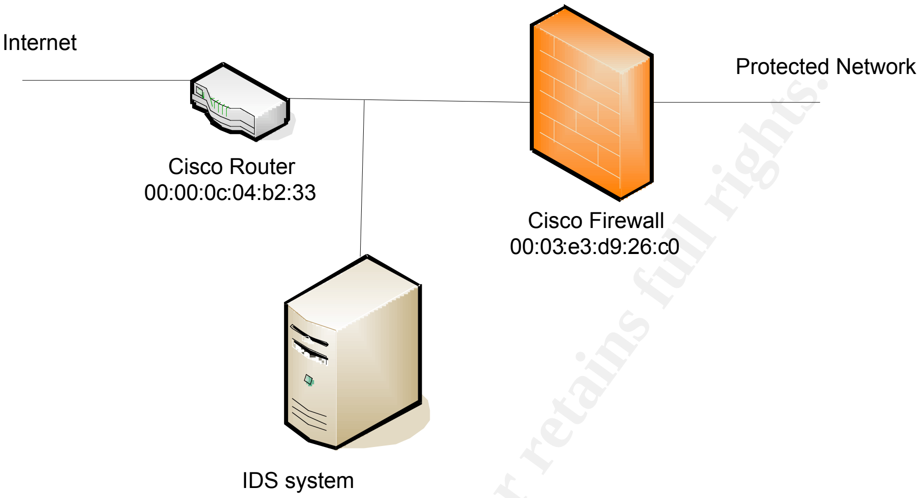
Finally there is a summary of the analysis process used to perform this analysis.

From the analysis performed, there is no evidence of a compromise on your network. There is ample evidence of active scanning for compromised devices on your network. There is also evidence of reconnaissance in the form of Operating System (OS) fingerprinting, which allows an attacker to gain information as to what operating system is running at a given destination IP address.

It is advised to ensure that all hosts are patched with the latest vendor patches. And that the firewalls have proper ingress filtering in place. An acceptable use policy should implemented for all IT resources.

Network Topology

The following network topology has been inferred, since there was no network information provided with the assignment. This topology applies to all three of the following network detects.



Network Detects

Detect 1 Backdoor Q Access

Description of the detect

This back door is considered to be “remote shell and admin tool” (1) by its author Mixter. It allows for the creation of a secure tunnel for communication. This tool is freely downloadable from the following website http://mixter.void.ru/Q-2.4.tgz. This file is not a true Trojan in the sense that when it is installed it does not get started every time the computer is restarted. In its present form it is more likely that a malicious user will keep Q in their bag of tricks, and use it as a secure/encrypted backdoor in to the compromised host. This could easily be changed by someone with moderate knowledge. The consequences of this tool being installed and used would allow for a very hard if not impossible to detect communication channel between a malicious user and the compromised host.

These packets were coming from source 255.255.255.255 with a source port of 31337 which is known as the “eleet” port in the hacker lexicon. This port has been associated with Trojans in the past, any traffic on this port should warrant further investigation.

Reason this detect was selected

This series of detects was selected due to the evidence of packet crafting and the possible consequences of an installation of a backdoor/Trojan on one or more of your devices.

Detect was generated by

This set of detects was generated by a Win32 system running Snort 2.2.0 with the 2.2.0 rules set installed with all rules enabled. This set of detects were generated by the following rule:

```
alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q
access"; flags:A+; dsize: >1; reference:arachnids,203;
sid:184;classtype:misc-activity; rev:3;)
```

