



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

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

## Interested in learning more?

Check out the list of upcoming events offering  
"Intrusion Detection In-Depth (Security 503)"  
at <http://www.giac.org/registration/gcia>

\*\*\* Northcutt, I know Gao worked really hard on this and the work shows it. ESL grading is on for this one. The accuracy is fine, though I am not sure 5 is fingerprinting, kinda hard to say with a single packet, I would classify this as an out of spec. 6 looks a bit more like a long day in cablemodem land. 75 \*

## 10 Detects for SANS GIAC Intrusion Analyst Certification

Gao ZhiXing  
April 20, 2000

The practical analyses is to fulfill the practical requirements for the SANS GIAC Intrusion Analyst Certification.

### Trace 1

This detect is taken from the GIAC page, 18-Apr-2000.

```
Apr 17 07:01:09.188513 www.olech.pl.0 > .edu.sunrpc: SF 3426811904:3426811904(0) win 512 (ttl 232, id 46593)
Apr 17 07:01:09.201824 www.olech.pl.0 > .edu.sunrpc: SF 3426811904:3426811904(0) win 512 (ttl 232, id 48129)
```

**Type:** SF+port0+sunrpc scan

**Active targeting:** Yes

**History:** Unknown

**Analysis:**

Source port 0, TCP flag SF and unchanged sequence number indicate they are crafted packets probably generated by automated tool. They are not duplicated packets because of different id. It looks like a SYN-FIN-Port-0 scan and may be used for OS fingerprinting. The attacker also seems have interest in whether port 111 is alive. Scanning port 111 normally is the first step in scanning a system looking for all the RPC services. RPC is vulnerable protocol and those services based on RPC such as NFS are the hacker's favor. I guess the hacker is looking for linux machine and from hacker point of view, this combination of port 0+SF+portmapper is quite efficient.

The source address exists and belongs to a telecommunication company.

**Severity:** The severity of this attack is medium.

### Trace 2

This detect is taken from the GIAC page, 18-Apr-2000.

```
Apr 16 02:22:06 picard tcplog: pop3 connection attempt from 207.61.128.71
Apr 16 02:23:31 medusa tcplog: pop2 connection attempt from 207.61.128.71
Apr 16 02:24:45 medusa tcplog: pop3 connection attempt from 207.61.128.71
Apr 16 02:22:06 bigfoot tcplog: pop2 connection attempt from 207.61.128.71
Apr 16 02:22:06 bigfoot tcplog: pop3 connection attempt from 207.61.128.71
Along with these TCP connect attempts appears to be a LOT of UDP attempts too...but only on medusa (our
secondary name server)
Apr 16 02:23:32 medusa icmplog: destination
unreachable from medusa.csihq.com to medusa.csihq.com
```

Michael D. Black

**Type: POP scan**

**Active targeting:** Yes

**History:** Unknown

**Analysis:**

The short time interval indicates this maybe a scripted attack. POP2 and POP3 are used for mail clients to access server, there are many holes in these services, and many implementations are vulnerable to buffer overflow attack. I do not know which one is his real mail server, but attacker appears to have strong interest in those machines. The attack address exists.

**Severity:** Severity should be medium.

### Trace 3

This detect is taken from the GIAC page, 17-Apr-2000.

The date is 0412 (MMDD) and the filter is tcp and dst port 20034

```
18:45:15.283384 pop09-1-ras1-p178.barak.net.il.2478 > @home.com.20034:
S 2633956517:2633956517(0) win 8760 <mss 1460,nop,nop,sackOK> (DF)
20:56:53.204028 212.68.154.152.1170 > @home.com.20034:
S 5881092:5881092(0) win 8192 <mss 1460> (DF)
20:56:54.189287 212.68.154.152.1170 > @home.com.20034:
S 5881092:5881092(0) win 8192 <mss 1460> (DF)
20:56:55.171289 212.68.154.152.1170 > @home.com.20034:
S 5881092:5881092(0) win 8192 <mss 1460> (DF)
20:56:56.162634 212.68.154.152.1170 > @home.com.20034:
S 5881092:5881092(0) win 8192 <mss 1460> (DF)
```

**Type: Netbus scan**

**Active targeting:** Yes

**History:** Unknown

**Analysis:**

These packets consists of two activities. The second activities (packet 2-5) probably are crafted packets generated by automated tool because of even time space and unchanged timestamp.

