Global Information Assurance Certification Paper

SANS Intrusion Detection Practical (Version 2.8)

Glenn Davis

Detect 1

FW-1 Logs

```
17:36:08 accept fw1 >lan3 proto udp src 209.91.102.126
         dst A.B.C.70 service 53 s_port 68 len 66
         rule 19
17:36:14 accept fw1 >lan3 proto udp src 209.91.102.126
         dst A.B.C.69 service 53 s_port 68 len 66
         rule 19
```

NFR Alerts

```
Wed May 16 17:36:06 s4 DHCP_MONITOR[27277]:
    Malformed DHCP/BOOTP packet.
    Source: 209.91.102.126 Dest: A.B.C.70
    Payload Length: 38
Wed May 16 17:36:07 s4 DHCP_MONITOR[27277]:
    Malformed DHCP/BOOTP packet.
    Source: 209.91.102.126 Dest: A.B.C.70
    Payload Length: 38
Wed May 16 17:36:12 s4 DHCP_MONITOR[27277]:
    Malformed DHCP/BOOTP packet.
    Source: 209.91.102.126 Dest: A.B.C.69
    Payload Length: 38
Wed May 16 17:36:13 s4 DHCP_MONITOR[27277]:
    Malformed DHCP/BOOTP packet.
    Source: 209.91.102.126 Dest: A.B.C.69
    Payload Length: 38
```

NFR Data

Time	**xid**	**Source IP**	**Dest IP**
2001/05/16 17:36	65536	A.B.C.70	209.91.102.126
	operation		Boot Reply
	htype		unknown
	hardware address length		132
	hops from server		131
	age of acquisition		1
	process		

Time	xid	Source IP	Dest IP
	flags		0
	client address(ciaddr)		7.100.110.115
	client address(yiaddr)		104.111.115.116
	bootstrap server address(siaddr)		8.109.121.100
	relay address		111.109.97.105
	client hardware address		6e:03:63:6f:6d:00
	server host name		
	bootfile		
2001/05/16 17:36	65536	A.B.C.70	209.91.102.126
	operation		Boot Reply
	htype		unknown
	hardware address length		132
	hops from server		131
	age of acquisition		1
	process		
	flags		0
	client address(ciaddr)		7.100.110.115
	client address(yiaddr)		104.111.115.116
	bootstrap server address(siaddr)		8.109.121.110
	relay address		111.109.97.105
	client hardware address		6e:03:63:6f:6d:00
	server host name		
	bootfile		
2001/05/16 17:36	65536	A.B.C.69	209.91.102.126
	operation		Boot Reply
	htype		unknown
	hardware address length		132
	hops from server		131
	age of acquisition		1
	process		
	flags		0

Time	xid	Source IP	Dest IP
		client address(ciaddr)	7.100.110.115
		client address(yiaddr)	104.111.115.116
		bootstrap server address(siaddr)	8.109.121.110
		relay address	111.109.97.105
		client hardware address	6e:03:63:6f:6d:00
		server host name	
		bootfile	
2001/05/16 17:36	65536	A.B.C.69	209.91.102.126
		operation	Boot Reply
		htype	unknown
		hardware address length	132
		hops from server	131
		age of acquisition	1
		process	
		flags	0
		client address(ciaddr)	7.100.110.115
		client address(yiaddr)	104.111.115.116
		bootstrap server address(siaddr)	8.109.121.110
		relay address (giaddr)	111.109.97.105
		client hardware address (chaddr)	6e:03:63:6f:6d:00
		server host name (sname)	
		bootfile	

1.1 Source of Trace:

My network.

1.2 Detect was generated by:

NFR v5 IDS, and Checkpoint Firewall-1 firewall logs. Note that the Checkpoint FW-1 is stateful, and this is reflected in the network logs. Only the inbound packet that was accepted was logged; the reply is not logged because it was assumed to be part of the original DNS query transaction.

1.3 Probability the source address was spoofed:

High. The intent was a DoS against DHCP clients, so for this attack to work the source address must be spoofed. Other evidence that these are crafted packets is: source port = 68 and there were 4 packets with the same source port number.

1.4 Description of attack:

This is a DoS against DHCP clients; they receive DHCP replies containing invalid IP addresses from a targeted name server.

Target of this attack appears to be ISP customers:

> nslookup 209.91.102.126
> Name: h-209-91-102-126.gen.cadvision.com
> Address: 209.91.102.126
>
> whois.arin.net: 209.91.102.126
>
> Cadvision Development Corp. (NETBLK-CADVISION-BLK2)
> Netname: CADVISION-BLK2
> Netblock: 209.91.64.0 - 209.91.127.255

1.5 Attack mechanism:

This attack works by sending a valid UDP DNS query to a name server, setting the source IP as the intended target of the DoS and the source port of 68. When the DNS server responds, the source and destination ports are swapped making this packet appear to be a DHCP or BOOTP reply message. When the intended target receives this unsolicited DHCP reply, it may use the data to set a new IP (invalid) address resulting in the loss of network connectivity. The DNS query ID in the spoofed packet must be set so that only the least significant bit in the high byte is set to 1; when the DNS response is interpreted as a DHCP reply the opcode field is set to 1 (Boot Reply).

Mapping DHCP packet fields to DNS packet fields is done as follows:

```
op + htype => ID
hlen => QR, Opcode, AA, TC, RD
hops => RA, Z, RCODE
xid => QDCOUNT, ANCOUNT
secs => NSCOUNT
flags => ARCOUNT
ciaddr, yiaddr, siaddr, etc => reply
```

This mapping of the fields, using the data in the NFR report above, shows that the original DNS packet was a valid query and the response is an authoritative reply that 'dnshost.mydomain.com' does not exist. (Note: the host name data was modified to mask the real domain name)

Field	Value	Interpretation
ID		Insufficient data to determine ID
QR	1	Query reply
Opcode	0	
AA	1	Authoritative answer
TC	0	
RD	0	
RA	1	Recursion available
Z	000	
RCODE	0011	3 = NXDOMAIN
data	'dnshost.mydomain.com'	hostname queried

1.6 Correlations:

This particular detect has never been seen before. However, there have been discussions in vuln-dev mailing list of DoS attacks against cable modem providers targeting the exhaustion of the DHCP IP scopes.

1.7 Evidence of active targeting:

These packets were sent only to DNS servers, no scan for DNS servers seen from this IP address. The packets were directly targeted at DNS servers, suggesting an earlier reconnaissance was done to find DNS servers.

1.8 Severity:

Severity=(Criticality+Lethality)-(System Countermeasures+Network Countermeasures)

(1 + 1) – (2 + 2) = -2

Severity: 1 Target is a DHCP client of an ISP
Lethality: 1 Lethal to intended victim, not us
System Countermeasures: 2 attack is a second order DoS
Network Countermeasures: 2

1.9 Defensive recommendation:

Modify firewall rule restrict DNS query source port to 53 and >1024

1.10 Multiple choice test question:

What are valid values for the source port in a UDP dns query?