The following is a sample of the actual alerts that were generated:

```
[**] BACKDOOR Q access [**]
11/11-17:26:59.016507 255.255.255.255:31337 -> 207.166.51.236:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] BACKDOOR Q access [**]
11/11-17:27:46.996507 255.255.255.255:31337 -> 207.166.44.172:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] BACKDOOR Q access [**]
11/11-18:10:40.326507 255.255.255.255:31337 -> 207.166.2.52:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

This rule is looking for any traffic inbound to the protected network with the source address having a network address of 255.255.255.0 regardless of what value is in the last octet.

Probability the source address was spoofed

With the source IP address of 255.255.255.255, the probability of the source being spoofed is absolute. This IP address is reserved and is not routable per RFC 919 October 1984.

“The address 255.255.255.255 denotes a broadcast on a local hardware network, which must not be forwarded. This address may be used, for example, by hosts that do not know their network number and are asking some server for it.

Thus, a host on net 36, for example, may:

- broadcast to all of its immediate neighbors by using 255.255.255.255
- broadcast to all of net 36 by using 36.255.255.255

(Note that unless the network has been broken up into subnets, these two methods have identical effects.)

If the use of "all ones" in a field of an IP address means "broadcast", using "all zeros" could be viewed as meaning "unspecified". There is probably no reason for such addresses to appear anywhere but as the source address of an ICMP Information Request datagram. However, as a notational convention, we refer to networks (as opposed to hosts) by using addresses with zero fields. For example, 36.0.0.0 means "network number 36" while 36.255.255.255 means "all hosts on network number 36".”

This basically means that any IP router should drop/not forward packets with this address.

Attack mechanism

There is no general consensus as to what this traffic is actually performing or attempting to perform. It has been speculated that is might be part of a worm related to Internet Relay Chat (IRC) (2).

I believe that these packets may possibly be stimulus packets designed to initiate a connection between the client and server. Below is a copy of one of the packets captured.

```
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 2B 00 00 00 00 0F 06 24 D5 FF FF FF FF CF A6  .+......$.......
0x0020: 03 9C 7A 69 02 03 00 00 00 00 00 00 00 00 50 14  ..zi..........P.
0x0030: 00 00 D9 FC 00 00 63 6B 6F 00 00 00              ......cko...
```

There were a total of 78 inbound packets over the two day period analyzed. All of these packets had a sequence number of 0, which is not seen in normal traffic (3) as it should fall between 1 and 2^32 -1 in normal traffic. (4) This is a strong evidence of crafted packets which should always be viewed as hostile. All packets had the ACK and RESET flags set which may be set to allow them to pass thru the filtering on the firewall, the firewall believing that the packets are part of a previously established session that was disrupted. All packets have a Time To Live (TTL) of 15 which is likely an indication of further crafting of the packet, and may indicate the fact that all of these packets are sourced from a single machine. In normal traffic the TTL value is determined by the operating system, and this value is decremented by a value of 1 for each hop it takes through a router. Most modern operating systems will have either an initial TTL of 64 or 128. In normal traffic the TTL value could be an indication of the location/distance to the source, but already knowing that this packet has been crafted, it is unlikely that any other intelligence can be garnered from this information. All packets contained the characters cko in the payload, which may be the stimulus for a response from a compromised host.

Without additional logs from an internal sensor, or the firewall, I am not able to determine if these packets were able to reach their intended targets. A complete TCPdump capture of the network traffic would also enable me to determine if the destination machines responded.

Correlations

Source code for Q from the authors' website.

http://mixter.void.ru/

Security Focus discussion of similar traffic.

http://www.securityfocus.com/archive/75/182244

SANS FAQ on the Q Trojan

http://www.sans.org/resources/idfaq/qtrojan.php

Author Reference for the Snort signature:

http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids203&view=signatures

SANS GCIA papers covering the Q Trojan

http://www.giac.org/practical/Trenton_Riddell_GCIA.doc

http://www.giac.org/practical/GCIA/Al_Maslowski-Yerges_GCIA.pdf

RFC that covers TCP sequence number:

http://www.faqs.org/rfcs/rfc1379.html

Evidence of active targeting

This traffic does not show any evidence of active targeting. The packets arrive over a two day period to apparently random IP address on the 207.166.x.x network. This may be indicative of slow scanning. The attacker may have input a network range in to a tool which randomly scans the range to avoid detection.

Severity
Severity is calculated with the following formula

severity = (criticality + lethality) - (system countermeasures + network countermeasures)

2 = (3 + 5) – (3 + 3)

Criticality = 3 No where in the logs is there evidence of any of the destinations replying back, although if a tunnel were created, it is unlikely that the IDS system would have detected this traffic.

Lethality = 5 If Backdoor Q is installed on a device on your network it is no longer owned/controlled by you.

System countermeasures = 3 There is no information provided on system countermeasures. It is not known what services are running (or are supposed to be running) on the destination system.

Network countermeasures = 3 There is no information provided on network countermeasures. It has been assumed that there is a firewall and it is believed that this traffic was captured by an IDS system.

Defensive recommendations

Ensure virus definitions are up to date.

Ensure all hosts have been patched with the latest vendor patches.

Ensure ingress filtering is in place on the firewall, rejecting all inbound traffic with a broadcast address as the source.

Detect #2 BAD-TRAFFIC tcp port 0 traffic

Description of the detect

These groups of 16 packets arrived over an approximately 2 second period from each source. All packets in this detect are identical. Each packet has a sequence number of 0, the SYN, ACK and Do Not Fragment flags are also set on each packet. First I thought the repeating packets were retransmit attempts of an original packet. The TCP protocol does require a retransmit of any data for which it did not receive a response. These events arrive more rapidly and in greater numbers than would be expected from normal retransmit attempts. This traffic looks like the script/tool in use is making rapid attempts to elicit a response from the destination. This could be a SYN flood/DOS attempt, but this is unlikely as the attacker would need to send a much higher volume of these events to crash a server. This is most likely an attempt at mapping your network, and OS fingerprinting.

The Time To Live (TTL) value for each packet is 47, this value can be easily crafted along with the rest of the packet. In normal traffic the TTL value is determined by the operating system, and this value is decremented by a value of 1 for each hop it takes through a router. These packets will elicit different responses from different Operating Systems (OS). This is known as OS fingerprinting, which is a very common form of reconnaissance.

Reason this detect was selected

This series of detects was selected due to the evidence of packet crafting by a tool such as hping or Nmap against your network. This is indication of active reconnaissance against your network. These detects are also suspicious in the fact that they consistently arrive in groups of 16 from one source IP address, with the source address changing for the next set of 16, with the source address for each grouping in the same class C address space.

Detect was generated by

This set of detects was generated on a Win32 system running Snort 2.2.0 with the 2.2.0 rules set installed.

This set of detects were generated by the following rule:

```
alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD-TRAFFIC tcp port 0
traffic"; flow:stateless; classtype:misc-activity; sid:524; rev:8;)
```

This signature will fire on any TCP traffic directed towards port 0, which is not used in normal network traffic.

The following is a sample of the actual alerts that were generated:

```
[**] BAD-TRAFFIC tcp port 0 traffic [**]
11/12-08:37:49.576507 211.47.255.20:35927 -> 207.166.93.224:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xCCBF8BD4  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] BAD-TRAFFIC tcp port 0 traffic [**]
11/12-08:37:52.556507 211.47.255.20:35927 -> 207.166.93.224:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xCCBF8BD4  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] BAD-TRAFFIC tcp port 0 traffic [**]
11/12-08:37:58.576507 211.47.255.20:35927 -> 207.166.93.224:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xCCBF8BD4  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+

[**] BAD-TRAFFIC tcp port 0 traffic [**]
11/12-08:38:10.586507 211.47.255.20:35927 -> 207.166.93.224:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xCCBF8BD4  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Probability the source address was spoofed
There is a possibility of the source address being spoofed. This is a TCP connection and the attacker would need to receive a response from the destination to make the effort worthwhile. A whois search on the APNIC whois server http://www.apnic.net/ for the source reveals:

```
inetnum:     211.46.0.0 - 211.49.255.255
netname:     KRNIC-KR
descr:       KRNIC
descr:       Korea Network Information Center
country:     KR
```

A further whois search on the KRNIC whois server http://www.nic.or.kr/www/english/ reveals that the source is address space that is being held in reserve.

query: 211.47.255.20
KRNIC is not a ISP but a National Internet Registry similar to APNIC.
The IPv4 address is allocated from APNIC to KRNIC.
KRNIC is holding the IPv4 address for further allocation to its member ISPs in the furture. If you have any question with the IPv4 address,
Please contact at hostmaster@nic.or.kr

Which is interesting, as to why a reserved address is being used against this network, but it does not give us any really usable information, as the whois information may not be completely current. The only real option is to send an abuse letter to the KRNIC, and see what their response if any is.