Both of the activities are looking for trojan horse Netbus which is running in port 20034. The destination address looks like at home, adding a firewall is definitely helpful to those home user.

**Severity:** The severity could be either low to medium depends on whether he has protection.

### Trace 4

This detect is taken from the GIAC page, 17-Apr-2000.

Type, date, time, source, destination, transport

```
FWIN,2000/04/16,10:44:46 -5:00 GMT,210.68.177.120:13365,x.x.x.x:98,TCP
```

**Type: Linuxconf scan**

**Active targeting:** Yes

**History:** Only this packet is provided by reporter.

**Analysis:**

Linuxconf has a web-enabled interface at port 98 through a integrated HTTP server. It has a number of security issues and may vulnerable to many of buffer overflow or HTTP attacks. The source address is in Taiwan.

**Severity:** low or medium

## Trace 5

This detect is taken from the GIAC page, 17-Apr-2000.

```
04/14-06:21:23.402593 MY.NET.202.98:0 -> 207.172.3.46:3194
TCP TTL:126 TOS:0x0 ID:56306 DF
2*SF**A* Seq: 0x770335 Ack: 0x643DFA07 Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 2424 3031 3233
3435 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

**Type:** OS fingerprinting

**Active targeting:** Yes

**History:** Unknown

**Analysis:**

Notice prot 0 and the illegal TCP flags, and according to RFC 793 and RFC 1323, that TCP option also is not legal. Because TCP 3194 is not a well-known port. I think this packet is used for OS fingerprinting. The combination of port 0+illegal\_TCP\_flag+illegal\_TCP\_option is most possibly used to trigger more different types of responses from different OS.

**Severity:** I think the severity could be low or medium.

## Trace 6

This detect and the one that follows are from the GIAC page, 17-Apr-2000.

```
Apr 13 23:45:55 cc1014244-a kernel: securityalert: tcp if=ef0 from
24.200.32.216:4988 to 24.3.21.199 on unserved port 27374
Apr 14 00:26:09 cc1014244-a kernel: securityalert: udp if=ef0 from
24.24.101.170:1699 to 24.3.21.199 on unserved port 137
Apr 14 01:04:58 cc1014244-a kernel: securityalert: tcp if=ef0 from
63.77.199.123:3690 to 24.3.21.199 on unserved port 111
Apr 14 18:10:24 cc1014244-a kernel: securityalert: tcp if=ef0 from
142.165.185.81:4462 to 24.3.21.199 on unserved port 1243
Apr 14 19:46:00 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:1085 to 24.3.21.199 on unserved port 2
Apr 15 00:47:48 cc1014244-a kernel: securityalert: udp if=ef0 from
24.1.97.9:1029 to 24.3.21.199 on unserved port 9200
Apr 15 07:36:54 cc1014244-a kernel: securityalert: tcp if=ef0 from
63.21.198.242:2806 to 24.3.21.199 on unserved port 1080
Apr 15 12:25:38 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:4891 to 24.3.21.199 on unserved port 5632
```

**Type:** Coordinated attack?

**Active targeting:** Yes

**History:** Unknown

**Analysis:**

Those are “hot” ports to hackers. TCP/27374 and TCP/1243 for SubSeven, UDP/137 for NetBIOS, TCP/111 for portmap, TCP/1080 for socks, UDP/22 and TCP/5632 for PCAnywhere, and UDP/9200 is unknown.

I was shocked when I first saw this trace. **Could it be a coordinated attack?** I do not know whether all captured packets are listed here. But notice the firewall log, only one or two malicious packets were sent in 2 days time (quite stealth way) from each host, and there was no redundant scan. There were two packets from same host 24.3.21.225 and all targeted at PCAnywhere, but their time interval was 15 hours. That gives me strong impression, **they were intentionally trying to hide something.**

**Severity:** Severity should be medium to high.

## Trace 7

