Intrusion Detection In Depth

GCIA Practical Assignment

Version 3.1

Dongmei Huang

Table of Content

Assignment 1 – Describe the state of Intrusion Detection

For this assignment, I will focus on two IDS evasion tool – snot and stick. These two tools implement the same theory for evading the radar of intrusion detection system – starving the resource of IDS system [1]. Both tools make shear number of alarms to consume the hardware resource of intrusion detection system and the human resource of IDS analyst.

One of the IDS weaknesses these tools utilize was stateless analysis to TCP protocol. After these tools demonstrate their power at 2000 and 2001, IDS product, such as snort has developed their stateful analysis ability by implementing TCP stream4 preprocessor. The goal of my paper is to verify whether these tools can still defeat the current evolved IDS product.

Stick and Snot

Stick was developed by Coretez Giovanni at 2000. His white paper *fun with packet* [2] stated that this tool is written to demonstrate some flaws in IDS product. It uses the Snort rule set and produces a C program via lex that when compiled will produce an IP packet capable of triggering that rule from a spoofed IP range (or all possible IP addresses) into a target IP range. The tool currently produces these at about 250 alarms per second [4].

Snot was developed by Sniph in 2001. It is an arbitrary packet generator that uses snort rules files as its source of packet information. It can trigger alerts by sending packets with the signatures that snort is looking for and evade the detection by randomize the information in the packets.

Why these tools work

Intrusion detection system, as crucial security device, unfortunately, has resource limitation just as other computer systems. When the resource is exhausted, it becomes a fail-open system; attacks can happen stealthy just like the IDS is not there. Especially IDS is a passive packet sniffer listening on all the packets on the network, which make it inherently vulnerable to denial of service. When the IDS product run out of resource, it will start to drop packets and become blind to the coming real attack.

When attacker intentionally makes noise – craft huge amount of packets with highest speed to trigger the IDS system to generate alert, he can eventually exhaust the resource of the IDS system so that the real attack can go without notice of IDS product and IDS analyst.

The ability of triggering false alert with crafted packet relies on the weakness of most signature based IDS product. These weakness include:

- Stateless analysis of TCP packets. Old IDS systems have design flaw of not monitoring the state of TCP connection. This result in crafted TCP packet can trigger alert despite the fact that the TCP session is not actually established and no harm could be made [2].

- Lack of recording. Most IDS systems do not log the packets transmitted before and after the packet that trigger the alert. Only seeing the individual packet without context makes IDS analyst hard to validate the attack and determine the attacking result [2].

Thus, the attackers can make use of these weaknesses to trigger huge amount of false alerts to keep the IDS system running out of resource and keep the IDS analyst busy with identifying real alert from large amount of false alert. When this happen, the real attack can go unnoticed much easier.

Create a simulate environment for testing

I installed both the tools in Redhat Linux 7.3 on an IBM T21 laptop with a 3com 3cXFE574BT 10/100 PC LAN card. Snot requires *libnet* but did not work with the latest *libnet version 1.1*. I downloaded the *libnet-1.0.2* from packetfacoty [3]. After *./configure, make, make install*, it is ready to go. Then I downloaded the Snot version 0.91 package from SecurityFocus tools archive [5]. It was nicely compiled and the snot executable was generated. Stick can be downloaded from SecurityFocus archive [6] as well. After just executing *./create-stick*, the stick binary was created.

Both the tools provide a snort rule file with all the snort 1.7.0 ruleset dated at 03/15/2001(Snot) or 01/08/2001(stick). But they both can work with new snort rule set as long as the tool can understand their format. I made a new rule file with the latest snort ruleset for snot and it was working fine with it. But I could not get Stick worked with the new ruleset, I have to use the old ruleset coming with Stick.

My plan is targeting a snort sensor in the Internet. I wanted to test the tools in real world instead of in a local network environment. Testing in this way will provide a much more accurate measurement of how effective the tools are. My friend has a snort sensor connecting to the Internet, it thus became my testing target. Plus the target network is pretty good firewalling, no any service open to Internet and inbound ICMP is drop. It is pretty quiet and rarely gets attack. The snort sensor is installed in a Dell GX150 workstation, installed with snort version 1.8.7(Build 128) and snort.conf v1.77.2.20 dated 2002/08/12, all the rule files are included in snort.conf.

Testing methodology

I put the Laptop directly connect to Internet through a cable modem. Initially I put the Laptop behind my home firewall, but after I found out the firewall blocks lots of TCP outgoing packets so that the snort senor could not see any incoming TCP packets.

I used the LINUX utility “top” to measure the CPU busy rate in the target sensor. I executed command “top –d 1 –b > tophistory” to record the system performance for the whole testing process.

I ran snort daemon with and without the “-z” option. This option is added for defeating the DoS attack against the IDS system itself [7]. Through this test, I will be able to examine how effective is the snort stateful inspection feature.

To make snort working with best performance, I use binary log format instead of the traditional human readable log format.

The snort ruleset in my target sensor enable the following preprocessors and options:

```
preprocessor frag2
preprocessor stream4: detect_scans, disable_evasion_alerts
preprocessor stream4_reassemble
preprocessor http_decode: 80 -unicode -cginull
preprocessor rpc_decode: 111 32771
preprocessor bo
preprocessor telnet_decode
```

Stick and Snot in action

With my hardware and software resource, snot is able to generate 20,000 – 60,000 packets per minute; stick is able to generate 45,000 – 90,000 packets per minute in my local machine. Around 3/4 of the forge packets are TCP packets because most of the snort rule dealing with TCP packets. While both tools read the snort rules as input, they generate corresponding amount of TCP, UDP and ICMP packets. I ran tcpdump in the remote target to check how many packets are able to reach the other end. I found only around 1/5 -1/10 generated packets could be delivered to my target. The "high speed" Internet connection through cable modem cannot catch up the speed of these tools generating packets.

I ran "top" in batch mode to record the system performance while it is under attack. Then I started the snort daemon with command:
Snort –D –c /path/snort.conf –d –b –I eth1. After each test, I restart the daemon so that the alert from different tool can be stored separately.

Each of the tools ran 5 minutes against the target. The following table shows the alerts each tool generated in just 5 minutes:

	Stick	Snot
Total amount of alerts	41337	20441
ICMP alerts	36035	5758
UDP alerts	4186	4198
TCP alerts	1116	10485
Kinds of Signatures	144	512

The TCP alerts are related to TCP based application such as web, SMTP, POP, FTP and database connection. These alarm are definitely false alarms because there is no such service listening on the target. The prerequisite of attack over TCP, establishing session through three ways handshake, is impossible to be satisfied, thus the attack over TCP is impossible to happen.

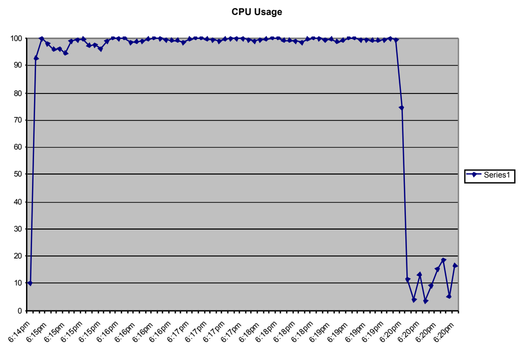
Following are 2 sample false alerts triggered by the tools:

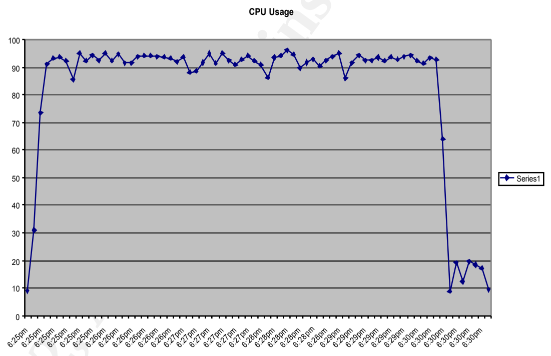
```
[**] [1:1695:3] ORACLE truncate table attempt [**]
```

```
[Classification: Generic Protocol Command Decode] [Priority: 3]
08/29-18:25:01.952624 my.home.net.ip:55408 -> my.friend.net.ip:1521
TCP TTL:48 TOS:0x0 ID:42104 IpLen:20 DgmLen:1444
***A**** Seq: 0x1DA3C3AC  Ack: 0x4E3055C6  Win: 0x1440  TcpLen: 20

[**] [1:340:3] FTP EXPLOIT overflow [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
08/29-18:25:02.050022 my.home.net.ip:6821 -> my.friend.net.ip:21
TCP TTL:202 TOS:0x0 ID:40825 IpLen:20 DgmLen:739
***A**** Seq: 0x9FF32EFC  Ack: 0xBBAF00  Win: 0xF558  TcpLen: 20
```

The following figures show the CPU usage while snort sensor was under attack:





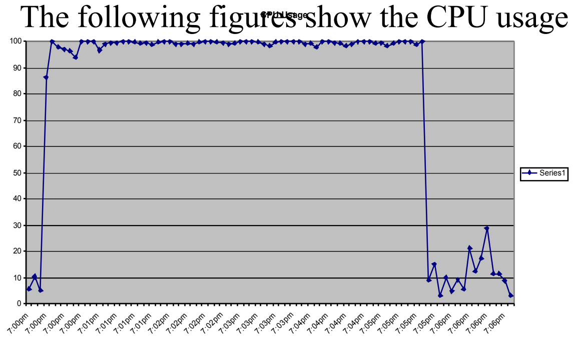
CPU Usage when stick in action CPU Usage when snot in action

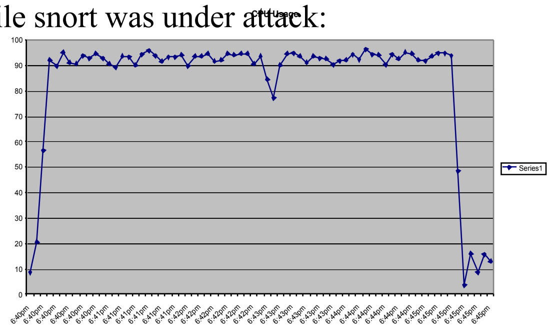
Now, I started to test the '-z' option and see how effective it is. I restart the daemon with the command *Snort –D –c /path/snort.conf –d –b –I eth1 –z.*
Then I ran each of the tools against the target for 5 minutes. The following table shows the alerts each tool generated:

	Stick	Snot
Total amount of alerts	42991	10290
ICMP alerts	38620	5493
UDP alerts	4371	4121
TCP alerts	0	676
Kind of Signatures	91	194

The TCP Alert triggered by Snot are all related to spp_stream4 scan detection and spp_stream4: TTL EVASION (reassemble) detection, none of them are related to TCP applications such as web, POP, SMTP, Oracle database connection etc.

The following figures show the CPU usage while snort was under attack:





CPU Usage when stick in action CPU Usage when snot in action

Explanation of testing result

The test results demonstrated that snort successfully defeat the attacking with forge TCP packet. It does not generate alerts while the TCP session is not pre-established. By having the "-z" option, snort defeats denial of service attack to the IDS analyst.

But the ICMP and UDP packets are still able to trigger huge amount of alerts and keep the system very busy. In this means, snort is still vulnerable to denial of service attack to its hardware resource it relies on. The figures showed that the amount of the forge packets from just one source could keep the CPU usage to almost 100% all the time. If more instances of these tools are coordinated together to attack the sensor from different sources, they can definitely bring the sensor down and leave thousands of false alert for the IDS analyst to verify.

Countermeasure

As the test has illustrated, Snort has developed its capability to deal with the DoS attack the NIDS itself by adding the stream4 preprocessor with the "-z est" option. This allows snort to track the state of the TCP connection before sending the packets to the detection engine. Another wonderful feature is that with this option, only attacks getting cooperative response will be alerted. Cooperative response means the target response to the attack with neither RESET nor FIN packets [7]. This is so called stateful inspection of IDS product. It greatly increases the accuracy of the alarm system and decreases the manpower for verifying the impact of attacks. Thus the IDS analyst can concentrate in the real threat in terms of attack over TCP. The stateful inspection can cooperate with protocol analysis to provide better attack impact validation [7]. For example, the snort sensor can examine the return code of the web server elicited by a malicious HTTP request, if the return code from the web server is "403 forbidden" or "404 not found", then the sensor will not fire an alert or still alert but give the event a lower priority.

But at the same time, the huge amount of ICMP packets and UDP packets generated from both tools are still able to hammer the sensor. As the tables show, 53-89% of the packets that triggered alerts were ICMP packets. To protect the sensor from Denial of Service attack, the border router should block most of the incoming ICMP packets so that they cannot reach the sensor.

In terms of defeating attacks over UDP, one of the possible strategies is applying the UDP stateful inspection idea from Checkpoint firewall if the attack relies on two-ways communication. If the destination of the UDP packets does not response in certain amount of time, the sensor can treat this server is not cooperative with the attack thus do not raise the alert. This will again defeat the denial of service attack to the IDS analyst.

But the UDP stateful inspection idea cannot defeat the attack with one-way UDP connection attack such as wake up call or remote commands sent to zombies. It is hard to distinguish the attack from this kind of tools and the real attacks. Only by sending this kind of packet can trigger huge amount of false alert and keep the IDS system very busy. In this case, the IDS system has to seek protection from firewall or router by blocking incoming UDP packets.

List of Reference:

[1] Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection
URL: http://www.robertgraham.com/mirror/Ptacek-Newsham-Evasion-98.html

[2]. Fun with packet
URL: http://www.eurocompton.net/stick/papers/Peopledos.pdf

[3]. Packetfactory – download libnet
URL: http://www.packetfactory.net/libnet/dist/deprecated/libnet-1.0.2.tgz .

[4] SecurityFocus tools archive - IDS -Evasion
URL:http://online.securityfocus.com/tools/category/112

[5] SecurityFocus tools archive snotv0.91
URL: http://online.securityfocus.com/tools/1983

[6] SecurityFocus tools archive - Stick
http://online.securityfocus.com/data/tools/stick.tgz

[7] Snort User Manual
http://www.snort.org/docs/writing_rules/chap2.html#tth_sEc2.4.7

[8] Check Point FireWall-1TM White Paper
URL: http://www.firstvpn.com/papers/chkpnt/CheckpointWhitepaper.pdf

< -- End of Reference -- >

Assignment 2 – Network Detects

Detect #1 is from a message posted in intrusion archive of Incidents.org, subjected Sqlsnake variant or hack attempt over port 139 [1]. There was only one reply to the message saying that this seemed like infected by spida worm [2]. After going through a detail analyze to the posted log, I conclude that this could be a variant of sqlsnake trying to infect the target by Named Pipe instead of by TCP/IP port 1433, which is the common way; in addition, the target machine was not infect eventually due to it did not authenticate to the user sa with blank password.

Detect #2 is from http://www.incidents.org/logs/raw. It is a subset of tcpdump log from the tcpdump binary raw log of July 7, 2002 and July 8, 2002. The detects are packets from source address 255.255.255.255, the snort rule it triggered is "Backdoor Q access". Since the packets can quite confuse the analyst at the first glance, what I did is excluding all the other possibility to reveal the real purpose of the attack.

Detect #3 is from my home network. It is a formmail perl script and cgi probe by the email spammer. As the Formmail package has become a favorite tool of spammers, the formmail attack was ranked the third of the top 10 attacks in quarter one, 2002 in security focus [3].

Detect #1: A variant of Sqlsnake

Note 1: The packet trace is separated to several steps for the convenience of referring. The key binary codes are highlighted in color for convenience of interpreting. They do not come with the original log posted on the web.

Note 2: The Netbios and SMB packets are interpreted according to NetBios specification, CIFS specification and some other documents [4][5][6][7][8].

```
< -- Beginning of the log -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
< -- Step 0 -- >
05/30-05:55:36.550992 203.235.44.100:3399 -> myip:1433
TCP TTL:116 TOS:0x0 ID:35252 IpLen:20 DgmLen:44 DF
******S* Seq: 0x8B5E2E6A  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 2C 89 B4 40 00 74 06 15 95 CB EB 2C 64 xx xx  .,..@xxxxxxx,d.,
0x0020: xx xx 0D 47 05 99 8B 5E 2E 6A 00 00 00 00 60 02  ...G...^.j....`.
0x0030: 20 00 43 FB 00 00 02 04 05 B4 00 00              .C.........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-05:55:36.551096 myip:1433 -> 203.235.44.100:3399
TCP TTL:128 TOS:0x0 ID:3256 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x484F7906  Ack: 0x8B5E2E6B  Win: 0x4470  TcpLen: 24
TCP Options (1) => MSS: 1460
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 2C 0C B8 40 00 80 06 86 91 xx xx  xx xx  CB EB  .,..@xxxxxx,....
```

```
0x0020: 2C 64 05 99 0D 47 48 4F 79 06 8B 5E 2E 6B 60 12  ,d...GHOy..^.k`.
0x0030: 44 70 5E 24 00 00 02 04 05 B4 00 00              Dp^$........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-05:55:36.806116 203.235.44.100:3399 -> myip:1433
TCP TTL:116 TOS:0x0 ID:41908 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x8B5E2E6B  Ack: 0x484F7907  Win: 0x2238  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 28 A3 B4 40 00 74 06 FB 98 CB EB 2C 64 xx xx  .(..@xxxxxxx,d.,
0x0020: xx xx  0D 47 05 99 8B 5E 2E 6B 48 4F 79 07 50 10  ...G...^.kHOy.P.
0x0030: 22 38 98 19 00 00 00 00 00 00 00 00              "8..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-05:55:36.806215 203.235.44.100:3399 -> myip:1433
TCP TTL:116 TOS:0x0 ID:42164 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x8B5E2E6B  Ack: 0x484F7907  Win: 0x2238  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 28 A4 B4 40 00 74 06 FA 98 CB EB 2C 64 xx xx  .(..@xxxxxxx,d.,
0x0020: xx xx  0D 47 05 99 8B 5E 2E 6B 48 4F 79 07 50 11  ...G...^.kHOy.P.
0x0030: 22 38 98 18 00 00 00 00 00 00 00 00              "8..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-05:55:36.806298 myip:1433 -> 203.235.44.100:3399
TCP TTL:128 TOS:0x0 ID:3257 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x484F7907  Ack: 0x8B5E2E6C  Win: 0x4470  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 28 0C B9 40 00 80 06 86 94 xx xx  xx xx  CB EB  .(..@xxxxxx,....
0x0020: 2C 64 05 99 0D 47 48 4F 79 07 8B 5E 2E 6C 50 10  ,d...GHOy..^.lP.
0x0030: 44 70 75 E0 00 00 00 00 00 00 00 00              Dpu.........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-05:55:36.807144 myip:1433 -> 203.235.44.100:3399
TCP TTL:128 TOS:0x0 ID:3258 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x484F7907  Ack: 0x8B5E2E6C  Win: 0x4470  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 28 0C BA 40 00 80 06 86 93 xx xx  xx xx  CB EB  .(..@xxxxxx,....
0x0020: 2C 64 05 99 0D 47 48 4F 79 07 8B 5E 2E 6C 50 11  ,d...GHOy..^.lP.
0x0030: 44 70 75 DF 00 00 00 00 00 00 00 00              Dpu.........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-05:55:37.062100 203.235.44.100:3399 -> myip:1433
TCP TTL:116 TOS:0x0 ID:50868 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x8B5E2E6C  Ack: 0x484F7908  Win: 0x2238  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 28 C6 B4 40 00 74 06 D8 98 CB EB 2C 64 xx xx  .(..@xxxxxxx,d.,
0x0020: xx xx  0D 47 05 99 8B 5E 2E 6C 48 4F 79 08 50 10  ...G...^.lHOy.P.
0x0030: 22 38 98 17 00 00 00 00 00 00 00 00              "8..........
< -- End of step 0 -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
< -- Step 1 -- >
05/30-06:51:08.257111 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:22801 IpLen:20 DgmLen:44 DF
******S* Seq: 0x8CA2DFE1  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 2C 59 11 40 00 74 06 46 38 CB EB 2C 64 xx xx  .,Y.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 DF E1 00 00 00 00 60 02  ...;..........`.
0x0030: 20 00 97 59 00 00 02 04 05 B4 00 00              ..Y........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:08.257226 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3310 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x79E6C7B5  Ack: 0x8CA2DFE2  Win: 0x4470  TcpLen: 24
TCP Options (1) => MSS: 1460
0x0000:xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 2C 0C EE 40 00 80 06 86 5B xx xx  xx xx  CB EB  .,..@xxxx[.,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C7 B5 8C A2 DF E2 60 12  ,d...;y.......`.
0x0030: 44 70 31 3C 00 00 02 04 05 B4 00 00             Dp1<........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:08.511975 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:23057 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x8CA2DFE2  Ack: 0x79E6C7B6  Win: 0x2238  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 28 5A 11 40 00 74 06 45 3C CB EB 2C 64 xx xx  .(Z.@xxxx<..,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 DF E2 79 E6 C7 B6 50 10  ...;......y...P.
0x0030: 22 38 6B 31 00 00 00 00 00 00 00 00             "8k1........

< -- End of step 1 -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

< -- Step 2 -- >
Text filled in Green: Netbios Session Command
05/30-06:51:08.513002 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:23313 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x8CA2DFE2  Ack: 0x79E6C7B6  Win: 0x2238  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 70 5B 11 40 00 74 06 43 F4 CB EB 2C 64 xx xx  .p[.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 DF E2 79 E6 C7 B6 50 18  ...;......y...P.
0x0030: 22 38 25 0C 00 00 81 00 00 44 20 43 4B 46 44 45  "8%......D CKFDE
0x0040: 4E 45 43 46 44 45 46 46 43 46 47 45 46 46 43 43  NECFDEFFCFGEFFCC
0x0050: 41 43 41 43 41 43 41 43 41 43 41 00 20 45 49 45  ACACACACACA. EIE
0x0060: 50 46 44 46 45 46 44 46 47 46 43 43 41 43 41 43  PFDFEFDFGFCCACAC
0x0070: 41 43 41 43 41 43 41 43 41 43 41 41 41 00        ACACACACACAAA.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

Text filled in Green: Netbios Session Response
05/30-06:51:08.513119 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3311 IpLen:20 DgmLen:44 DF
```

```
***AP*** Seq: 0x79E6C7B6  Ack: 0x8CA2E02A  Win: 0x4428  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 2C 0C EF 40 00 80 06 86 5A xx xx  xx xx  CB EB  .,..@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C7 B6 8C A2 E0 2A 50 18  ,d...;y......*P.
0x0030: 44 28 C6 EC 00 00 82 00 00 00 00 00              D(..........

< -- End of step 2 -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

< -- Step 3 -- >
Text filled in Green: SMB session request and response
05/30-06:51:08.770488 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:23569 IpLen:20 DgmLen:214 DF
***AP*** Seq: 0x8CA2E02A  Ack: 0x79E6C7BA  Win: 0x2234  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 D6 5C 11 40 00 74 06 42 8E CB EB 2C 64 xx xx  ..\.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E0 2A 79 E6 C7 BA 50 18  ...;.....*y...P.
0x0030: 22 34 72 19 00 00 00 00 00 AA FF 53 4D 42 72 00  "4r........SMBr.
0x0040: 00 00 00 18 03 00 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 00 FE CA 00 00 00 00 00 87 00 02 50 43  ..............PC
0x0060: 20 4E 45 54 57 4F 52 4B 20 50 52 4F 47 52 41 4D   NETWORK PROGRAM
0x0070: 20 31 2E 30 00 02 58 45 4E 49 58 20 43 4F 52 45   1.0..XENIX CORE
0x0080: 00 02 4D 49 43 52 4F 53 4F 46 54 20 4E 45 54 57  ..MICROSOFT NETW
0x0090: 4F 52 4B 53 20 31 2E 30 33 00 02 4C 41 4E 4D 41  ORKS 1.03..LANMA
0x00A0: 4E 31 2E 30 00 02 57 69 6E 64 6F 77 73 20 66 6F  N1.0..Windows fo
0x00B0: 72 20 57 6F 72 6B 67 72 6F 75 70 73 20 33 2E 31  r Workgroups 3.1
0x00C0: 61 00 02 4C 4D 31 2E 32 58 30 30 32 00 02 4C 41  a..LM1.2X002..LA
0x00D0: 4E 4D 41 4E 32 2E 31 00 02 4E 54 20 4C 4D 20 30  NMAN2.1..NT LM 0
0x00E0: 2E 31 32 00                                      .12.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
05/30-06:51:08.770653 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3312 IpLen:20 DgmLen:153 DF
***AP*** Seq: 0x79E6C7BA  Ack: 0x8CA2E0D8  Win: 0x437A  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 99 0C F0 40 00 80 06 85 EC xx xx  xx xx  CB EB  ....@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C7 BA 8C A2 E0 D8 50 18  ,d...;y.......P.
0x0030: 43 7A 65 92 00 00 00 00 00 6D FF 53 4D 42 72 00  Cze......m.SMBr.
0x0040: 00 00 00 98 03 00 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 00 FE CA 00 00 00 00 11 07 00 03 32 00  ..............2.
0x0060: 01 00 04 41 00 00 00 00 01 00 00 00 00 00 FD F3  ...A............
0x0070: 00 00 F0 7C 57 98 CE 07 C2 01 2C 01 08 28 00 84  ...|W.....,..(..
0x0080: 86 7E F6 5C B4 6B 98 57 00 4F 00 52 00 4B 00 47  .~.\.k.W.O.R.K.G
0x0090: 00 52 00 4F 00 55 00 50 00 00 00 xx 00 xx 00 xx  .R.O.U.P...x.x.x
0x00A0: 00 xx 00 xx 00 00 00                             .x.x...

< -- End of step 3 -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
< -- Step 4 -- >
Text filled in Green: SMB command in the chain
SMB_COM_SESSION_SETUP_ANDX: 0x73 SMB_COM_TREE_CONNECT_ANDX: 0x75
Text filled in Red: (1) UserID (2)Security Blod Length =0x0001
(3) Security Blod  (4) Password length = 0x0001 (5): password = 0x00
Text filled in Yellow: ServerName, Text filled in Blue:resource name
```