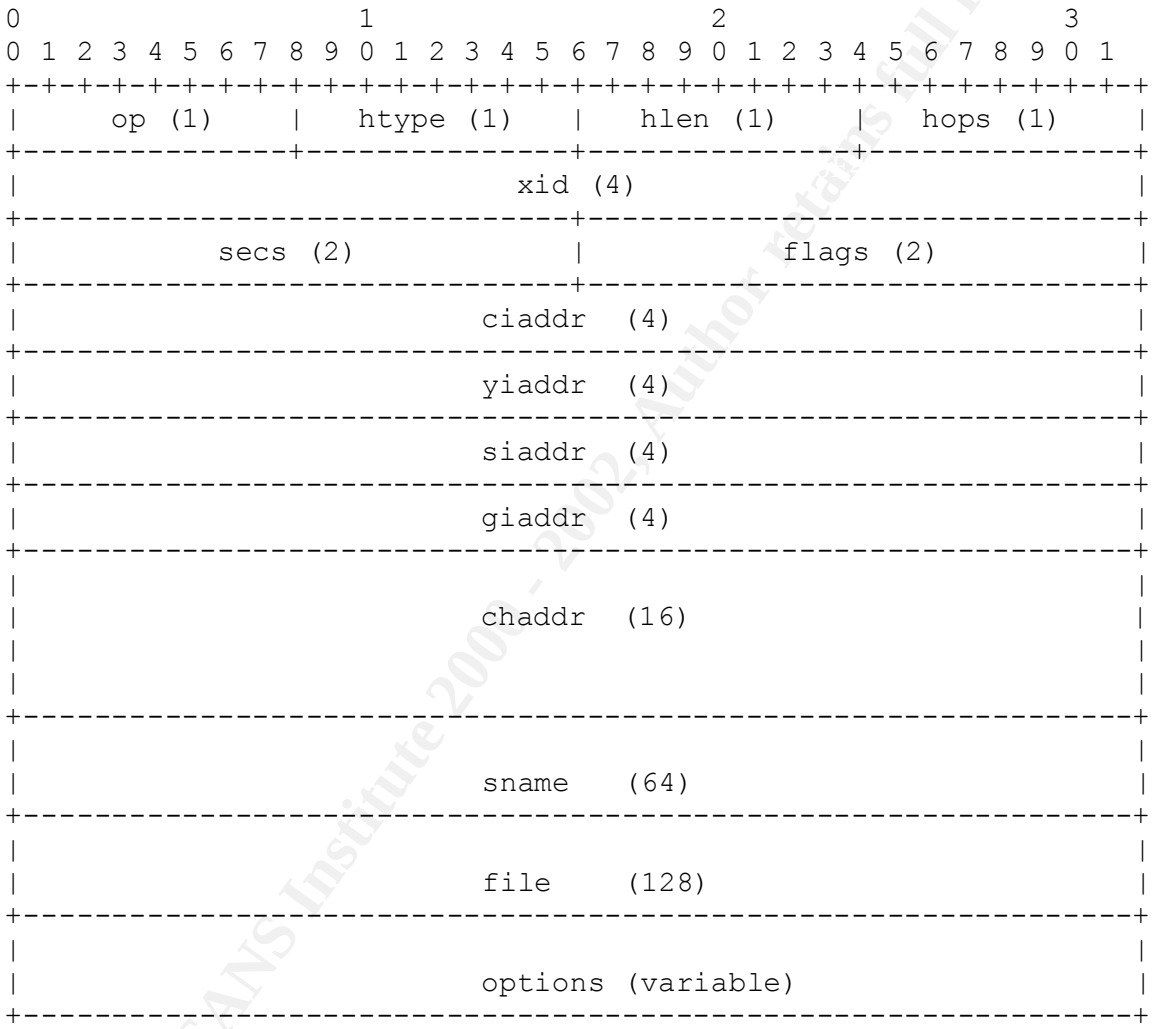
a) 53
b) < 32766
c) < 1023 or 53

d) < 1023

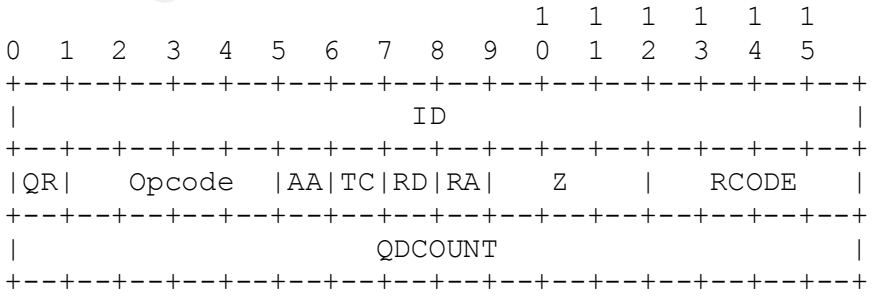
Answer: c

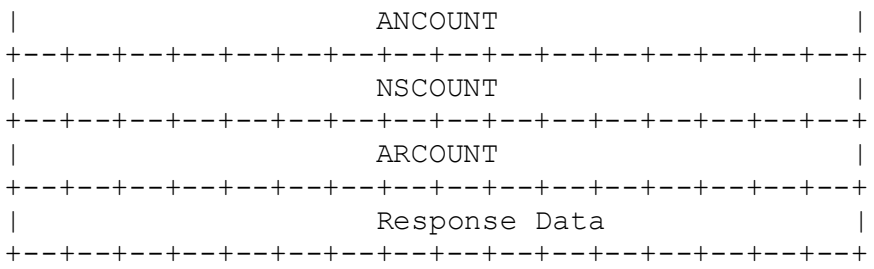
1.11 Appendix

DHCP Packet format



DNS Packet format





Detect 2

Checkpoint FW-1 Logs

15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.32 service domain s_port 3138 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.34 service domain s_port 3140 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.33 service domain s_port 3139 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.36 service domain s_port 3142 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.38 service domain s_port 3144 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.35 service domain s_port 3141 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.40 service domain s_port 3146 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.37 service domain s_port 3143 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.39 service domain s_port 3145 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.42 service domain s_port 3148 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.41 service domain s_port 3147 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.43 service domain s_port 3149 len60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.44 service domain s_port 3150 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.46 service domain s_port 3152 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.45 service domain s_port 3151 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.48 service domain s_port 3154 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.50 service domain s_port 3156 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.47 service domain s_port 3153 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.52 service domain s_port 3158 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.49 service domain s_port 3155 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.51 service domain s_port 3157 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.54 service domain s_port 3160 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.53 service domain s_port 3159 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.56 service domain s_port 3162 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.55 service domain s_port 3161 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.58 service domain s_port 3164 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.57 service domain s_port 3163 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.60 service domain s_port 3166 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.62 service domain s_port 3168 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.59 service domain s_port 3165 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.64 service domain s_port 3170 len 60 rule 39
15:22:58 accept fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.70 service domain s_port 3176 len 60 rule 18
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.61 service domain s_port 3167 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.76 service domain s_port 3182 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.82 service domain s_port 3188 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.88 service domain s_port 3194 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.128 service domain s_port 3234 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.134 service domain s_port 3241 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.140 service domain s_port 3247 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.146 service domain s_port 3253 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.152 service domain s_port 3259 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.159 service domain s_port 3266 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.194 service domain s_port 3301 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.198 service domain s_port 3305 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.202 service domains_port 3310 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.197 service domain s_port 3304 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.199 service domain s_port 3307 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.201 service domain s_port 3309 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.204 service domain s_port 3312 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.203 service domain s_port 3311 len 60 rule 39
15:22:58 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.195 service domain s_port 3302 rule 39
15:22:58 accept fw1 >lan3 proto udp src 216.25.136.196 dst A.B.C.70 service domain s_port 3611 len 58 rule 18
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.34 service domain s_port 3140 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.36 service domain s_port 3142 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.63 service domain s_port 3169 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.66 service domain s_port 3172 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.65 service domain s_port 3171 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.68 service domain s_port 3174 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.67 service domain s_port 3173 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.72 service domain s_port 3178 len 60 rule 39
15:23:01 accept fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.69 service domain s_port 3175 len 60 rule 18

```
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.71 service domain s_port 3177 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.73 service domain s_port 3179 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.75 service domain s_port 3181 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.81 service domain s_port 3187 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.87 service domain s_port 3193 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.93 service domain s_port 3199 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.131 service domain s_port 3237 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.137 service domain s_port 3244 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.158 service domain s_port 3265 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.145 service domain s_port 3252 len 60 rule 39
15:23:01 drop fw1 >lan3 proto tcp src 216.25.136.196 dst A.B.C.194 service domain s_port 3301 rule 39
15:23:01 accept fw1 >lan3 proto udp src 216.25.136.196 dst A.B.C.69 service domain s_port 3611 len 58 rule 18
```

NFR Alerts

```
Thu Mar  8 15:22:57 s4 dns_alerts[16209]:DNS Version Request.
                    Source: 216.25.136.196 Dest: A.B.C.70
Thu Mar  8 15:23:00 s4 dns_alerts[16209]:DNS Version Request.
                    Source: 216.25.136.196 Dest: A.B.C.69
```

NFR Data

Time	Source IP	Src Port	Dest. IP	Dst Port	Protocol
2001/03/08 15:22	216.25.136.196	3611	A.B.C.70	53	UDP
>\x92\x09\x80\x00\x00\x00\x01\x00\x00\x00\x00>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAA>BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB BBBBBBBBBBBBBB>CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC >\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\ x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20!""#$%&'()*+,\x34./0123 456789:;<=>EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE>FFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF=GGGGGGGGGGGGGGGGGGGG GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG\x00\x00\x01\x00\x01\x00\x00\x00\x01 \x00\xff\xbfT					
2001/03/08 15:22	216.25.136.196	3611	A.B.C.70	53	UDP
>\x92\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x07version\x04bind\x00\x00\x10\x 00\x03					
2001/03/08 15:23	216.25.136.196	3611	A.B.C.69	53	UDP
>\x94\x09\x80\x00\x00\x00\x01\x00\x00\x00\x00>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAA>BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB BBBBBBBBBBBBBB>CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC >\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\ x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20!""#$%&'()*+,\x34./0123 456789:;<=>EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE>FFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF=GGGGGGGGGGGGGGGGGGGG GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG\x00\x00\x01\x00\x01\x00\x00\x00\x01 \x00\xff\xbfT					
2001/03/08 15:23	216.25.136.196	3611	A.B.C.69	53	UDP
>\x94\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x07version\x04bind\x00\x00\x10\x 00\x03					

2.1 Source of Trace:

My network.

2.2 Detect was generated by:

A NFR IDS is on the external (Internet) segment of a CheckPoint FW-1 firewall. The logs are from the firewall, and alerts generated by NFR. The NFR data provides only the date/time, source IP, source port, destination IP, destination port, protocol, and payload.

2.3 Probability the source address was spoofed:

Low. The TCP port 53 scan needs a three way handshake response for the scanner to know he has found a system with an open port. The UDP exploit attempt source address could be spoofed because feedback to the originator is not required, and UDP does not use a three way handshake. However, since the source IP address of the TCP scans is the same as the IP address of the UDP Iquery, it is probably not spoofed.

2.4 Description of attack:

This appears to be a reconaissance scan for bind servers vulnerable to the Iquery buffer overflow attack.

Correlations:

arachNIDS: IDS277 named-probe-iquery
CVE: CVE-1999-0009
BUGTRAQ ID: BugtraqID 134
BLACK ICE: 2000409

2.5 Attack mechanism:

This a scan to find dns servers with the "fake iquery" option set in the configuration that are vulnerable to a buffer; pre 8.1.2 / 4.9.8 named nameservers. The attack works by opening scanning a sequence of IP addresses and attempting to connect to port 53. When a name server is found, a version request and a Iquery request are sent to the name server.

There is no executable code present in the data, so the only possible result would be to crash a vulnerable named - the length of the data segment is 467 bytes which is sufficient to generate an overflow condition. The signature does not match known known Iquery buffer overflow attacks: LSD TSIG, ADM, or . LSD's TSIG buffer overflow contains the values 0xab 0xcd in the first two bytes. ADM

Signature from these traces show an identical messsage id in the iquery and version request, with subsequent message id's incrementing by 2

Observations:
- source ports increment with time
- source ports always start approx 3000 in 8 observed scans
- scan sequence of hosts similar for each scan
- version request and iquery occur immediatly after sucsessfull tcp probe to port 53

This query is probably a pre-attack probe, prior to an attempted overflow of named:

2.6 Correlations:

Crist Clark reported Iquery packets with the same signature (AAAAABBBBCCC etc) on the Incidents mailing list, but there has been no other reports of this pattern. However, bind buffer overflow exploits are well known.

```
Subject: DNS Probe and (?) Exploit Attempt
Date:    Tue Mar 06 2001 12:01:59
Author:  Crist Clark < crist.clark@globalstar.com >
http://www.securityfocus.com/templates/archive.pike?list=75&mid=166850
```

I have seen 6 identical traces, at random times over the period Feb 19 through May 6, 2001.

2.7 Evidence of active targeting:

General scan of entire subnet for DNS servers, probably not active targeting.

2.8 Severity:

Severity=(Criticality+Lethality)-(System Countermeasures+ Network Countermeasures)

(5 + 2) – (2 + 4) = 1

Criticality: 5 DNS server is a critical target
Lethality: 2 attack is a reconnaissance scan
System Countermeasures: 2 traffic is allowed to target
Network Countermeasures: 4 target is running bind 9.1, and does not respond to iquery or version request

2.9 Defensive recommendation:

Defenses are fine, attack was blocked by firewall - but the perimeter router should have an ACL that permits port 53 traffic only to the name servers.

2.10 Multiple choice test question:

```
Thu Mar  8 15:22:57 s4 dns_alerts[16209]:DNS Version Request.
                    Source: 216.25.136.196 Dest: A.B.C.1
```

What does this alert message represent?
a) A reconnaissance scan
b) Attempted buffer overflow exploit
c) Normal dns traffic
d) Load balancing

Answer: a

Detect 3

Router Logs

Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.4(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.14(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.24(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.27(23), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.40(23), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.50(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.54(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.60(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.70(23), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.74(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.78(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.83(23), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.87(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.93(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.100(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.106(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.109(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.112(23), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.118(23), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.122(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.127(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.131(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.134(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.140(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.149(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.152(23), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.156(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.162(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.166(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.172(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.175(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.184(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.188(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.194(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.198(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.201(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.211(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.220(21), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.224(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.233(111), 1 packet
Nov 10 01:40:10: denied tcp 24.68.122.76(21305) -> A.B.C.243(111), 1 packet

Firewall Logs

```
01:40:10 drop fw1 >lan0 proto tcp src 24.68.122.76
         dst A.B.C.229 service 31337 s_port 21305
         len 40 rule 49

01:40:10 drop fw1 >lan0 proto tcp src 24.68.122.76
         dst A.B.C.214 service 31337 s_port 21305
```

```
         len 40 rule 49

01:40:10 drop fw1 >lan0 proto tcp src 24.68.122.76
         dst A.B.C.236 service 31337 s_port 21305
         len 40 rule 49

01:44:57 reject fw1 >lan0 proto tcp src 24.68.122.76
         dst A.B.C.143 service 31337 s_port 21305
         len 40 rule 10

01:44:57 reject fw1 >lan0 proto tcp src 24.68.122.76
         dst 255.255.255.255 service 31337 s_port 21305
         len 40 rule 10
```

3.1 Source of Trace:

My network.