```
22:11:06.789808 somewhere.com > my.com: (frag 46914:1480@1480+)
22:11:06.791040 somewhere.com > my.com: (frag 46915:1480@2960+)
22:11:06.792272 somewhere.com > my.com: (frag 46916:1480@4440+)
22:11:06.793504 somewhere.com > my.com: (frag 46917:1480@5920+)
22:11:06.794744 somewhere.com > my.com: (frag 46918:1480@7400+)
22:11:06.796043 somewhere.com > my.com: (frag 46919:1480@8880+)
22:11:06.797280 somewhere.com > my.com: (frag 46920:1480@10360+)
22:11:06.798476 somewhere.com > my.com: (frag 46921:1480@11840+)
22:11:06.799710 somewhere.com > my.com: (frag 46922:1480@13320+)
22:11:06.800932 somewhere.com > my.com: (frag 46923:1480@14800+)
22:11:06.802211 somewhere.com > my.com: (frag 46924:1480@16280+)
22:11:06.803431 somewhere.com > my.com: (frag 46925:1480@17760+)
22:11:06.803431 somewhere.com > my.com: (frag 46925:1480@17760+)
22:11:06.804634 somewhere.com > my.com: (frag 46926:1480@19240+)
22:11:06.805866 somewhere.com > my.com: (frag 46927:1480@20720+)
22:11:06.807091 somewhere.com > my.com: (frag 46928:1480@22200+)
22:11:06.808356 somewhere.com > my.com: (frag 46929:1480@23680+)
22:11:06.809594 somewhere.com > my.com: (frag 46930:1480@25160+)
22:11:06.810812 somewhere.com > my.com: (frag 46931:1480@26640+)
22:11:06.812039 somewhere.com > my.com: (frag 46932:1480@28120+)
22:11:06.813240 somewhere.com > my.com: (frag 46933:1480@29600+)
22:11:06.814538 somewhere.com > my.com: (frag 46934:1480@31080+)
22:11:06.815758 somewhere.com > my.com: (frag 46935:1480@32560+)
22:11:06.816943 somewhere.com > my.com: (frag 46936:1480@34040+)
22:11:06.818148 somewhere.com > my.com: (frag 46937:1480@35520+)
22:11:06.819379 somewhere.com > my.com: (frag 46938:1480@37000+)
22:11:06.820664 somewhere.com > my.com: (frag 46939:1480@38480+)
22:11:06.821894 somewhere.com > my.com: (frag 46940:1480@39960+)
22:11:06.823138 somewhere.com > my.com: (frag 46941:1480@41440+)
22:11:06.824898 somewhere.com > my.com: (frag 46942:1480@42920+)
22:11:06.825562 somewhere.com > my.com: (frag 46943:1480@44400+)
22:11:06.826842 somewhere.com > my.com: (frag 46944:1480@45880+)
22:11:06.828047 somewhere.com > my.com: (frag 46945:1480@47360+)
22:11:06.829287 somewhere.com > my.com: (frag 46946:1480@48840+)
22:11:06.830547 somewhere.com > my.com: (frag 46947:1480@50320+)
22:11:06.831731 somewhere.com > my.com: (frag 46948:1480@51800+)
22:11:06.833007 somewhere.com > my.com: (frag 46949:1480@53280+)
22:11:06.834217 somewhere.com > my.com: (frag 46950:1480@54760+)
22:11:06.835441 somewhere.com > my.com: (frag 46951:1480@56240+)
22:11:06.836716 somewhere.com > my.com: (frag 46952:1480@57720+)
```

```
22:11:06.837894 somewhere.com > my.com: (frag 46953:1480@59200+)
22:11:06.839176 somewhere.com > my.com: (frag 46954:1480@60680+)
22:11:06.840378 somewhere.com > my.com: (frag 46955:1480@62160+)
22:11:06.841603 somewhere.com > my.com: (frag 46956:1480@63640+)
22:11:06.842863 somewhere.com > my.com: (frag 46957:1480@65120)
```

### **Type: Ping of Death**

**Active targeting:** Yes

**History:** Found in old traffic log

### **Analysis:**

I found this ping of death attack in my old traffic logs. It is quite interesting. This exploit weakness of fragmentation. It should be an outdated technique, since most of new OS are patched.

**Severity:** Low

## **Trace 8 and 9**

This detect is taken from the GIAC page, 13-Apr-2000.

DIAL-A-MATTRESS, NEW YORK NY, USA

```
Apr 11 18:21:38 dns1 portsentry[438328]: attackalert:
Connect from host:12.20.24.133/12.20.24.133 to TCP port: 1524
Apr 11 18:21:40 dns2 portsentry[2259]: attackalert:
Connect from host:12.20.24.133/12.20.24.133 to TCP port: 1524
Apr 11 18:21:40 dns3 portsentry[6017]: attackalert:
Connect from host:12.20.24.133/12.20.24.133 to TCP port: 1524
```

This detect is taken from the GIAC page, 15-Apr-2000.

DIAL-A-MATTRESS, NEW YORK NY, USA

First was scanning 1524, now scanning 600.....