Text filled in Teal: Service Name

```
05/30-06:51:09.030192 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:25873 IpLen:20 DgmLen:232 DF
***AP*** Seq: 0x8CA2E0D8  Ack: 0x79E6C82B  Win: 0x21C3  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 E8 65 11 40 00 74 06 39 7C CB EB 2C 64 xx xx  ..e.@xxxx|..,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E0 D8 79 E6 C8 2B 50 18  ...;......y..+P.
0x0030: 21 C3 9A 6D 00 00 00 00 00 BC FF 53 4D 42 73 00  !..m.......SMBs.
0x0040: 00 00 00 18 03 80 00 00 B3 66 F8 3F A5 63 6B 6F  .........f.?.cko
0x0050: 00 00 00 00 FE CA 00 00 00 00 0D 75 00 84 00 04  ...........u....
0x0060: 41 32 00 00 00 00 00 00 00 01 00 00 00 00 00 00  A2.............
0x0070: 00 D4 00 00 00 47 00 00 00 00 00 00 57 00 69 00  .....G......W.i.
0x0080: 6E 00 64 00 6F 00 77 00 73 00 20 00 4E 00 54 00  n.d.o.w.s. .N.T.
0x0090: 20 00 31 00 33 00 38 00 31 00 00 00 00 00 57 00   .1.3.8.1.....W.
0x00A0: 69 00 6E 00 64 00 6F 00 77 00 73 00 20 00 4E 00  i.n.d.o.w.s. .N.
0x00B0: 54 00 20 00 34 00 2E 00 30 00 00 00 00 00 04 FF  T. .4...0.......
0x00C0: 00 00 00 00 00 01 00 2D 00 00 5C 00 5C 00 xx 00  .......-..\.\.x.
0x00D0: xx 00 xx 00 2E 00 xx 00 xx 00 xx 00 xx 00 xx 00  x.x...x.x...x.x.
0x00E0: xx 00 2E 00 xx 00 5C 00 49 00 50 00 43 00 24 00  x...x.\.I.P.C.$.
0x00F0: 00 00 49 50 43 00                                ..IPC.
```

=+

Text in Grey: Return code -indicate successful connection
Text in Red: (1).Tree ID:2048 (2)User ID: 2048
Text in Teal: Security Blod Length = 0x5E 0x00
Text in Blue:Obtained service: IPC

```
05/30-06:51:09.030529 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3313 IpLen:20 DgmLen:194 DF
***AP*** Seq: 0x79E6C82B  Ack: 0x8CA2E198  Win: 0x42BA  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 C2 0C F1 40 00 80 06 85 C2 xx xx  xx xx  CB EB  ....@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C8 2B 8C A2 E1 98 50 18  ,d...;y..+....P.
0x0030: 42 BA 8A 8B 00 00 00 00 00 96 FF 53 4D 42 73 00  B..........SMBs.
0x0040: 00 00 00 98 03 80 00 00 B3 66 F8 3F A5 63 6B 6F  .........f.?.cko
0x0050: 00 00 00 08 FE CA 00 08 00 00 03 75 00 87 00 00  ...........u....
0x0060: 00 5E 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00  .^..W.i.n.d.o.w.
0x0070: 73 00 20 00 35 00 2E 00 30 00 00 00 57 00 69 00  s. .5...0...W.i.
0x0080: 6E 00 64 00 6F 00 77 00 73 00 20 00 32 00 30 00  n.d.o.w.s. .2.0.
0x0090: 30 00 30 00 20 00 4C 00 41 00 4E 00 20 00 4D 00  0.0. .L.A.N. .M.
0x00A0: 61 00 6E 00 61 00 67 00 65 00 72 00 00 00 57 00  a.n.a.g.e.r...W.
0x00B0: 4F 00 52 00 4B 00 47 00 52 00 4F 00 55 00 50 00  O.R.K.G.R.O.U.P.
0x00C0: 00(03 FF 00 96 00 01 00 06 00 49 50 43 00 00 00 ..........IPC...
```

< -- End of Step 4 -- >
=+

< -- Step 5 -- >
Text filled in Green: SMB command in the chain- NT Create AndX: 0xA2
Text in Red: (1).Tree ID:2048 (2)User ID: 2048
Text in Blue: File name: \sql\query
Text in Grey: Indicate successful connection

```
05/30-06:51:09.288582 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:26385 IpLen:20 DgmLen:150 DF
***AP*** Seq: 0x8CA2E198  Ack: 0x79E6C8C5  Win: 0x2129  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 96 67 11 40 00 74 06 37 CE CB EB 2C 64 xx xx  ..g.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E1 98 79 E6 C8 C5 50 18  ...;......y...P.
0x0030: 21 29 86 FF 00 00 00 00 00 6A FF 53 4D 42 A2 00  !).......j.SMB..
0x0040: 00 00 00 18 03 80 D3 F8 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 00 47 00 08 40 00 18 FF 00 00 00 00  .....G..@xxxxxxx
0x0060: 14 00 06 00 00 00 00 00 00 00 9F 01 02 00 00 00  ................
0x0070: 00 00 00 00 00 00 80 00 00 00 01 00 00 00 03 00  ................
0x0080: 00 00 40 00 00 00 02 00 00 00 03 17 00 00 5C 00  ..@xxxxxxxxxxx\.
0x0090: 73 00 71 00 6C 00 5C 00 71 00 75 00 65 00 72 00  s.q.l.\.q.u.e.r.
0x00A0: 79 00 00 00                                      y...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
05/30-06:51:09.289349 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3314 IpLen:20 DgmLen:147 DF
***AP*** Seq: 0x79E6C8C5  Ack: 0x8CA2E206  Win: 0x424C  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 93 0C F2 40 00 80 06 85 F0 xx xx  xx xx  CB EB  ....@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C8 C5 8C A2 E2 06 50 18  ,d...;y.......P.
0x0030: 42 4C 9A 44 00 00 00 00 00 67 FF 53 4D 42 A2 00  BL.D.....g.SMB..
0x0040: 00 00 00 98 03 80 D3 F8 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 00 47 00 08 40 00 22 FF 00 67 00 00  .....G..@x"..g..
0x0060: 00 40 01 00 00 00 00 00 00 00 00 00 00 00 00 00  .@xxxxxxxxxxxxxx
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0080: 00 00 00 00 00 00 80 00 00 00 00 20 00 00 00 00  ........... ....
0x0090: 00 00 00 00 00 00 00 00 00 00 00 02 00 FF 05 00 00  ................
0x00A0: 00                                               .

< -- End of Step 5 -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
< -- Step 6 -- >
Text filled in Green: SMB command in the chain- SMB_COM_WRITE_ANDX = 0x2F
Text filled in Grey: open mode – 0x008: Write mode
Text in Teal: Data length = 0x0200
Text starting from Blue to the end: Data to write

05/30-06:51:09.563507 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:26897 IpLen:20 DgmLen:615 DF
***AP*** Seq: 0x8CA2E206  Ack: 0x79E6C930  Win: 0x20BE  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 02 67 69 11 40 00 74 06 33 FD CB EB 2C 64 xx xx  .gi.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E2 06 79 E6 C9 30 50 18  ...;......y..0P.
0x0030: 20 BE D0 41 00 00 00 00 02 3B FF 53 4D 42 2F 00   ..A.....;.SMB/.
0x0040: 00 00 00 18 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 80 00 0C FF 00 00 00 00  ................
0x0060: 40 00 00 00 00 FF FF FF FF 08 00 00 02 00 00 00  @...............
0x0070: 02 3B 00 00 02 02 00 02 00 00 00 01 00 48 4F 53  .;...........HOS
0x0080: 54 53 56 52 00 00 00 00 00 00 00 00 00 00 00 00  TSVR............
0x0090: 00 00 00 00 00 00 00 00 00 00 00 07 73 61 00 00  ............sa..
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00B0: 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00  ................
```

```
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00D0: 00 00 00 00 00 00 00 00 00 00 30 30 30 30 30 32  ..........000002
0x00E0: 38 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00  82..............
0x00F0: 00 00 00 00 4C 99 62 CF 08 03 01 06 0A 09 01 01  ....L.b.........
0x0100: 00 00 00 00 00 00 00 00 00 4D 69 63 72 6F 73 6F  .........Microso
0x0110: 66 74 28 52 29 20 57 69 00 00 00 00 00 00 00 00  ft(R) Wi........
0x0120: 00 00 00 00 00 00 00 0F xx xx xx 2E xx xx 2E xx  ........xxx.xx.x
0x0130: xx xx 2E xx 00 00 00 00 00 00 00 00 00 00 00 00  xx.x............
0x0140: 00 00 00 00 00 00 0C 00 00 00 00 00 00 00 00 00  ................
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0240: 00 00 00 00 00 00 02 04 02 00 00 4F 4C 45 44 42  ...........OLEDB
0x0250: 00 00 00 00 00 05 06 00 00 00 00 0D 11 00 00 00  ................
0x0260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0270: 00 00 00 00 00                                   .....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:09.563630 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3315 IpLen:20 DgmLen:91 DF
***AP*** Seq: 0x79E6C930  Ack: 0x8CA2E445  Win: 0x400D  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 5B 0C F3 40 00 80 06 86 27 xx xx  xx xx  CB EB  .[..@xxxx'.,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C9 30 8C A2 E4 45 50 18  ,d...;y..0...EP.
0x0030: 40 0D 3F 5D 00 00 00 00 00 2F FF 53 4D 42 2F 00  @.?]...../.SMB/.
0x0040: 00 00 00 98 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 80 00 06 FF 00 2F 00 00  ............./..
0x0060: 02 FF FF 00 00 00 00 00 00                       .........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:09.822652 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:27153 IpLen:20 DgmLen:174 DF
***AP*** Seq: 0x8CA2E445  Ack: 0x79E6C963  Win: 0x208B  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 AE 6A 11 40 00 74 06 34 B6 CB EB 2C 64 xx xx  ..j.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E4 45 79 E6 C9 63 50 18  ...;.....Ey..cP.
0x0030: 20 8B 1F 32 00 00 00 00 00 82 FF 53 4D 42 2F 00   ..2.......SMB/.
0x0040: 00 00 00 18 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 C0 00 0C FF 00 00 00 00  ................
0x0060: 40 00 00 00 00 FF FF FF FF 08 00 47 00 00 00 47  @..........G...G
```

```
0x0070: 00 3B 00 47 00 02 01 00 47 00 00 02 00 00 00 00  .;.G....G.......
0x0080: 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00B0: 00 00 34 30 39 36 00 00 04 00 00 00              ..4096......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:09.822790 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3316 IpLen:20 DgmLen:91 DF
***AP*** Seq: 0x79E6C963  Ack: 0x8CA2E4CB  Win: 0x3F87  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 5B 0C F4 40 00 80 06 86 26 xx xx  xx xx  CB EB  .[..@xxxx&.,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C9 63 8C A2 E4 CB 50 18  ,d...;y..c....P.
0x0030: 3F 87 00 E3 00 00 00 00 00 2F FF 53 4D 42 2F 00  ?......../.SMB/.
0x0040: 00 00 00 98 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 C0 00 06 FF 00 2F 00 47  ............/.G
0x0060: 00 FF FF 00 00 00 00 00 00                       .........

< -- End of step 6 -- >

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
< -- Step 7 -- >
Text filled in Green: SMB command in the chain- SMB_COM_READ_ANDX: 0x2E

05/30-06:51:10.078796 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:27409 IpLen:20 DgmLen:104 DF
***AP*** Seq: 0x8CA2E4CB  Ack: 0x79E6C996  Win: 0x2058  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 68 6B 11 40 00 74 06 33 FC CB EB 2C 64 xx xx  .hk.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E4 CB 79 E6 C9 96 50 18  ...;......y...P.
0x0030: 20 58 98 BA 00 00 00 00 00 3C FF 53 4D 42 2E 00   X.......<.SMB..
0x0040: 00 00 00 18 00 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 00 01 0C FF 00 00 00 00  ................
0x0060: 40 00 00 00 00 00 00 02 00 02 FF FF FF FF 00 02 00  @..............
0x0070: 00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] MS-SQL/SMB sa login failed [**]
05/30-06:51:10.078956 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3317 IpLen:20 DgmLen:163 DF
***AP*** Seq: 0x79E6C996  Ack: 0x8CA2E50B  Win: 0x3F47  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 A3 0C F5 40 00 80 06 85 DD xx xx  xx xx  CB EB  ....@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C9 96 8C A2 E5 0B 50 18  ,d...;y.......P.
0x0030: 3F 47 21 A4 00 00 00 00 00 77 FF 53 4D 42 2E 00  ?G!......w.SMB..
0x0040: 00 00 00 98 00 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 00 01 0C FF 00 00 00 00  ................
0x0060: 00 00 00 00 00 3B 00 3C 00 00 00 00 00 00 00 00  .....;.<........
0x0070: 00 00 00 3C 00 00 04 01 00 3B 00 00 01 00 AA 27  ...<.....;.....'
0x0080: 00 18 48 00 00 01 0E 1B 00 4C 6F 67 69 6E 20 66  ..H......Login f
0x0090: 61 69 6C 65 64 20 66 6F 72 20 75 73 65 72 20 27  ailed for user '
0x00A0: 73 61 27 2E 00 00 00 00 FD 02 00 00 00 00 00 00  sa'.............
0x00B0: 00                                             .

< -- End of step 7 -- >

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

< -- Step 8 -- >
Text filled in Green: SMB command: SMBClose = 0x04
                                   SMB_COM_TREE_DISCONNECT = 0x71
                                   SMB_COM_LOGOFF_ANDX = 0x74

05/30-06:51:10.335613 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:27665 IpLen:20 DgmLen:86 DF
***AP*** Seq: 0x8CA2E50B  Ack: 0x79E6CA11  Win: 0x1FDD  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 56 6C 11 40 00 74 06 33 0E CB EB 2C 64 xx xx  .Vl.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E5 0B 79 E6 CA 11 50 18  ...;......y...P.
0x0030: 1F DD 8F 9D 00 00 00 00 00 2A FF 53 4D 42 04 00  .........*.SMB..
0x0040: 00 00 00 18 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 40 01 03 00 40 FF FF FF  ........@xxx@...
0x0060: FF 00 00 00                                      ....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:10.335768 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3318 IpLen:20 DgmLen:79 DF
***AP*** Seq: 0x79E6CA11  Ack: 0x8CA2E539  Win: 0x3F19  TcpLen: 20
0x0000:xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 4F 0C F6 40 00 80 06 86 30 xx xx  xx xx  CB EB  .O..@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 CA 11 8C A2 E5 39 50 18  ,d...;y......9P.
0x0030: 3F 19 B2 C1 00 00 00 00 00 23 FF 53 4D 42 04 00  ?........#.SMB..
0x0040: 00 00 00 98 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 40 01 00 00 00           ........@xxxx

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:10.658689 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:27921 IpLen:20 DgmLen:79 DF
***AP*** Seq: 0x8CA2E539  Ack: 0x79E6CA38  Win: 0x1FB6  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 4F 6D 11 40 00 74 06 32 15 CB EB 2C 64 xx xx  .Om.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E5 39 79 E6 CA 38 50 18  ...;.....9y..8P.
0x0030: 1F B6 25 7D 00 00 00 00 00 23 FF 53 4D 42 71 00  ..%}.....#.SMBq.
0x0040: 00 00 00 18 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 80 01 00 00 00           .............

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:10.658795 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3319 IpLen:20 DgmLen:79 DF
***AP*** Seq: 0x79E6CA38  Ack: 0x8CA2E560  Win: 0x3EF2  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 4F 0C F7 40 00 80 06 86 2F xx xx  xx xx  CB EB  .O..@xxxx/.,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 CA 38 8C A2 E5 60 50 18  ,d...;y..8...`P.
```

```
0x0030: 3E F2 05 9A 00 00 00 00 00 23 FF 53 4D 42 71 00  >........#.SMBq.
0x0040: 00 00 00 98 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 80 01 00 00 00           .............

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:10.916123 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:28177 IpLen:20 DgmLen:83 DF
***AP*** Seq: 0x8CA2E560  Ack: 0x79E6CA5F  Win: 0x1F8F  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 53 6E 11 40 00 74 06 31 11 CB EB 2C 64 xx xx  .Sn.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E5 60 79 E6 CA 5F 50 18  ...;.....`y.._P.
0x0030: 1F 8F DF 4E 00 00 00 00 00 27 FF 53 4D 42 74 00  ...N.....'.SMBt.
0x0040: 00 00 00 18 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 C0 01 02 FF 00 FF FF 00  ................
0x0060: 00                                               .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:10.916252 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3320 IpLen:20 DgmLen:83 DF
***AP*** Seq: 0x79E6CA5F  Ack: 0x8CA2E58B  Win: 0x3EC7  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 53 0C F8 40 00 80 06 86 2A xx xx  xx xx  CB EB  .S..@xxxx*.,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 CA 5F 8C A2 E5 8B 50 18  ,d...;y.._....P.
0x0030: 3E C7 BF 44 00 00 00 00 00 27 FF 53 4D 42 74 00  >..D.....'.SMBt.
0x0040: 00 00 00 98 03 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 C0 01 02 FF 00 27 00 00  .............'..
0x0060: 00                                               .

< -- End of Step 8 -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
< -- Step 9 -- >

05/30-06:51:11.179325 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:28433 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x8CA2E58B  Ack: 0x79E6CA8A  Win: 0x1F64  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 28 6F 11 40 00 74 06 30 3C CB EB 2C 64 xx xx  .(o.@xxxx<..,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E5 8B 79 E6 CA 8A 50 11  ...;......y...P.
0x0030: 1F 64 65 87 00 00 00 00 00 00 00 00              .de.........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/30-06:51:11.179436 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3321 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x79E6CA8A  Ack: 0x8CA2E58C  Win: 0x3EC7  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 28 0C F9 40 00 80 06 86 54 xx xx  xx xx  CB EB  .(..@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 CA 8A 8C A2 E5 8C 50 11  ,d...;y.......P.
0x0030: 3E C7 46 23 00 00 00 00 00 00 00 00              >.F#........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
05/30-06:51:11.444433 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:28689 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x8CA2E58C  Ack: 0x79E6CA8B  Win: 0x1F64  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 28 70 11 40 00 74 06 2F 3C CB EB 2C 64 xx xx  .(p.@xxx/<..,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E5 8C 79 E6 CA 8B 50 10  ...;......y...P.
0x0030: 1F 64 65 86 00 00 00 00 00 00 00 00              .de.........

< -- End of step 9 -- >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

< -- End of the log -- >
```

1. **Source of Trace**

This detect is from the following URL: http://www.incidents.org/archives/intrusions/msg14574.html. The packets were captured by Robert Wagner.

The source of the attack is 203.235.44.100 and the destination is myIP. Only one source and one destination IP are involved in this attack. Obviously, the TCP port 1433 and 139 are open in the target. Since this attack was not captured from my own network or my corporate network, other details such as whether there are firewalls in between the source and destination, why the ports are open, are unknown to me.

2. **Detect was generated by:**

This detect was generated by snort and the format of the log comply with the snort ASCII log format.

The whole communication between the attacker and the target on ports 1433 and 139 were fully logged, it leads me to elect that there were snort rules to log all the packets from or to TCP port 1433 and 139.

3. **Probability the source address was spoofed:**

Based on the following two reasons, the probability of spoofing source address is low:

- The attacker was communicating with MYIP over TCP. In order to be a successful attack, the attacker must be able to receive the response packets.
- The source of the attack might be a victim of the SQLspida worm. The owner of the machine might not have any knowledge of what the machine was doing. Thus there was no intension of spoofing.

4. **Description of attack:**

The attack was the probing activity of a variant of Sqlspida worm. The log trace shows that the worm attempted to login to a Microsoft SQL database using a SQL user account “sa” with Blank password over Named pipe on TCP port 139. Fortunately, the target server rejected the login request. If the authentication succeeds, the worm will start its propagation right away.

The sqlspida worm was discovered on May 21,2002. Internet Security Service issued alert warning that the sqlspida worm was propagating [9]. The detail description of the worm is available at Symantec's web site [10] and NAI's web site [11].

This worm targets Microsoft SQL servers. It probes the Internet for SQL servers on port 1433 and compromises those servers using the default SQL administrator account "sa". Once a SQL server has been accessed, the worm modifies the NT user "guest" by changing the password on that account, adding that user to the local administrators group and adds the user to the "Domain Admins" group. The worm then writes several files to the compromised server and kicks off the propagation routine. Finally, the worm disables the guest account, and removes it from the local administrators group and the "Domain Admins" group.

The virus description from both sites stated that once a computer is infected, it will searches for vulnerable computers on networks whose IP addresses do not begin with 10, 127, 172, or 192. The port the worm scans is TCP port 1433.

However, base the fact that Microsoft SQL server default installation enables TCP/IP and Named Pipe protocols, it is also possible that the worm can try the SQL user account "sa" login over named Pipe, which is the attack mechanism used by the attack showed in the above log. The named pipe protocol uses TCP port 139.

The attacking machine could be a compromised system by Sqlspida. It became another launch pad of the worm and started to scan other machines after it was infected.

There are several clues in the trace packets enabling me to determine the OS of the attacking machine. In the SYN packet it sent out, we can see the window size = 0x2000=8192, TCP Option = MSS only. MSS = 1460. These are what we can normally see from a SYN packet sent out by Windows NT, Windows 95 and Widows for workgroup 3.1. [12] In addition, we can see from the SMB protocol negotiation packet that the attacking machine support XENIX core which again is supported by the aforementioned OS. Windows 2000 does not support this protocol. Finally, from in the SMB_COM_SESSION_SETUP_ANDX command, the attacking machine claim itself it is a Windows NT 4.0. All these clues told us that the attacking machine is a windows NT 4.0 box rather than a samba client on *nix box or a Windows 2000 machine.

5. **Attack mechanism:**

The above probe happened at May 30, 2002, 5:55:36, 9 days after the worm was discovered. But the attacking machine did not try to login to SQL server over TCP/IP, instead, it try to login to SQL server by Named Pipe over TCP port 139. The probe for TCP port 1433 might just a reconnaissance to identify whether the SQL service on MYIP is available.

Almost one hour later, the attacker came back. But it looked for another port this time. Following are the attacking steps I identified from the packet trace:

Step 0: The attacking machine first probed for TCP port 1433 on MYIP by completing the three-way handshake. After, the attacking machine killed the connection by sending a FIN packet, the whole disconnection process was completely normally. This is just a probe for opening SQL service.

Step 1: The attacking machine completed three-way handshake with TCP port

139 on MYIP.

Step 2: The attacking machine established the Netbios session by connecting directly to the IP address of the server. Typically, the Netbios name resolve proceeds before creating a Netbios session since Netbios normally rely on Netbios name instead of IP address. But in our case, the worm used called name "*SMBSERVER" and calling name "HOSTSVR". According to the CIFS specification [4], "*SMBSERVER" is a generic called name, allowing requester to connect to the SMB server without knowing its real, registered NETBIOS name. The server accepted the called name and granted the session to the attacker.

Step 3: The attacking machine sent out a Netbios protocol negotiation packet, the SMB command is SMB_COM_NEGOTITATE. It is worth to notice that one of the protocols supported is XENIX core. If this is not a craft packet, it gives me a clue that the OS of the attacking machine could be Windows 95, Windows NT 3.1 or Windows NT 4.0. Windows 2000 does not support this protocol anymore. The following packets in SMB_COM_SESSION_SETUP_ANDX further demonstrate this.

Step 4: After the negotiation is finished, the worm sent out a request for setting up the SMB communication session and connected to the named pipe or IPC. The request contains a chain of two commands. The first command is SMB_COM_SESSION_SETUP_ANDX to establish the session. The second command is SMB_COM_TREE_CONNECT_ANDX to connect to the IPC$ tree over the SMB session setup by the previous command. Transmits the user's name and credentials to the server for verification through the first command. Successful server response has Uid field set in SMB header used for subsequent SMBs on behalf of this user. In the above packets, the SMB client sent a NULL (0x00) password to the SMB server. Note this is so called NULL session for anonymous user with a NULL password. Attacker can view the share resource of the target once the NULL session is created. The access right on the target system granted to the attacker will rely on the system's authorization to anonymous user. Here we can see that the attacker claim its native OS is Windows NT which is consistent with the protocols it claimed it supported, but all of this can be faked out. The server allowed anonymous accessing and thus granted a null session to the attacker. The server also tells the attacker that itself is a windows 2000 server in the response packet.

Step 5: The attacking machine sent off a SMB_COMM_NT_CREATE_ANDX command and request to open the named pipe \sql\query to it. Unfortunately, with the default installation of Windows 2000, this named pipe allow anonymous access thus the server was happy to open the named pipe for the attack.

Step 6: Although the null session has been created, but the worm did not want to take any advantage of it except connecting to the named pipe \sql\query. It did not hope to be able to authenticate himself to the SQL server by this anonymous login to the named pipe. Instead, it did a SMB_COM_Write_ANDX to send the default SQL user account "sa" with a NULL password to the SQL server. We can see the following from the packet:

1. The attacking machine's Netbios name: HOSTSVR
2. The default SQL account "sa"

3. Blank password. The bytes after the "sa" are all 0x00. If the attacker did supply a non-blank password, those bytes should not be 0x00.