3.2 Detect was generated by:

Cisco router ACL logs, Checkpoint FW-1 logs.

3.3 Probability the source address was spoofed:

Low. This is a TCP scan that requires a three way handshake to complete, so a spoofed source IP address would not work. The packets are crafted because the source port is identical in every packet.

3.4 Description of attack:

Subnet scan using tcp on ports 21, 23, and 111 (blocked by router ACL) followed by a scan of four hosts on port 31337 (eleet) – back orifice (blocked by Firewall). Source port identical on all packets.

The source of the attack is an @Home cable modem system running SyGate.

```
ARIN Output (from whois.arin.net)

Shaw Fiberlink ltd. (NETBLK-FIBERLINK-CABLE)
Netname: FIBERLINK-CABLE
Netblock: 24.64.0.0 - 24.71.255.255

# net view \\24.68.122.76
Shared resources at \\24.68.122.76
Share name   Type         Used as  Comment
-------------------------------------------
HP DESKJET 5 Print                SyGate Share
STAR NX-1020 Print                SyGate Share
The command completed successfully.

# nbtstat -A 24.68.122.76
NetBIOS Remote Machine Name Table
   Name               Type        Status
---------------------------------------
CS40576-A      <00>  UNIQUE       Registered
```

```
CS-14976        <00>  GROUP         Registered
CS40576-A       <03>  UNIQUE        Registered
CS40576-A       <20>  UNIQUE        Registered
CS-14976        <1E>  GROUP         Registered
CS-14976        <1D>  UNIQUE        Registered
..__MSBROWSE__.<01>  GROUP         Registered

MAC Address = 00-80-C8-C2-16-89
```

CVE: CVE-2000-0113 The SyGate Remote Management program does not properly restrict access to its administration service, which allows remote attackers to cause a denial of service, or access network traffic statistics.

3.5 Attack mechanism:

This is a network scan for open ports: 21 (ftp), 23 (telnet), and 111 (portmap). Random hosts in the subnet are scanned sequentially checking one port per host. Host identification can be made by looking at the ftp and telnet banners returned on sucessfull connections.

The scan originated from a host running SyGate, a tool that runs on windows platforms to enable multiple computers to simultaneously share a single Internet connection by using NAT (network address translation). Early versions of SyGate had vulnerabilities that permitted remote users to obscure network scans by bouncing them off of a SyGate server.

3.6 Correlations:

There are many well known vulnerabilites in RPC services, FTP, and TELNET as indicated in the following list of CVE entries.

CVE Entries for FTP Vulnerabilities

Name	Description
CVE-1999-0017	FTP servers can allow an attacker to connect to arbitrary ports on machines other than the FTP client, aka FTP bounce.
CVE-1999-0035	Race condition in signal handling routine in ftpd, allowing read/write arbitrary files.
CVE-1999-0054	Sun's ftpd daemon can be subjected to a denial of service.
CVE-1999-0075	PASV core dump in wu-ftpd daemon when attacker uses a QUOTE PASV command after specifying a username and password.
CVE-1999-0079	Remote attackers can cause a denial of service in FTP by issuing multiple PASV commands, causing the server to run out of available ports.
CVE-1999-0080	wu-ftp FTP server allows root access via "site exec" command.
CVE-1999-0082	CWD ~root command in ftpd allows root access.
CVE-1999-0083	getcwd() file descriptor leak in FTP
CVE-1999-0097	The AIX FTP client can be forced to execute commands from a malicious server through shell metacharacters (e.g. a pipe character).
CVE-1999-0185	In SunOS or Solaris, a remote user could connect from an FTP server's data port to an rlogin server on a host that trusts the

Name	Description
	FTP server, allowing remote command execution.
CVE-1999-0201	A quote cwd command on FTP servers can reveal the full path of the home directory of the "ftp" user.
CVE-1999-0202	The GNU tar command, when used in FTP sessions, may allow an attacker to execute arbitrary commands.
CVE-1999-0219	Buffer overflow in Serv-U FTP server when user performs a cwd to a directory with a long name.
CVE-1999-0302	SunOS/Solaris FTP clients can be forced to execute arbitrary commands from a malicious FTP server.
CVE-1999-0349	A buffer overflow in the FTP list (ls) command in IIS allows remote attackers to conduct a denial of service and, in some cases, execute arbitrary commands.
CVE-1999-0351	FTP PASV "Pizza Thief" denial of service and unauthorized data access. Attackers can steal data by connecting to a port that was intended for use by a client.
CVE-1999-0368	Buffer overflows in wuarchive ftpd (wu-ftpd) and ProFTPD lead to remote root access, a.k.a. palmetto.
CVE-1999-0432	ftp on HP-UX 11.00 allows local users to gain privileges.
CVE-1999-0707	The default FTP configuration in HP Visualize Conference allows conference users to send a file to other participants without authorization.
CVE-1999-0777	IIS FTP servers may allow a remote attacker to read or delete files on the server, even if they have "No Access" permissions.
CVE-1999-0789	Buffer overflow in AIX ftpd in the libc library.
CVE-1999-0838	Buffer overflow in Serv-U FTP 2.5 allows remote users to conduct a denial of service via the SITE command.
CVE-1999-0878	Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via MAPPING_CHDIR.
CVE-1999-0879	Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via macro variables in a message file.
CVE-1999-0914	Buffer overflow in the FTP client in the Debian GNU/Linux netstd package.
CVE-1999-0950	Buffer overflow in WFTPD FTP server allows remote attackers to gain root access via a series of MKD and CWD commands that create nested directories.
CVE-1999-0955	Race condition in wu-ftpd and BSDI ftpd allows remote attackers gain root access via the SITE EXEC command.
CVE-1999-0997	wu-ftp with FTP conversion enabled allows an attacker to execute commands via a malformed file name that is interpreted as an argument to the program that does the conversion, e.g. tar or uncompress.
CVE-2000-0150	Firewall-1 allows remote attackers to bypass port access restrictions on an FTP server by forcing it to send malicious packets which Firewall-1 misinterprets as a valid 227 response to a client's PASV attempt.
CVE-2000-0462	ftpd in NetBSD 1.4.2 does not properly parse entries in /etc/ftpchroot and does not chroot the specified users, which allows those users to access other files outside of their home directory.
CVE-2000-0514	GSSFTP FTP daemon in Kerberos 5 1.1.x does not properly

Name	Description
	restrict access to some FTP commands, which allows remote attackers to cause a denial of service, and local users to gain root privileges.
CVE-2000-0573	The lreply function in wu-ftpd 2.6.0 and earlier does not properly cleanse an untrusted format string, which allows remote attackers to execute arbitrary commands via the SITE EXEC command.
CVE-2000-0577	Netscape Professional Services FTP Server 1.3.6 allows remote attackers to read arbitrary files via a .. (dot dot) attack.
CVE-2000-0636	HP JetDirect printers versions G.08.20 and H.08.20 and earlier allow remote attackers to cause a denial of service via a malformed FTP quote command.
CVE-2000-0640	Guild FTPd allows remote attackers to determine the existence of files outside the FTP root via a .. (dot dot) attack, which provides different error messages depending on whether the file exists or not.
CVE-2000-0641	Savant web server allows remote attackers to execute arbitrary commands via a long GET request.
CVE-2000-0674	ftp.pl CGI program for Virtual Visions FTP browser allows remote attackers to read directories outside of the document root via a .. (dot dot) attack.
CVE-2000-0676	Netscape Communicator and Navigator 4.04 through 4.74 allows remote attackers to read arbitrary files by using a Java applet to open a connection to a URL using the "file", "http", "https", and "ftp" protocols, as demonstrated by Brown Orifice.
CVE-2000-0717	GoodTech FTP server allows remote attackers to cause a denial of service via a large number of RNTO commands.
CVE-2000-0761	OS2/Warp 4.5 FTP server allows remote attackers to cause a denial of service via a long username.
CVE-2000-0813	Check Point VPN-1/FireWall-1 4.1 and earlier allows remote attackers to redirect FTP connections to other servers ("FTP Bounce") via invalid FTP commands that are processed improperly by FireWall-1, aka "FTP Connection Enforcement Bypass."
CVE-2000-0837	FTP Serv-U 2.5e allows remote attackers to cause a denial of service by sending a large number of null bytes.
CVE-2000-1027	Cisco Secure PIX Firewall 5.2(2) allows remote attackers to determine the real IP address of a target FTP server by flooding the server with PASV requests, which includes the real IP address in the response when passive mode is established.
CVE-2000-1182	WatchGuard Firebox II allows remote attackers to cause a denial of service by flooding the Firebox with a large number of FTP or SMTP requests, which disables proxy handling.
CVE-2001-0053	One-byte buffer overflow in replydirname function in BSD-based ftpd allows remote attackers to gain root privileges.
CVE-2001-0054	Directory traversal vulnerability in FTP Serv-U before 2.5i allows remote attackers to escape the FTP root and read arbitrary files by appending a string such as "/..%20." to a CD command, a variant of a .. (dot dot) attack.
CVE-2001-0318	Format string vulnerability in ProFTPD 1.2.0rc2 may allow attackers to execute arbitrary commands by shutting down the

Name	Description
	FTP server while using a malformed working directory (cwd).