Attack mechanism

This attack works by sending TCP traffic to port 0 on each destination. According to RFC 1700 Port 0 is a reserved port. (3) According to computernetworking.com

> " port 0 sometimes takes on a special meaning in network programming, particularly Unix socket programming. In this environment, port 0 is a programming technique for specifying system-allocated (dynamic) ports.
>
> Instead of "hard-coding" a particular port number, or writing code that searches for an open port, Unix programmers simply specify port 0 as a connection parameter. That triggers the operating system to automatically search for and return the next available port in the dynamic port number range. "
>
> This programming technique does not work the same way in Microsoft Windows as it does in Unix." Normal traffic should never be directed to this port." (4)

There are freely downloadable tools available such as hping and Nmap v. 3.50 and newer will generate packets such as these in this detect. As Susan Kovacevich noted in her GCIA practical "The attacker is probably using a Windows machine since the window default for the Windows platform is 5000-9000 and in this instance it is 5840 which is hex 0x16D0." (5) The exact tool used to create these packets is not known, Nmap will run on Windows® but Nmap v.3.50 was not available until early 2004, and earlier versions of Nmap were not able to perform port 0 scans. According to the hping website "hping will run on Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MacOs X". (6)

Correlations

Snort.org signature database

http://www.snort.org/snort-db/sid.html?id=524

Everything you ever wanted to know about Nmap:

www.insecure.org

Hping available at:

www.hping.org

Information regarding port 0 is available at:

http://compnetworking.about.com/library/ports/blports_0.htm

List of assigned port numbers

http://www.iana.org/assignments/port-numbers

SANS GCIA paper covering similar traffic:

http://www.giac.org/practical/GCIA/Susan_Kovacevich_GCIA.pdf

Evidence of active targeting

This traffic is does not necessarily indicate active targeting. The fact that the events are happening over a two day period suggests a very slow scan against your network, and these events may be part of a much larger scan hitting random IP address in hopes of avoiding detection.

Severity

Severity is calculated with the following formula:

severity = criticality + lethality) - (system countermeasures + network countermeasures)

-1 = (3 + 1) – (2 +3)

Criticality = 2 These events are evidence of reconnaissance against your network.

Lethality = 2 These events are not able to compromise any system, they are only used for OS discovery. The attacker will potentially come back and attempt exploits specific to the operating systems discovered.

System countermeasures = 2 There is no information provided on system countermeasures. It is not know what services are running (or are supposed to be running) on the destination systems.

Network countermeasures = 3 There is no information provided on network countermeasures. No where in the files is there evidence of any of the destinations replying back, but without a full TCPdump of the days traffic I can not state with any certainty.

Defensive recommendations:

Ensure virus definitions are up to date.

Ensure all hosts have been patched with the latest vendor patches

Ensure ingress filtering is in place on the firewall, rejecting all inbound port 0 traffic.

Shun the source of the scan at the firewall.

Detect #3 WEBROOT DIRECTORY TRAVERSAL

Description of the detect

This series of detects indicate a scan for a very well known security hole in un-patched versions of Microsoft IIS®. These events are attempting to exploit the same security hole that was exploited by the Nimda virus. What really raises the suspicions about these events is that there are no other packets attempting the other IIS events commonly seen with Nimda such as:

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
c+dir
GET
/msadc/..%5c../..%5c../..%5c/..55../..c1../../.../winnt/system32/cmd
.exe?/c+dir 32/cmd.exe?/c+dir
```

*The entire series of Nimda related payloads has been cut for brevity

The lack of other events was verified, as there is no other traffic which generated alerts from the source going to any other destinations on the network over the time period analyzed.

Reason this detect was selected

This series of detects was selected due to the consequences of a successful response to the payload of each event, and the fact that the events walk/scan a small range of IP addresses on your network.

Detect was generated by

This set of detects was generated on a Win32 system running Snort 2.2.0 with the 2.2.0 rules set installed. The addresses in this file appear to have the IP addresses obfuscated as all packets, including those in this detect have bad checksums.

These detects were generated by the generic http inspect decoder. The snort documentation included with the installation states:

> “HttpInspect is a generic HTTP decoder for user applications. Given a data buffer, HttpInspect will decode the buffer, find HTTP fields, and normalize the fields. HttpInspect works on both client requests and server responses.” (7)

This specific alert is generated when a URL directory traversal moves past the webroot directory as stated in the documentation available from www.snort.org

> “This option generates an alert when a directory traversal traverses past the web server root directory. This generates much less false positives than the directory option, because it doesn't alert on directory traversals that stay within the web server directory structure. It only alerts when the directory traversals go past the web server root directory, which is associated with certain web attacks.” (8)

The following is a sample of the actual alerts that were generated:

```
[**] (http_inspect) WEBROOT DIRECTORY TRAVERSAL [**]
11/12-13:34:01.316507 208.45.79.122:51471 -> 207.166.87.40:80
TCP TTL:113 TOS:0x0 ID:2396 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xBB8F97F1  Ack: 0x49E4CBF  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] (http_inspect) WEBROOT DIRECTORY TRAVERSAL [**]
11/12-13:34:11.876507 208.45.79.122:51589 -> 207.166.87.157:80
TCP TTL:113 TOS:0x0 ID:2729 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xBC10C238  Ack: 0x2B93D6A5  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] (http_inspect) WEBROOT DIRECTORY TRAVERSAL [**]
11/12-13:34:01.336507 208.45.79.122:51472 -> 207.166.87.41:80
TCP TTL:113 TOS:0x0 ID:2398 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xBB906E1B  Ack: 0x4C2632A  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] (http_inspect) WEBROOT DIRECTORY TRAVERSAL [**]
11/12-13:34:04.796507 208.45.79.122:51473 -> 207.166.87.42:80
TCP TTL:113 TOS:0x0 ID:2501 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xBB9E88F4  Ack: 0x48EDC74  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] (http_inspect) WEBROOT DIRECTORY TRAVERSAL [**]
11/12-13:34:04.796507 208.45.79.122:51482 -> 207.166.87.51:80
TCP TTL:113 TOS:0x0 ID:2503 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xBBA4A0CD  Ack: 0x765A1E1D  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Probability the source address was spoofed

There is no indication of the source address being spoofed. This attack relies on a TCP response from the destinations. There is a possibility that the source is a previously compromised device, or that the attacker is using a compromised wireless network to perform the attack. Both of these two latter possibilities would allow the attacker to remain anonymous.

An ARIN look up of the source IP provides the following results:

```
Qwest Communications QWEST-DSL-BLK-01 (NET-208-45-0-0-1)
```

208.45.0.0 - 208.45.127.255

It is apparent from the results that the address belong to a block of addresses which Qwest communications uses for its' DSL service. This is very likely the address of a home user. It is not known if Qwest uses static or dynamic IP's for their DSL service.

Attack mechanism

The alerts generated all had the identical malicious payload of:

```
***AP*** Seq: 0xBC10C238  Ack: 0x2B93D6A5  Win: 0x4470  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 63 0A A9 40 00 71 06 02 4B D0 2D 4F 7A CF A6  .c..@.q..K.-Oz..
0x0020: 57 9D C9 85 00 50 BC 10 C2 38 2B 93 D6 A5 50 18  W....P...8+...P.
0x0030: 44 70 C7 4F 00 00 47 45 54 20 2F 73 63 72 69 70  Dp.O..GET /scrip
0x0040: 74 73 2F 2E 2E 25 35 63 25 35 63 2E 2E 2F 77 69  ts/..%5c%5c../wi
0x0050: 6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D 64  nnt/system32/cmd
0x0060: 2E 65 78 65 3F 2F 63 2B 64 69 72 0D 0A 69 72 0D  .exe?/c+dir..ir.
```

This attack works by getting around the security on un-patched versions of Microsoft IIS®. When IIS receives a request it checks to see if the request is attempting to access protected directories, including the parent directories of the file being accessed. A request to /scripts/../../ would be denied by this security check as it is attempting to access the directories above the /scripts directory.