4. The Database driver: OLEDB.

Because lack of material about OLEDB, I could not decode all data payloads the attacker sent to the server in this and the next packets. But I did a trace to a normal connection from a MS SQL analyzer to a SQL database through named pipe and compare the normal packet with the one the attacker sent and made the above best guess. Following is the normal packet when MSSQL analyzer in a Windows 2000 box connects to SQL server over named piped. I used the “sa” account with blank password to login.

```
                                        ff 53 4d 42 2f 00          .SMB/.
0040  00 00 00 18 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 ff fe 00 08 f0 00 0e ff 00 de de 01   ................
0060  40 00 00 00 00 ff ff ff ff 08 00 a4 00 00 00 a4   @...............
0070  00 40 00 00 00 00 00 a5 00 ee 10 01 00 a4 00 00   .@..............
0080  01 00 9c 00 00 00 00 00 00 71 00 00 00 00 00 00   .........q......
0090  00 07 98 04 00 00 00 00 00 00 e0 03 00 00 e0 01   ................
00a0  00 00 09 04 00 00 56 00 00 00 56 00 02 00 5a 00   ......V...V...Z.
00b0  00 00 5a 00 12 00 7e 00 0b 00 00 00 00 00 94 00   ..Z...~.........
00c0  04 00 9c 00 00 00 9c 00 00 00 00 04 5a 77 e6 b5   ............Zw..
00d0  00 00 00 00 9c 00 00 00 73 00 61 00 53 00 51 00   ........s.a.S.Q.
00e0  4c 00 20 00 51 00 75 00 65 00 72 00 79 00 20 00   L. .Q.u.e.r.y. .
00f0  41 00 6e 00 61 00 6c 00 79 00 7a 00 65 00 72 00    A.n.a.l.y.z.e.r.
0100  31 00 30 00 2e 00 32 00 30 00 2e 00 33 00 30 00   1.0...2.0...3.0.
0110  2e 00 32 00 31 00 4f 00 44 00 42 00 43 00         ..2.1.O.D.B.C.
```

If a password is supplied, the encoded password will be put between “sa” and “SQL query Analyzer”. We can see that in this packet, the tool (SQL query Analyzer), the destination server IP (10.20.30.21) and the Database driver (ODBC) are visible. But in the attacker’s packet, we did not see the tool and the destination IP or name, only see the Database driver (OLEDB).

Step 7: Finally, the attacking machine can see whether he is good luck enough to have the login passed. It sent out a SMB_COM_READ_ANDX to see whether he can get database information from the server, but the server tell him - login failed for user 'sa', which trigger the snort rule "MS-SQL/SMB sa login failed".

Step 8: The attacking machine did not want to waste its time to brute force the password; this is not the sqlspida’s mission. It decided to search for another victim instead of sticking with this server. He sent SMB_COM_Close, SMB_COM_Tree_DIsconnect, SMB_COM_LOGOFF_ANDX and lastly the FIN packet to close its conversation with the server.

Step 9: The attacking machine closes the TCP connection to port 139.

It is worthy to know why the worm is interesting in connecting to SQL server. The worm takes advantage of Microsoft SQL default installations with administrator accounts that have no passwords defined. Although Microsoft recommends that the "sa" account be set upon installation, many servers are not properly secured. Microsoft SQL server prior to the 2000 version installation setup a sa account with blank password by default [13]. Starting with the MS SQL 2000 version, the installation asks for a password to be setup, but blank sa password is still allowed. In addition, the SQL server does not log the login failed event by default [14], which makes the malicious attempt unnoticed.

In addition, after login as sa, the user can execute arbitrary code in the context of sa by running the "xp_cmdshell" extended stored procedure, which is the SQL call used to execute system commands within SQL queries. Normally, the SQL server should run in the context of a Domain user, but many servers run the SQL service as administrator account or local system account. With this privilege, the sqlspida worm can infect the victim and start its propagate and scanning process [15].

6. Correlations:

The incidents.org site has reported there were many scans against port 1433 and becomes the second active port (August 14), as well, the scans to TCP port 139 becomes the fourth or fifth most active port one month after this detects was caught.

Some blame the increasing of the TCP port 139 scan to a new virus called Datcom. But after knowing the fact that the SQLspida worm can propagate itself through the Named Pipe, not only through TCP port 1433, I can elect the active of the SQLspida worm should be one of the reason for the increasing of TCP port 139 scan.

Actually if the attacking machine is a Windows 2000 box, it could be very possible that it will use TCP port 445 to attack other machine over named pipe. Windows 2000 can connect to TCP port 139 and 445 to access named pipe, but it prefer TCP port 445. However, Windows NT can only use TCP port 139 to access named pipe, that is why in the above detect we see the attacking machine is working on TCP port 139.

There are 4 threads in intrusion mailing list of incidents.og including Sqlsnake variant or hack attempt over port 139? (where this detect is posted),SQLSnake - 6 Questions,Abrupt end to MSSQL probes?, 135, 139 going up, 80, 1433 staying even. There are a lot of discussions of SQLsnake in this two thread pages:
http://www.incidents.org/archives/intrusions/thrd3.html
http://www.incidents.org/archives/intrusions/thrd4.html

The attack mechanism showed in this detect have not been seen before and after. It might not be a common way to propagate sqlspida because using named pipe to connect to SQL server increase traffic overload comparing with using TCP/IP port 1433.

On May 21, Robert Wagner posted two messages on the intrusion mailing list. The messages are available at
http://www.incidents.org/archives/intrusions/msg12944.html and
http://www.incidents.org/archives/intrusions/msg12975.html.

The packet capture showed in the above two message shows the SQL worm in the attacking phase after successfully find out a vulnerable SQL server. The worm activates the guest account, changes the password of it and adds the guest account to the administrator group. All these action was done by the xp_cmdshell SQL extended procedure.

7. **Evidence of active targeting:**
The log trace definitely shows the evidence of active targeting. The attacking machine first reconnaissance to the SQL service, connect to it via \sql\query named pipe and try to login to the database server with default SQL account “sa” and a blank password.

8. **Severity:**
The severity is different in different circumstance:

- The database server is a production server

Criticality: 5.The database server normally hosts the most important data in a company.
Lethality: 1. This kind of attack is unlikely to be successful because the sa password is not blank.
System countermeasures: 5. In the context of this attack, the server is secure because the sa password is set which can defeat the attacker.
Network countermeasures: 1. The SQL service and NETBIOS is reachable from Internet. No firewall protects the database or the firewall allows these traffics to come in.

Severity = (criticality + lethality) -(system countermeasures + network countermeasures) = (5+1) – (5+1) = 0

- The database server is a honeypot, which was setup to attract the worm and analyze it.

Criticality: 1.The database won’t have any important data on it in this case.
Lethality: 1. This kind of attack is unlikely to be successful because the sa password is not blank
System countermeasures: 5. In the context of this attack, the server is secured because the sa password is set which can defeat the attacker.
Network countermeasures: 1. The SQL service and NETBIOS is reachable from Internet. No firewall protects the database or the firewall allows these traffics to come in.

Severity = (criticality + lethality) -(system countermeasures + network countermeasures) = (1+1)-(5+1) = -4

The severity in this case is –4, which means we took the advantage in this attack because we can analyze the worm and see how it works. After we investigate its propagation mechanism, we can publish the countermeasure to this worm. We take all the advantage as long as we take care of this machine and won’t let it be a launch pad to further attack.

From the message posted at this URL: http://www.incidents.org/archives/intrusions/msg12975.html somehow indicates that Robert Wagner, who is the owner of this detect, setup the SQL server as honeypot.

9. **Defensive recommendation:**

Block the ports at the perimeter. Database service and NetBIOS should not be exposed to Internet. Only necessary service should be opened to Internet.

Another option is to configure IP filter in Windows 2000 only allowing certain trusted IP to be able to access its service.

Harden the Windows 2000 OS according to hardening guideline [16]. Securing the SQL server by following the advisories given by the vendor [17] [18].

Change the TCP/IP port the SQL service listens on, enforce complex password for the service.

10. **Multiple choice test question:**

```
05/30-06:51:09.288582 203.235.44.100:3131 -> myip:139
TCP TTL:116 TOS:0x0 ID:26385 IpLen:20 DgmLen:150 DF
***AP*** Seq: 0x8CA2E198  Ack: 0x79E6C8C5  Win: 0x2129  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  .P...i..c.....E.
0x0010: 00 96 67 11 40 00 74 06 37 CE CB EB 2C 64 xx xx  ..g.@xxxxxxx,d.,
0x0020: xx xx  0C 3B 00 8B 8C A2 E1 98 79 E6 C8 C5 50 18  ...;......y...P.
0x0030: 21 29 86 FF 00 00 00 00 00 6A FF 53 4D 42 A2 00  !).......j.SMB..
0x0040: 00 00 00 18 03 80 D3 F8 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 00 47 00 08 40 00 18 FF 00 00 00 00  .....G..@xxxxxxx
0x0060: 14 00 06 00 00 00 00 00 00 00 9F 01 02 00 00 00  ................
0x0070: 00 00 00 00 00 00 80 00 00 00 01 00 00 00 03 00  ................
0x0080: 00 00 40 00 00 00 02 00 00 00 03 17 00 00 5C 00  ..@xxxxxxxxxxx\.
0x0090: 73 00 71 00 6C 00 5C 00 71 00 75 00 65 00 72 00  s.q.l.\.q.u.e.r.
0x00A0: 79 00 00 00                                      y...
05/30-06:51:09.289349 myip:139 -> 203.235.44.100:3131
TCP TTL:128 TOS:0x0 ID:3314 IpLen:20 DgmLen:147 DF
***AP*** Seq: 0x79E6C8C5  Ack: 0x8CA2E206  Win: 0x424C  TcpLen: 20
0x0000: xx xx xx xx xx xx xx xx xx xx xx xx xx xx 45 00  ..c....P...i..E.
0x0010: 00 93 0C F2 40 00 80 06 85 F0 xx xx  xx xx  CB EB  ....@xxxxxx,....
0x0020: 2C 64 00 8B 0C 3B 79 E6 C8 C5 8C A2 E2 06 50 18  ,d...;y.......P.
0x0030: 42 4C 9A 44 00 00 00 00 00 67 FF 53 4D 42 A2 00  BL.D.....g.SMB..
0x0040: 00 00 00 98 03 80 D3 F8 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 00 47 00 08 40 00 22 FF 00 67 00 00  .....G..@x"..g..
0x0060: 00 40 01 00 00 00 00 00 00 00 00 00 00 00 00 00  .@xxxxxxxxxxxxxx
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0080: 00 00 00 00 00 00 80 00 00 00 00 20 00 00 00 00  ........... ....
0x0090: 00 00 00 00 00 00 00 00 00 00 02 00 FF 05 00 00  ................
0x00A0: 00                                               .
```

Which of the following could not be determined from the response packet from MYIP:
(A) The SQL service is open in MYIP.
(B) The SMB connection to the \sql\squery pipe is successful.
(C) The SMB client is able to login to the database using the same credential it use to connect to the \sql\query named pipe.

(D) The SMB client can send login to the SQL serivice over the created named pipe.

The answer is C.

Detect # 2

```
< -- Date: 2002-07-17 -- >

20:21:30.374488 IP (tos 0x0, ttl 14, id 0, len 43) 255.255.255.255.31337 >
46.5.51.34.515: R [bad tcp cksum 6d6 (->ffce)!] 0:3(3) ack 0 win 0 [RST
cko]bad cksum 52ae (->4ba7)!
0x0000       4500 002b 0000 0000 0e06 52ae ffff ffffE..+......R.....
0x0010       2e05 3322 7a69 0203 0000 0000 0000 0000..3"zi..........
0x0020       5014 0000 06d6 0000 636b 6f00 0000     P.......cko...

23:47:30.444488 IP (tos 0x0, ttl 14, id 0, len 43) 255.255.255.255.31337 >
46.5.40.206.515: R [bad tcp cksum 1228 (->a23)!] 0:3(3) ack 0 win 0 [RST
cko]bad cksum 5e00 (->55fb)!
0x0000       4500 002b 0000 0000 0e06 5e00 ffff ffffE..+......^.....
0x0010       2e05 28ce 7a69 0203 0000 0000 0000 0000..(.zi..........
0x0020       5014 0000 1228 0000 636b 6f00 0000     P....(..cko...

< -- Date: 2002-07-18 -- >

00:55:42.454488 IP (tos 0x0, ttl 14, id 0, len 43) 255.255.255.255.31337 >
46.5.26.21.515: R [bad tcp cksum 1fe3 (->18dc)!] 0:3(3) ack 0 win 0 [RST
cko]bad cksum 6bbb (->64b4)!
0x0000       4500 002b 0000 0000 0e06 6bbb ffff ffffE..+......k.....
0x0010       2e05 1a15 7a69 0203 0000 0000 0000 0000....zi..........
0x0020       5014 0000 1fe3 0000 636b 6f00 0000     P.......cko...
...

19:18:37.464488 IP (tos 0x0, ttl 14, id 0, len 43) 255.255.255.255.31337 >
46.5.51.19.515: R [bad tcp cksum 6e5 (->ffdd)!] 0:3(3) ack 0 win 0 [RST
cko]bad cksum 52bd (->4bb6)!
0x0000       4500 002b 0000 0000 0e06 52bd ffff ffffE..+......R.....
0x0010       2e05 3313 7a69 0203 0000 0000 0000 0000..3.zi..........
0x0020       5014 0000 06e5 0000 636b 6f00 0000     P.......cko...
```

1. Source of Trace

This detect was extracted from the tcpdump binary file posted on http://www.incidents.org/logs/Raw/2002.6.8. The dates of the packets are 2002-07-17 and 2002-07-18.

The above packets are extracted by windump from the file, the windump command is as following:

windump -n -X -vvv -s 0 -r c:\giacout\snort\raw\2002.6.8 host 255.255.255.255 > C:\mydump.txt

Note not all packets with source address 255.255.255.255 are posted here. The packets are all the same except the time, destination IP address and checksums. Only the first packet and the last packet of each day are posted.

The destination IP addresses of the packets are random and unique. All the related destination IP addresses are listed here:

```
    46.5.51.34, 46.5.193.12, 46.5.223.37, 46.5.191.74, 46.5.94.29,
46.5.204.134, 46.5.228.131, 46.5.173.53, 46.5.172.154, 46.5.48.173,
46.5.40.206, 46.5.26.21, 46.5.190.218, 46.5.49.218, 46.5.184.214,
46.5.199.138, 46.5.9.211, 46.5.165.51, 46.5.75.2, 46.5.230.149,
46.5.174.19, 46.5.90.180, 46.5.63.186, 46.5.174.43, 46.5.21.209,
46.5.95.30, 46.5.69.99, 46.5.148.11, 46.5.197.236, 46.5.84.215,
46.5.231.194, 46.5.22.176, 46.5.225.102, 46.5.14.153, 46.5.6.34,
46.5.51.19
```

2. Detect was generated by:

As stated at the GCIA practical assignment requirement v3.2, the log files contain tcpdump binary logs was generated by a snort ruleset. The snort ruleset was not provided.

I start snort to execute a post analyze to this tcpdump binary log file. The snort binary version is 1.8-WIN32 (Build 103). The snort rule file snort.conf is version 1.84 dated 2002/03/02. The command is: snort -de -r raw\2002.6.8 -l log -c rules\snort.conf. The purpose of the –de option is to include the packet payload in the alert file and display the layer 2 header information.

The rule triggered by the above packets is “Backdoor Q access”. Following is one of the alert triggered by the last packets of the above log:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] BACKDOOR Q access [**]
07/07-20:34:09.424488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
255.255.255.255:31337 -> 46.5.193.12:515 TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
63 6B 6F                                  cko

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

The rule itself in Backdoor.rules file is as following:

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

3. Probability the source address was spoofed:

The source address in this interesting detect is absolutely spoofed. The source address is 255.255.255.255, which is a limited broadcast address. It should never be forwarded by any network routers [19]. The attacker who generated these packets did not expect to establish TCP session by them.

4. Description of attack:

All these packets targeted the internal network: 46.5.0.0/24.The packets are almost identical except the destination IP address and the IP and TCP checksum. The checksum is calculated by the whole packet thus is subjected to change when the destination is changed. Notes that all the checksum in the tcpdump binary file downloaded from incidents.org/logs/raw are incorrect. This might be caused by the packets owner changing the destination IP address in the packets they provided.

All the packets followed the same pattern outlined below:

- The source address is limited broadcast address 255.255.255.255
- The destination addresses are random without repeat.
- Total length = IP header (20) + TCP header (20) + payload (3) =43
- IPID = 0
- TTL =14
- Protocol = TCP
- Source port = 31337
- Destination port = 515
- Sequence number = 0
- Ack number = 0
- Flags = RST, ACK
- Win size = 0
- TCP payload = "cko"

With the above characteristic, all IDS analyst will think they are highly suspicious. The source port is the famous Back Orffice Trojan port. The destination port 515 is for the printer service in multiple OS. CERT had issued several advisories to address the vulnerabilities with the printer service in the last tow years [20][21][22]. Since the vulnerability existed in multiple implementations of many popular OS, the 515 port became one of the favorite port hackers would like to exploit.

But before we conclude that these packets were sent from an external attacker, we need to exclude 2 possibilities:

- The packets were captured on the internal network since the 255.255.255.255 source address would not be forwarded in or out by routers. Maybe there are someone on the internal network wanted to scan the printer port 515.
- The packet is a response of RESET to out bound packets looking for port 31337. Maybe there is someone on internal network looking for Trojans.

I do the following analyze to exclude the above 2 possibilities:

- We can observer the packets MAC address, comparing with other packets seen on the tcpdump binary file and determine the direction of the packet.

(1).Incoming packet

```
07/07-20:20:52.344488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33
type:0x800 len:0x3C
62.153.209.202:21 -> 46.5.180.241:21 TCP TTL:30 TOS:0x0 ID:39426
IpLen:20 DgmLen:40
```

The source MAC address is *0:3:E3:D9:26:C0* and the destination MAC address is *0:0:C:4:B2:33*

(2). Outgoing packet

```
07/08-11:43:31.144488 0:0:C:4:B2:33 -> 0:3:E3:D9:26:C0
type:0x800 len:0x4D
46.5.180.250:62259 -> 207.68.167.253:6667 TCP TTL:124 TOS:0x2C
ID:39540 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0x171BB72  Ack: 0x82540D56  Win: 0x21C0  TcpLen:
32
TCP Options (3) => NOP NOP TS: 224727 5789140
```

The source MAC address is *0:0:C:4:B2:33* and the destination MAC address is *0:3:E3:D9:26:C0*

Now we can determine the MAC address of the external gateway is *0:3:E3:D9:26:C0* and the MAC address of the internal gateway is *0:0:C:4:B2:33*

We compare the packet from 255.255.255.255 with above two at the second layer MAC address, we can be pretty sure that this is an incoming packet from the external network.

```
[**] BACKDOOR Q access [**]
07/07-20:34:09.424488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
255.255.255.255:31337 -> 46.5.193.12:515 TCP TTL:14 TOS:0x0 ID:0
IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
63 6B 6F                                         cko
```

- There are several signs in the packets which illustrated themselves are not response to outgoing Trojan port scan although the RST and ACK flag made them look like response.
  1. If scanning packets were sent to broadcast address 255.255.255.255, they could not be forwarded by the border routers thus could not go to external network and get any response. [19]
  2. Even we assumed that the border router did forward the packet (in which way it did not follow the RFC 1812), we still have other evidences to illustrate that the packets are not response at all. They are crafted to pretend to be response. The following item are not consistent with normal IP behavior.
     - ACK = 0

     Acknowledgement number represents the next byte of data receiving host expects: (last received sequence number + 1), the valid acknowledgement number should greater than 0.
     - IPID = 0 in every packet.

     According to RFC 791 [23], “the sender must choose the Identifier to be unique for this source, destination pair and protocol for the time the datagram (or any fragment of it) could be alive in the Internet.” The IP ID should not be identical in every packet. Plus IP ID=0 is not common being seen.

- TTL = 14
  This TTL is unbelievable low when considering the smallest TTL generated by OS is 64. How can a broadcast packet cross over 50 hops?

Now I am pretty sure that these packets are neither internal packet nor response packets to outbound Trojan scan, they are definitely crafted packet sent from external. This conclusion excludes the possibility of these packets are response to some source spoofed packet which set 46.5.0.0/24 network as source address.

But what can these packets achieve?

Even the border router forward this packet to the internal network and the internal hosts can receive them, they will just be discard silently because the ACK number is not correct and the hosts do not response to a RST/ACK packet even their ACK number is correct. These packet can not elicit any response from the internal host thus the attacker can not get any result if he is really want to scan port 515. Base on this, these packets do not aim to port scan.

Since these packets trigger the snort rule " Backdoor Q access", I did some further investigation to identify whether these packets could be associated with Backdoor Q.

Whitehat.com [24] explain the signature as following:

IDS203/TROJAN-ACTIVE-Q-TCP

[redacted]

```
01/04-02:51:15.622040 255.255.255.255:59564 -> target:15579
TCP TTL:240 TOS:0x0 ID:2323
****A* Seq: 0x5F9B8911   Ack: 0x204F1C8C   Win: 0x61C4
70 69 6E 67 20 32 33 2E 32 33 2E 32 33 2E 32 33  ping 23.23.23.23
00                                               .
```

[redacted]

```
Q is a remote access and redirection Trojan that employs strong encryption.
It allows for the execution of remote commands as root by sending a raw
tcp/icmp/udp packet. This signature watches for the source address
255.255.255.255, which should not appear in normal traffic. The content of
the packet is the command to run as root - and is arbitrary.
```

I downloaded the source code of Backdoor Q [25]. After review the program and its documentation, I found that there are three key parts in this software.

Qd – the backdoor daemon in the remote system. It is stealth until the client activates it and asks it to listen on the specified port.

Qs – the activation program. It can send a command to the backdoor to execute without opening a shell, or it can activate the backdoor to listen on certain port with an encrypted or unencrypted shell.

Q – the client who can interact with the backdoor by the shell.

There are two kinds of communication between the client and the backdoor, one is the activation, which is one way traffic; another is two way traffic when the client talks with the Backdoor system with the shell. Since a spoofed source address will not work in the second kind of communication, the only chance that the above packet would be useful is in the first communication.

The source code of activation server control is qs.c in the Q-1.0.tgz package. It calls forge.c, stega.c. conf.h is the configure file for this software. Here the client can configure the activate codes including CODE_EXEC, CODE_SSHD, CODE_RED for remote command execute and two different shells, the Q_ID and other options.

After reading the above 3 source codes, I have a clear picture about what a Q packet should look like:

- IPID = htons(Q_ID). If the Q_ID is 1, the IPID is 1. But if Q_ID is 0, the IPID is not 0.
- If the source IP address of the packet is a.b.c.d, each integer follow the rule listed:
    - i. a = (rand () % 220) + 1. It is a random number less than 222. But it cannot be 10 or 127.
    - ii. b = (rand () % 220) + 1. It is a random number less than 222.
    - iii. c = a/b
    - iv. d is the activation code defined in conf.h. That is why the code author Mixter suggested the activation code should be an integer between 1 and 254.
- TCP flag = ACK
- Win = htonl (random());
- Sequence number = htonl (random())
- Acknowledge number = htonl (random())
- TTL = 0xf 0 =240
- Source port can be specified by the client.
- Destination port can be specified by the client.
- Payload: The last binary in the payload is 0x00 because it indicates the end of a string, which contains the activation command.

In addition, the Q client can activate the server by TCP, UDP or ICMP. In the activation call, many fields in the IP payload and header do not matter to the remote backdoor since the Q server listens on the raw IP layer. It just cares about the activation code in the last digit of the source IP address and the command stored in the payload.

The conclusion is the packets in detect#2 do not match the pattern of Q packets. They have nothing to do with Backdoor Q access. Thus this is a false positive.

Another guess is that these packets are generated by a network device or OS with broken IP stack. But the packets are very obviously a scan because the destination addresses are different in every packet and do not repeat within two days.

The packets must be generated by other means since they are so artificially crafted. What is the purpose of the packets, what the attacker can achieve by them and what could be the tool the attacker use? I will give my best guess of the questions in the coming session – attack mechanism.

5. Attack mechanism:

The packets cannot elicit any response from TCP layer because the RST and ACK flag are set. But it could be possible that a poorly configured router is kindly to send an ICMP host unreachable response when the internal host does not exist. Thus the attacker can map the internal network stealthily by excluding the IP addresses the router response. This is called inverse mapping. There is a wonderful explanation of this mechanism in S.Northcutt's *Security Analyst Handbook:*

"How does the reset scan work? The reset scan and many other scans rely on ICMP to help them with their mapping. … Under many circumstance there will be no reply. However, if the host or network does not exist and the packet in not blocked before it reaches a helpful router, the router will send back information via ICMP that the target is unreachable."

This method may work because the router does not care about whether the RESET flag reasonable or not, i.e. whether it resets a previous established session. It just examines the destination address and checks whether it is reachable. When the access control list in the router does not block itself to leak the internal network information, it will just broadcast to the outside what it knows.

But in this special case where the source address of the packets are 255.255.255.255. To get this mechanism work, the hacker must make one more assumption that the router does not care the source address, it will reply to the source address as is. If this assumption is correct, then the router will send an ICMP broadcast message to outside when the scanned host is unreachable. Because the response will be a broadcast, as long as the hacker is at the same local network at the router, he will be able to receive it. The attacker can hide its identity by this mean.

These packets could have been generated by the tool hping2 from www.hping.org. The following command can generate the packets same as the one in the trace log:

```
hping 46.5.51.34 -a 255.255.255.255 -t 14 -N 0 -w 0 -y -s 31337 -p 515 -k -M 0 -L 0 -
R -A -d 3 -E cko -I eth1
```

cko is a file, which only contains three letters: cko. Hping read the file and put the beginning 3 characters as the packet's payload.

Still, I am not 100% sure that this is a RESET scan because the purpose of RESET scan and other means of inverse mapping technology is being stealthy. By having that much suspicious characteristics can catch any intrusion detection systems attention. It could not be stealthy at all. If the attacker is so aggressive and does not care whether its action get notice, why doesn't he just do a dirty nmap SYN scan or TCP scan so that he can map the network much faster instead of having unbelievable patience to wait for the result from inverse mapping to a class A network.