CVE Entries for TELNET vulnerabilities

Name	Description
CVE-1999-0073	Telnet allows a remote client to specify environment variables including LD_LIBRARY_PATH, allowing an attacker to bypass the normal system libraries and gain root access.
CVE-1999-0087	Denial of service in AIX telnet can freeze a system and prevent users from accessing the server.
CVE-1999-0192	Buffer overflow in telnet daemon tgetent routing allows remote attackers to gain root access via the TERMCAP environmental variable.
CVE-1999-0230	Buffer overflow in Cisco 7xx routers through the telnet service.
CVE-1999-0273	Denial of service through Solaris 2.5.1 telnet by sending ^D characters.
CVE-1999-0290	The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.
CVE-1999-0416	Vulnerability in Cisco 7xx series routers allows a remote attacker to cause a system reload via a TCP connection to the router's TELNET port.
CVE-1999-0740	Remote attackers can cause a denial of service on Linux in.telnetd telnet daemon through a malformed TERM environmental variable.
CVE-1999-0749	Buffer overflow in Microsoft Telnet client in Windows 95 and Windows 98 via a malformed Telnet argument.
CVE-1999-0817	Lynx WWW client allows a remote attacker to specify command-line parameters which Lynx uses when calling external programs to handle certain protocols, e.g. telnet.
CVE-1999-0889	Cisco 675 routers running CBOS allow remote attackers to establish telnet sessions if an exec or superuser password has not been set.
CVE-1999-0991	Buffer overflow in GoodTech Telnet Server NT allows remote users to cause a denial of service via a long login name.
CVE-2000-0113	The SyGate Remote Management program does not properly restrict access to its administration service, which allows remote attackers to cause a denial of service, or access network traffic statistics.
CVE-2000-0166	Buffer overflow in the InterAccess telnet server TelnetD allows remote attackers to execute commands via a long login name.
CVE-2000-0212	InterAccess TelnetID Server 4.0 allows remote attackers to conduct a denial of service via malformed terminal client configuration information.
CVE-2000-0268	Cisco IOS 11.x and 12.x allows remote attackers to cause a denial of service by sending the ENVIRON option to the Telnet daemon before it is ready to accept it, which causes the system to reboot.
CVE-2000-0598	Fortech Proxy+ allows remote attackers to bypass access restrictions for to the administration service by redirecting their connections through the telnet proxy.
CVE-2000-0665	GAMSoft TelSrv telnet server 1.5 and earlier allows remote attackers to cause a denial of service via a long username.

Name	Description
CVE-2000-0733	Telnetd telnet server in IRIX 5.2 through 6.1 does not properly cleans user-injected format strings, which allows remote attackers to execute arbitrary commands via a long RLD variable in the IAC-SB-TELOPT_ENVIRON request.
CVE-2000-0991	Buffer overflow in Hilgraeve, Inc. HyperTerminal client on Windows 98, ME, and 2000 allows remote attackers to execute arbitrary commands via a long telnet URL, aka the "HyperTerminal Buffer Overflow" vulnerability.
CVE-2000-1184	telnetd in FreeBSD 4.2 and earlier, and possibly other operating systems, allows remote attackers to cause a denial of service by specifying an arbitrary large file in the TERMCAP environmental variable, which consumes resources as the server processes the file.
CVE-2001-0041	Memory leak in Cisco Catalyst 4000, 5000, and 6000 series switches allows remote attackers to cause a denial of service via a series of failed telnet authentication attempts.
CVE-2001-0185	Netopia R9100 router version 4.6 allows authenticated users to cause a denial of service by using the router's telnet program to connect to the router's IP address, which causes a crash.

CVE Entries for RPC Services

Name	Description
CVE-1999-0003	Execute commands as root via buffer overflow in Tooltalk database server (rpc.ttdbserverd)
CVE-1999-0008	Buffer overflow in NIS+, in Sun's rpc.nisd program
CVE-1999-0208	rpc.ypupdated (NIS) allows remote users to execute arbitrary commands.
CVE-1999-0212	Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server.
CVE-1999-0228	Denial of service in RPCSS.EXE program (RPC Locator) in Windows NT.
CVE-1999-0320	SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files.
CVE-1999-0353	rpc.pcnfsd in HP gives remote root access by changing the permissions on the main printer spool directory.
CVE-1999-0493	rpc.statd allows remote attackers to forward RPC calls to the local operating system via the SM_MON and SM_NOTIFY commands, which in turn could be used to remotely exploit other bugs such as in automountd.
CVE-1999-0687	The ToolTalk ttsession daemon uses weak RPC authentication, which allows a remote attacker to execute commands.
CVE-1999-0696	Buffer overflow in CDE Calendar Manager Service Daemon (rpc.cmsd)
CVE-1999-0900	Buffer overflow in rpc.yppasswdd allows a local user to gain privileges via MD5 hash generation.
CVE-1999-0969	The Windows NT RPC service allows remote attackers to conduct a denial of service using spoofed malformed RPC packets which generate an error message that is sent to the spoofed host, potentially setting up a loop, aka Snork.
CVE-1999-0974	Buffer overflow in Solaris snoop allows remote attackers to gain root privileges via GETQUOTA requests to the rpc.rquotad

Name	Description
	service.
CVE-2000-0508	rpc.lockd in Red Hat Linux 6.1 and 6.2 allows remote attackers to cause a denial of service via a malformed request.

3.7 Evidence of active targeting:

This represents a subnet scan, not actively targeted.

3.8 Severity:

Severity=(Criticality+Lethality)-(System Countermeasures+Network Countermeasures)

(2 + 2) – (5 + 5) = -6

Criticality: 2 Not actively targeted
Lethality: 2 attack is a reconnaissance scan
System Countermeasures: 5 traffic is blocked at perimeter router and firewall
Network Countermeasures 5 Internet connected systems are not running portmap, telnet, or ftp

3.9 Defensive recommendation:

The firewall rules and router ACL's are configured correctly. However, a router or network device is generating a "255.255.255.255" packet in response to a subnet broadcast address - this feature can be used in DoS attacks known as "Smurf Amplification" and should be disabled.

3.10 Multiple choice test question:

SyGate is a useful tool for Corporate Internet users because:
a) It has no known vulnerabilites
b) It is free
c) It acts as a personal firewall
d) None of the above

Answer: d

Detect 4

13:34:48 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.132 rule 38 icmp-type 3 icmp-code 1
15:14:54 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.158 rule 38 icmp-type 3 icmp-code 1
17:36:54 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.151 rule 38 icmp-type 3 icmp-code 1
18:12:02 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.136 rule 38 icmp-type 3 icmp-code 1
18:56:33 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.154 rule 38 icmp-type 3 icmp-code 1
19:12:11 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.139 rule 38 icmp-type 3 icmp-code 1
19:14:36 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.204 rule 38 icmp-type 3 icmp-code 1
20:48:31 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.156 rule 38 icmp-type 3 icmp-code 1
22:29:37 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.154 rule 38 icmp-type 3 icmp-code 1
22:42:50 accept fw1 >lan3 proto icmp src 157.130.241.17 dst A.B.C.202 rule 38 icmp-type 3 icmp-code 1

4.1 Source of Trace:

My network.

4.2 Detect was generated by:

Script to scan checkpoint firewall logs for destination IP addresses that are not in use

4.3 Probability the source address was spoofed:

The source address of the ICMP packets are probably not spoofed, but these ICMP host unreachable packets were generated by spoofed packets because the destination IP addresses in the trace are not in use. This is a third-party effect of a scan or DoS using spoofed source IP addresses.

4.4 Description of attack:

A denial of service attack or host scan targeting 157.130.241.17 using random spoofed source addresses to mask the real origin. This is not a mapping of our network because ICMP messages are never generated in response to an ICMP message. We see the ICMP host unreachable messages because our IP addresses are being spoofed.

4.5 Attack mechanism:

Routers generate ICMP host unreachable messages (ICMP type 3 code 1) in response to a packet whose destination IP address is on a directly connected subnet, but the destination IP address does not reply to a ARP request by the router.

This is either a very slow, stealthy scan, or a large number of random spoofed addresses were used by the scanner. This may also be TF2N (Tribe Flood Network) looking for valid spoofed IP address to use. The source if the ICMP messages was confirmed to be a router:

nslookup 157.130.241.17: 500.Serial2-11.GW5.LAX4.ALTER.NET
whois source: whois.arin.net
UUNET Technologies, Inc. (NET-UUNETCUSTB40)
Netname: UUNETCUSTB40
Netblock: 157.130.0.0 - 157.130.255.255

4.6 Correlations:

CVE-1999-0214: Denial of service by sending forged ICMP unreachable packets.

CAN-1999-0454: A remote attacker can sometimes identify the operating system of a host based on how it reacts to some IP or ICMP packets, using a tool such as nmap or queso.

CAN-2000-0138: A system has a distributed denial of service (DDOS) attack master, agent, or zombie installed, such as: Trinoo, Tribe Flood Network (TFN), Tribe Flood Network 2000 (TFN2K), stacheldraht, mstream, or shaft.

4.7 Evidence of active targeting:

Third party effect

4.8 Severity:

Severity=(Criticality+Lethality)-(System Countermeasures+Network Countermeasures)

(2 + 2) – (2 + 2) = 0

Criticality: 2 Target IP addresses are not in use
Lethality: 2 Second order effect
System Countermeasures: 2 traffic is allowed to target
Network Countermeasures: 2

4.9 Defensive recommendation:

Defenses are good, but additional measures to improve security are: block ICMP messages at the external perimieter router and firewall, and configure the perimeter router to not generate ICMP error messages. However, ICMP unreachable - fragmentation required messages are needed to support path MTU discovery.

Installation of a NID on the external network segment that is capable of capturing the entire packet (e.g. snort, shadow) would permit examination of the data portion of the ICMP message which would have included the offending IP header and first 8 bytes of data so we would know what the actual cause was.

4.10 Multiple choice test question:

ICMP packets should be blocked by a perimeter router or firewall because:
a) They can be used as a covert channel
b) ICMP is often used in DoS attacks
c) Network scanners require ICMP replies for successful reconnaissance
d) All of the above

Answer: d

Detect 5

```
Fri May  4 08:46:08 s4 BAD_WEB[27277]:Possible Attack URL:
GET /cgi-bin/phf?Qalias=x
/bin/cat /etc/passwd HTTP/1.0  Source: A.B.C.72 Dest:
216.93.104.34

A.B.C.15 - xxxxxxx [04/May/2001:08:47:55 -0600] "GET
http://www.grex.org/cgi-
bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd HTTP/1.0" 504 282
- - - - 448 108 - - 107
```

5.1 Source of Trace:

My network.

5.2 Detect was generated by:

NFR IDS, and Netscape http proxy server logs.

5.3 Probability the source address was spoofed:

Low. The source address is internal to our network, forwarded by an internal proxy server. If the address was spoofed the TCP 3 way handshake would not have been completed, and the proxy log entry would not have been generated. For this attack to work a TCP session must be established between the attacker and the target host.

5.4 Description of attack:

CGI phf program allows remote command execution through shell metacharacters.

CVE: CVE-1999-0067
CERT:CA-96.06.cgi_example_code
XForce:http-cgi-phf
Bugtraq ID:629

5.5 Attack mechanism:

phf is a directory service that was distributed with NCSA httpd and Apache web servers. In the vulnerable versions of phf, it passed unchecked the newline (hex 0x0a) characters to the unix shell allowing remote command execution. In this case it tries to display a unix password file (/etc/password).