```
Apr 12 20:53:58 hosth snort[87556]: spp_portscan:
PORTSCAN DETECTED from 12.20.24.133
Apr 12 20:54:04 hosth snort[87556]: spp_portscan:
portscan status from 12.20.24.133: 22 connections across 22 hosts:
TCP(22), UDP(0)
Apr 12 20:54:10 hosth snort[87556]: spp_portscan:
portscan status from 12.20.24.133: 4 connections across 4 hosts:
TCP(4), UDP(0)
Apr 12 20:54:16 hosth snort[87556]: spp_portscan:
End of portscan from 12.20.24.133
-----
Apr 12 20:53:56 12.20.24.133:4973 -> a.b.e.13:600 SYN **S*****
Apr 12 20:54:01 12.20.24.133:1054 -> a.b.e.51:600 SYN **S*****
Apr 12 20:53:58 12.20.24.133:1060 -> a.b.e.58:600 SYN **S*****
Apr 12 20:53:58 12.20.24.133:1066 -> a.b.e.63:600 SYN **S*****
Apr 12 20:53:58 12.20.24.133:1076 -> a.b.e.73:600 SYN **S*****
Apr 12 20:53:58 12.20.24.133:1082 -> a.b.e.79:600 SYN **S*****
Apr 12 20:53:58 12.20.24.133:1086 -> a.b.e.83:600 SYN **S*****
Apr 12 20:53:58 12.20.24.133:1091 -> a.b.e.88:600 SYN **S*****
Apr 12 20:53:58 12.20.24.133:1094 -> a.b.e.91:600 SYN **S*****
```

etc.....

**Type: Backdoor scan**

**Active targeting:** Yes

**History:** Unknown

**Analysis:**

I found second trace(in GIAC 15/04) first, because the reporter mentioned about previous attack, so I search the web manually and found it in (GIAC 13/04).

The attack is quick and should be scripted scan. TCP 1524 is the ingreslock backdoor, many attack scripts install a backdoor shell at this port, especially those against Sun systems via holes in sendmail and RPC services like in statd, ttdbserver and cmsd(refers to

[www.robertgraham.com/pubs/firewall-seen.html](http://www.robertgraham.com/pubs/firewall-seen.html)). TCP/600 is pserver backdoor has the same problem as ingreslock. Notice two attacks are launched by same IP address which is assigned to PC WARE International. I do don't have details about first attack, but from second trace we know the attacker should already have the knowledge about target network.

**Severity:** Medium

**Trace 10**

```
[**] spp_portscan: PORTSCAN DETECTED from 10.1.1.1 [**]
```

```
04/14-17:43:17.847024
```

```
[**] XMAS Scan [**]
```

```
04/14-17:43:17.846194 10.1.1.1:60987 -> x.y.z.77:583
```

```
TCP TTL:50 TOS:0x0 ID:14355
```

```
***F*P*U Seq: 0x0 Ack: 0x0 Win: 0xC00
```

```
[**] XMAS Scan [**]
```

```
04/14-17:43:17.848268 10.1.1.1:60987 -> x.y.z.77:24
```

```
TCP TTL:50 TOS:0x0 ID:8273
```

```
***F*P*U Seq: 0x0 Ack: 0x0 Win: 0xC00
```

```
[**] XMAS Scan [**]
```

```
04/14-17:43:17.849045 10.1.1.1:60987 -> x.y.z.77:432
```

```
TCP TTL:50 TOS:0x0 ID:30089
```

```
***F*P*U Seq: 0x0 Ack: 0x0 Win: 0xC00
```

```
[**] XMAS Scan [**]
```

```
04/14-17:43:17.849809 10.1.1.1:60987 -> x.y.z.77:517
```

```
TCP TTL:50 TOS:0x0 ID:27229
```

```
***F*P*U Seq: 0x0 Ack: 0x0 Win: 0xC00
```

```
<skipped>
```

**Type: XMAS scan**

**Active targeting:** Yes

**History:** captured from intranet.

**Analysis:** The attack captured by snort. The close timestamp, illegal TCP flag, and unchanged source port number indicate that these packets are generated by automated tool(perhaps nmap). The attacker has interest in the target host but he seems inexperienced and does the work in an inefficient way.

**Severity:** Low

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



# Upcoming Training

Click Here to  
**{Get CERTIFIED!}**



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