Based on the reason that almost every element in the packet can make an intrusion analyst to pay attention to it, it could be very possible that the purpose of these suspicious packets are to disturb the IDS system and the analyst. It firstly makes noise to consume the security analyst's energy to analyze it and secondly make smoke screen to hide other attacks. It has the same idea as the IDS evasion tool stick, which was well documented in the white paper *Fun with Packets* [26] by the tool's author Coretez Giovanni.

As Coretez Giovanni stated in his white paper, People are the essential element in intrusion Detection …Organizations hire just enough, people as to accomplish handling a normal load of alarms. Therefore, when a high number of false alarms are produced, finding the actual attack becomes impossible due to the lack of resources to investigate actual from spoofed attacks. The attacks cause a denial-of-service attack against not the computer network, but the human processes that support intrusion detection.

When the analysts are busy with analyzing this suspicious activity, it could be possible that they ignore one or two actual attacks that happened at the same time. In this case, these suspicious packets effectively make a smoke screen for the actual attacks.

6. Correlations:

Dshield.org [27] provide a mean of correlation by submitting the IP address and check how many attacks originated from it. From the following report generated on August 16, 2002, there have been 4992 attacks from the broadcast address 255.255.255.255. One of the ports this IP attack is 515, same as the one we have here.

IP Address:

255.255.255.255

HostName:

255.255.255.255

DShield Profile:

Country:
XX

Contact E-mail:
RESERVED

Total Records against IP:
4992

Number of targets:
1307

Date Range:
2002-08-14 to 2002-08-14

Ports Attacked (up to 10):

Port	Attacks
515	7

Fightback:

sent to RESERVED on 2001-09-03 07:41:09
no reply received

Whois:

```
No match for "255.255.255.255".

%%%%%%%%%%%%%%%%%% NO MATCH TIP
%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                             %
%  ALL OF THE POINT OF CONTACT HANDLES IN THE ARIN       %
%  WHOIS END WITH "-ARIN", IF YOU ARE QUERYING A POINT   %
%  OF CONTACT HANDLE PLEASE ADD -ARIN TO YOUR QUERY.     %
%                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%


The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.
```

Other security analyst of IDS system or firewall system saw the similar attacks. In 2001, there are several posts of the similar packet in intrusion mailing list and incident mailing list [28] [29] [30] [31]. The analysis posted in the mailing list shed some lights on my analysis in this paper.

7. Evidence of active targeting:

Almost every element of the packets is suspicious enough to cause the IDS system's attention. These packets are active targeting the IDS system. If the border router is not properly configured, the attacker can inversely map the internal network as well.

8. Severity:

a. Criticality : 5. The IDS system and the security analyst are the front-end of detecting attacks.

b. Lethality : 3. The suspicious packets can catch the IDS and security analyst's attention by every means. In this context, it has been successful. But since the rate of the packets is not high enough to cause real Denial of service to IDS system and annoy too much to the analyst, so I give 3 to lethality.

c. System countermeasures: 5. The IDS system caught these suspicious packets. I rate the countermeasure to the IDS system as 5.

d. Network countermeasures: 1. Assuming the IDS sensor is setup between the border router and perimeter firewall or internal network, seeing these packets from the IDS sensor means the border router does not block access from 255.255.255.255.

Severity = (criticality + lethality) - (system countermeasures + network countermeasures) = (5+3)-(5+1) =2

9. Defensive recommendation

a. The router should be properly configured not to forward limited broadcast and directed broadcast. The router should be further hardened according the security best practice and vendor's hardening guideline. The following links are valuable resource in terms of router security:

- Juniper Networks Router Security - Best Common Practices for Hardening the Infrastructure http://www.juniper.net/techcenter/app_note/350013.html
- SANS router security policy: http://www.sans.org/newlook/resources/policies/ Router_Security_Policy.pdf
- Router security configuration guide: http://nsa2.www.conxion.com/cisco/download.htm

b. Snort rules for Q access should be refined to properly capture the real

attempts.
c. Use stateful technology with IDS system so that the unsolicited packet cannot overwhelm the IDS system.
d. IDS system should still catch this kind of attack, but it should be given a lower priority based on its nature so that this kind of events cannot overwhelm the IDS analyst.

10. Multiple choice test question

```
[**] BACKDOOR Q access [**]
07/07-20:34:09.424488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
255.255.255.255:31337 -> 46.5.193.12:515 TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
63 6B 6F                                cko

[**] [1:184:3] BACKDOOR Q access [**]
07/08-17:14:52.414488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
255.255.255.255:31337 -> 46.5.22.176:515 TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
63 6B 6F                                cko

 [**] [1:184:3] BACKDOOR Q access [**]
07/08-18:35:04.434488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
255.255.255.255:31337 -> 46.5.6.34:515 TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
63 6B 6F                                cko
```

Which field of the above Reset packets is wrong?
A. The sequence number
B. TCP Flag Reset
C. Acknowledgement number.
D. TCP Flag ACK.

The answer is A.