5.6 Correlations:

This attack is common, for example:

- http://www.sans.org/y2k/practical/Don_Kendrick.doc
- http://www.sans.org/y2k/practical/Potheri_Mohan.doc
- http://www.sans.org/y2k/practical/Donald_Tomczak.doc
- http://www.sans.org/y2k/practical/Allison_Miller.doc

- http://www.sans.org/y2k/practical/Randall_Heck.doc

5.7 Evidence of active targeting:

Yes. The exploit attempt was directed to a specific host.

5.8 Severity:

Severity=(Criticality+Lethality)-(System Countermeasures+Network Countermeasures)

(1 + 2) – (2 + 4) = -3

Criticality: 1. Target is not one of our hosts
Lethality: 2.
System Countermeasures: 2. Traffic is allowed to target
Network Countermeasures: 4. This is an old, well known vulnerability

5.9 Defensive recommendation:

Our defenses are fine.

5.10 Multiple choice test question:

```
GET http://www.targetwebserver.org/cgi-
bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd HTTP/1.0
```

What is being attempted with the above http command?
a) buffer overflow
b) searching for an alias in the password file
c) view the password file
d) unicode attack

Answer: c

Large Scale Distributed Intrusion Detection

Large scale Internet wide aggregation of intrusion events is a recent phenomenon driven by SANS (and others) to improve the security of the Internet by sharing information between security professionals. These event correlation systems have been attributed with the early detection of worms that infected a large number of Internet sites. This approach is valuable for detecting broad based attacks, but has limitations that severely diminish the value of the data collected.

Introduction

More attention is given to the technology and tactics of deploying and managing Intrusion Detection Systems (IDS) rather than the strategic aspects. What is the strategic goal of implementing an IDS? The purpose is to identify activity defined as: any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource. Ultimately the goal is a CyberSpace Situational Awareness [5] where attacks are detected and repelled in realtime.

The challenges of distributed intrusion detection include [11]: "widely distributed heterogeneous environment, voluminous noisy and volatile data, incomplete information for decision making, diverse variety of probes, difficulty in communication coordination command and control, lack of trust between entities, and changing attack patterns".

An 1998 evaluation of ID systems by DARPA [15] found that: research IDS were better that the commercial systems, older exploits were the most likely to be detected, and even the best systems failed to detect approximately half of the newer attacks. They suggested that research should focus on techniques to find new attacks rather than extending existing rule based ID systems

In 1987 Dorothy Denning [9] described the first model of an Intrusion Detection System. This is one of the most widely cited papers in the intrusion detection literature, and is a seminal work. To detect coordinated attacks against many hosts NADIR (Network Anomaly Detection and Intrusion Reporter) and DIDS (Distributed Intrusion Detection System) were developed in 1991. [1],[3]

Scalability issues were addressed in1994 by Mark Crosbie and Gene Spafford [8] who suggested the use of autonomous agents. These agents improved the scalability, maintainability, efficiency and fault tolerance of the IDS. In 1996 GrIDS [14] tried another approach to solve scalability problem. This system facilitates the detection of large-scale automated or coordinated attacks used a graph based correlation approach.

As the volume of data generated by IDS became unmanageable, Ross Anderson and Abida Khattak [2] suggested the use of information retrieval, distinct from data mining, techniques into intrusion detection tools. The concept was to provide an "AltaVista" like

engine for viewing and searching audit trails using Glimpse. This tool would most likely to be used in forensic applications rather than a real time alert system.

The history of ID development has been directed to addressing problems of scalability, decreasing false positives, data fusion, and creating visualization tools.

Current attempts at Large Scale ID

Distributed Intrusion Detection Systems (DIDS) within organizations has become a fact of life. As more firewalls and sensors are installed in an environment, there is an increasing incentive for integration and automation. Manually reviewing raw log files has become impossible because of the volume of data. Open software projects like ACID (Analysis Console for Intrusion Detection) http://www.cert.org/kb/acid and formatting tools for SNORT ID data http://www.snort.org/ like SnortSnarf are good examples of tools to manage ID data. Many of the commercial ID tools also have a central consoles, or aggregation points, (RealSecure, NFR, etc), and there are several enterprise security console products to manage security events from diverse sensors (e.g. Intelitactics NSM, Axent ESM, etc.)

Large scale Internet based aggregation of intrusion events is a newer phenomenon. ID data is contributed by Internet connected sites, stored at a central repository, and then searched for activity that corresponds with broad-based attacks such as the ramen and red worms. The best known examples include: Global Incident Analysis Center (GIAC) http://www.sans.org/giac.htm/, Attack Registry Intelligence Service (ARIS) http://aris.securityfocus.com/, DShield Distributed Intrusion Detection System http://www.dshield.org/, myNetWatchman http://www.mynetwatchman.com/, and Consensus Intrusion Database (CID) http://www.incidents.org/.

Other implementations of this approach exist in closed environments (e.g. military), but this information is not available to the public.

Problems with the Data

For ID systems to be useful, the data from diverse sensors must be reduced to a common format. CID and DShield use the TCP quad format: date/time, source IP, source port, destination IP, destination port, protocol, and TCP flags. ARIS collects TCP quad data and sensor information formatted using XML, and also allows common events like port scans to be represented as a single entity.

Data submitted to these services originates from traffic that was blocked at a firewall or router, or events triggered by a filter to detect abnormal (anomalous) packets (e.g. SYN and RST TCP flags both set). The source of the events is normally obscured, presumably to prevent others using the data as a source of recon information, or because of a lack of trust. Data is sent to CID and GIAC via email, while AFIDS uses an encrypted SSL channel.

Performing collection of network data in a host other than the one to which the data is destined can provide the attacker the possibility of performing insertion and evasion attacks [12]. The data reduction results in a loss of fidelity, so that correlation based on source IP, sequence number, fragment identifier, or TTL, is not possible. No validation checking, in done by the clients so data will have a high level of false positives, or noise.

Worms may be a method used by attackers to generate sufficient noise so that the real attacks are more difficult to detect. Projects like CID targeting large scale integration may be harmful because they cause the users of these services to focus on the top 5 ports/source IP, rather that newer vulnerabilities or exploits that are more likely to be successful.

Types of Inference		Level of Inference
- Cyberspace Situational Awareness		
- Threat Analysis		- HIGH
- Situational Assessment		
- Behavior of Intruder		- MEDIUM
- Identity of Intruder		
- Rate of Intrusion		
- Existence of Intrusion		- LOW

Hierarchy of IDS Data Fusion Inferences [6]

In order to achieve higher levels of inference, some form of data fusion is required; the process of refining raw data so that it can be used to make inferences, or assertions, about the intent of an attacker. One approach for formatting the data is using the ASN.1 notation. [5] (e.g. tcpSYNFlood OID ::== { iso 3.6.1.5.1.3.1.1 }) A security MIB could be devised that would permit implementation of a relational database for storing the data. Performing first level aggregation, or filtering, at source to minimize the centralized processing and transport overhead. Including more detail in submitted data would make the database much more useful. Of course, a fast mechanism for querying this database would also be required.

CID/ARIS/etc are limited to low levels of inference: existence, rate, and possibly identity of intruder because of the low data fidelity and low signal to noise ratio. These large-scale efforts are therefore not useful for: tracking hackers to their source, real-time response, or identifying new security flaws. These systems only identify the wide spread exploitation of a vulnerability, which may occur many months after the development of a new exploit. Individuals or organizations with sufficient security literacy to use services like GIAC/ARIS/DShield will already have protected themselves from the common attacks of the day (e.g. dns, portmap, lpr) so derive little value from the service.

Monolithic versus Distributed Systems

Monolithic systems can not scale. At some point the data collection overwhelms the capacity of the network being monitored. The CID approach is to use hierarchical aggregation from Internet Storm Center analysis and coordination centers (SACCs) to

The Global Analysis and Coordination Center (GISWACC). The SACC's serve industries or communities permitting some source IP information to be deduced. This type of hierarchical model was shown to scale in GrIDS [14].

The participation rate of Internet connected sites would need to be high to achieve an accurate understanding of the current attack methods but this would overwhelm the data collection techniques in use.

Conclusions

Event correlation systems like CID are useful because they allow the detection of compromised machines, which can then be shut down or shunned by adding filters on routers/firewalls. Given the large number of systems compromised by well-known exploits, this is valuable information if the information could be disseminated effectively.

Given the difficulty of convincing Internet sites to share ID data, the compromises made by CID, etc., seem reasonable. . If they had attempted to address all issues (trust, obscure data formats, etc) out front - it would never be up and running and been able to show positive results quickly.

Large scale aggregation systems don't detect new, or novel, attacks because of the low of data fidelity, and low signal/noise ratio. Correlation based on TTL, TCP sequence, etc., is not possible. The data collection system is vulnerable to insertion and evasion attacks, has data volume and scaling problems. It is impossible to have an effective situational awareness system (with high resolution spatial and temporal correlation) using in-band communication [5].

LSID systems can be improved by inclusion of: data fusion techniques increased data fidelity, filtering/aggregation of ID data at source, dynamically configurable autonomous agents [8], and an inverted text search engine (e.g. Glimpse).

Overall, systems like CID are a necessary, but not sufficient, piece of the ID landscape. They fill the role of identifying large-scale intrusions be need substantial improvement to permit detection of new attacks.

References

1. Anderson, D., Frivold, T., Valdes, A., "Next-generation intrusion detection expert system (NIDES)", Technical report, SRI-CSL-95-07, SRI International, Computer Science Lab, May 1995.

2. Anderson, Ross, Khattak, Abida ,"The Use of Information Retrieval Techniques for Intrusion Detection", Proceedings of RAID '98, Louvain-la-Neuve, Belgium, September 1998.

3. Axelsson, Stefan, "Research in Intrusion-Detection systems: A Survey". Technical Report 98--17, Dept. of Computer Eng. Chalmers Univ. of Tech, SE-412 96 Goteborg, Sweden, December 1998. URL: http://www.ce.chalmers.se/staff/sax.

4. Balasubramaniyan, Jai, Jose Omar Garcia-Fernandez, E. H. Spafford, and Diego Zamboni. "An Architecture for Intrusion Detection using Autonomous Agents." Department of Computer Sciences, Purdue University; Coast TR 98-05; 1998.

5. Bass, Tim, "Intrusion detection systems & multisensor data fusion: Creating cyberspace situational awareness." Communications of the ACM, April 2000. URL http://www.silkroad.com/papers/acm.fusion.ids.ps

6. Bass, Tim, "Multisensor data fusion for next generation distributed intrusion detection systems". In Proceedings, 1999 IRIS National Symposium on Sensor and Data Fusion, May 1999.

7. Bishop. Matt, "A Standard Audit Trail Format." In Proceedings of the 18th National Information Systems Security Conference, pages 136-145, Baltimore, Maryland, USA, October 10-13, 1995.

8. Crosbie, Mark, Spafford, Gene, "Defending a Computer System using Autonomous Agents", Technical report No. 95-022, COAST Laboratory, Department of Computer Sciences, Purdue University, March 1994.