This attack is attempting to exploit a hole in the above security check. Direct your attention to the following request:

```
        "GET: /scripts/..%5c%5c../winnt/system32/cmd.exe?/c+dir"
```

This request is attempting to take advantage of the fact that the Unicode %5c is the equivalent of a "/" in ASCII. An un-patched IIS system will examine the string, compare it to the rules in place and allow it to be processed. The reason this request will be processed is the way un-patched IIS systems handled Unicode characters. IIS will read this request as

```
        "GET: /scripts/..//../winnt/system32/cmd.exe?/c+dir"
```

which will be allowed to be processed, as IIS checks to ensure if a ../ appears before the Unicode character, at this point IIS has not done any Unicode decoding. As far as and un-patched IIS system is concerned, the request is not malicious as it does not violate any of its security rules so it is ok to be processed. This exploit takes advantage of the sequence in which IIS processes Unicode encoding. IIS will ignore the extra / in the string which will result in the request being processed. The Nimda virus exploited this by using placing the string "..%5c.." in it's requests. This attack is attempting to build on that exploit by placing additional "%5c" characters in the string. IDS rules had been created to detect this "..%5c.." string. The placement of additional %5c, is most likely an attempt to bypass any IDS rules looking to catch this traffic.

Correlations

Cert advisory on Nimda

http://www.cert.org/advisories/CA-2001-26.html

Analysis of the Nimda worm.

http://aris.securityfocus.com/alerts/nimda/010919-analysis-nimda.pdf

Vendor patch

http://www.microsoft.com/technet/security/bulletin/MS01-026.asp

Snort documentation regarding the httpinspect from Snort.org.

http://www.snort.org/docs/snort_manual/node10.html

SANS practical on the Nimda worm

http://www.giac.org/practical/Mark_Embrich_GCIA.htm#_Toc1531241

Evidence of active targeting
This traffic is does not necessarily indicate active targeting. These events may be directed only against this network, or may be part of a broader scan.

Severity
Severity is calculated with the following formula:

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Severity = (3+5) – (3+3) = 2

Criticality = 3 It is not known what operating system resides on the destinations.

Lethality = 5 These events could lead to a compromise of the destinations, if they have not been patched with all the latest patches available.

System countermeasures = 3 There is no information provided on system countermeasures. This is very well known exploit, and the destinations “should” be patched. Looking through the raw log, there is no indication that the destination hosts ever replied to these events. The hosts may be running a non vulnerable version, or may not be running IIS at all.

Network countermeasures = 3 There is no information provided on network countermeasures. I have made the assumption that there is a firewall in place in the above diagram

Defensive recommendations

Ensure virus definitions are up to date.

Ensure all hosts have been patched with the latest vendor patches

Ensure ingress filtering is in place on the firewall, allowing port 80 traffic to authorized web servers.

Shun the source of the scan at the firewall.

Summary of network traffic observed

The following is a summary of the traffic in the files for the two days analyzed.

Chart of top 10 talkers by IP address

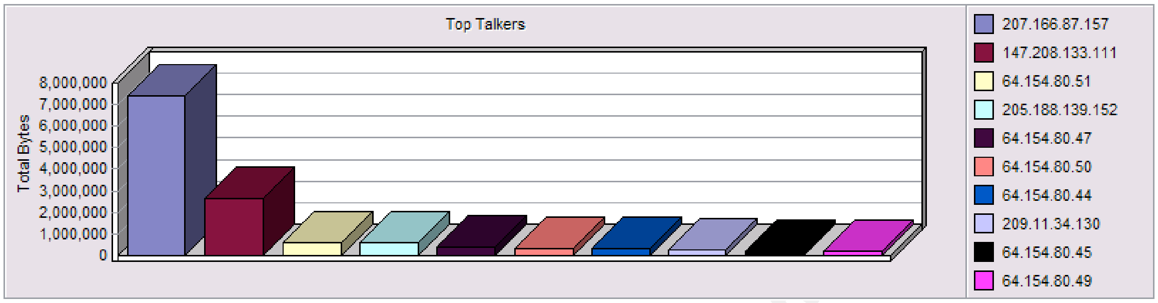
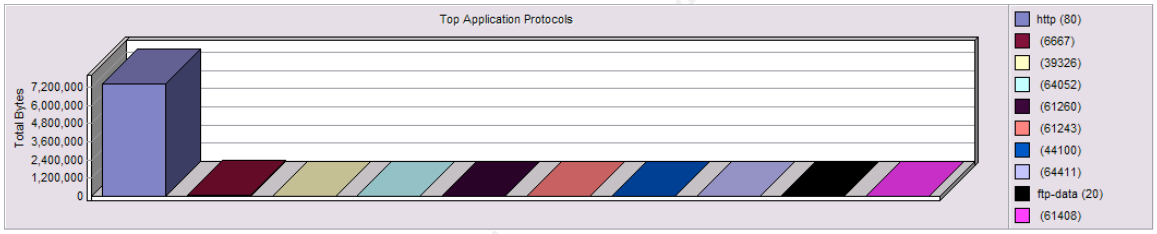
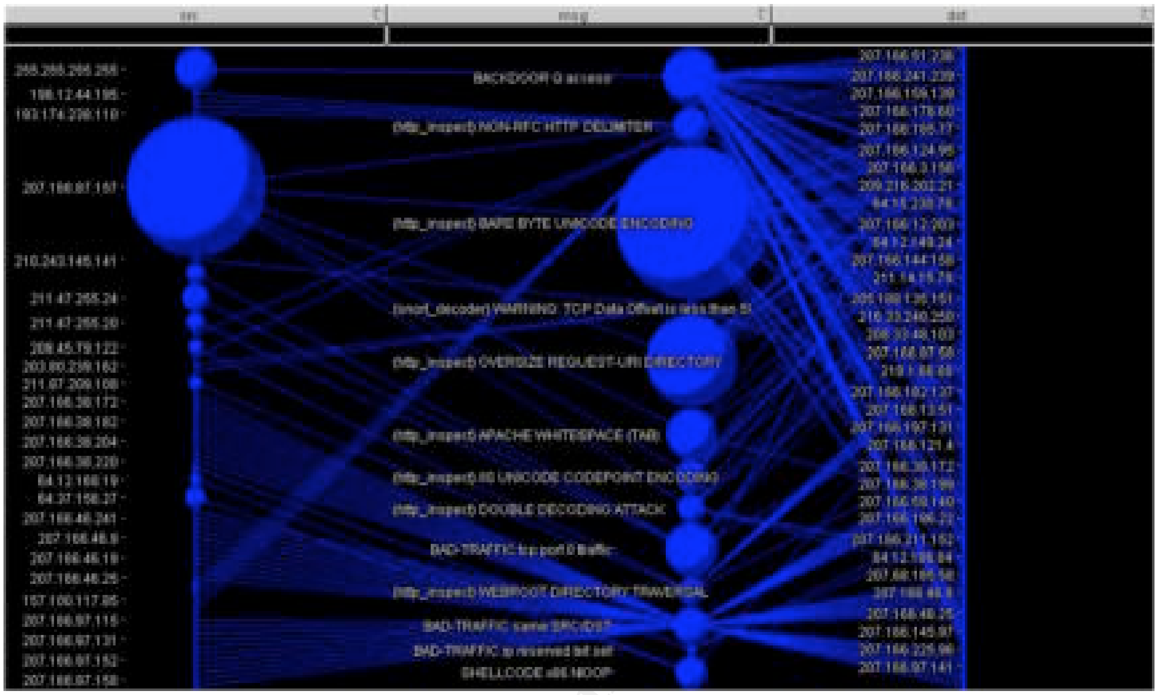


Chart of top 10 protocols observed on the network



The majority of the traffic in the files analyzed was traffic http traffic bound for port 80 and was generated outbound by host 207.166.87.157. Most other traffic was using ephemeral ports for communications. The three most suspicious IP address were 255.255.255.255, a broadcast address that should never be seen inbound on the network. Source 208.45.79.122, which generated a IIS vulnerability scan against a number of hosts on the network. Source 211.47.255.20 generated a scan against the network using port 0 which is a reserved port. All three of these suspicious sources were investigated in depth in the second portion of this paper.

Link graph visualization of traffic included in the files analyzed



This image shows the relationship between source IP's the alerts that were triggered, and the destination IP's. By looking at this data I was able to get a better understanding of the traffic analyzed.

Analysis Process

This analysis was conducted as part of the SANS GCIA 4.0 certification attempt as a result of SANS Rocky Mountain 2004, Denver Colorado June 10th – 16th.

The log files for this analysis were downloaded from http://isc.sans.org/logs/Raw As required for the assignment. Two days worth of log files were downloaded `2002.110.12` and `2002.10.13`. Two days logs were chosen to provide me with a larger set of traffic to analyze. By using a larger dataset, I was able to get a better view of the traffic on the network.

These logs were merged in to one file using the merge function in Ethereal version 0.10.7©. When using the merge function, I loaded the later file first, and then merged it with the earlier file. This was done as my first attempt to merge the data in the reverse order resulted in Ethereal not being able to properly sort the data by date/time of the events.

Winpcap version 2.3 had also been downloaded and installed from http://winpcap.polito.it.

Once the files were merged they were run through Snort version 2.2.0 with the 2.2.0 rule set installed. All rules were enabled for the analysis. The snort command that was used on the merged file was:

```
snort –X –c c:\Snort\rules\snort.conf -l c:\Snort\log –r
c:\2002.10.1213
```

The switches in the command string are as follows

`-X` Tells Snort to generate output in hex format.

`-c` Tells Snort which snort.conf file to read from. *Note: Originally I was not able to run snort with the default snort.conf file created during the installation. A co-worker advised me that he had previously found a reference to this problem in a posting on the Internet. The solution was to place the string `config checksum_mode: none` in the snort.conf file immediately after the absolute path to the rules directory. Once this was added to the snort.conf file snort worked properly.*

`-l` Tells Snort what directory the results should be logged to. An absolute path is required

`-r` Tells Snort which file to read from. An absolute path is required

The string `output alert_csv: snort.csv default` had been placed in the output section of the snort.conf file. This was done to enable output to a .csv file which could be opened in Microsoft Excel®. This was done to allow easier sorting of the events for analysis.

This analysis was conducted on an IBM Thinkpad® T30 running Windows XP Professional Edition®. It is not known what hardware or software was used to capture the files being analyzed.

Snort generated 1138 alerts from 7348 packets in the merged files. I confirmed the alert and packet numbers by running each file individually through Snort® and Ethereal® and adding the results manually.

The Top 10 talkers and top ten port graphs were generated by importing the merged file in to Distinct Network Monitor specifically for this purpose. The graphs were taken from the HTML report generated from the data.

The link graph showing connections between the top 10 talkers was created in Visual Advisor Analyst Workbench to show a graphical representation of the traffic patterns in the files analyzed.

References Cited

Source code for Q from the authors' website.

(1) http://mixter.void.ru/ (Sept 10, 2004)

Security Focus discussion of similar traffic.

(2) http://www.securityfocus.com/archive/75/182244 (Nov 4, 2004)

A good overview of the TCP/IP protocols:

(3) http://www.linuxsecurity.com/resource_files/documentation/tcpip-security.html (Nov 15, 2004)

The RFC that covers TCP sequence numbers

(4) http://www.faqs.org/rfcs/rfc1379.html (Nov 4, 2004)

(5) Susan Kovacevich's practial regarding port 0 traffic
http://www.giac.org/practical/GCIA/Susan_Kovacevich_GCIA.pdf
(Nov 4, 2004)

(6) Information on hping
http://www.hping.org/ (Nov 15, 2004)

Snort documentation from Snort.org
(7) http://www.snort.org/docs/snort_manual/node10.html (Sept 1, 2004)

Snort documentation from Snort.org
(8) http://www.snort.org/docs/snort_manual/node10.html (Sept 1, 2004)