Detect # 3

Note: The packets were captured on August 17, 2002. Every time a HTML request was sent from the attacker, a new TCP session was created by three ways handshake. Except in the first connection, all the TCP three ways handshakes packets are ommited for briefing.

```
04:03:25.554953 IP (tos 0x0, ttl 112, id 3194, len 48) 67.28.92.222.4789 >
my.home.net.ip.80: S [tcp sum ok] 2051556865:2051556865(0) win 8760 <mss
1460,nop,nop,sackOK> (DF)
0x0000       4500 0030 0c7a 4000 7006 0651 431c 5cde E..0.z@.p..QC.\.
0x0040: 00 00 00 98 00 80 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 08 FE CA 00 08 00 01 0C FF 00 00 00 00  ................
0x0060: 00 00 00 00 00 3B 00 3C 00 00 00 00 00 00 00 00  .....;.<........
0x0070: 00 00 00 3C 00 00 04 01 00 3B 00 00 01 00 AA 27  ...<.....;.....'
0x0080: 00 18 48 00 00 01 0E 1B 00 4C 6F 67 69 6E 20 66  ..H......Login f
0x0090: 61 69 6C 65 64 20 66 6F 72 20 75 73 65 72 20 27  ailed for user '
```

```
0x00A0: 73 61 27 2E 00 00 00 00 FD 02 00 00 00 00 00 00  sa'........0x0010  xxxx xxxx 12b5 0050
7a48 4601 0000 0000    A0.....PzHF.....
0x0020       7002 2238 959b 0000 0204 05b4 0101 0402 p."8............

04:03:25.556349 IP (tos 0x0, ttl 127, id 4087, len 48) my.home.net.ip.80 >
67.28.92.222.4789: S [tcp sum ok] 3332950809:3332950809(0) ack 2051556866
win 5840 <mss 1460,nop,nop,sackOK> (DF)
0x0000       4500 0030 0ff7 4000 7f06 f3d3 xxxx xxxx E..0..@.....A0..
0x0010       431c 5cde 0050 12b5 c6a8 cb19 7a48 4602 C.\..P......zHF.
0x0020       7012 16d0 0f30 0000 0204 05b4 0101 0402 p....0..........

04:03:25.764481 IP (tos 0x0, ttl 112, id 3201, len 40) 67.28.92.222.4789 >
my.home.net.ip.80: . [tcp sum ok] 1:1(0) ack 1 win 8760 (DF)
0x0000       4500 0028 0c81 4000 7006 0652 431c 5cde E..(..@.p..RC.\.
0x0010       xxxx xxxx 12b5 0050 7a48 4602 c6a8 cb1a A0.....PzHF.....
0x0020       5010 2238 308c 0000 0000 0000 0000      P."80.........

04:03:25.807177 IP (tos 0x0, ttl 112, id 3202, len 360) 67.28.92.222.4789 >
my.home.net.ip.80: P [tcp sum ok] 1:321(320) ack 1 win 8760 (DF)
0x0000       4500 0168 0c82 4000 7006 0511 431c 5cde E..h..@.p...C.\.
0x0010       xxxx xxxx 12b5 0050 7a48 4602 c6a8 cb1a A0.....PzHF.....
0x0020       5018 2238 3420 0000 504f 5354 202f 6367 P."84...POST./cg
0x0030       692d 6269 6e2f 666f 726d 6d61 696c 2e63 i-bin/formmail.c
0x0040       6769 2048 5454 502f 312e 300d 0a61 6363 gi.HTTP/1.0..acc
0x0050       6570 743a 2069 6d61 6765 2f67 6966 2c20 ept:.image/gif,.
0x0060       696d 6167 652f 782d 7862 6974 6d61 702c image/x-xbitmap,
0x0070       2069 6d61 6765 2f6a 7065 672c 2069 6d61 .image/jpeg,.ima
0x0080       6765 2f70 6a70 6567 2c20 2a2f 2a0d 0a61 ge/pjpeg,.*/*..a
0x0090       6363 6570 742d 6c61 6e67 7561 6765 3a20 ccept-language:.
0x00a0       656e 2d75 730d 0a63 6f6e 7465 6e74 2d74 en-us..content-t
0x00b0       7970 653a 2061 7070 6c69 6361 7469 6f6e ype:.application
0x00c0       2f78 2d77 7777 2d66 6f72 6d2d 7572 6c65 /x-www-form-urle
0x00d0       6e63 6f64 6564 0d0a 5573 6572 2d41 6765 ncoded..User-Age
0x00e0       6e74 3a20 4d6f 7a69 6c6c 612f 342e 3020 nt:.Mozilla/4.0.
0x00f0       2863 6f6d 7061 7469 626c 653b 204d 5349 (compatible;.MSI
0x0100       4520 352e 3031 3b20 5769 6e64 6f77 7320 E.5.01;.Windows.
0x0110       3935 290d 0a48 6f73 743a 20xx xx2e xxxx 95)..Host:.my.home
0x0120       2exx xx2e xxxx xx0d 0a52 6566 6572 6572 .net.ip..Referer
0x0130       3a20 6874 7470 3a2f 2fxx xx2e xxxx 2exx :.http://my.home.ne
0x0140       xx2e xxxx xx2f 636f 6e74 6163 742e 6874 t.ip/contact.ht
0x0150       6d0d 0a43 6f6e 7465 6e74 2d4c 656e 6774 m..Content-Lengt
0x0160       683a 2033 3135 0d0a                     h:.315..

04:03:25.921426 IP (tos 0x0, ttl 127, id 4088, len 40) my.home.net.ip.80 >
67.28.92.222.4789: . [tcp sum ok] 1:1(0) ack 321 win 5520 (DF)
0x0000       4500 0028 0ff8 4000 7f06 f3da xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 12b5 c6a8 cb1a 7a48 4742 C.\..P......zHGB
0x0020       5010 1590 3bf4 0000 8888 8888 8888      P...;.........

04:03:26.217050 IP (tos 0x0, ttl 112, id 3218, len 359) 67.28.92.222.4789 >
my.home.net.ip.80: P [tcp sum ok] 321:640(319) ack 1 win 8760 (DF)
0x0000       4500 0167 0c92 4000 7006 0502 431c 5cde E..g..@.p...C.\.
0x0010       xxxx xxxx 12b5 0050 7a48 4742 c6a8 cb1a A0.....PzHGB....
0x0020       5018 2238 4f70 0000 0d0a 2665 6d61 696c P."8Op....&email
0x0030       3d42 737a 4068 6f6d 652e 636f 6d26 7265 =Bsz@home.com&re
0x0040       6369 7069 656e 743d 414c 4943 454e 504f cipient=ALICENPO
0x0050       5553 494e 4540 464c 4153 484d 4149 4c2e USINE@FLASHMAIL.
0x0060       434f 4d26 7375 626a 6563 743d 4276 6974 COM&subject=Bvit
```

```
0x0070       0d0a 436f 6e74 656e 742d 5479 7065 3a20 ..Content-Type:.
0x0080       7465 7874 2f68 746d 6c3b 0d0a 3c48 544d text/html;..<HTM
0x0090       4c3e 3c48 4541 443e 3c2f 6865 6164 3e3c L><HEAD></head><
0x00a0       626f 6479 3e3c 4252 3e49 6b78 6b71 6e77 body><BR>Ikxkqnw
0x00b0       2067 7264 7a78 646d 7465 2063 6f6d 6162 .grdzxdmte.comab
0x00c0       7363 6569 6420 7879 787a 6c6f 7368 206e sceid.xyxzlosh.n
0x00d0       7678 7a75 6820 6c64 7576 6674 2027 3432 vxzuh.lduvft.'42
0x00e0       2e36 352e 3030 2e30 3333 2f65 756f 2d77 .65.00.033/euo-w
0x00f0       6f66 2f79 676b 6464 716f 732e 6575 6f27 of/ygkddqos.euo'
0x0100       2051 554b 4b20 5349 3c42 523e 3c2f 424f .QUKK.SI<BR></BO
0x0110       4459 3e3c 2f48 544d 4c3e 0d0a 3c21 2d2d DY></HTML>..<!--
0x0120       0d0a 2663 6f6d 6d65 6e74 733d 205a 656b ..&comments=.Zek
0x0130       7969 206b 7466 7479 7665 7768 7a20 6967 yi.ktftyvewhz.ig
0x0140       2075 6266 676c 2075 6a6a 6164 6e72 6c62 .ubfgl.ujjadnrlb
0x0150       207a 7962 7478 6169 7776 6520 2d2d 3e20 .zybtxaiwve.-->.
0x0160       203c 212d 2d0d 0a                            .<!--..

04:03:26.218686 IP (tos 0x0, ttl 127, id 4089, len 40) my.home.net.ip.80 >
67.28.92.222.4789: R [tcp sum ok] 3332950810:3332950810(0) win 0 (DF)
0x0000       4500 0028 0ff9 4000 7f06 f3d9 xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 12b5 c6a8 cb1a 7a48 4742 C.\..P......zHGB
0x0020       5004 0000 5190 0000 8888 8888 8888      P...Q.........

< -- Three ways handshake -- >

04:03:26.867146 IP (tos 0x0, ttl 112, id 3252, len 359) 67.28.92.222.4808 >
my.home.net.ip.80: P [tcp sum ok] 1:320(319) ack 1 win 8760 (DF)
0x0000       4500 0167 0cb4 4000 7006 04e0 431c 5cde E..g..@.p...C.\.
0x0010       xxxx xxxx 12c8 0050 7a55 f420 c6ad 25bc A0.....PzU....%.
0x0020       5018 2238 cfed 0000 504f 5354 202f 6367 P."8....POST./cg
0x0030       692d 6269 6e2f 666f 726d 6d61 696c 2e70 i-bin/formmail.p
0x0040       6c20 4854 5450 2f31 2e30 0d0a 6163 6365 l.HTTP/1.0..acce
0x0050       7074 3a20 696d 6167 652f 6769 662c 2069 pt:.image/gif,.i
0x0060       6d61 6765 2f78 2d78 6269 746d 6170 2c20 mage/x-xbitmap,.
0x0070       696d 6167 652f 6a70 6567 2c20 696d 6167 image/jpeg,.imag
0x0080       652f 706a 7065 672c 202a 2f2a 0d0a 6163 e/pjpeg,.*/*..ac
0x0090       6365 7074 2d6c 616e 6775 6167 653a 2065 cept-language:.e
0x00a0       6e2d 7573 0d0a 636f 6e74 656e 742d 7479 n-us..content-ty
0x00b0       7065 3a20 6170 706c 6963 6174 696f 6e2f pe:.application/
0x00c0       782d 7777 772d 666f 726d 2d75 726c 656e x-www-form-urlen
0x00d0       636f 6465 640d 0a55 7365 722d 4167 656e coded..User-Agen
0x00e0       743a 204d 6f7a 696c 6c61 2f34 2e30 2028 t:.Mozilla/4.0.(
0x00f0       636f 6d70 6174 6962 6c65 3b20 4d53 4945 compatible;.MSIE
0x0100       2035 2e30 313b 2057 696e 646f 7773 2039 .5.01;.Windows.9
0x0110       3529 0d0a 486f 7374 3a20 xxxx 2exx xx2e 5)..Host:.my.home.
0x0120       xxxx 2exx xxxx 0d0a 5265 6665 7265 723a net.ip..Referer:
0x0130       2068 7474 703a 2f2f xxxx 2exx xx2e xxxx .http://my.home.net
0x0140       2exx xxxx 2f63 6f6e 7461 6374 2e68 746d .ip/contact.htm
0x0150       0d0a 436f 6e74 656e 742d 4c65 6e67 7468 ..Content-Length
0x0160       3a20 3333 340d 0a                            :.334..

04:03:29.969748 IP (tos 0x0, ttl 112, id 3375, len 697) 67.28.92.222.4808 >
my.home.net.ip.80: P [tcp sum ok] 1:658(657) ack 1 win 8760 (DF)
0x0000       4500 02b9 0d2f 4000 7006 0313 431c 5cde E..../@.p...C.\.
0x0010       xxxx xxxx 12c8 0050 7a55 f420 c6ad 25bc A0.....PzU....%.
0x0020       5018 2238 ea34 0000 504f 5354 202f 6367 P."8.4..POST./cg
0x0030       692d 6269 6e2f 666f 726d 6d61 696c 2e70 i-bin/formmail.p
0x0040       6c20 4854 5450 2f31 2e30 0d0a 6163 6365 l.HTTP/1.0..acce
```

```
0x0050       7074 3a20 696d 6167 652f 6769 662c 2069 pt:.image/gif,.i
0x0060       6d61 6765 2f78 2d78 6269 746d 6170 2c20 mage/x-xbitmap,.
0x0070       696d 6167 652f 6a70 6567 2c20 696d 6167 image/jpeg,.imag
0x0080       652f 706a 7065 672c 202a 2f2a 0d0a 6163 e/pjpeg,.*/*..ac
0x0090       6365 7074 2d6c 616e 6775 6167 653a 2065 cept-language:.e
0x00a0       6e2d 7573 0d0a 636f 6e74 656e 742d 7479 n-us..content-ty
0x00b0       7065 3a20 6170 706c 6963 6174 696f 6e2f pe:.application/
0x00c0       782d 7777 772d 666f 726d 2d75 726c 656e x-www-form-urlen
0x00d0       636f 6465 640d 0a55 7365 722d 4167 656e coded..User-Agen
0x00e0       743a 204d 6f7a 696c 6c61 2f34 2e30 2028 t:.Mozilla/4.0.(
0x00f0       636f 6d70 6174 6962 6c65 3b20 4d53 4945 compatible;.MSIE
0x0100       2035 2e30 313b 2057 696e 646f 7773 2039 .5.01;.Windows.9
0x0110       3529 0d0a 486f 7374 3a20 xxxx 2exx xx2e 5)..Host:.my.home.
0x0120       xxxx 2exx xxxx 0d0a 5265 6665 7265 723a net.ip..Referer:
0x0130       2068 7474 703a 2f2f xxxx 2exx xx2e xxxx .http://my.home.net
0x0140       2exx xxxx 2f63 6f6e 7461 6374 2e68 746d .ip/contact.htm
0x0150       0d0a 436f 6e74 656e 742d 4c65 6e67 7468 ..Content-Length
0x0160       3a20 3333 340d 0a0d 0a26 656d 6169 6c3d :.334....&email=
0x0170       5664 7779 4059 6168 6f6f 2e63 6f6d 2672 Vdwy@Yahoo.com&r
0x0180       6563 6970 6965 6e74 3d41 4c49 4345 4e50 ecipient=ALICENP
0x0190       4f55 5349 4e45 4046 4c41 5348 4d41 494c OUSINE@FLASHMAIL
0x01a0       2e43 4f4d 2673 7562 6a65 6374 3d42 7669 .COM&subject=Bvi
0x01b0       740d 0a43 6f6e 7465 6e74 2d54 7970 653a t..Content-Type:
0x01c0       2074 6578 742f 6874 6d6c 3b0d 0a3c 4854 .text/html;..<HT
0x01d0       4d4c 3e3c 4845 4144 3e3c 2f68 6561 643e ML><HEAD></head>
0x01e0       3c62 6f64 793e 3c42 523e 5469 2079 7364 <body><BR>Ti.ysd
0x01f0       2074 6166 7878 7171 6569 6f20 6d63 2078 .tafxxqqeio.mc.x
0x0200       6478 7774 6320 6d20 6a6d 2071 797a 6b6b dxwtc.m.jm.qyzkk
0x0210       2062 7267 6f65 6820 7671 6764 6464 6620 .brgoeh.vqgdddf.
0x0220       7579 6e61 7462 6465 616e 2072 2067 626c uynatbdean.r.gbl
0x0230       6720 6920 6666 7973 6671 2027 3432 2e36 g.i.ffysfq.'42.6
0x0240       352e 3030 2e30 3333 2f65 756f 2d77 6f66 5.00.033/euo-wof
0x0250       2f79 676b 6464 716f 732e 6873 2720 5155 /ygkddqos.hs'.QU
0x0260       4b4b 2053 493c 4252 3e3c 2f42 4f44 593e KK.SI<BR></BODY>
0x0270       3c2f 4854 4d4c 3e0d 0a3c 212d 2d0d 0a26 </HTML>..<!--..&
0x0280       636f 6d6d 656e 7473 3d20 4562 6b20 7566 comments=.Ebk.uf
0x0290       7275 2079 6168 7374 7863 6120 7767 717a ru.yahstxca.wgqz
0x02a0       7772 206c 6d6c 2067 6367 7663 6220 2d2d wr.lml.gcgvcb.--
0x02b0       3e20 203c 212d 2d0d 0a                       >..<!--..

04:03:29.971756 IP (tos 0x0, ttl 127, id 4092, len 40) my.home.net.ip.80 >
67.28.92.222.4808: R [tcp sum ok] 3333236156:3333236156(0) win 0 (DF)
0x0000       4500 0028 0ffc 4000 7f06 f3d6 xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 12c8 c6ad 25bc 7a55 f55f C.\..P....%.zU._
0x0020       5004 0000 48ac 0000 8888 8888 8888      P...H.........

< -- Three ways handshake -- >

04:03:30.647061 IP (tos 0x0, ttl 112, id 3405, len 358) 67.28.92.222.4892 >
my.home.net.ip.80: P [tcp sum ok] 1:319(318) ack 1 win 8760 (DF)
0x0000       4500 0166 0d4d 4000 7006 0448 431c 5cde E..f.M@.p..HC.\.
0x0010       xxxx xxxx 131c 0050 7a89 88c6 c6bc 847b A0.....Pz......{
0x0020       5018 2238 9465 0000 504f 5354 202f 6367 P."8.e..POST./cg
0x0030       6962 696e 2f66 6f72 6d6d 6169 6c2e 706c ibin/formmail.pl
0x0040       2048 5454 502f 312e 300d 0a61 6363 6570 .HTTP/1.0..accep
0x0050       743a 2069 6d61 6765 2f67 6966 2c20 696d t:.image/gif,.im
0x0060       6167 652f 782d 7862 6974 6d61 702c 2069 age/x-xbitmap,.i
0x0070       6d61 6765 2f6a 7065 672c 2069 6d61 6765 mage/jpeg,.image
```

```
0x0080       2f70 6a70 6567 2c20 2a2f 2a0d 0a61 6363 /pjpeg,.*/*..acc
0x0090       6570 742d 6c61 6e67 7561 6765 3a20 656e ept-language:.en
0x00a0       2d75 730d 0a63 6f6e 7465 6e74 2d74 7970 -us..content-typ
0x00b0       653a 2061 7070 6c69 6361 7469 6f6e 2f78 e:.application/x
0x00c0       2d77 7777 2d66 6f72 6d2d 7572 6c65 6e63 -www-form-urlenc
0x00d0       6f64 6564 0d0a 5573 6572 2d41 6765 6e74 oded..User-Agent
0x00e0       3a20 4d6f 7a69 6c6c 612f 342e 3020 2863 :.Mozilla/4.0.(c
0x00f0       6f6d 7061 7469 626c 653b 204d 5349 4520 ompatible;.MSIE.
0x0100       352e 3031 3b20 5769 6e64 6f77 7320 3935 5.01;.Windows.95
0x0110       290d 0a48 6f73 743a 20xx xx2e xxxx 2exx )..Host:.my.home.n
0x0120       xx2e xxxx xx0d 0a52 6566 6572 6572 3a20 et.ip..Referer:.
0x0130       6874 7470 3a2f 2fxx xx2e xxxx 2exx xx2e http://my.home.net.
0x0140       xxxx xx2f 636f 6e74 6163 742e 6874 6d0d ip/contact.htm.
0x0150       0a43 6f6e 7465 6e74 2d4c 656e 6774 683a .Content-Length:
0x0160       2033 3633 0d0a                                .363..

04:03:30.836936 IP (tos 0x0, ttl 127, id 4094, len 40) my.home.net.ip.80 >
67.28.92.222.4892: . [tcp sum ok] 1:1(0) ack 319 win 5522 (DF)
0x0000       4500 0028 0ffe 4000 7f06 f3d4 xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 131c c6bc 847b 7a89 8a04 C.\..P.....{z...
0x0020       5010 1592 3f13 0000 8888 8888 8888      P...?.........

04:03:31.108060 IP (tos 0x0, ttl 112, id 3429, len 407) 67.28.92.222.4892 >
my.home.net.ip.80: P [tcp sum ok] 319:686(367) ack 1 win 8760 (DF)
0x0000       4500 0197 0d65 4000 7006 03ff 431c 5cde E....e@.p...C.\.
0x0010       xxxx xxxx 131c 0050 7a89 8a04 c6bc 847b A0.....Pz......{
0x0020       5018 2238 32e3 0000 0d0a 2665 6d61 696c P."82.....&email
0x0030       3d51 6464 6c67 4059 6168 6f6f 2e63 6f6d =Qddlg@Yahoo.com
0x0040       2672 6563 6970 6965 6e74 3d41 4c49 4345 &recipient=ALICE
0x0050       4e50 4f55 5349 4e45 4046 4c41 5348 4d41 NPOUSINE@FLASHMA
0x0060       494c 2e43 4f4d 2673 7562 6a65 6374 3d42 IL.COM&subject=B
0x0070       7669 740d 0a43 6f6e 7465 6e74 2d54 7970 vit..Content-Typ
0x0080       653a 2074 6578 742f 6874 6d6c 3b0d 0a3c e:.text/html;..<
0x0090       4854 4d4c 3e3c 4845 4144 3e3c 2f68 6561 HTML><HEAD></hea
0x00a0       643e 3c62 6f64 793e 3c42 523e 476e 6b63 d><body><BR>Gnkc
0x00b0       7475 6573 2063 6872 7973 6c79 6a6a 206f tues.chryslyjj.o
0x00c0       7770 7265 7278 7564 6320 796b 6765 6b74  wprerxudc.ykgekt
0x00d0       206c 206a 7674 2069 7a6a 6c20 6b61 6566 .l.jvt.izjl.kaef
0x00e0       6567 7320 7a6f 7675 6365 6762 6f20 736b egs.zovucegbo.sk
0x00f0       6263 6d62 646b 6920 6774 7468 6c72 6670 bcmbdki.gtthlrfp
0x0100       6672 206e 2066 206c 6420 676d 706e 7362 fr.n.f.ld.gmpnsb
0x0110       206d 6e73 6420 7120 686b 7579 7369 6720 .mnsd.q.hkuysig.
0x0120       6669 6120 2734 322e 3635 2e30 302e 3033 fia.'42.65.00.03
0x0130       332f 6575 6f77 6f66 2f79 676b 6464 716f 3/euowof/ygkddqo
0x0140       732e 6873 2720 5155 4b4b 2053 493c 4252 s.hs'.QUKK.SI<BR
0x0150       3e3c 2f42 4f44 593e 3c2f 4854 4d4c 3e0d ></BODY></HTML>.
0x0160       0a3c 212d 2d0d 0a26 636f 6d6d 656e 7473 .<!--..&comments
0x0170       3d20 5065 2073 6a64 6f6a 656c 7120 6862 =.Pe.sjdojelq.hb
0x0180       6270 207a 6e78 6e7a 696f 6920 2d2d 3e20 bp.znxnzioi.-->.
0x0190       203c 212d 2d0d 0a                             .<!--..

04:03:31.109710 IP (tos 0x0, ttl 127, id 4095, len 40) my.home.net.ip.80 >
67.28.92.222.4892: R [tcp sum ok] 3334243451:3334243451(0) win 0 (DF)
0x0000       4500 0028 0fff 4000 7f06 f3d3 xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 131c c6bc 847b 7a89 8a04 C.\..P.....{z...
0x0020       5004 0000 54b1 0000 8888 8888 8888      P...T.........

< -- Three ways handshake -- >
04:03:40.687883 IP (tos 0x0, ttl 112, id 3821, len 359) 67.28.92.222.4917 >
my.home.net.ip.80: P [tcp sum ok] 1:320(319) ack 1 win 8760 (DF)
0x0000       4500 0167 0eed 4000 7006 02a7 431c 5cde E..g..@.p...C.\.
0x0010       xxxx xxxx 1335 0050 7a96 8eb9 c6e3 5df3 A0...5.Pz.....].
0x0020       5018 2238 0801 0000 504f 5354 202f 6367 P."8....POST./cg
0x0030       6962 696e 2f66 6f72 6d6d 6169 6c2e 6367 ibin/formmail.cg
0x0040       6920 4854 5450 2f31 2e30 0d0a 6163 6365 i.HTTP/1.0..acce
0x0050       7074 3a20 696d 6167 652f 6769 662c 2069 pt:.image/gif,.i
0x0060       6d61 6765 2f78 2d78 6269 746d 6170 2c20 mage/x-xbitmap,.
0x0070       696d 6167 652f 6a70 6567 2c20 696d 6167 image/jpeg,.imag
0x0080       652f 706a 7065 672c 202a 2f2a 0d0a 6163 e/pjpeg,.*/*..ac
0x0090       6365 7074 2d6c 616e 6775 6167 653a 2065 cept-language:.e
0x00a0       6e2d 7573 0d0a 636f 6e74 656e 742d 7479 n-us..content-ty
0x00b0       7065 3a20 6170 706c 6963 6174 696f 6e2f pe:.application/
0x00c0       782d 7777 772d 666f 726d 2d75 726c 656e x-www-form-urlen
0x00d0       636f 6465 640d 0a55 7365 722d 4167 656e coded..User-Agen
0x00e0       743a 204d 6f7a 696c 6c61 2f34 2e30 2028 t:.Mozilla/4.0.(
0x00f0       636f 6d70 6174 6962 6c65 3b20 4d53 4945 compatible;.MSIE
0x0100       2035 2e30 313b 2057 696e 646f 7773 2039 .5.01;.Windows.9
0x0110       3529 0d0a 486f 7374 3a20 xxxx 2exx xx2e 5)..Host:.my.home.
0x0120       xxxx 2exx xxxx 0d0a 5265 6665 7265 723a net.ip..Referer:
0x0130       2068 7474 703a 2f2f xxxx 2exx xx2e xxxx .http://my.home.net
0x0140       2exx xxxx 2f63 6f6e 7461 6374 2e68 746d .ip/contact.htm
0x0150       0d0a 436f 6e74 656e 742d 4c65 6e67 7468 ..Content-Length
0x0160       3a20 3337 330d 0a                              :.373..

04:03:40.868938 IP (tos 0x0, ttl 127, id 4099, len 40) my.home.net.ip.80 >
67.28.92.222.4917: . [tcp sum ok] 1:1(0) ack 320 win 5521 (DF)
0x0000       4500 0028 1003 4000 7f06 f3cf xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 1335 c6e3 5df3 7a96 8ff8 C.\..P.5..].z...
0x0020       5010 1591 5f5b 0000 8888 8888 8888      P..._[........

04:03:41.178381 IP (tos 0x0, ttl 112, id 3848, len 417) 67.28.92.222.4917 >
my.home.net.ip.80: P [tcp sum ok] 320:697(377) ack 1 win 8760 (DF)
0x0000       4500 01a1 0f08 4000 7006 0252 431c 5cde E.....@.p..RC.\.
0x0010       xxxx xxxx 1335 0050 7a96 8ff8 c6e3 5df3 A0...5.Pz.....].
0x0020       5018 2238 9573 0000 0d0a 2665 6d61 696c P."8.s....&email
0x0030       3d48 404d 534e 2e63 6f6d 2672 6563 6970 =H@MSN.com&recip
0x0040       6965 6e74 3d41 4c49 4345 4e50 4f55 5349 ient=ALICENPOUSI
0x0050       4e45 4046 4c41 5348 4d41 494c 2e43 4f4d NE@FLASHMAIL.COM
0x0060       2673 7562 6a65 6374 3d42 7669 740d 0a43 &subject=Bvit..C
0x0070       6f6e 7465 6e74 2d54 7970 653a 2074 6578 ontent-Type:.tex
0x0080       742f 6874 6d6c 3b0d 0a3c 4854 4d4c 3e3c t/html;..<HTML><
0x0090       4845 4144 3e3c 2f68 6561 643e 3c62 6f64 HEAD></head><bod
0x00a0       793e 3c42 523e 4375 7a63 6167 7020 6161 y><BR>Cuzcagp.aa
0x00b0       6672 6977 2075 7920 6c67 7720 7574 766f friw.uy.lgw.utvo
0x00c0       6b6f 6471 2072 7475 7120 7268 2078 7262 kodq.rtuq.rh.xrb
0x00d0       7773 7a65 7676 7020 7668 6e75 2074 6d6f wszevvp.vhnu.tmo
0x00e0       6276 2072 617a 6b76 686b 6920 7178 696a bv.razkvhki.qxij
0x00f0       6e73 2065 6d77 6770 6e68 2063 6362 6577 ns.emwgpnh.ccbew
0x0100       7477 6c73 7220 6264 7a6c 7470 6879 7a6a twlsr.bdzltphyzj
0x0110       2027 3432 2e36 352e 3030 2e30 3333 2f65 .'42.65.00.033/e
0x0120       756f 776f 662f 7967 6b64 6471 6f73 2e65 uowof/ygkddqos.e
0x0130       756f 2720 5155 4b4b 2053 493c 4252 3e3c uo'.QUKK.SI<BR><
0x0140       2f42 4f44 593e 3c2f 4854 4d4c 3e0d 0a3c /BODY></HTML>..<
0x0150       212d 2d0d 0a26 636f 6d6d 656e 7473 3d20 !--..&comments=.
0x0160       4478 7165 706f 6374 6320 786b 7863 7a6a Dxqepoctc.xkxczj
```

```
0x0170       6873 6420 6d6b 7179 6671 6b20 6279 6e6f hsd.mkqyfqk.byno
0x0180       656f 2076 7066 6b6b 6a6d 2062 6575 6161 eo.vpfkkjm.beuaa
0x0190       696b 6c68 7420 2d2d 3e20 203c 212d 2d0d iklht.-->..<!--.
0x01a0            0a                                                .

04:03:41.180121 IP (tos 0x0, ttl 127, id 4100, len 40) my.home.net.ip.80 >
67.28.92.222.4917: R [tcp sum ok] 3336789491:3336789491(0) win 0 (DF)
0x0000       4500 0028 1004 4000 7f06 f3ce xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 1335 c6e3 5df3 7a96 8ff8 C.\..P.5..].z...
0x0020       5004 0000 74f8 0000 8888 8888 8888      P...t.........

< -- Three ways handshake -- >

04:03:41.828054 IP (tos 0x0, ttl 112, id 3883, len 361) 67.28.92.222.1101 >
my.home.net.ip.80: P [tcp sum ok] 1:322(321) ack 1 win 8760 (DF)
0x0000       4500 0169 0f2b 4000 7006 0267 431c 5cde E..i.+@.p..gC.\.
0x0010       xxxx xxxx 044d 0050 7b17 6615 c6e8 ea8f A0...M.P{.f.....
0x0020       5018 2238 3b36 0000 504f 5354 202f 6367 P."8;6..POST./cg
0x0030       692d 6c6f 6361 6c2f 666f 726d 6d61 696c i-local/formmail
0x0040       2e70 6c20 4854 5450 2f31 2e30 0d0a 6163 .pl.HTTP/1.0..ac
0x0050       6365 7074 3a20 696d 6167 652f 6769 662c cept:.image/gif,
0x0060       2069 6d61 6765 2f78 2d78 6269 746d 6170 .image/x-xbitmap
0x0070       2c20 696d 6167 652f 6a70 6567 2c20 696d ,.image/jpeg,.im
0x0080       6167 652f 706a 7065 672c 202a 2f2a 0d0a age/pjpeg,.*/*..
0x0090       6163 6365 7074 2d6c 616e 6775 6167 653a accept-language:
0x00a0       2065 6e2d 7573 0d0a 636f 6e74 656e 742d .en-us..content-
0x00b0       7479 7065 3a20 6170 706c 6963 6174 696f type:.applicatio
0x00c0       6e2f 782d 7777 772d 666f 726d 2d75 726c n/x-www-form-url
0x00d0       656e 636f 6465 640d 0a55 7365 722d 4167 encoded..User-Ag
0x00e0       656e 743a 204d 6f7a 696c 6c61 2f34 2e30 ent:.Mozilla/4.0
0x00f0       2028 636f 6d70 6174 6962 6c65 3b20 4d53 .(compatible;.MS
0x0100       4945 2035 2e30 313b 2057 696e 646f 7773 IE.5.01;.Windows
0x0110       2039 3529 0d0a 486f 7374 3a20 xxxx 2exx .95)..Host:.my.home
0x0120       xx2e xxxx 2exx xxxx 0d0a 5265 6665 7265 .net.ip..Refere
0x0130       723a 2068 7474 703a 2f2f xxxx 2exx xx2e r:.http://my.home.
0x0140       xxxx 2exx xxxx 2f63 6f6e 7461 6374 2e68 net.ip/contact.h
0x0150       746d 0d0a 436f 6e74 656e 742d 4c65 6e67 tm..Content-Leng
0x0160       7468 3a20 3337 340d 0a                       th:.374..

04:03:41.972538 IP (tos 0x0, ttl 127, id 4103, len 40) my.home.net.ip.80 >
67.28.92.222.1101: . [tcp sum ok] 1:1(0) ack 322 win 5519 (DF)
0x0000       4500 0028 1007 4000 7f06 f3cb xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 044d c6e8 ea8f 7b17 6756 C.\..P.M....{.gV
0x0020       5010 158f 09c5 0000 8888 8888 8888      P.............

04:03:42.237918 IP (tos 0x0, ttl 112, id 3904, len 418) 67.28.92.222.1101 >
my.home.net.ip.80: P [tcp sum ok] 322:700(378) ack 1 win 8760 (DF)
0x0000       4500 01a2 0f40 4000 7006 0219 431c 5cde E....@@.p...C.\.
0x0010       xxxx xxxx 044d 0050 7b17 6756 c6e8 ea8f A0...M.P{.gV....
0x0020       5018 2238 80bb 0000 0d0a 2665 6d61 696c P."8......&email
0x0030       3d42 7264 6d73 6d6c 6972 7062 4059 6168 =Brdmsmlirpb@Yah
0x0040       6f6f 2e63 6f6d 2672 6563 6970 6965 6e74 oo.com&recipient
0x0050       3d41 4c49 4345 4e50 4f55 5349 4e45 4046 =ALICENPOUSINE@F
0x0060       4c41 5348 4d41 494c 2e43 4f4d 2673 7562 LASHMAIL.COM&sub
0x0070       6a65 6374 3d42 7669 740d 0a43 6f6e 7465 ject=Bvit..Conte
0x0080       6e74 2d54 7970 653a 2074 6578 742f 6874 nt-Type:.text/ht
0x0090       6d6c 3b0d 0a3c 4854 4d4c 3e3c 4845 4144 ml;..<HTML><HEAD
0x00a0       3e3c 2f68 6561 643e 3c62 6f64 793e 3c42 ></head><body><B
```

```
0x00b0       523e 4771 6620 6f6d 6520 7878 6c70 7620 R>Gqf.ome.xxlpv.
0x00c0       746a 7665 2065 6b67 7168 746b 2074 7278 tjve.ekgqhtk.trx
0x00d0       666e 7172 2069 6675 766c 6f79 6378 206b fnqr.ifuvloycx.k
0x00e0       6b6a 6d65 206c 7568 6820 6b6c 6874 6978 kjme.luhh.klhtix
0x00f0       206e 6873 6e69 7075 6d6d 2066 7478 2027 .nhsnipumm.ftx.'
0x0100       3432 2e36 352e 3030 2e30 3333 2f65 756f 42.65.00.033/euo
0x0110       2d73 6765 7173 2f79 676b 6464 716f 732e -sgeqs/ygkddqos.
0x0120       6873 2720 5155 4b4b 2053 493c 4252 3e3c hs'.QUKK.SI<BR><
0x0130       2f42 4f44 593e 3c2f 4854 4d4c 3e0d 0a3c /BODY></HTML>..<
0x0140       212d 2d0d 0a26 636f 6d6d 656e 7473 3d20 !--..&comments=.
0x0150       427a 666e 6f7a 7a65 7167 2064 7477 676b Bzfnozzeqg.dtwgk
0x0160       6676 7373 2075 6e6a 6e63 7069 2073 7470 fvss.unjncpi.stp
0x0170       6371 6779 2071 6176 7278 6475 756f 2075 cqgy.qavrxduuo.u
0x0180       666d 206b 736c 6e61 7574 7120 796a 7567 fm.kslnautq.yjug
0x0190       6a68 6e70 7768 202d 2d3e 2020 3c21 2d2d jhnpwh.-->..<!--
0x01a0            0d0a                                              ..

04:03:42.239684 IP (tos 0x0, ttl 127, id 4104, len 40) my.home.net.ip.80 >
67.28.92.222.1101: R [tcp sum ok] 3337153167:3337153167(0) win 0 (DF)
0x0000       4500 0028 1008 4000 7f06 f3ca xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 044d c6e8 ea8f 7b17 6756 C.\..P.M....{.gV
0x0020       5004 0000 1f60 0000 8888 8888 8888      P....`........

< -- Three ways handshake -- >

04:03:42.927935 IP (tos 0x0, ttl 112, id 3934, len 362) 67.28.92.222.1114 >
my.home.net.ip.80: P [tcp sum ok] 1:323(322) ack 1 win 8760 (DF)
0x0000       4500 016a 0f5e 4000 7006 0233 431c 5cde E..j.^@.p..3C.\.
0x0010       xxxx xxxx 045a 0050 7b25 95e4 c6ed ee69 A0...Z.P{%.....i
0x0020       5018 2238 63c1 0000 504f 5354 202f 6367 P."8c...POST./cg
0x0030       692d 6c6f 6361 6c2f 666f 726d 6d61 696c i-local/formmail
0x0040       2e63 6769 2048 5454 502f 312e 300d 0a61 .cgi.HTTP/1.0..a
0x0050       6363 6570 743a 2069 6d61 6765 2f67 6966 ccept:.image/gif
0x0060       2c20 696d 6167 652f 782d 7862 6974 6d61 ,.image/x-xbitma
0x0070       702c 2069 6d61 6765 2f6a 7065 672c 2069 p,.image/jpeg,.i
0x0080       6d61 6765 2f70 6a70 6567 2c20 2a2f 2a0d mage/pjpeg,.*/*.
0x0090       0a61 6363 6570 742d 6c61 6e67 7561 6765 .accept-language
0x00a0       3a20 656e 2d75 730d 0a63 6f6e 7465 6e74 :.en-us..content
0x00b0       2d74 7970 653a 2061 7070 6c69 6361 7469 -type:.applicati
0x00c0       6f6e 2f78 2d77 7777 2d66 6f72 6d2d 7572 on/x-www-form-ur
0x00d0       6c65 6e63 6f64 6564 0d0a 5573 6572 2d41 lencoded..User-A
0x00e0       6765 6e74 3a20 4d6f 7a69 6c6c 612f 342e gent:.Mozilla/4.
0x00f0       3020 2863 6f6d 7061 7469 626c 653b 204d 0.(compatible;.M
0x0100       5349 4520 352e 3031 3b20 5769 6e64 6f77 SIE.5.01;.Window
0x0110       7320 3935 290d 0a48 6f73 743a 20xx xx2e s.95)..Host:.my.
0x0120       xxxx 2exx xx2e xxxx xx0d 0a52 6566 6572 home.net.ip..Refer
0x0130       6572 3a20 6874 7470 3a2f 2fxx xx2e xxxx er:.http://my.home
0x0140       2exx xx2e xxxx xx2f 636f 6e74 6163 742e .net.ip/contact.
0x0150       6874 6d0d 0a43 6f6e 7465 6e74 2d4c 656e htm..Content-Len
0x0160       6774 683a 2033 3032 0d0a                          gth:.302..

04:03:43.076090 IP (tos 0x0, ttl 127, id 4106, len 40) my.home.net.ip.80 >
67.28.92.222.1114: . [tcp sum ok] 1:1(0) ack 323 win 5518 (DF)
0x0000       4500 0028 100a 4000 7f06 f3c8 xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 045a c6ed ee69 7b25 9726 C.\..P.Z...i{%.&
0x0020       5010 158e d5fb 0000 8888 8888 8888      P.............

04:03:43.327701 IP (tos 0x0, ttl 112, id 3948, len 346) 67.28.92.222.1114 >
```

```
my.home.net.ip.80: P [tcp sum ok] 323:629(306) ack 1 win 8760 (DF)
0x0000       4500 015a 0f6c 4000 7006 0235 431c 5cde E..Z.l@.p..5C.\.
0x0010       xxxx xxxx 045a 0050 7b25 9726 c6ed ee69 A0...Z.P{%.&...i
0x0020       5018 2238 e8bb 0000 0d0a 2665 6d61 696c P."8......&email
0x0030       3d4f 7977 6973 6662 7a66 6e76 4059 6168 =Oywisfbzfnv@Yah
0x0040       6f6f 2e63 6f6d 2672 6563 6970 6965 6e74 oo.com&recipient
0x0050       3d41 4c49 4345 4e50 4f55 5349 4e45 4046 =ALICENPOUSINE@F
0x0060       4c41 5348 4d41 494c 2e43 4f4d 2673 7562 LASHMAIL.COM&sub
0x0070       6a65 6374 3d42 7669 740d 0a43 6f6e 7465 ject=Bvit..Conte
0x0080       6e74 2d54 7970 653a 2074 6578 742f 6874 nt-Type:.text/ht
0x0090       6d6c 3b0d 0a3c 4854 4d4c 3e3c 4845 4144 ml;..<HTML><HEAD
0x00a0       3e3c 2f68 6561 643e 3c62 6f64 793e 3c42 ></head><body><B
0x00b0       523e 5562 6c20 6563 776d 7272 7174 6c20 R>Ubl.ecwmrrqtl.
0x00c0       6c61 6868 2071 736f 6169 6520 6e6f 796e lahh.qsoaie.noyn
0x00d0       7077 626d 7320 6674 6564 7220 726b 6d7a pwbms.ftedr.rkmz
0x00e0       2027 3432 2e36 352e 3030 2e30 3333 2f65 .'42.65.00.033/e
0x00f0       756f 2d73 6765 7173 2f79 676b 6464 716f uo-sgeqs/ygkddqo
0x0100       732e 6575 6f27 2051 554b 4b20 5349 3c42 s.euo'.QUKK.SI<B
0x0110       523e 3c2f 424f 4459 3e3c 2f48 544d 4c3e R></BODY></HTML>
0x0120       0d0a 3c21 2d2d 0d0a 2663 6f6d 6d65 6e74 ..<!--..&comment
0x0130       733d 2045 716e 6364 7464 676b 6620 7a7a s=.Eqncdtdgkf.zz
0x0140       7962 6d71 746a 7020 787a 6177 6971 202d ybmqtjp.xzawiq.-
0x0150       2d3e 2020 3c21 2d2d 0d0a                         ->..<!--..
04:03:43.329495 IP (tos 0x0, ttl 127, id 4107, len 40) my.home.net.ip.80 >
67.28.92.222.1114: R [tcp sum ok] 3337481833:3337481833(0) win 0 (DF)
0x0000       4500 0028 100b 4000 7f06 f3c7 xxxx xxxx E..(..@.....A0..
0x0010       431c 5cde 0050 045a c6ed ee69 7b25 9726 C.\..P.Z...i{%.&
0x0020       5004 0000 eb95 0000 8888 8888 8888      P.............
```

1. **Source of Trace**
   This detects is from my home network. The address my.home.net.ip was the address assigned by ISP at the time the packets were captured. Another IP address 67.28.92.222 in the log is the attacker's IP address.

2. **Detect was generated by:**
   Detect was generated by Ethereal. Ethereal captured the network traffic passing the external interface of my home firewall.

3. **Probability the source address was spoofed:**
   As the attacker must maintain TCP sessions to be able to launch the attack, the probability the source address was spoofed is low.

4. **Description of attack:**
   This is an attack looking for web server with vulnerable formmail perl or cgi script. If the attack succeeds, spammer will use this web server to send spam e-mail anonymously.
   Overview of formmail [32]: FormMail is a generic HTML form to e-mail gateway that parses the results of any form and sends them to the specified users. … This also makes FormMail the perfect system-wide solution for allowing users form-based user feedback capabilities without the risks of allowing freedom of CGI access. … FormMail is quite possibly the most used CGI program on the Internet,

having been downloaded over 2,000,000 times since 1997.

The following HTML source code for formmail can show us how formmail works.

```
<form method="POST" action="http://www.myexample.com/cgi-
bin/formmail.pl">
     <p>
            <input type="hidden" name="recipient"
            value="staff@myexample.com,webmaster@myexample.com">
            <input type="hidden" name="required" value="email,subject">
            <input type="hidden" name="env_report"
            value="HTTP_USER_AGENT">
            Fill out the form and click on the &quot;Submit&quot; button
     </p>
     <table>
            <tr><td class="trebuchet11" align="right">Your name:</td>
            </tr>
            <tr><td class="trebuchet11" align="right">Your
            email:</td></tr>
            <tr><td class="trebuchet11" align="right">Your
            subject:</td></tr>
     </table>
     <p align="center" class="trebuchet11"> Please to hear your comment:
     <br><textarea name="comments"></textarea></p>
     <p><input type="submit" name="Submit" value="Submit">&nbsp; </p>
</form>
```

In normal situation, the recipient of the e-mail should always be the people behind staff@myexample.com and webmaster@myexample.com. The hidden “recipient” varible highlighted in Bold is not subjected to change by the web users. However, bugs existing in formmail script prior to version 1.92 cause the script to be hijacked to send e-mail to the e-mail accounts spammer want, instead of the recipient specified by the web master. To do this, a malicious user can simply include the list of target email addresses in an HTTP request to Formmail. This behavior makes tracking down the origin of the spam difficult because the only place the spammers IP address is saved is in the Web logs of the affected site [33].

On March 07, 2001, the vulnerability was announced at BUGTRAQ ID 2469 named FormMail Recipient CGI Variable Spamming Vulnerability [34]. On January 23, 2002, monkeys.com published their advisory [35] detailed the various methods and mechanisms for the remote abuse of the formmail script.

5. Attack mechanism:

The attack mechanism of this detect could be clearly revealed when the TCP session is replayed by Ethereal. Here is one of the tcp streams the attacker sent to the web server:

```
POST /cgi-local/formmail.cgi HTTP/1.0
accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
accept-language: en-us
content-type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows 95)
Host: my.home.net.ip
Referer: http://my.home.net.ip/contact.htm
```

```
Content-Length: 302

&email=Oywisfbzfnv@Yahoo.com&recipient=ALICENPOUSINE@FLASHMAIL.COM&subject=
Bvit
Content-Type: text/html;
<HTML><HEAD></head><body><BR>Ubl ecwmrrqtl lahh qsoaie noynpwbms ftedr rkmz
'42.65.00.033/euo-sgeqs/ygkddqos.euo' QUKK SI<BR></BODY></HTML>
<!--&comments= Eqncdtdgkf zzybmqtjp xzawiq -->  <!—
```

First of all, my web server needs authentication and does not have a web page called /cgi-local/formmail.cgi. There is no form for the attacker to fill and sent back to me. The attacker brute forces the web server probing for the formmail.cgi script.

Secondly, the referrer was crafted to http://my.home.net.ip/contact.htm. This is one of the key points for the attack to work. FormMail relies on the HTTP_REFERER header to establish the identity of the user. Forged HTTP_REFERERS may circumvent the measures employed by FormMail to validate the authenticity of the user. It is trivial for a remote attacker to craft his own HTTP_REFERER header [36]. The IP address of the web server is always one of the referrer configured in formmail. As long as the spoof referrer matches the one in formmail's configuration, the script will be happy to deliver message for the user.

Thirdly, the recipient's e-mail address is set to ALICENPOUSINE@FLASHMAIL.COM. The recipient variable in the formmail script is for the web master to specify the e-mail account to receive feedback or questions from the web user. It should not be changed by the web users. However, the bugs existed in the script allow the recipient variable to be overwritten to arbitrary e-mail account which cause the script became a tool of open relay. If the attack succeeds, the spammer behind e-mail account ALICENPOUSINE@FLASHMAIL.COM will receive an e-mail from my web server and know that the web server can be his launch pad for the next spam.

The detect has 18 minutes activity from 4:03:25 – 4:03:43 August 17,2002. The attacker's tool tried 3 different directory names and two formmail script names as following sequence:

1. /cgi-bin/formmail.cgi 2. /cgi-bin/formmail.pl 3. /cgibin/formmail.pl
4. /cgibin/formmail.cgi 5. /cgi-local/formmail.pl 6. /cgi-local/formmail.cgi

Obviously, since the attacker does not have the knowledge of the name of the directory which host the script and does not know whether the suffix of the script is a cgi or pl. The attacking tool has to try its best to do the best guess and try all the possibility it can think of.

In every attempt, the recipient e-mail account is the same - ALICENPOUSINE@FLASHMAIL.COM, meaning that the attacks are from the same attacker and he want to harvest the result from the aforementioned e-mail. But the email variable changed in every attempt. The e-mail accounts are all with the free e-mail server, such as yahoo.com, home.net. The account names look like something randomly generated. What is the email variable does not matter at all in this kind of attack. It is presented just because it is a required field for the formmail script.

The e-mail content in each attempt looks alike. It does not mean anything to me because it is not in English. But inside each e-mail, there is an IP address like string “42.65.00.033”. A whois lookup in www.arin.net told me that it is an IANA reserved address.

All the attacks did not succeed because my web server does not have formmail script and RESETs were sent to end the TCP connection.

6. Correlations:

My web server just has very limited functionality. It has no access log. Thus I do not have web server log to correlate with the attack seen from the network sniffer. But there are other web servers that had logged the same kind of formmail probe.

Thomas Luethi (pubtom@bigfoot.com) post his web server log data at newsgroup ch.admin

Here is the log he posted:

Error.log:

```
[Sun Jul  7 03:59:34 2002] [error] [client 216.126.156.176] script
  not found or unable to stat: /usr/local/httpd/cgi-bin/formmail.cgi
[Sun Jul  7 03:59:35 2002] [error] [client 216.126.156.176] script
  not found or unable to stat: /usr/local/httpd/cgi-bin/formmail.pl
[Sun Jul  7 03:59:37 2002] [error] [client 216.126.156.176]
  File does not exist: /home/*domain*.ch/html/cgi-local/formmail.pl
[Sun Jul  7 03:59:37 2002] [error] [client 216.126.156.176]
  File does not exist: /home/*domain*.ch/html/cgi-local/formmail.cgi
[Sun Jul  7 03:59:39 2002] [error] [client 216.126.156.176]
  File does not exist: /home/*domain*.ch/html/cgibin/formmail.pl
[Sun Jul  7 03:59:39 2002] [error] [client 216.126.156.176]
  File does not exist: /home/*domain*.ch/html/cgibin/formmail.cgi
```

Aus transfer.log:

```
04-176.119.popsite.net - - [07/Jul/2002:03:59:35 +0200] "GET
/cgi-bin/formmail.cgi?recipient=tandyhull@yahoo.com&subject=your%20open
%20formmail%20should%20be%20closed&email=williamhardy@yahoo.com&=http://
www.*domain*.ch/cgi-bin/formmail.cgi
HTTP/1.1" 404 296 "-" "Microsoft URL Control - 6.00.8862"
04-176.119.popsite.net - - [07/Jul/2002:03:59:35 +0200] "GET
/cgi-bin/formmail.pl?recipient=tandyhull@yahoo.com&subject=your%20open
%20formmail%20should%20be%20closed&email=williamhardy@yahoo.com&=http://
www.*domain*.ch/cgi-bin/formmail.pl
HTTP/1.1" 404 295 "-" "Microsoft URL Control - 6.00.8862"
04-176.119.popsite.net - - [07/Jul/2002:03:59:37 +0200] "GET
/cgi-
local/formmail.pl?recipient=tandyhull@yahoo.com&subject=your%20open%
20formmail%20should%20be%20closed&email=williamhardy@yahoo.com&=http://w
ww.*domain*.ch/cgi-local/formmail.pl
HTTP/1.1" 302 291 "-" "Microsoft URL Control - 6.00.8862"
04-176.119.popsite.net - - [07/Jul/2002:03:59:37 +0200] "GET
/cgi-
local/formmail.cgi?recipient=tandyhull@yahoo.com&subject=your%20open
%20formmail%20should%20be%20closed&email=williamhardy@yahoo.com&=http://
www.*domain*.ch/cgi-local/formmail.cgi
HTTP/1.1" 302 291 "-" "Microsoft URL Control - 6.00.8862"
04-176.119.popsite.net - - [07/Jul/2002:03:59:39 +0200] "GET
/cgibin/formmail.pl?recipient=tandyhull@yahoo.com&subject=your%20open%20
formmail%20should%20be%20closed&email=williamhardy@yahoo.com&=http://www
```

```
.*domain*.ch/cgibin/formmail.pl
HTTP/1.1" 302 291 "-" "Microsoft URL Control - 6.00.8862"
04-176.119.popsite.net - - [07/Jul/2002:03:59:39 +0200] "GET
/cgibin/formmail.cgi?recipient=tandyhull@yahoo.com&subject=your%20open%2
0formmail%20should%20be%20closed&email=williamhardy@yahoo.com&=http://ww
w.*domain*.ch/cgibin/formmail.cgi
HTTP/1.1" 302 291 "-" "Microsoft URL Control - 6.00.8862"
```

We can see from the log that the attacker was probing a Microsoft IIS server. He tried the same directory name and script name as what I have in the detect #3. This leads me to guess that the spammers use the same tool looking around for the vulnerable formmail. The date of the above log is less than 2 months away from my captures.

There is a mailing list thread [38] discuss the exploit against formmail. One of the discussion mentioned that a vulnerable old formmail script was exploited and used by the spammer to send bulk mail so that the web server was black listed in the spam.net. This is another impact of having a vulnerable formmail script. Not only the web server help the spammers send bulk mails and hide their identifies, but also influence the normal functionality and the reputation of the organization.

IP report in Dshield.org provides a mean of correlation by checking whether the same IP address seen from this detect attacks other targets on Internet or not; how many attacks it has tried; which ports it attacks and who is it. Dshield.org shows the IP has attacked 4 targets on HTTP port. But since the hostname indicates that the address is assigned to dialup user, the machine that attacked my web server might not be the same one that attacked the other targets.

IP Address:

67.28.92.222

HostName:

dialup-67.28.92.222.Dial1.SaintLouis1.Level3.net

DShield Profile:

Country:
US

Contact E-mail:
security@level3.com

Total Records against IP:
4

Number of targets:
4

Date Range:
2002-08-17 to 2002-08-17

Ports Attacked (up to 10):

Port	Attacks
80	4

Fightback:

not sent

Whois:

```
Level 3 Communications, Inc. (NETBLK-LC-ORG-ARIN-BLK3)
   1025 Eldorado Boulevard,
   Broomfield, CO 80021
   US

   Netname: LC-ORG-ARIN-BLK3
   Netblock: 67.24.0.0 - 67.31.255.255
   Maintainer: LVLT

   Coordinator:
      level Communications  (LC-ORG-ARIN)  ipaddressing@level3.com
      +1 (877) 453-8353

   Domain System inverse mapping provided by:

   NS1.LEVEL3.NET               209.244.0.1
   NS2.LEVEL3.NET               209.244.0.2

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
```

7. **Evidence of active targeting:**

The attacker connected to my web server over TCP session, sending crafted packet to probe whether my web server has the formmail script and whether the formmail script is vulnerable. It is very obvious that the attack was targeting my web server.

8. **Severity**

Criticality: 2. The web server is just a personal server.
Lethality: 1. No formmail script is installed. The attack is unlikely to succeed.
System countermeasures: 5. The system countermeasure is reasonable. Authentication is required to access. Formmail is not installed at all.
Network countermeasures: 5. The network countermeasure is reasonable. The web service is desire to be reachable from Internet.

Severity = (criticality + lethality) - (system countermeasures + network countermeasures) = (2+1) – (5+5) = -7

9. **Defensive recommendation:**

Upgrade!

A newer version of Matt Wright's FormMail script, Version 1.92, is now available http://www.scriptarchive.com/formmail.html.

To this kind of formmail probing, we should send the evidence (log) with the e-mail address in the recipient field to the mail service provider and ask them to cancel the account.

10. **Multiple choice test question:**

Which of the following element is not the key point when probing the formmail vulnerability:

A. E-mail variable of the script
B. Recipient variable of the script.
C. Referer variable of the script.
D. Directiory name of the script.

The answer is A.

List of Reference:

[1]. Intrusion Mailing list
URL:http://www.incidents.org/archives/intrusions/msg14574.html
[2]. Intrusion Mailing list
URL: http://www.incidents.org/archives/intrusions/msg14683.html
[3]. SecurityFocus Research Top Attacks for the 1st Quarter 2002
URL: http://www.securityfocus.com/corporate/research/top10attacks_q1_2002.shtml
[4]. Common Internet File System Protocol (CIFS/1.0)
URL: http://ftp.cerias.purdue.edu/pub/lists/best-of-security/115
[5].CIFS: Common Insecurities Fail Scrutiny
URL: http://web.textfiles.com/computers/cifs.txt
[6]. Samba Documentation - negprot.c
URL: http://samba.org/doxygen/samba/appl-head/negprot_8c-source.html
[7]: NETBIOS
URL: http://www.rware.demon.co.uk/netbios.htm
[8]: RFC 1002 PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: DETAILED SPECIFICATIONS
URL: http://www.ietf.org/rfc/rfc1002.txt
[9]. ISS Security Center Alert - Microsoft SQL Spida Worm Propagation
URL: http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise118
[10]. Symantec Security Response - Digispid.B.Worm
http://securityresponse.symantec.com/avcenter/venc/data/digispid.b.worm.html
[11] McAfee Virus information library - JS/SQLSpida.a.worm
URL: http://vil.nai.com/vil/content/v_99500.htm
[12]. Microsoft Knowledge Base Article - Q172983
Explanation of the Three-Way Handshake via TCP/IP
URL: http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q172983&
[13]. CERT knowledge database Vulnerability Note VU#635463 -Microsoft SQL Server and Microsoft Data Engine (MSDE) ship with a null default password
URL: http://www.kb.cert.org/vuls/id/635463
[14]. MSDN SQL server properties (Security Tab)
URL: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/entrmgr/em_6z21.asp
[15]. CERT® Incident Note IN-2002-04
Exploitation of Vulnerabilities in Microsoft SQL Server
URL: http://www.cert.org/incident_notes/IN-2002-04.html
[16]. System Expert – Hardening Windows 2000
URL: http://www.systemexperts.com/tutors/HardenW2K101.pdf
[17]. Microsoft SQL Server Technical Resources – administration -security
URL: http://www.microsoft.com/sql/techinfo/administration/2000/security.asp
[18]. SQL server 2000 security
URL: http://www.microsoft.com/sql/evaluation/features/security.asp
[19]. Requirement for IP version 4 Routers
URL: http://www.ietf.org/rfc/rfc1812
[20]. CERT Advisory CA-2000-22 Input Validation Problems in LPRng

URL: http://www.cert.org/advisories/CA-2000-22.html
[21]. CERT Advisory CA-2001-30 Multiple Vulnerabilities in lpd
URL: http://www.cert.org/advisories/CA-2001-30.html
[22]. CERT Advisory CA-2001-15 Buffer Overflow In Sun Solaris in.lpd Print Daemon
URL: http://www.cert.org/advisories/CA-2001-15.html
[23]. RFC 791 INTERNET PROTOCOL
URL: http://www.ietf.org/rfc/rfc791.txt
[24]. Whitehat.com arachNIDS - The Intrusion Event Database – IDS203
URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids203&view=research
[25]. PacketStorm Security
URL: http://packetstormsecurity.nl/groups/mixter/Q-1.0.tgz
[26]. White paper: Fun with Packet
URL: http://www.eurocompton.net/stick/papers/Peopledos.pdf
[27]. Dshield.org IP info (august 16, 2002)
URL: http://www1.dshield.org/ipinfo.php?ip=255.255.255.255&Submit=Submit
[28]. Re: Deny IP spoof from 255.255.255.255
URL: http://archives.neohapsis.com/archives/incidents/2001-07/thread.html#17
[29]. More weird scans
URL: http://lists.jammed.com/incidents/2001/04/index.html#92
[30]. Backdoor Q access in intrusion mailing list:
URL: http://lists.jammed.com/incidents/2001/04/0153.html
[31]. Re: Strange log line...
URL: http://maclux-rz.uibk.ac.at/~maillists/security-asics/thrd228.shtml#01945
[32]. Matt's Script Archive - Formmail
URL: http://www.scriptarchive.com/formmail.html
[33]. SecurityFocus Research - Top Attacks for the 1st Quarter 2002
URL:http://www.securityfocus.com/corporate/research/top10attacks_q1_2002.shtml
[34]. FormMail Recipient CGI Variable Spamming Vulnerability
URL: http://online.securityfocus.com/bid/2469/discussion/.
[35]. Anonymous Mail Forwarding Vulnerabilities in FormMail 1.9
URL: http://www.monkeys.com/anti-spam/formmail-advisory.pdf
[36]. SecurityFocus –Vulnerability - FormMail HTTP_Referer Spoofing Vulnerability
URL: http://online.securityfocus.com/bid/3954/discussion/
[37]. Re: SPAM-Versuche auf meinen SMTP-Server
URL: http://groups.google.ca/groups?dq=&hl=en&lr=&ie=UTF-8&selm=3d304947.14152147%40news.cis.dfn.de
[38]. Re: <victim>server formmail.pl exploit in the wild
http://lists.insecure.org/incidents/2002/Apr/0060.html

< End of Reference >

Assignment #3 Analyze this

Executive Summary

The following report analyzes the 5 days log from the intrusion detection system of the university. The purpose is to provide a clear view of critical intrusion attempts against the university network. Statistics of intrusion attempts will be exhibited and compromised system, critical intrusion attempts and network configuration problem will be identified; recommendation will be given out.

During the period of August 1 to August 5, 2002, 13 critical events occurred and 5 hosts are identified being compromised. The university administrator should take the hosts offline and further investigate them.

To sum up the analysis, the following table lists hosts that show obvious sign of compromised, system rebuild or recovery for these hosts is needed immediately.

IP address	Vulnerable service	IP address	Vulnerable service
my.net.82.130	File share TCP port 139	my.net.84.234	HTTP service port 80
my.net.100.208	HTTP service port 80		
my.net.111.116	SNMP service port 161	my.net.111.159	SNMP service port 161

The following table lists hosts that should be fully investigated for possible system compromise:

IP address	IP address	IP address	IP address
my.net.83.146	my.net.117.25	my.net.190.26	my.net.168.27
my.net.91.122	my.net.111.230	my.net.111.231	my.net.109.105
my.net.111.219	my.net.70.69		

List of the snort log files:

Data was analyzed for August 1 through August 5, 2002.

alert.020801.gz	scans.020801.gz	oos_Aug.1.2002.gz
alert.020802.gz	scans.020802.gz	oos_Aug.2.2002.gz
alert.020803.gz	scans.020803.gz	oos_Aug.3.2002.gz
alert.020804.gz	scans.020804.gz	oos_Aug.4.2002.gz
alert.020805.gz	scans.020805.gz	oos_Aug.5.2002.gz
Total: 2503012 entries	Total: 4110184 entries	Total: 1637 entries

Microsoft SQL Desktop Edition and UNIX shell commands were utilized to generate analysis reports, which were then manually reviewed. A full description of the analysis process is provided at the end of this report.

Network and Host profile

- Network Profile:

The university uses a class B network of my.net.0.0. Hosts that provide public service are not put centrally in several network, instead, they are distributed in every subnet of the network.

In addition, the border router forward IP packets with reserved private IP addresses, which cause lots of "SRC and Dest outside network" alerts.

Snort engine is expected to place at the border of the network. If there are firewalls protecting the entire internal network, the snort engine should be placed between the router and the firewall. This placement allow all the egress and ingress traffic to be monitored.

- Host Profile

After a thorough analysis of the alert file and the scans file, the following 4 lists of are generated. These list help a lot with identifying false positive.

1. List of hosts that provide web public services. This list is generated by looking for hosts that get attacking alert on HTTP service. The whole list has 512 entries, but in the following table only the ones that get more than 100 hits are listed.

HostIP	Hits on Port 80	HostIP	Hits on Port 80
my.net.111.140	370	my.net.134.10	216
my.net.5.96	322	my.net.137.7	213
my.net.105.10	303	my.net.157.11	206
my.net.91.154	303	my.net.113.208	205
my.net.70.58	295	my.net.130.34	205
my.net.91.8	294	my.net.130.14	203
my.net.106.228	271	my.net.130.86	203
my.net.104.177	261	my.net.83.247	203
my.net.106.191	260	my.net.134.11	202
my.net.130.122	259	my.net.130.92	199
my.net.5.14	258	my.net.130.17	194
my.net.5.92	252	my.net.136.2	189
my.net.70.103	250	my.net.150.228	186
my.net.130.27	249	my.net.150.101	176
my.net.105.204	248	my.net.114.116	173
my.net.86.19	247	my.net.114.42	173
my.net.104.104	240	my.net.179.77	168
my.net.111.21	234	my.net.111.221	162
my.net.70.164	232	my.net.21.27	145
my.net.106.222	229	my.net.117.132	140
my.net.130.91	229	my.net.130.123	137

my.net.157.49	227	my.net.114.98	132
my.net.130.166	221	my.net.110.76	131
my.net.5.95	219	my.net.27.3	129
my.net.157.239	217	my.net.21.43	105
my.net.130.40	216		

2. List of hosts that provide non-HTTP services.

Host IP	Service	Host IP	Service
my.net.137.7	DNS port 53	my.net.83.146	Ident port 113
my.net.137.7	Kerberos Port 88 open to 204.183.84.240	my.net.6.40	SMTP port 25
my.net.137.7	LDAP Port 389 open to 204.183.84.240	my.net.139.230	SMTP port 25
my.net.137.7	MS Netbios name service Port 137	my.net.100.217	SMTP port 25
my.net.100.214	MS file share port 139	my.net.145.9	SMTP port 25
my.net.116.53	MS file share port139	my.net.70.50 (help desk)	FTP port 21
my.net.70.28	MS file share port139	my.net.182.98	FTP port 21
my.net.87.15	MS file share port139	my.net.70.49 (help desk)	FTP port 21
my.net.115.86	MS file share port 445	my.net.163.97	SSH port 22
my.net.111.195	MS file share port 445	my.net.87.50	Service unknown. Port 888,999
my.net.70.133	AFS Server port 7002, 7003, 7004,7021	my.net.140.179	AFS port 7001
my.net.70.34	AFS port 7001	my.net.152.158	AFS port 7001
my.net.111.116	SNMP port 161	my.net.111.159	SNMP port 161
my.net.130.200	SNMP port 161	my.net.154.26	SNMP port 161
my.net.87.44	Unknown service Port 1299,1243,1244,1245,1246,2143,2144		

3. List of hosts with file sharing and gaming program installed

Host IP	Program and Port	Host IP	Program and port
my.net.81.27	MS Gaming Zone port 28800	my.net.70.180	Bluster Music port 41170
my.net.70.200	Bluster Music port 41170	my.net.165.24	WInMX P2P file sharing system port 6257
my.net.117.137	KaZaa port 1214	my.net.83.146	WInMX P2P file sharing system port 6257
my.net.70.210	KaZaa port 1214	my.net.81.27	MSN gaming zone
my.net.198.204	KaZaa port 1214	my.net.83.150	WInMX P2P file sharing system port 6257
my.net.116.69	KaZaa port 1214	my.net.84.130	WInMX P2P file sharing system port 6257

my.net.100.220	KaZaa port 1214	my.net.111.145	Edonkey port 4662
my.net.88.162	KaZaa port 1214	my.net.104.204	KaZaa port 1214
my.net.153.191	KaZaa port 1214	my.net.108.42	KaZaa port 1214
my.net.81.27	MS Gaming Zone port 28800		

4. List of hosts with IRC XDCC installed

Host IP	Count of Alert	Host IP	Count of Alert
my.net.82.130	1170	my.net.178.172	101
my.net.178.199	372	my.net.80.149	82
my.net.82.87	160	my.net.163.78	37
my.net.163.93	131	my.net.162.226	1

Identified false positives and their reason.

We all knew that IDS data analysis just like looking for gold in sands. The real attack will not become especially outstanding before the false positive are filtered.

There are multiple false positives were identified as listed below. Since data are analyzed after the false positives were identified, there is no false positive in the data analyzed in the next sections.

1. The requests of webcasting from Yahoo cause more than 32,000 alert of “UDP source and destination outside the network”. The alerts are related to source net 63.250.213.0 and destination net 233, 229, 239 networks. The source network belongs to Yahoo broadcast Inc. The destination networks are IANA reserved multicast net.
2. Some internal users visited the web site of the Chinese Society of Theoretical and Applied Mechanics and Chinese Academy of Mathematics and System Sciences. The IP addresses of these web sites fall into the IP range of Watchlist 000222 NET-NCFC. The responses HTTP traffic caused the alert of “Watchlist 000222 NET-NCFC”. Other activities associated with this watchlist are legitimate HTTP and SMTP client incoming traffic. Adding all this together, there are approximately 1300 false positive related to this alert.
3. Some KaZaa file sharing hosts causes alert of “Watchlist 000220 IL-ISDNNET-990517” when communicating with other Kazaa client and web server in Israel.
   Some of the internal hosts share files to other universities including khuang.bus.usu.edu and harlem.ebiz.washington.edu. 1100 alerts of “EXPLOIT x86 NOOP” and 11 “SMB CD…” were generated. The alerts were triggered when the transferred files containing the contents match the snort signature.
4. KaZaa, Napster streaming video traffics as well caused 36 false alerts of Exploit x86 NOOP, Exploit x86 setuid, Exploit x86 setgid.
5. Some web surfing traffic also caused false alert of Exploit x86 NOOP,

Exploit x86 setuid, Exploit x86 setgid. There are 48 these kinds of alerts are triggered by return packets from port 80, 85% of the sources IP are live web servers.

6. Some SSH traffic again cause false alert of Exploit x86 setuid, Exploit x86 setgid. The internal host at my.net.163.97 appear to serve as SSH server and client. The other end of the traffic are from Jet Propulsion Laboratory and NASA Ames Research Center, both belong to US government. The IP addresses are 137.78.58.62 and 198.118.229.166 respectively. SSH traffic are encrypted traffic, since the snort engine can not decrypt the traffic, finding shell code in the encryption text is totally not related to having shell code in the decrypted traffic.
7. Response traffic from the internal e-mail server cause alert of "Port 55850 tcp - Possible myserver activity - ref. 010313-1". The e-mail client happened to use port 58850 as ephemeral port, which matches the signature of myserver activity.
8. There are 47 alerts of "EXPLOIT x86 stealth noop" related to 140.172.180.165 – noaa.gov. They are possibly false positive. The program that triggered this alert is unknown. It pays to be safe to investigate the internal host at my.net.158.70 for further clarification. The following table shows the first alert and the last alert of the traffic lasting 3 minutes.

08/02-19:18:48.461687	EXPLOIT x86 stealth noop	140.172.180.165:54884 -> my.net.158.70:2438
08/02-19:21:04.628281	EXPLOIT x86 stealth noop	140.172.180.165:54884 -> my.net.158.70:2438

9. E-mail traffic from vger.kernel.org cause false positive of "Queso fingerprint" because they set the previous reserved bit ECN in TCP packet header, which is legitimate now.
10. The DNS client my.net.154.27 always use port 32771 as ephemeral port when it sent out query to other DNS server, which cause the response traffic triggered the false alert of "Attempted Sun RPC high port access". Some web surfing response traffics cause this false positive as well.
11. File sharing program installed in host my.net.83.150 triggered false positive of "High port 65535 udp - possible Red Worm – traffic" when the other end use 65535 as ephemeral port.

The following registration information from www.arin.net has help to determine the false positive:

Search results for: 63.250.213.0

Yahoo! Broadcast Services, Inc. (NETBLK-NETBLK2-YAHOOBS)
   701 First Avenue

```
   Sunnyvale, California 94089
   US

   Netname: NETBLK2-YAHOOBS
   Netblock: 63.250.192.0 - 63.250.223.255
   Maintainer: YAHO

   Coordinator:
      Admin, Netblock  (NA258-ARIN)  netblockadmin@yahoo-inc.com
      1-408-349-7183

   Domain System inverse mapping provided by:

   NS1.YAHOO.COM                66.218.71.63
   NS2.YAHOO.COM                209.132.1.28
   NS3.YAHOO.COM                217.12.4.104
   NS4.YAHOO.COM                63.250.206.138
   NS5.YAHOO.COM                64.58.77.85

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 27-Mar-2002.
   Database last updated on  23-Aug-2002 16:56:03 EDT.
```

Search results for: 137.78.58.62

```
National Aeronautics and Space Administration (NET-JPL-NET2)
   4800 Oak Grove Drive
   Pasadena, CA 91109-8099
   US

   Netname: JPL-NET2
   Netblock: 137.78.0.0 - 137.78.255.255
   Maintainer: NASA

   Coordinator:
      Wieclawek, Joseph A.  (JAW16-ARIN)  Joseph.A.Wieclawek-Jr@jpl.nasa.gov
      (818) 354-2419

   Domain System inverse mapping provided by:

   NSE.JPL.NASA.GOV                      192.138.85.66
   NSE2.JPL.NASA.GOV                     192.138.85.69
   MX.NSI.NASA.GOV              128.102.18.31
   TRANTOR.UMD.EDU                       128.8.10.14

   Record last updated on 23-Aug-2002.
   Database last updated on  23-Aug-2002 16:56:03 EDT.

National Aeronautics and Space Administration (NETBLK-NETBLK-NSI198)
   NASA Ames Research Center
   MS 233-8
   Moffett Field, CA 95014
```

```
US

Netname: NETBLK-NSI
Netblock: 198.116.0.0 - 198.123.255.255
Maintainer: NASA

Coordinator:
   National Aeronautics and Space Administration  (ZN7-ARIN)  dns.support@nasa.gov
   256-544-5623

Domain System inverse mapping provided by:

NASANS1.NASA.GOV                         192.77.84.32
MX.NSI.NASA.GOV                128.102.18.31
NS.ARC.NASA.GOV                128.102.16.2

Record last updated on 26-Sep-2001.
Database last updated on  23-Aug-2002 16:56:03 EDT.
```

Search results for: 198.118.229.166

```
National Aeronautics and Space Administration (NETBLK-NETBLK-NSI198)
   NASA Ames Research Center
   MS 233-8
   Moffett Field, CA 95014
   US

   Netname: NETBLK-NSI
   Netblock: 198.116.0.0 - 198.123.255.255
   Maintainer: NASA

   Coordinator:
      National Aeronautics and Space Administration  (ZN7-ARIN)  dns.support@nasa.gov
      256-544-5623

   Domain System inverse mapping provided by:

   NASANS1.NASA.GOV                         192.77.84.32
   MX.NSI.NASA.GOV                128.102.18.31
   NS.ARC.NASA.GOV                128.102.16.2

   Record last updated on 26-Sep-2001.
   Database last updated on  23-Aug-2002 16:56:03 EDT.
```

Detects Prioritized by Severity

As outlined in the topic, the following detects are listed according to the priority of severity; thus the detects associated with compromised host are listed first, and then the attacks could possibly compromise host listed after, the detect of scans listed the last.

High Severity Alert # 1 EXPLOIT x86 NOOP and Samba Client access

There are 678 alerts of Samba client access from IP 64.81.195.164 destined to part of the my.net network on port 139. These alerts appear a reconnaissance to Microsoft Windows system's open shares or Samba server. my.net.82.130 appears to be scanned twice. The first scan happened within the random scanning process, the second scan happened after the whole scan was finished. The second scan could be an attempt of verifying about whether the target found was desired. After the verification, the attacker launched the exploit phase, which last 15 seconds. The followed 'CD..' command indicated the attack could have succeeded. After, the victim my.net.82.130 joined several IRC channel to "phone home" about the successful attack. The attacker then connected from 24.209.230.143 to the victim to upload the malicious code, further infect the victim and install backdoor etc.

The following table briefs the whole reconnaissance, exploitation, notification, and remote access hacking process. Only significant packets are shown here.

08/01-06:01:01.735115	Samba client access	64.81.195.164:35987 -> my.net.100.126:139
08/01-06:23:10.075026	Samba client access	64.81.195.164:39250 -> my.net.other.IPs:139
08/01-06:23:10.075026	Samba client access	64.81.195.164:39250 -> my.net.82.130:139
08/01-06:23:10.075026	Samba client access	64.81.195.164:39250 -> my.net.other.IPs:139
08/01-06:26:24.896144	Samba client access	64.81.195.164:39866 -> my.net.91.80:139
08/01-06:33:21.107442	Samba client access	64.81.195.164:40659 -> my.net.82.130:139
08/01-06:33:52.978640	EXPLOIT x86 NOOP	64.81.195.164:40662 -> my.net.82.130:139
Ongoing	EXPLOIT x86 NOOP	64.81.195.164:40662 -> my.net.82.130:139
08/01-06:34:07.859996	EXPLOIT x86 NOOP	64.81.195.164:40662 -> my.net.82.130:139
08/01-06:34:13.850195	SMB CD...	64.81.195.164:40662 -> my.net.82.130:139
08/01-06:34:17.151546	SMB CD...	64.81.195.164:40662 -> my.net.82.130:139
08/01-06:38:54.211855	IRC evil - running XDCC	my.net.82.130:1948 -> 146.20.20.20:6667
Ongoing for days and nights	IRC evil - running XDCC	my.net.82.130:1948 -> 146.20.20.20:6667
08/02-09:00:05.984209	IRC evil - running XDCC	my.net.82.130:2628 -> 216.152.65.144:6667
08/02-11:47:57.156227	IRC evil - running XDCC	my.net.82.130:2628 -> 216.152.65.144:6667
08/02-11:52:00.980988	IRC evil - running XDCC	my.net.82.130:2538 -> 63.98.19.242:6667
08/04-02:11:30.774117	IRC evil - running XDCC	my.net.82.130:2538 -> 63.98.19.242:6667

08/04-02:35:47.942449	IRC evil - running XDCC	my.net.82.130:3697 -> 213.162.129.10:6667
08/04-23:12:22.885980	IRC evil - running XDCC	my.net.82.130:3697 -> 213.162.129.10:6667
08/04-23:17:44.214422	TFTP - External TCP connection to internal tftp server	24.209.230.143:1544 -> my.net.82.130:69
08/04-23:17:44.214677	TFTP - External TCP connection to internal tftp server	my.net.82.130:69 -> 24.209.230.143:1544
08/04-23:17:44.677593	TFTP - External TCP connection to internal tftp server	24.209.230.143:1544 -> my.net.82.130:69
08/04-23:17:44.677886	TFTP - External TCP connection to internal tftp server	my.net.82.130:69 -> 24.209.230.143:1544
08/04-23:17:45.189181	TFTP - External TCP connection to internal tftp server	24.209.230.143:1544 -> my.net.82.130:69
08/04-23:17:45.189601	TFTP - External TCP connection to internal tftp server	my.net.82.130:69 -> 24.209.230.143:1544
08/04-23:18:48.277388	SMB Name Wildcard	24.209.230.143:137 -> my.net.82.130:137
08/04-23:21:29.834182	IRC evil - running XDCC	my.net.82.130:3697 -> 213.162.129.10:6667
08/05-10:04:26.777615	IRC evil - running XDCC	my.net.82.130:3697 -> 213.162.129.10:6667
08/05-10:14:18.950329	IRC evil - running XDCC	my.net.82.130:1027 -> 63.98.19.242:6667
08/05-17:32:51.835943	IRC evil - running XDCC	my.net.82.130:1027 -> 213.162.129.10:6667

After going through all the alert file, scans file and oos files in the 5 days period, no other suspicious traffic related to 24.209.230.143 and 64.81.195.164 are found. It looks like the victim was not installed with some kind of worms or the worm was not active immediately, otherwise active outgoing scan would have been seen.

Dshield.org provide correlation information as shown in following table indicating there is a past history of this IP attacking other machines on port 139.

IP Address:	64.81.195.164
HostName:	dsl081-195-164.nyc2.dsl.speakeasy.net

Dshield Profile: Country:
US

Contact E-mail:
abuse@speakeasy.net

Total Records against IP:
1074

Number of targets:
1052

Date Range:
2002-08-20 to 2002-08-20

Ports Attacked (up to 10):

Port	Attacks
139	130
445	896

Fightback: sent to abuse@speakeasy.net on 2002-06-15 20:02:34
no reply received

Whois:

```
Speakeasy Network (NETBLK-SPEAKEASY-3)
  2304 2nd Ave
  Seattle, WA 98121
  US

  Netname: SPEAKEASY-3
  Netblock: 64.81.0.0 - 64.81.255.255
  Maintainer: SPEK

  Coordinator:
    Stollar, Andreas  (AS3414-ARIN)  abuse@speakeasy.net
    +1-206-728-9770 (FAX) (206)728-1500

  Domain System inverse mapping provided by:

  NS1.SPEAKEASY.NET                    216.254.0.9
  NS2.SPEAKEASY.NET                    216.231.41.22

  ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

  Record last updated on 05-Jan-2001.
  Database last updated on  11-Aug-2002 19:59:36 EDT.

----------

Speakeasy Network DSL (NETBLK-SPEK-DSL-NYC-BR6-0)
  25 Broadway, 5th Floor
  New York, NY 10004
  US

  Netname: SPEK-DSL-NYC-BR6-0
  Netblock: 64.81.192.0 - 64.81.215.255

  Coordinator:
    Stollar, Andreas  (AS3414-ARIN)  abuse@speakeasy.net
    +1-206-728-9770 (FAX) (206)728-1500

  Record last updated on 01-Mar-2001.
  Database last updated on  11-Aug-2002 19:59:36 EDT.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.
```

Information about 24.209.230.143 provided by Dshield.org show this IP does not have bad history of attacking. But it is very obvious the people behind this IP have used host 64.81.195.164 as his zombie to hide its malicious activity.

IP Address: 24.209.230.143

HostName: dhcp024-209-230-143.cinci.rr.com

DShield Profile:	Country: US Contact E-mail: abuse@rr.com Total Records against IP: Number of targets: Date Range: to Ports Attacked (up to 10): **Port** **Attacks**
Fightback:	not sent

Whois:

```
ROADRUNNER (NETBLK-RR-CENTRAL-3BLK)
   13241 Woodland Park Road
   Herndon, VA 20171
   US

   Netname: RR-CENTRAL-3BLK
   Netblock: 24.208.0.0 - 24.211.31.255
   Maintainer: RRMA

   Coordinator:
      ServiceCo LLC  (ZS30-ARIN)  abuse@rr.com
      1-703-345-3416

   Domain System inverse mapping provided by:

   DNS1.RR.COM                         24.30.200.3
   DNS2.RR.COM                         24.30.201.3
   DNS3.RR.COM                         24.30.199.7
   DNS4.RR.COM                         65.24.0.172

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 14-Aug-2002.
   Database last updated on  21-Aug-2002 20:01:34 EDT.
```

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.

There are three vulnerabilities associated with port 139, two of them are related to Windows system and can cause denial or service. Another one is CVE-1999-0182 - Samba has a buffer overflow which allows a remote attacker to obtain root access by specifying a long password. This is very possibly the vulnerability the attacker use to exploit the box. [1]

Recommendation: The victim machine should be taken off the network immediately. If possible, investigate the vulnerabilities in the box and what are changed in the system. Otherwise, install anti-virus software to remove the Trojan, which could have been installed by the attacker after compromised. If possible, rebuild the whole system is more desire. Key point is installation of latest patch to the system and applications, harden the open service and do not open port 139 to the Internet anymore, or at least strictly control the incoming access by firewalling or use of TCPwrapper.

abuse@speakeasy.net and abuse@rr.com should be contacted to stop the attack which could happen to other victim in the future.

High Severity Alert # 2 IRC evil - running XDCC

The computers in the following list have been infected with the rogue IRC bot software. They may be unknowingly participating in a massive underground warez file-sharing network and consuming big amount of network bandwidth. In addition, the bot software might have installed Trojan horse software in these computers to allow a remote attacker to gain access to the system. The system could thus become a zombie of Distributed Denial of Service. [2] [3] This is a very serious security issue. XDCC has spread in many US universities. [4]

From the 5 days of alert, scans and oos data, only the host my.net.82.130 run the XDCC client after compromised. Other hosts do not have the same story as it. They might be installed by the users or infected without knowledge of the user.

Source IP	# of Alerts	Source IP	# of Alerts
my.net.82.130	1170	my.net.178.172	101
my.net.178.199	372	my.net.80.149	82
my.net.82.87	160	my.net.163.78	37
my.net.163.93	131	my.net.162.226	1

Recommendation: An analysis document [5] from Christopher E. Cramer in duke.edu provides the details of malicious file installed on the infected computer and can be clean up accordingly. Anti-virus software with the latest signature can help to clean up the system. [6]

High Severity Alert # 3 NIMDA - Attempt to execute cmd from campus host
High Severity Alert # 3 NIMDA - Attempt to execute root from campus host

HTTP server my.net.100.208 has been infected by Nimda worm [7] and become a launch point of attacking other HTTP servers. Starting from August 5, it sent out one million malicious HTTP execute cmd request to thousands of different hosts on Internet. No attack resulted to the infection of worm could be found from the log of this 5 days period, The machine could have been infected before August 1, 2002.

This HTTP server is an IIS 4.0/5.0 unpatched server. Nimda worm infected it by exploiting the directory traversal vulnerabilities. [8]

Recommendation: CERT provided detail information of the worm and the solution [7]. This host should be isolated from the rest of the network and a system rebuild is desired. Otherwise, system could be recovered according the instruction from CERT [9]. Do not put IIS service if web service is not a requirement to this machine. Otherwise, install the latest patch from Microsoft and anti-virus software to prevent from infecting by the Nimda worm again.

<u>High Severity Alert # 4 IDS552/web-iis IIS ISAPI Overflow ida INTERNAL nosize</u>

Host my.net.84.234 was infected by Code Red Worm [10] and sent out 482387 malicious HTTP requests to the Internet to attack other machines starting from August 4, 2002.

This system is a Windows system without applying any latest patch. The code Red worm can infect Windows NT, Windows 2000, Windows XP system if they are left unpatched. The worm infected the machine by exploit the Buffer Overflow vulnerability in IIS Indexing Service DLL. [11] CVE number of the vulnerability is CVE-2001-0500 [12]

Recommendation: CERT provided detail information of the worm and the solution [10]. This host should be isolated from the rest of the network and a system rebuild is desired. Otherwise, system could be recovered according the instruction from CERT [9]. Install the latest patch from Microsoft and anti-virus software to prevent from infecting by the Code Red worm again.

<u>High severity Alert #5 EXPLOIT x86 setuid 0, EXPLOIT x86 setgid 0, EXPLOIT x86 NOOP</u>
<u>Medium Severity Alert: Queso Finger Print</u>

my.net.83.146 is a system with WinMX P2P and Napster file sharing program installed and has lots of file sharing traffics. It appears to be unusual high port opened at port 54879. There is no well-known service or Backdoor listen on this port indicating from google search engine [13]. On the other hand, ident service port 113 seems to be open in this machine as well. Snort detects two major attacks to this machine at the aforementioned two ports.
The alerts warn that one of the shell codes of setuid0, setgid0 and NOOP were found in the packets payload. Since the signatures only concern binary "|b0b5 cd80|" or

"|b017 cd80|" or a series of |90| or |61| in the payload, sometimes it could be false positive when the payload is a binary file or image being transferred. But it is unusual to find this towards such a high port, especially when not only one attacker - 144.118.192.42, but there is still another attacker 130.74.32.130 know this port and trigger the same alert as well.

08/03-04:01:52.429864	EXPLOIT x86 setuid 0	144.118.192.42:1417 -> my.net.83.146:54879
08/03-04:35:43.122524	EXPLOIT x86 setuid 0	144.118.192.42:1455 -> my.net.83.146:54879
08/03-04:37:38.044915	EXPLOIT x86 setuid 0	144.118.192.42:1426 -> my.net.83.146:54879
08/03-04:40:35.971277	EXPLOIT x86 setuid 0	144.118.192.42:1441 -> my.net.83.146:54879
08/03-04:51:06.651812	EXPLOIT x86 setuid 0	144.118.192.42:1434 -> my.net.83.146:54879
08/03-05:11:42.120241	EXPLOIT x86 setuid 0	144.118.192.42:1488 -> my.net.83.146:54879
08/03-05:20:54.220336	EXPLOIT x86 setuid 0	144.118.192.42:1424 -> my.net.83.146:54879
08/03-05:27:19.054893	EXPLOIT x86 setgid 0	144.118.192.42:1476 -> my.net.83.146:54879
08/03-05:46:07.774662	EXPLOIT x86 NOOP	144.118.192.42:1437 -> my.net.83.146:54879
08/03-06:07:37.659411	EXPLOIT x86 setuid 0	144.118.192.42:1434 -> my.net.83.146:54879
08/03-06:36:28.835400	EXPLOIT x86 setuid 0	144.118.192.42:1413 -> my.net.83.146:54879
08/03-06:55:39.642612	EXPLOIT x86 setuid 0	144.118.192.42:1404 -> my.net.83.146:54879
08/03-07:29:10.346363	EXPLOIT x86 setgid 0	144.118.192.42:1438 -> my.net.83.146:54879
08/05-14:03:06.123527	EXPLOIT x86 setgid 0	130.74.32.130:4345 -> my.net.83.146:54879

No other activity against other machine in the university's network was found from the above two IP addresses in the 5 days period. No other unusual activity was found with my.net.83.146 as well.

The alert "Queso fingerprint" warns that incoming packets with old reserved and unused bits and high TTL are found. This is the character of OS fingerprint tool Queso. In our case, this tool was used against port 113 for ident service, which is common to be used identify operating systems. Usually after the OS fingerprint, attacker can figure out the exploitable vulnerability with the specific OS and come back for hacking. So far, No other activity was detected from the source IP in the period.

08/04-21:10:00.629339	Queso fingerprint	213.48.150.1:51596 -> my.net.83.146:113
08/04-21:10:01.340804	Queso fingerprint	213.48.150.1:51600 -> my.net.83.146:113
Ongoing	Queso fingerprint	77 packets with different source port
08/05-15:50:18.115391	Queso fingerprint	213.48.150.1:32866 -> my.net.83.146:113
08/05-15:50:18.115391	Queso fingerprint	213.48.150.1:32866 -> my.net.83.146:113

After query the IP info from Dshield.org, no attacking history was found from the above 3 source IP addresses. [14] [15] [16]

Correlation with other GCIA: Previous analysts have reviewed data from the site, however, none of them find the attacks on port 54879.

Recommendation: Identify whether the port 54879 and port 113 are open in the machine. If so, the machine should be checked for the sign of compromised,

High Severity Alert # 6
EXPLOIT NTPDX buffer overflow
High port 65535 udp - possible Red Worm – traffic
Back Orifice

Snort detected two connections to Back Orifice port, two NTPDX buffer overflow and 6 possible Red worm traffic. After consult the scans file, I found that the NTPDX attacks are dangerous, the other two alerts are generated by a random port scans.

Snort triggered an alert for "EXPLOIT NTPDX buffer overflow" because it observed an NTP packet that had a UDP payload greater than 128 bytes. Most NTP packets have a datagram size much less than 128 bytes. [17]

Older NTP daemons have a vulnerability that allows remote root access, which could be the target of this attacker, see the CVE database entry CVE-2001-0414 [18] and securityfocus vulnerability database. [19]

08/05-16:36:39.582295	Back Orifice	63.240.142.227:18672 -> my.net.117.25:31337
08/05-16:36:39.707788	Back Orifice	63.240.142.227:18672 -> my.net.117.25:31337
08/05-16:47:27.164335	EXPLOIT NTPDX buffer overflow	63.240.142.227:4239 -> my.net.117.25:123
08/05-16:47:27.929508	EXPLOIT NTPDX buffer overflow	63.240.142.227:4239 -> my.net.117.25:123
08/05-16:48:29.150115	High port 65535 udp - possible Red Worm – traffic	63.240.142.227:65535 -> my.net.117.25:65535
08/05-16:50:44.325979	High port 65535 udp - possible Red Worm – traffic	63.240.142.227:65535 -> my.net.117.25:65446
08/05-17:03:46.053683	High port 65535 udp - possible Red Worm – traffic	63.240.142.227:51402 -> my.net.117.25:65535
08/05-17:03:54.319184	High port 65535 udp - possible Red Worm – traffic	63.240.142.227:65535 -> my.net.117.25:5591
08/05-16:53:23.158383	High port 65535 udp - possible Red Worm – traffic	63.240.142.227:65535 -> my.net.117.25:65534
08/05-16:53:48.612595	High port 65535 udp - possible Red Worm – traffic	63.240.142.227:65535 -> my.net.117.25:49155

The scans file indicated that the attacker from IP 63.240.142.227 has conducted a UDP scan to the target my.net.117.25. Random UDP ports were scanned from

```
scans:08/05-16:45:56 63.240.142.227:0 -> my.net.117.25:0 UDP
scans:08/05-16:45:54 63.240.142.227:45308 -> my.net.117.25:23675 UDP
scans:08/05-16:45:54 63.240.142.227:16212 -> my.net.117.25:6993 UDP
…
scans:08/05-16:47:28 63.240.142.227:0 -> my.net.117.25:0 UDP
scans:08/05-16:47:27 63.240.142.227:4239 -> my.net.117.25:123 UDP
```

```
scans:08/05-16:47:28 63.240.142.227:13621 -> my.net.117.25:1410 UDP
…
scans:08/05-16:48:29 63.240.142.227:65535 -> my.net.117.25:65535 UDP
scans:08/05-16:48:30 63.240.142.227:0 -> my.net.117.25:0 UDP
scans:08/05-16:48:30 63.240.142.227:39974 -> my.net.117.25:1610 UDP
…
scans:08/05-16:53:23 63.240.142.227:65535 -> my.net.117.25:65534 UDP
scans:08/05-16:53:48 63.240.142.227:65535 -> my.net.117.25:49155 UDP
scans:08/05-16:54:48 63.240.142.227:0 -> my.net.117.25:0 UDP
scans:08/05-16:54:47 63.240.142.227:4065 -> my.net.117.25:139 UDP
scans:08/05-16:54:48 63.240.142.227:42 -> my.net.117.25:21226 UDP
scans:08/05-16:54:48 63.240.142.227:16948 -> my.net.117.25:23814 UDP
scans:08/05-16:54:48 63.240.142.227:13621 -> my.net.117.25:1410 UDP
```

The UDP port scan to port 123 happened at Aug 5 16:47:27, the NTPDX exploit happened at 08/05-16:47:27.164335. The exploit could have happened right after the scan found the live UDP port.

All of the UDP scans to or from port 65535 triggered snort alert of "possible red worm traffic" 16:48:29. But after the UDP scans stopped at 16:54:48, snort captured two "possible red worm" packets. These packets are not part of the scans. They could be dangerous exploit after live port 65534 and 49155 were identified by the scans.

Correlation with other GCIA analyst: I found a match with the NTPDX overflow pattern showed up in Kyle Haugsness's practical assignment [17]. A NTPDX exploit happened around the same time of random UDP scans. But the UDP scans in his paper does not indicate scan to or from port 65535, thus the red worm alert was not triggered.

High Severity Alert # 7 EXPLOIT x86 NOOP to port 2011

Totally 60 exploits were detected towards port 2011. 2011/TCP port is assigned to Raid-cc Service, 2011/UDP is assigned to servserv. No known vulnerabilities were found with this port so far [20]. As well, it could be possible that some kind of game server setup at port 2011[21]

```
08/05-12:46:56.908226  [**] SMB Name Wildcard [**] 66.57.117.222:137 -> my.net.190.52:137
08/05-12:47:59.040663  [**] SMB Name Wildcard [**] 66.57.117.222:137 -> my.net.190.100:137
08/05-12:55:48.171814  [**] EXPLOIT x86 NOOP [**] 66.57.117.222:2254 -> my.net.190.26:2011
08/05-12:55:48.483521  [**] EXPLOIT x86 NOOP [**] 66.57.117.222:2254 -> my.net.190.26:2011
08/05-12:55:48.543862  [**] EXPLOIT x86 NOOP [**] 66.57.117.222:2254 -> my.net.190.26:2011
08/05-12:55:48.779130  [**] EXPLOIT x86 NOOP [**] 66.57.117.222:2254 -> my.net.190.26:2011
…
08/05-12:56:06.873494  [**] EXPLOIT x86 NOOP [**] 66.57.117.222:2308 -> my.net.190.26:2011
08/05-12:56:07.133393  [**] EXPLOIT x86 NOOP [**] 66.57.117.222:2308 -> my.net.190.26:2011
08/05-12:56:07.607512  [**] EXPLOIT x86 NOOP [**] 66.57.117.222:2308 -> my.net.190.26:2011
```

No other alert was found with the internal host. From 66.57.117.222, there are only a few SMB Name wildcard alert were found against other IP at the my.net.190 subnet. No records could be found from scans file.

The questions here are whether port 2011 is open in the internal host, and if it is the case, what is the service bound to it.

The university's administrator is recommended to investigate the machine and answer the above question. If the port is open, check whether the machine was being attacked and compromised. It could be even better if the investigation result could be posted on the Internet so that other people know what is going on when they see port 2011 in their log. So far, there is no any detail related information about it available from the google search engine. [13]

<u>High Severity Alert # 8 EXPLOIT x86 NOOP to high port</u>

There are 7 alerts of attacks against high port 49195 in host my.net.168.27; 20 alerts of attacks against high port 32934 – 32960 in host my.net.91.122. IP address 140.221.9.138 belongs to Argonne National Laboratory; 208.209.50.17 belongs to UUNET. [22]

```
08/04-13:53:21.218352  [**] EXPLOIT x86 NOOP [**] 208.209.50.17:13825 -> my.net.168.27:49195
08/04-13:53:21.219766  [**] EXPLOIT x86 NOOP [**] 208.209.50.17:13825 -> my.net.168.27:49195
08/04-13:53:31.900970  [**] EXPLOIT x86 NOOP [**] 208.209.50.17:13825 -> my.net.168.27:49195
08/04-13:53:31.901649  [**] EXPLOIT x86 NOOP [**] 208.209.50.17:13825 -> my.net.168.27:49195
08/04-14:23:06.868283  [**] EXPLOIT x86 NOOP [**] 208.209.50.17:13825 -> my.net.168.27:49195
08/04-14:23:06.869170  [**] EXPLOIT x86 NOOP [**] 208.209.50.17:13825 -> my.net.168.27:49195
08/04-14:23:06.869419  [**] EXPLOIT x86 NOOP [**] 208.209.50.17:13825 -> my.net.168.27:49195

08/05-17:16:18.921024  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:9921 -> my.net.91.122:32934
08/05-17:16:18.921566  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:9921 -> my.net.91.122:32934
08/05-17:30:57.898455  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:4471 -> my.net.91.122:32940
08/05-18:29:01.328403  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:29:01.329496  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:29:01.329911  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:29:05.800947  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:51:17.817417  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:24189 -> my.net.91.122:32960
08/05-18:52:13.028911  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:24189 -> my.net.91.122:32960
08/05-18:52:13.029335  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:24189 -> my.net.91.122:32960
08/05-17:16:18.921024  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:9921 -> my.net.91.122:32934
08/05-17:16:18.921566  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:9921 -> my.net.91.122:32934
08/05-17:30:57.898455  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:4471 -> my.net.91.122:32940
08/05-18:29:01.328403  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:29:01.329496  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:29:01.329911  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:29:05.800947  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:48642 -> my.net.91.122:32950
08/05-18:51:17.817417  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:24189 -> my.net.91.122:32960
08/05-18:52:13.028911  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:24189 -> my.net.91.122:32960
08/05-18:52:13.029335  [**] EXPLOIT x86 NOOP [**] 140.221.9.138:24189 -> my.net.91.122:32960
```

After a search from google[13], I did not find any known service bound to these high ports.

So far, after a thorough search of the alerts, scans and oos file, no other unusual activities against the internal network devices were found. No other malicious activities were found from the aforementioned source IP and subnet.

No attacking history was found from Dshield.org regarding the two source IP addresses. [23] [24]

Correlation with other GCIA: Previous analysts have reviewed data from this site, however none of them observed similar activity.

Recommendation: The administrator is encouraged to check whether there are services listening on those ports and whether the machines are compromised by the attacks listed above.

<u>High Severity Alert # 9 TFTP – External UDP connection to internal tftp server</u>

The 4 hosts in the following table sent massive outbound packets from port 69 to 192.168.0.216. No stimulus was found from 192.168.0.216. Although 192.168.0.0 is reserved private subnet, the university does not use it internally according to the title of the rule.

Source	Occurrences	Destip
my.net.111.230	6090	192.168.0.216
my.net.111.231	6059	192.168.0.216
my.net.109.105	6054	192.168.0.216
my.net.111.219	6007	192.168.0.216

The following are sample of those packets, the destination port are ephemeral port, the source port is always the same: 69.

08/01-00:07:11.963569	TFTP - External UDP connection to internal tftp server	my.net.111.231:69 -> 192.168.0.216:9695
08/01-00:07:20.015017	TFTP - External UDP connection to internal tftp server	my.net.111.231:69 -> 192.168.0.216:9695
08/01-00:07:28.062229	TFTP - External UDP connection to internal tftp server	my.net.111.231:69 -> 192.168.0.216:3596
08/01-00:07:30.497374	TFTP - External UDP connection to internal tftp server	my.net.111.231:69 -> 192.168.0.216:9281

These traffics could be seen from the log file of each day. They happened all day long. This traffic could be transfer of files requested by the attacker. It could lead to sensitive information leak if the internal hosts are not desired to be accessed by external users.

The reason why these packets were sent to a private IP is unknown yet. It is pretty common that tftp service is open after host is compromised by worm so that next victim can download malicious code from the infected host. TFTP service open is a very dangerous sign to a server, which is not deemed to provide TFTP service. According to CVE-1999-0183 [25], Linux implementations of TFTP would allow access to files outside the restricted directory. This vulnerability brings more convenience to the attacker to transfer their file without limitation and allow more serious information leak.

Recommendation: A snort rule should be implemented to monitor the incoming traffic to port 69 on the above internal hosts. Snort tcpdump log files should be consulted to investigate the payload of the packets. The hosts should be taken offline to check whether TFTP services are running and whether hosts had been compromised.

The packets with reserved private address are able to reach the border of the network illustrate that there are configuration problem in the router. The border router should not forward any packet with private reserved address according to "Best current practice" defined in RFC 1918. [26]

High Severity Alert # 10 SNMP public access

It seems like several external hosts including my.net1.109.84 and some host from my.net2.250 network have gained access to 4 internal network devices, which IP are my.net.111.159,my.net.111.116, my.net.130.200 and my.net.154.26. The alerts of "SNMP public access" were trigger every day. After a whois lookup to arin.net,[22] I found that the my.net1 subnet and my.net2.250 subnet are belonged to the same University.

To identify the nature of the access, the university administrator should verify whether there are some kinds of network management relationship between the two universities. Even there are some kinds of relationship exist, the "public" SNMP community string should never be used. This string is a default "password" for every insecure SNMP service for gaining read access. It is pretty common that attackers try this string to gain access to those insecure SNMP service.

Before knowing any "relationship", I will treat the people who accessed the devices with "public" string as attacker.

The attack leveraged the vulnerability of SNMP – each SNMP service has a default read access community string: public. After the exploit, the attacker might already have walked through all the SNMP MIB and gain the knowledge of all the network topology of the university. Even worst, if the attacker leveraged Multiple Vulnerabilities in Many Implementations of the SNMP [27], the attacker can gain unauthorized privileged access to the vulnerable device, cause denial-of-service or unstable behavior to the network device or even the network.

Each attacker came back after the first attack. Each access time is long enough to gain the information they want. They can use tools such as WS Ping Pro Pack[28] and snmpwalk[29] to get all the MIB stored inside the device and learn everything about the devices themselves and the whole network topology[30]. Further more, they can use PROTOS c06-snmpv1 test suite [31] to gain privilege access and/or cause more damage to the device and/or the network.[27]

The following table shows part of the alerts.

08/01-09:17:22.378549	SNMP public access	my.net1.109.84:1026 -> my.net.111.116:161
08/01-09:26:59.747217	SNMP public access	my.net1.109.84:1026 -> my.net.111.116:161
08/01-09:27:23.093066	SNMP public access	my.net1.109.84:1026 -> my.net.111.159:161
08/01-09:36:39.724477	SNMP public access	my.net1.109.84:1026 -> my.net.111.116:161
08/01-09:37:23.196523	SNMP public access	my.net1.109.84:1026 -> my.net.111.159:161
08/01-11:40:18.682826	SNMP public access	my.net1.109.84:1026 -> my.net.111.116:161
08/01-11:40:24.786362	SNMP public access	my.net1.109.84:1026 -> my.net.111.116:161
08/01-12:42:45.421165	SNMP public access	my.net1.109.84:1026 -> my.net.111.159:161
08/01-12:42:45.564575	SNMP public access	my.net1.109.84:1026 -> my.net.111.116:161
08/01-12:51:18.055797	SNMP public access	my.net1.109.84:1026 -> my.net.111.116:161
08/04-11:04:43.144859	SNMP public access	my.net2.250.150:1026 -> my.net.111.116:161
08/04-11:04:43.993496	SNMP public access	my.net2.250.150:1026 -> my.net.111.159:161
08/04-11:38:44.405227	SNMP public access	my.net2.250.150:1026 -> my.net.111.159:161
08/04-11:38:47.959594	SNMP public access	my.net2.250.150:1026 -> my.net.111.116:161
08/04-20:29:32.632123	SNMP public access	my.net2.250.208:1026 -> my.net.111.116:161
08/04-20:29:32.983885	SNMP public access	my.net2.250.208:1026 -> my.net.111.159:161
08/04-21:06:33.086478	SNMP public access	my.net2.250.208:1026 -> my.net.111.116:161
08/04-21:06:33.422347	SNMP public access	my.net2.250.208:1026 -> my.net.111.116:161

So far, after a thorough search of the alerts, scans and oos file, no other unusual activities against the internal network devices were found. No other malicious activities were found from the aforementioned source IP and subnet.

Recommendation: Set complex read and write access community strings of the 4 network devices immediately and keep them secret. Set strict access control to the device. Add snort rule to monitor next attempts from the IP my.net1.109.84 and the my.net2.250 subnet.

High Severity Alert 11 – Beetle.ucs

There were approximately 166 alerts for "Beetle.ucs" from the Snort alert data over 5 days. This could be a customized signature defined by the university's administrator to monitor the activity from and to my.net.70.69. According to another GCIA analyst Edward Peck's paper [32], Beetle.ucs is a host that houses a CD-R. This alert indicates that users are copying information form the Internet and saving it to a CD-R. However, There are other activities going on as well. I saw some incoming connection to port 139, 445, 1433, 80, 21,23 and the host response to the request. After a close look, All the responses, except the ones posted in the following table, could be RESET packet telling the requester the ports are not open. But there are two strange outbound requests from my.net.70.69. One is to 203.146.102.21 port 13000 and another one is too 198.109.0.73 port 43981. The port 13000 is used by Trojan Senna Spy. Port 43981 is Netware for IP; it is the ninth most probed port in August, 2002. These could be signs of compromised.

08/01-01:11:03.622809	beetle.ucs	65.211.134.134:2189 -> my.net.70.69:139
08/01-01:11:03.622936	beetle.ucs	my.net.70.69:139 -> 65.211.134.134:2189
08/01-01:11:04.676576	beetle.ucs	my.net.70.69:139 -> 65.211.134.134:2189
08/01-01:11:05.190374	beetle.ucs	my.net.70.69:139 -> 65.211.134.134:2189
08/03-02:38:50.325377	beetle.ucs	my.net.70.69:6112 -> 198.109.0.73:43981
08/05-19:56:38.655611	beetle.ucs	my.net.70.69:13000 -> 203.146.102.211:13000

Other events related to this host have been searched, no other more significant events are found.

Administrator should close investigate the host and

1. Consult the snort tcpdump log file to check whether the outgoing packets are Resets or response with data payload.
2. Check whether the host was infected by senna spy or other virus. The senna spy Trojan is only effective to windows machine. The removal intrution of the Trojan is available at http://securityresponse.symantec.com/avcenter/venc/data/pf/sennaspy.generator.html and http://www.dark-e.com/archive/trojans/sennaspy/2000/index.shtml

High Severity Alert # 12 UDP source and destination outside the network

There are approximately 51396 UDP traffics from 3.0.0.99 to 10.0.0.1. According to the snort rule, these two addresses are not part of the university's network. Both of the source and destination ports are 137 – Microsoft Netbios Service.
The traffics are seen every day during the 5 days period and they are unidirectional, all from 3.0.0.99 to 10.0.0.1. If one of the addresses felled into the university's IP range, the traffic might have trigger the "SMB name wildcard" alert.

```
08/05-23:50:17.354819  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
08/05-23:50:20.359196  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
08/05-23:50:23.363616  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
08/05-23:50:24.865813  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
08/05-23:50:27.870195  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
08/05-23:50:29.372338  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
08/05-23:50:30.874527  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
```

The destination IP address 10.0.0.1 is IANA reserved address for private use. It is unusual to see packets with this destination IP because router (generally) do not route these packets [26].

The source IP address belongs to GE, it is possible a spoofed IP. This could be determined by contacting GENICTech@GE.COM and they are willing to investigate their outgoing traffic in the same period as the incoming traffics were captured by snort.

The address 10.0.0.1 is not unique, but has its counterpart in the massive SMB name wildcard alerts, which were triggered in the same period. Following are alerts originated from 10.0.0.1 to internal hosts.

08/03-21:45:49.890966	SMB Name Wildcard	10.0.0.1:137 -> my.net.157.241:137
08/03-21:45:51.385905	SMB Name Wildcard	10.0.0.1:137 -> my.net.157.241:137
08/03-21:45:52.885491	SMB Name Wildcard	10.0.0.1:137 -> my.net.157.241:137
08/04-14:39:47.164528	SMB Name Wildcard	10.0.0.1:137 -> my.net.99.44:137
08/04-14:39:48.662955	SMB Name Wildcard	10.0.0.1:137 -> my.net.99.44:137
08/04-14:39:50.162506	SMB Name Wildcard	10.0.0.1:137 -> my.net.99.44:137
08/04-17:26:04.766555	SMB Name Wildcard	10.0.0.1:137 -> my.net.157.241:137

Now the 10.0.0.1 address sometimes looks like from external and sometimes looks like used by internal. Determine which address, 10.0.0.1 or 3.0.0.99, is spoofed is very important. But no matter what, seeing this 10.0.0.1 IP address on the border indicate network configuration problem.

Recommendation: If the link layer is dumped when snort log the traffic, the MAC addresses in the above "UDP SRC and DST outside network" log should be consulted to determine the direction of the traffic. Due to the high volume of the traffic and its suspicion of UDP flooding, the administrator should take a close look at the configuration of the border router to make sure it dose not forward packets with private IP addresses.

The following IP info of the source address is provided by www.arin.net. No hacking history was found from Dshield.org regarding this IP:
Search results for: 3.0.0.99

```
General Electric Company (NET-GE-INTERNET)
  1 Independence Way
  Princeton, NJ 08540
  US

  Netname: GE-INTERNET
  Netblock: 3.0.0.0 - 3.255.255.255

  Coordinator:
    General Electric Company  (GET2-ORG-ARIN)  GENICTech@ge.com
    518-612-6672

  Record last updated on 12-Nov-1998.
  Database last updated on  23-Aug-2002 16:56:03 EDT.
```

High Severity Alert #13 64.194.26.225 sweep for port 1433

External IP address 211.232.192.153 had swept the university's network for port 1433. It hit 17730 different hosts in the internal network. Port 1433 is bound with Microsoft SQL Database service. This scan is very possibly related to the SQL worm

Spida activity. The worm was discovered in May, 2002 and has spread the Internet for a few months. It becomes the first most probed port according to the port report from isc.incidents.org at August 25, 2002[34].

The SQLspida worm [35] exploits the null password vulnerability with Microsoft SQL server and desktop edition [36]. It first scans for system listening on port 1433, Once it find out a open port, it will try to login the service with the sa account and blank password then start to propagate itself to the victim. The infected victim will repeat the same process and start to scan port 1433.

```
08/04-10:43:14 211.232.192.153:4700 -> my.net.5.31:1433 SYN ******S*
08/04-10:43:14 211.232.192.153:4752 -> my.net.5.83:1433 SYN ******S*
08/04-10:43:14 211.232.192.153:4756 -> my.net.5.87:1433 SYN ******S*
…
08/04-12:01:43 211.232.192.153:3898 -> my.net.199.252:1433 SYN ******S*
08/04-12:01:43 211.232.192.153:3900 -> my.net.199.254:1433 SYN ******S*
08/04-12:01:43 211.232.192.153:3899 -> my.net.199.253:1433 SYN ******S*
```

Correlation: No attacking history was found from Dshield.org. Other GCIA analysts have review the traffic from this university, however none of them observer similar activity because the worm is new. Mailing list threads http://cert.uni-stuttgart.de/archive/intrusions/2002/05/threads.html and http://cert.uni-stuttgart.de/archive/intrusions/2002/06/threads.html have multiple discussion and log related to the SQL spida worm.

Recommendation: Administrator should set strong password to all the sa account to protect the database. The access to the critical database should be strictly controlled. It is strongly recommended that administrator should use their own scanners to find out the vulnerable SQL server. The SQL Worm Scanner from eEye Digital Security is one of the choices [37].

Top Talkers List

1. Top 10 active source IPs and the destination port.

   The following data is extracted from the scans log. The most active source IP and the destination ports that these source IPs connect to are listed.

Hits	SourceIP	DestPort	Service	Reason
2436774	my.net.70.200	41170	Bluster Music share	Bluster Music share station
478405	my.net.84.234	80	HTTP	Infected by Code Red
169938	my.net.100.208	80	HTTP	Infected by Nimda worm
103512	my.net.165.24	6257	WInMX P2P file sharing	WInMX P2P file sharing station
89119	my.net.83.150	6257	WInMX P2P file sharing	WInMX P2P file sharing station
29492	my.net.81.27	28800	MS gaming zone	Game station
21019	24.138.61.171	80	HTTP	Scan port 80
20329	161.132.205.100	80	HTTP	Scan port 80

17730	211.232.192.153	1433	MS SQL	SQL Spida Worm
16929	my.net.137.7	53	DNS	DNS server recursively lookup Domain name for internal client

From this list, I can elect that the my.net.84.234 and my.net.100.208 are compromised; the "Exploit x86 xxx" alert could be false positive when it is associate with the file sharing program; the most active probing port from Internet are looking for HTTP and MSSQL service.

2. Top 10 active external source IPs and their primary attacks

The following data are extracted from the alert file. It list the 10 most active external IP and the primary attacks they conducted.

Type of Attacks from the SRCIP	Total Hits to internal Net	Source IP	Primary Alert Triggered by SrcIP
1	51396	3.0.0.99	UDP SRC and DST outside network
1	32161	63.250.213.12	UDP SRC and DST outside network *(Yahoo Broadcast to MultiCast IP)*
2	8375	194.98.189.139	External RPC call
2	6900	80.137.90.34	spp_http_decode: IIS Unicode attack detected
1	4980	63.250.213.73	UDP SRC and DST outside network *(Yahoo Broadcast to MultiCast IP)*
2	4529	61.182.50.241	External RPC call
1	3392	212.179.66.17	Watchlist 000220 IL-ISDNNET-990517 *(IMESH.com provide file sharing service)*
2	3214	216.228.171.81	SMB Name Wildcard
2	2482	151.203.178.36	spp_http_decode: IIS Unicode attack detected
1	2474	212.179.35.118	Watchlist 000220 IL-ISDNNET-990517 *(IMESH.com provide file sharing service)*

This list is not as helpful as the above one. It does help to figure out some false alert. But none of the IP in the list directly cause host compromised. The most noisy one is not necessary to be the most harmful one.

Analysis Process

The data was downloaded from http://www.incidents.org/logs. The total size of all the data files for 5 days is 700M bytes. All the files consist of 6,614,833 entries. Initially I was going to use snortsnarf to analyze the data. But even use the middle size server in our lab can never finish the perl script after two days and nights. I had to give up and start to use the tool, which at least can finish the calculating job in two days. I used Microsoft SQL Desktop Edition. I imported the plain text file to the database, trying to separate them to three column including eventdate, alertname and IP-Ports. After I write some SQL script to formalize the data, separate the source IP, source port, destination IP and destination port to different column. Here the trouble showed up.

The downloaded data have 6000-7000 error lines. Here I listed some error alerts to illustrate how painful I was when I was trying to rescue the data.

```
08/01-09:00:15.254484  [**] UDP SRC and DST outside network [**]
63.250.213.1208/01-08:52:55.544614  [**] UDP SRC and DST outside network
[**] 63.250.213.12:1031 -> 233.28.65.148:1031 -> 233.28.65.148:5779

08/04-19:10:19.478899  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL
nosize [**] my.net.84.234:3278 -> 47.58.98.4608/04-19:15:29.202409  [**]
IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize [**]
my.net.84.234:4769 -> 62.195.87.140:80

08/01-09:46:27.413652  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.99.18208/01-09:40:43.086153  [**] spp_http_decode: CGI Null Byte
attack detected [**] my.net.145.160:36869 -> 192.151.52.111:80

08/04-19:10:09.888060  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL
nosize [**] my.net.84.234:80
```

It is very often that the next alert overlaps with the previous alert. The 4 IP involved in 2 alerts confused together or miss.

The problem with the data might not be found out if use UNIX command. On contrast, Database requests the imported data to be very uniformed. A small irregular can cause failure. The SQL script I wrote to formalize the data has to consider all the possible irregular format of the alert data. I am not going to list them over here because I am going to run out of pages limitation,

After the whole formalization, I wrote some SQL scripts to generate statistics and query for specific alert when investigating. The simplest one
(Select * from alert where srcip like 'x.x.x.x' and destip like 'x.x.x.x') to investigate all the activity associate with a particular IP.

I did not import the scans data into database after experiment the pain with the alert file. I used unix command to generate some statistic.

Find out the internal server and their listening port:
cut -f2 -d' ' scans | grep 'my.net' | sort |uniq -c |sort -r -n > scans_int_server_port

Get the scan source IP and destination port it connect to
cut -f2,4 -d' ' scans | cut -f1,3 -d':' | sort |uniq -c |sort -r –n >scanned_srcip_destport

Sort a list of destination port of oos file
cut -f4 -d' ' oos-headers | cut -f2 -d':' | sort | uniq -c | sort -r -n > oos-headers-dst-port

After getting the statistics, other investigation processes are done manually. The most important step I went through are creating a service/client program profile for the most active internal host and filter out false positive. After this step, the real attack floated on the sea. Then I concentrate on the attacks with exploit ability and the anomaly activity. I always do a search for all the activities of the source IP and destination IP of the event of interest so that I can correlate all the events in all the log files.

List of References

[1]. ISC Port 139 Reports. URL:http://isc.incidents.org/port_details.html?port=139 (August 24, 2002)
[2]. SecurityFocus Mailing List Archive
World-wide distributed DoS and "warez" bot networks (fwd) May 3 2002 5:27PM
URL: http://online.securityfocus.com/archive/75/270867
[3]. ISS Security Center Alerts
Increased Hacking Activity Associated with Underground File-Sharing Networks May 3, 2002
URL: http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise117
[4]. TheoryGroup Unisog Mailing list Archive
ATTENTION MORE COMPROMISED COMPUTERS FOUND!
URL: http://www.theorygroup.com/Archive/Unisog/2002/msg00658.html
[5]. Hacked university machines
URL:http://staff.washington.edu/dittrich/misc/ddos/unisog-xdcc.txt
[6]. Symantec Security Response
URL:http://securityresponse.symantec.com/
[7] CERT® Advisory CA-2001-26 Nimda Worm
URL: http://www.cert.org/advisories/CA-2001-26.html
[8]. CERT® Advisory CA-2001-12 Superfluous Decoding Vulnerability in IIS
http://www.cert.org/advisories/CA-2001-12.html
[9]. CERT® Coordination Center
Steps for Recovering from a UNIX or NT System Compromise
URL: http://www.cert.org/tech_tips/win-UNIX-system_compromise.html
[10]. CERT® Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
URL: http://www.cert.org/advisories/CA-2001-19.html
[11]. CERT® Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL
URL: http://www.cert.org/advisories/CA-2001-13.html
[12]: CVE Vulnerability Database CVE-2001-0500 CVE Version: 20020625
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0500
[13] Google search engine
URL: www.google.com
[14] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=144.118.192.42 (August 24, 2002)
[15] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=130.74.32.130 (August 24, 2002)
[16] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=213.48.150.1(August 24, 2002)
[17]. Kyle Haugsness GCIA Practical Assignment
URL: http://www.giac.org/practical/Kyle_Haugsness_GCIA.zip
[18]. CVE Vulnerability Database. "CVE-2001-0414." CVE Version: 20020625
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0414
[19] SecurityFocus Vulnerability Database. "Bugtraq ID 2540."
URL: http://www.securityfocus.com/bid/2540

[20]. ISC Port 2011 Reports.
URL:http://isc.incidents.org/port_details.html?port=2011 (August 24, 2002)
[21] Game listen on Port 2011
URL: http://groups.google.ca/groups?q=port+2011+game&hl=en&lr=&ie=UTF-8&oe=UTF-8&selm=1991Apr10.200608.14756%40colorado.edu&rnum=1
URL: http://www.niksula.cs.hut.fi/~jkoi/gb/games/marvinII.announce.html
[22] Arin.net - Whois Database
URL: http://www.arin.net
[23] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=208.209.50.17(August 24 2002)
[24] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=140.221.9.138 (August 24 2002)
[25] CVE Vulnerability Database. "CVE-1999-0183." CVE Version: 20020625
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0183
[26] Address Allocation for Private Internets
URL: http://www.ietf.org/rfc/rfc1918.txt
[27] CERT® Advisory CA-2002-03 Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP)
URL: http://www.cert.org/advisories/CA-2002-03.html
[28] WS ping Pro
URL: www.ipswitch.com.
[29] Snmpwalk:
URL: ucd-snmp.ucdavis.edu.
[30]Using SNMP for Reconnaissance
URL: http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm
[31] PROTOS c06-snmpv1 test suite
URL: http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/0100.html
[32] Edward Peck GCIA practical assignment
URL: http://www.giac.org/practical/Edward_Peck_GCIA.doc
[33]. ISC port report for port 21
URL: http://isc.incidents.org/port_details.html?port=21 (August 25, 2002)
[34]: ISC top 10 ports report
URL: http://isc.incidents.org/top10.html
[35]: CERT® Incident Note IN-2002-04
Exploitation of Vulnerabilities in Microsoft SQL Server
URL: http://www.cert.org/incident_notes/IN-2002-04.html
[36]: CERT Vulnerability Note VU#635463
Microsoft SQL Server and Microsoft Data Engine (MSDE) ship with a null default password
URL: http://www.kb.cert.org/vuls/id/635463
[37]: SQL Worm Scanner from eEye Digital Security
URL: http://www.eeye.com/html/Research/Tools/sqlworm.html

[End of References]