9. Denning, Dorothy E., "An intrusion-detection model", IEEE Transactions on Software Engineering, vol. SE-13, pp. 222-232, February 1987.

10. Lee, Wenke, et. al.,"A Data Mining and {CIDF} Based Approach for Detecting Novel and Distributed Intrusions", Recent Advances in Intrusion Detection, pp 49-65, 2000.

11. Ming-Yuh Huang, Thomas M. Wicks, Robert J. Jasper,"A Large-scale Distributed Intrusion Detection Framework Based on Attack Strategy Analysis", Computer Networks, vol 31, no 23-24, pp2465-2475, 1999.

12. Thomas H. Ptacek and Timothy N. Newsham., "Insertion, Evasion, And Denial Of Service: Eluding Network Intrusion Detection," Technical Report, Secure Networks, Inc., January 1998.

13. Snapp S. R. , et al., "A system for distributed intrusion detection", Proceedings of the IEEE COMPCON 91, San Francisco, CA., February 1991.

14. Staniford-Chen, et. al., "GrIDS -- A Graph-Based Intrusion Detection System for Large Networks", The 19th National Information Systems Security Conference, Baltimore, MD., October 1996.

15. Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman, "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation", Lincoln Laboratory MIT, 244 Wood Street, Lexington, MA 02173-9108

Analyze This

Executive Summary

The GIAC Enterprises network has been frequently audited, and as result considerable improvements have been made to your security posture. This review has identified some new issues that require immediate attention. I will also point to previously identified problems that have not been corrected, and reference prior audits where appropriate to highlight these problem areas.

Significant attention should be paid to defining a security policy, and tightening the perimeter defenses of your network. Specific policy issues are gaming, mbone, and remote access. Your perimeter controls appear to be limited, as there is a large amount of traffic to and from several hosts on the Internet to your internal network; an assessment of what traffic is required to pass through the firewall should be performed and the firewall rules updated appropriately.

Description of dataset

The data provided for analysis was in 70 files generated by the Snort intrusion detection system: three were duplicates, 17 contained snort alerts based on signatures configured in the ID, 21 files containing out of spec (OOS) alerts, and 29 files contained scan alerts. The data spanned a time range from Jan 30 through Mar 10, 2001 - but data for a number of days was missing.

Duplicates	UMBCNI25.txt and UMBCNI31.txt UMBCNI3.txt and UMBCNI5.txt SnortA35.txt and SnortA36.txt
Alert Files	SnortA25.txt, SnortA3.txt, SnortA35.txt, SnortA6.txt, SnortAle.txt, UMBCNI27.txt, UMBCNI28.txt, UMBCNI3.txt, UMBCNI30.txt, UMBCNI32.txt, UMBCNI39.txt, UMBCNI41.txt, UMBCNI5.txt, UMBCNI52.txt, UMBCNI54.txt, UMBCNI58.txt, UMBCNI60.txt
OOSFiles	OOSche24.txt, OOSche26.txt, OOSche28.txt, OOSche29.txt, OOSche30.txt, OOSche31.txt, OOSche32.txt, OOSche33.txt, OOSche34.txt, OOSche4.txt, OOSche5.txt, OOScheck.txt, UMBCNI33.txt, UMBCNI37.txt, UMBCNI38.txt, UMBCNI42.txt, UMBCNI45.txt, UMBCNI48.txt, UMBCNI49.txt, UMBCNI50.txt, UMBCNI56.txt
Scan Files	SnortS2.txt, SnortS26.txt, SnortS27.txt, SnortS29.txt, SnortS32.txt, SnortS34.txt, SnortS7.txt, SnortS8.txt, SnortSca.txt, UMBCNI2.txt, UMBCNI25.txt, UMBCNI26.txt, UMBCNI29.txt, UMBCNI31.txt, UMBCNI34.txt, UMBCNI35.txt, UMBCNI36.txt, UMBCNI4.txt, UMBCNI40.txt, UMBCNI43.txt, UMBCNI44.txt, UMBCNI46.txt, UMBCNI47.txt, UMBCNI51.txt, UMBCNI53.txt, UMBCNI55.txt, UMBCNI57.txt, UMBCNI59.txt, UMBCNI61.txt

Alert Summary

After removing duplicate files, the alert data was analyzed using SnortSnarf v111500.1 giving the following results:

458703 alerts found among the files:
Earliest alert at **00:00:07**.303804 *on 01/30*
Latest alert at **23:51:59**.244669 *on 03/10*

Signature	# Alerts	Sources	Destinations
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	1	1	1
Russia Dynamo - SANS Flash 28-jul-00	1	1	1
Probable NMAP fingerprint attempt	2	2	2
Security 000516-1	4	2	2
TCP SMTP Source Port traffic	4	4	3
STATDX UDP attack	8	2	8
Back Orifice	25	2	25
ICMP SRC and DST outside network	79	22	17
SUNRPC highport access!	112	7	7
Null scan!	125	108	84
Tiny Fragments - Possible Hostile Activity	228	20	11
Queso fingerprint	431	57	106
WinGate 1080 Attempt	469	96	216
connect to 515 from inside	532	5	5
Attempted Sun RPC high port access	543	7	7
SMB Name Wildcard	729	307	425
SNMP public access	1155	4	8
External RPC call	1517	4	1466
TCP SRC and DST outside network	1714	62	102
NMAP TCP ping!	4817	12	3824
Watchlist 000222 NET-NCFC	5720	20	10
Possible RAMEN server activity	9851	2321	5031
SYN-FIN scan!	10499	7	9550
Watchlist 000220 IL-ISDNNET-990517	11874	49	71
UDP SRC and DST outside network	408263	660	1815

Ideally the alerts would be further categorized as: false positives, reconnaissance, attempted exploits, and successful exploits. Unfortunately due to the volume of data, I was unable to complete this categorization - instead dealing with the top 5 in each category. There is a risk that important events will be lost in the noise, and attempts should be made to update the snort ruleset to decrease the number of false positive alerts generated.

Alert Analysis

UDP SRC and DST outside network: 408263 alerts
TCP SRC and DST outside network: 1714 alerts

"SRC and DST outside network" alerts are triggered by network traffic with a source and destination that is not identified in the ID sensor (i.e. snort ruleset) as belonging to your network. This event was not reported in earlier evaluations or your network. I assume this represents a new rule added to the ID ruleset based on recommendations from these audits, rather than discovery of the mbone (multicast backbone).

There are several possible explanations for these alerts that depend on: where the snort sensor is installed, whether the snort configuration file includes all of the subnets in your network, Internet network traffic being routed through your network due to misconfiguration, dialup users with full time Internet network connections on the same computer (e.g. cable modem, ISDN), or misconfigured hosts (e.g. itinerant laptops, wireless devices, etc.).

In your case, these alerts all appear to be false positives caused by:
- Multicast traffic (mbone video conferencing, etc.),
- systems on your network without valid IP address assignments: using reserved IP addresses, or Windows 2000 systems with default IP address assignments (LINKLOCAL),
- misconfigured systems (e.g. laptops) connected to network,
- Backdoor network connections (e.g. dialup users with dedicated network links like cable modems),
- and game playing

Alert Breakdown:

Event Source	Event Count
Multicast Traffic: Netname: MCAST-NET Netblock: 224.0.0.0 - 239.255.255.255	395243 UDP alerts
Systems without IP address assignments: Netname: LINKLOCAL Netblock: 169.254.0.0 - 169.254.255.255	4858 UDP alerts
Reserved addresses: Netname: RESERVED-10 Netblock: 10.0.0.0 - 10.255.255.255 Netname: IANA-CBLK1 Netblock: 192.168.0.0 - 192.168.255.255 Netname: IANA-BBLK-RESERVED Netblock: 172.16.0.0 - 172.31.255.255 Netname: RESERVED-23 Netblock: 23.0.0.0 - 23.255.255.255	6630 UDP alerts

Event Source	Event Count
Potential backdoors (e.g. dialup user w/ cable modem), or misconfigured laptops connected to network. Destination ports: 53, 137, and 138.	1496 UDP alerts
Online role-playing game: Asheron's Call Destination port 9000 and 9004	44 UDP alerts

Recommendations

Identify hosts that are using IP addresses in the reserved address ranges, and if appropriate assign permanent addresses. Regardless, subnets using reserved addresses should be documented. Your security policy should identify the process, and controls, for connecting new hosts to the network - including IP address assignment.

To reduce the false positives generated by this snort rule, there are two actions required:
1. On internal ID sensors, add the multicast, reserved, and linklocal IP address ranges in the above table to the list of valid IP addresses on your network.
2. Block the reserved IP address ranges at the perimeter router or firewall.

The backdoor network connections due to @Home cable modem users dialing into your network is a bigger concern. I suggest placing a filtering router on the dialin lines, and permit traffic only to your network addresses from the DHCP/BOOTP assigned IP address of the dialin link. In addition, dialin users should be advised to disconnect cable modems, ensure that a virus scanner with up to date pattern files is installed, and consider the use of personal firewalls

The Internet Address Number Authority (IANA) has given the MBONE (which is largely used for teleconferencing) the Class D subset of 224.2.*.* . Mbone traffic has it's own set of risks. It uses UDP only, and no specific ports are reserved for it's use making it difficult to implement router filters. While it is unlikely that a virus, or trojan, will be introduced by using the mbone - it can use a significant ammount of network bandwidth. I suggest you configure a multicast router on a screened subnet and control access to this server.

Watchlist 000220 IL-ISDNNET-990517: 11874 alerts

This alert was generated in response to traffic to/from the ISDNNET network, an Israeli ISP. This network was identified in previous audits as a source of napster traffic, and network or port scans.

From whois.ripe.net:

inetnum: 212.179.0.0 - 212.179.255.255
netname: IL-ISDNNET-990517
descr: PROVIDER
country: IL (Israel)

Top 20 Source/Destination Port pairs:

This table shows that most of the traffic is related to the use of Napster. However a few connections are using unassigned ports which may be an indication of system compromise via trojans or backdoors. There was speculation in earlier audits that hosts from the ISDNNET had compromised some of your systems, so it may be prudent to block this network at the firewall and investigate the hosts that show communication on unknown ports.

Count	Source Port	Destination Port	Application
4061	1113	6688	Napster
2186	1172	6699	Napster
1246	63891	4718	Unknown
791	26835	6699	Napster
651	12708	6688	Napster
455	40109	4074	Unknown
436	21304	41003	Unknown
407	1546	6688	Napster
402	43313	6699	Napster
295	1073	6699	Napster
272	1572	6699	Napster
260	2226	6688	Napster
212	12701	6688	Napster
206	12693	4222	Unknown
196	63633	4718	Unknown
187	6699	2610	Napster
171	12702	6688	Napster
170	12587	6688	Napster
152	63255	6688	Napster
150	11124	6688	Napster

Top 10 Talkers

The Napster traffic is primarily inbound to your network. The following hosts should be evaluated for the existence of a Napster server or possible trojan:

Count	Source Host	Destination Host
4061	212.179.41.169	MY.NET.213.250
2186	212.179.21.179	MY.NET.207.226
1599	212.179.33.82	MY.NET.209.114
1442	212.179.125.114	MY.NET.207.126
791	212.179.72.226	MY.NET.220.42
615	212.179.79.2	MY.NET.222.2
436	212.179.44.62	MY.NET.210.34

Count	Source Host	Destination Host
413	212.179.29.250	MY.NET.217.42
407	212.179.41.14	MY.NET.225.50
402	212.179.79.2	MY.NET.217.206

In addition to the risks of unauthorized file sharing, and legal copyright issues - there are security flaws in some napster implementations as described in the following CVE entries:

Name	Description
CAN-2000-0281	** CANDIDATE (under review) ** Buffer overflow in the Napster client beta 5 allows remote attackers to cause a denial of service via a long message.
CAN-2000-0412	** CANDIDATE (under review) ** The gnapster and knapster clients for Napster do not properly restrict access only to MP3 files, which allows remote attackers to read arbitrary files from the client by specifying the full pathname for the file.

Recommendations

Establish policy for the use of napster, and other peer to peer file-sharing utilities, and block these ports on the firewall if possible.

Investigate systems reported using connections on unknown ports, they may be compromised and have backdoors installed.

Continue to monitor traffic from this network

SYN-FIN scan!: 10499 alerts

The SYN-FIN scan alert is triggered by hostile network scans using specially crafted packets that violate the TCP specification by setting both the SYN flag (used to initiate a TCP session) and the FIN flag (used to terminate a TCP session). The purpose of an SYN-FIN scan is to bypass a firewall, or to avoid logging of the scan attempt.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
130.234.184.112	9336	9336	8681	8681
128.61.136.233	1158	1159	1158	1159
24.50.25.5	1	1	1	1
4.35.4.244	1	2	1	1
66.25.174.123	1	1	1	1
128.206.176.25	1	1	1	1
209.255.180.130	1	1	1	1

Recommendations

Block non standard IP traffic at a perimeter firewall. All modern firewalls are capable of blocking these types of packets. This action will have many beneficial effects: reduce the likelihood of compromise from external sources, limit reconnaissance information available to hackers, and reduce the noise level in the ID system.

Possible RAMEN server activity : 9851 alerts

This alert is triggered by a TCP connection attempt to port 27374; the port used by the RAMEN worm to serve a copy of the code. The SubSeven trojan also uses Port 27374. This alert was not reported in earlier audits of your network.

The Ramen worm performs reconnaissance by scanning a range of Internet addresses and checking the FTP banner on machines with an active port 21 (ftp). Ramen compromised hosts are listening on port 27374, and will show an initial outbound connection to this port to download the trojan followed by incoming connections from newly compromised systems. Ramen exploits vulnerabilities in: wu-ftpd (port 21/tcp), rpc.statd (port 111/udp), and lprng (port 515/tcp).

For more information on the RAMEN worm, see:

> Houle, Kevin, 'CERT® Incident Note IN-2001-01: Widespread Compromises via "ramen" Toolkit', CERT Coordination Center, http://www.cert.org/incident_notes/IN-2001-01.html

Top 10 source/destination port pairs

Count	Source Port	Dest Port	Description of Activity
730	27374	4781	Inbound connection from compromised host
554	4781	27374	Inbound scans for compromised hosts
322	23	27374	inbound telnet sessions
210	27374	23	outbound telnet sessions
10	2154	27374	outbound from compromised host
6	2796	27374	outbound from compromised host
6	27374	3290	Inbound connection from compromised host
6	27374	2154	Inbound connection from compromised host
6	2493	27374	outbound from compromised host
5	4777	27374	outbound from compromised host

This is not normal traffic. There are connections inbound and outbound to port 27374 indicating systems compromised by the Ramen worm.

Top 10 talkers:

If any of the systems on MY.NET in this table are running the Linux operating system they should be considered at high risk of being compromised.

Count	Source Host	Dest Host
728	128.138.2.112	MY.NET.201.146
553	MY.NET.201.146	128.138.2.112
322	MY.NET.60.11	148.129.143.2
210	148.129.143.2	MY.NET.60.11
15	MY.NET.225.66	24.48.121.105
13	24.48.121.105	MY.NET.225.66
11	MY.NET.97.61	24.180.160.210
10	MY.NET.225.66	24.23.131.82
10	MY.NET.225.66	203.106.99.237
9	MY.NET.97.154	208.5.8.169

Alert traffic breakdown:

Unique local sources: 2130
Unique remote sources: 191
Unique local destinations: 332
Unique remote destinations: 4699

Signs of RAMEN compromise:

2 different signatures are present for 128.61.136.233 as a source
1 instances of SITE EXEC - Possible wu-ftpd exploit - GIAC000623
1158 instances of SYN-FIN scan!
Earliest: 16:07:53.847779 on 03/06
Latest: 16:44:02.658052 on 03/06

2 different signatures are present for MY.NET.219.22 as a destination
1 instances of SITE EXEC - Possible wu-ftpd exploit - GIAC000623
1 instances of SYN-FIN scan!
02/25-05:08:45.765154 [**] SYN-FIN scan! [**] 130.234.184.112:21-> MY.NET.219.22:21
03/06-16:44:02.658052 [**] SITE EXEC - Possible wu-ftpd exploit - GIAC000623 [**] 128.61.136.233:4705-> MY.NET.219.22:21

1 different signatures are present for MY.NET.219.22 as a source
2 instances of Possible RAMEN server activity
02/23-23:14:32.606383 [**] Possible RAMEN server activity [**] MY.NET.219.22:27374-> 24.67.186.244:3426
02/23-23:14:33.233285 [**] Possible RAMEN server activity [**] MY.NET.219.22:27374-> 24.67.186.244:3426

1 different signatures are present for 24.67.186.244 as a destination

1309 instances of Possible RAMEN server activity
Earliest: 22:57:56.786139 on 02/23
Latest: 23:03:40.770201 on 02/23

1 different signatures are present for 24.67.186.244 as a source
2438 instances of Possible RAMEN server activity
Earliest: 22:57:57.027022 on 02/23
Latest: 22:58:27.483022 on 02/23

Recommendations

While MY.NET.219.22 has not been identified as originating scans, some of the data is missing - this host should be considered at high risk of being compromised.

Use updated snort rules to reduce false positives:

> alert TCP $EXTERNAL any -> $INTERNAL 27374 (msg: "IDS460/worm-ramen-asp-retrieval-incoming"; flags: A+; content: "GET "; depth: 8; nocase;)
>
> alert TCP $INTERNAL any -> $EXTERNAL 27374 (msg: "IDS461/worm-ramen-asp-retrieval-outgoing"; flags: A+; content: "GET "; depth: 8; nocase;)

Perform a scan of your network TCP port 27374 to see if any RAMEN servers can be identified.

Patch all linux systems to remove current vulnerabilities, and establish an ongoing program to keep operating systems patches up to date.

Evaluate if the telnet traffic is appropriate, and block it at the firewall if possible.

Watchlist 000222 NET-NCFC: 5720 alerts

A match on the subnets assigned to NCFC, a Chinese school, generated this alert. This network has been previously identified in earlier audits as a source of SMTP traffic, but no significant security incidents were identified, and the alert count has not changed significantly.

From whois.arin.net:

> The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
> Netname: NCFC
> Netblock: 159.226.0.0 - 159.226.255.255

Top 5 Destination ports

Count	Port	Port Description

Count	Port	Port Description
4738	25	SMTP - email traffic
281	23	TELNET, also used by trojans (e.g. Blade Runner)
6	113	AUTH, also used by trojans (e.g. Kazimas)
1	9157	Back connections from SMTP session
1	804	Back connections from SMTP session

Top 5 NET-NCFC

Top 5 Destination Addresses		Top 5 source IP addresses	
Count	**Host**	**Count**	**Host**
5338	MY.NET.6.47	5362	159.226.81.1
203	MY.NET.6.7	170	159.226.45.204
80	MY.NET.60.11	111	159.226.45.108
43	MY.NET.253.43	35	159.226.39.4
36	MY.NET.100.230	10	159.226.210.6

MY.NET hosts with inbound SMTP traffic:

Count	Host
4659	MY.NET.6.47
39	MY.NET.253.43
33	MY.NET.100.230
6	MY.NET.253.41
1	MY.NET.6.34

MY.NET hosts with inbound TELNET traffic:

Count	Host
201	MY.NET.6.7
80	MY.NET.60.11

Most of this traffic appears to be normal SMTP mail traffic to host MY.NET.6.47, with alerts generated for SMTP (port 25) connections and AUTH (port 113) replies.

Recommendations

Verify that inbound telnet traffic to MY.NET.6.7 and MY.NET.60.11 is appropriate, and that the hosts listed in the above table of inbound SMTP connections are valid mail servers.

AUTH is commonly used by sendmail to determine the userid on the remote host making the connection. This service is inherently unreliable and should be disabled.

Setup a common mail gateway on a screened subnet, and permit mail traffic only to that host and not directly to/from the Internet. Virus scanning would be useful on the mail gateway.

Implement anti-spam measures such as: rbl (real time black list), domain name validation, and reverse IP address lookup verification with hostname.

Top 10 Sources of Alerts

After excluding false positives from the "SRC and DST outside network" alerts, the top 10 sources are:

Host	Count	Alerts
130.234.184.112	9336	9336 instances of "SYN-FIN scan!" to port 21 between 02/25 04:50:13 and 04:50:33
159.226.81.1	5362	5362 instances of "Watchlist 000222 NET-NCFC" between 02/11 05:44:46 and 05:53:01
MY.NET.70.38	4788	2 instances of SUNRPC highport access! 4786 instances of NMAP TCP ping! Between 02/20 00:00:46 and 02/23 13:56:55
212.179.41.169	4061	4061 instances of "Watchlist 000220 IL-ISDNNET-990517" between 03/10 19:01:45 and 21:35:46
24.67.186.244	2438	2438 instances of "Possible RAMEN server activity" between 02/23 22:57:57 and 22:58:27
24.48.226.183	1819	1074 instances of "Possible RAMEN server activity" between 02/11 23:03:20 and 23:20:50
212.179.33.82	1599	1599 instances of "Watchlist 000220 IL-ISDNNET-990517" on 02/28 08:45:15and 08:51:19
212.179.125.114	1444	1444 instances of "Watchlist 000220 IL-ISDNNET-990517" between 02/03 00:10:36 and 02/28 09:01:15
171.65.61.201	1274	7 instances of STATDX UDP attack 1267 instances of External RPC call between 02/20 19:41:05 and 19:50:27
128.61.136.233	1159	1 instances of SITE EXEC - Possible wu-ftpd exploit - GIAC000623 1158 instances of SYN-FIN scan! Between 03/06 16:07:53 and 16:44;02

Recommendations

Investigate systems reporting scan and attack alerts for possible compromise.

Implement a statefull perimeter firewall and consider blocking the top source IP addresses at the perimeter.

Top 10 Alert Destinations

After excluding false positives from the "SRC and DST outside network" alerts, the top 10 destinations are:

Host	Count	Status
MY.NET.6.47	5339	1 instances of SYN-FIN scan! 5338 instances of Watchlist 000222 NET-NCFC Earliest: 05:36:19.191114 on 02/11 Latest: 05:44:41.848940 on 02/11
MY.NET.213.250	4069	1 instances of Possible RAMEN server activity 4068 instances of Watchlist 000220 IL-ISDNNET-990517 Earliest: 23:14:07.765325 on 02/23 Latest: 21:35:46.336712 on 03/10
MY.NET.209.114	1599	1599 instances of Watchlist 000220 IL-ISDNNET-990517 Earliest: 08:45:15.767853 on 02/28 Latest: 08:45:36.029905 on 02/28
MY.NET.207.126	1451	1451 instances of Watchlist 000220 IL-ISDNNET-990517 Earliest: 01:28:14.213171 on 02/28 Latest: 01:43:02.109846 on 02/28
24.67.186.244	1309	2438 instances of "Possible RAMEN server activity" Earliest : 22:57:57 on 02/23 Latest: 22:58:27 on 02/23
24.48.226.183	1074	1074 instances of Possible RAMEN server activity Earliest: 23:03:20.894861 on 02/11 Latest: 23:20:50.063477 on 02/11
MY.NET.100.99	872	872 instances of SNMP public access Earliest: 11:56:55.782297 on 02/22 Latest: 16:30:30.082391 on 02/22
MY.NET.220.42	792	1 instances of SYN-FIN scan! 791 instances of Watchlist 000220 IL-ISDNNET-990517 Earliest: 05:21:30.936583 on 02/25 Latest: 09:13:58.963515 on 02/25
MY.NET.201.146	730	2 instances of Queso fingerprint 728 instances of Possible RAMEN server activity Earliest: 21:29:09.380432 on 02/03 Latest: 03:17:51.362339 on 02/23
MY.NET.222.2	619	619 instances of Watchlist 000220 IL-ISDNNET-990517 Earliest: 05:58:46.130907 on 02/20 Latest: 14:36:48.139051 on 03/06

Recommendations

Investigate systems reporting scan and attack alerts for possible compromise.

Implement a stateful firewall to block network scanning, and examine MY.NET hosts that have received a significant amount of external attention for possible compromise.

OOS Alerts

Top 5 sources

Host	Count	Status
129.104.19.94	11044	SYN-FIN scan of a large portion of your net
64.0.153.38	3665	SYN-FIN scan of hosts on your net
128.61.136.233	2967	SYN-FIN scan of your network
62.119.119.3	2276	Scan to port 317 with both reserved bits
MY.NET.217.150	2108	Packets to Internet hosts with odd TCP flags: 1SFRP, 21FR, 2SFRPAU, 21S.

Top 10 port pairs

Count	Source Port	Destination Port	Status
11045	109	109	Host 129.104.19.94 performing SF scan
8424	21	21	Host 64.0.153.38 performing SF scan Host 128.61.136.233 SF scan
2975	53	53	Network Scans
167	0	6346	Network Scans
148	0	2340	Network Scans
52	30973	50632	Network Scans

Recommendations

Investigate MY.NET.217.150. This may be a broken network or system.

Block scans, and out of spec. IP packets, at the perimeter firewall.

Analysis of Scan File Data

The scan files contained 1,191,592 alerts.

Top 10 scan source and destination:

Top 10 Sources		Top 10 Destinations	
Count	Host	Count	Host
34517	MY.NET.218.90	21060	129.2.246.94
24185	MY.NET.150.220	9995	MY.NET.160.109
21060	MY.NET.221.26	8814	MY.NET.60.8
20040	MY.NET.204.66	4079	216.155.34.54
19785	MY.NET.229.154	3879	169.197.49.83
15606	MY.NET.70.38	3032	MY.NET.218.86

Top 10 Sources		Top 10 Destinations	
Count	**Host**	**Count**	**Host**
15130	MY.NET.150.133	2341	24.157.10.197
14282	MY.NET.202.50	2180	63.71.84.102
12721	MY.NET.227.254	2172	24.156.151.85
12129	169.226.202.234	2112	24.21.239.107

Top 10 Talkers

Count	Source	Destination
21060	MY.NET.221.26	129.2.246.94
9992	206.112.192.106	MY.NET.160.109
8642	24.141.226.62	MY.NET.60.8
4079	MY.NET.208.10	216.155.34.54
3028	24.4.196.167	MY.NET.218.86
2180	MY.NET.179.78	63.71.84.102
2041	MY.NET.218.90	216.19.133.116
2012	MY.NET.218.90	172.132.71.130
1833	MY.NET.218.90	24.91.199.203
1815	MY.NET.179.78	63.71.84.104

Further examination of the scan data showed that a large portion of the scan traffic had identical source and destination ports. An analysis of the scan data for the top 10 revealed:

Port	Total Count	# Sources		# Destinations	
		Local	**Remote**	**Local**	**Remote**
28800	122066	55	0	0	16692
13139	51948	113	0	0	25519
0	36163	46	45	26	796
6112	32518	113	0	0	4911
21	29713	1	9	21252	2
53	9389	0	7	7751	0
9001	6723	19	0	0	5760
3283	3322	0	1	1157	0
5232	1631	0	2	1616	0
54321	1392	0	1	1392	0

The ports in use by these scans were identified as:

Port	
28800	MSN Gaming Zone. Host Directplay (GameZone, Mplayer, Boneyards)
13139	Unknown
0	Unused (probable hostile scan)

6112	Unknown
21	FTP
53	DNS
9001	KastenX Pipe
3283	Unknown
5232	Unknown
54321	Back Orifice 2000, School Bus

MY.NET.224.50	Large amount of traffic on source port 28800 and destination port 28800 to many external IP addresses. Probable MSN game server.
MY.NET.203.214	Large amount of traffic on source port 6112 and destination port 6112 to many external IP addresses. Probable Diablo game server.

Recommendations

Determine if policy permits game servers on the network, and block game ports at the firewall if possible.

Explanation of Analysis Techniques

Alert summaries were generated using SnortSnarf v111500.1. To count the total alerts all of the data was combined in one file for analysis and the string "MY.NET" replaced with a numeric to permit SnortSnarf to generate useful summaries. Due to the large volume of data, the analysis was run on a machine with 4GB memory, and the process memory limits set to 4 GB.

Identification of duplicate files. I used the unix commands *diff* and *cksum* to evaluate the files:

```
denali> diff UMBCNI25.txt UMBCNI31.txt
denali> cksum UMBCNI25.txt UMBCNI31.txt
3509280910 2115215 UMBCNI25.txt
3509280910 2115215 UMBCNI31.txt

denali> diff UMBCNI3.txt UMBCNI5.txt
denali> cksum UMBCNI3.txt UMBCNI5.txt
3163670079 4956801 UMBCNI3.txt
3163670079 4956801 UMBCNI5.txt

denali> diff SnortA35.txt SnortA36.txt
denali> cksum SnortA35.txt SnortA36.txt
812592916 4492808 SnortA35.txt
812592916 4492808 SnortA36.txt
```

Generating TCP quad format data from alert files:

```
#!/bin/perl
$ip='([0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-
9]{1,3}|MY.NET.[0-9]{1,3}.[0-9]{1,3})';
while (<>) {
  if (/[\*\*].*[\*\*] ($ip):([0-9]*) \-\> ($ip):([0-9]*)/)
{
    print "$1 $3 $4 $6\n";
 }
}
```

Top 10 lists were created from quad data using *awk*, *sort*, *uniq*, and *grep*:

```
# awk '{print $1}'<quaddata|sort|uniq -c|sort -rn|head -10
```

Summary

There are five areas that should be investigated: installation of a statefull firewall, creation of a screened subnet for managing services requiring Internet connectivity, policy defining what access to the Internet is permitted, ongoing problems with loss of ID alert data, and investigate potentially compromised systems.

A properly configured statefull firewall will block many of the scans that were identified in the snort ID data. A screened subnet on the firewall should be created to support the following services: mbone (multicast router), mail hub, and proxies for services that you decide will be permitted through the firewall (e.g. telnet). No direct traffic should be permitted through the firewall; it should be proxied or routed through an intermediary on a screened subnet.

Your security policy needs to address which services you will permit to the Internet. Services that are a potential concern are: games, napster, inbound telnet, and remote access.

After these issues have been addressed, I suggest you have another audit to determine how successful the improvement activities were at fixing the problems.

Last Updated: April 21st, 2018

Upcoming Training

SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201805,	May 02, 2018 - Jun 14, 2018	vLive
Community SANS Virginia Beach SEC503	Virginia Beach, VA	May 07, 2018 - May 12, 2018	Community SANS
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
Mentor Session - SEC503	Houston, TX	Jun 18, 2018 - Jul 18, 2018	Mentor
SANS Oslo June 2018	Oslo, Norway	Jun 18, 2018 - Jun 23, 2018	Live Event
Minneapolis 2018 - SEC503: Intrusion Detection In-Depth	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	vLive
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS London July 2018	London, United Kingdom	Jul 02, 2018 - Jul 07, 2018	Live Event
SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LA	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS San Antonio 2018	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	Live Event
San Antonio 2018 - SEC503: Intrusion Detection In-Depth	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	vLive
Community SANS Columbia SEC503	Columbia, MD	Aug 13, 2018 - Aug 18, 2018	Community SANS
SANS Virginia Beach 2018	Virginia Beach, VA	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, Netherlands	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, Japan	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS London September 2018	London, United Kingdom	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NV	Sep 23, 2018 - Sep 30, 2018	Live Event
SANS Brussels October 2018	Brussels, Belgium	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Northern VA Fall- Tysons 2018	Tysons, VA	Oct 13, 2018 - Oct 20, 2018	Live Event
SANS Denver 2018	Denver, CO	Oct 15, 2018 - Oct 20, 2018	Live Event
SANS October Singapore 2018	Singapore, Singapore	Oct 15, 2018 - Oct 28, 2018	Live Event
Mentor Session - SEC503	Ballston, VA	Nov 01, 2018 - Dec 06, 2018	Mentor
SANS Dallas Fall 2018	Dallas, TX	Nov 05, 2018 - Nov 10, 2018	Live Event
SANS San Diego Fall 2018	San Diego, CA	Nov 12, 2018 - Nov 17, 2018	Live Event
SANS Stockholm 2018	Stockholm, Sweden	Nov 26, 2018 - Dec 01, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced