Global Information Assurance Certification Paper

Level II Intrusion Detection GCIA Practical Assignment – Version 2.7

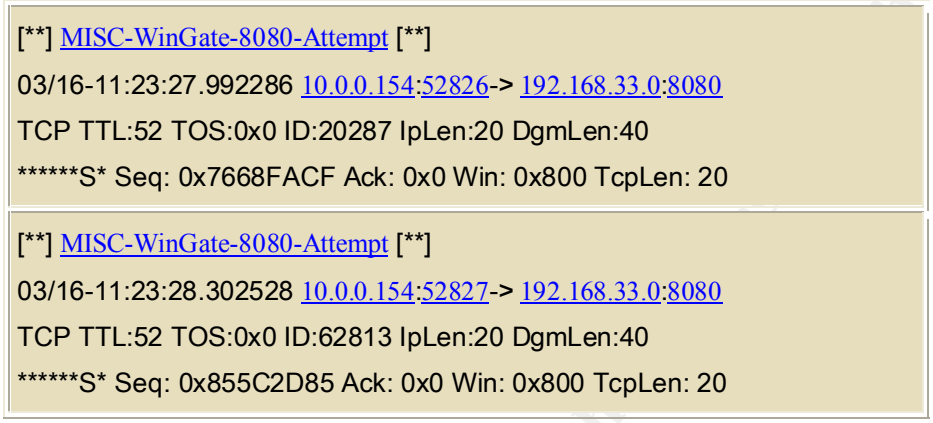
Submission by: John Garris

Table of Contents

Assignment 1 – Network Detects

Trace #1

1. Source of Trace



These logs were generated in a test network running multiple OSes (MS Win2000, MS WinNT4.0, Linux, and Solaris).

2. Detect was Generated By

Detects were generated by SNORT-1.7. HTML output obtained from SnortSnarf.

3. Probability the Source Address was Spoofed

Since this traffic was generated on a test network, I know for a fact it was not spoofed. However, it would be unlikely that "real world" attempts to connect to a WinGate server would be spoofed. This detect captured an attempt to determine if a WinGate server was available on 192.168.33.0. WinGate is a product that enables small offices to share a network interface to the Internet. WinGate servers, when configured to allow unauthenticated logins, are a favorite launching point for hackers. This allows hackers to mask their true point of origin by looping through these servers. Spoofing their source IP would not allow them to see a SYN-ACK response to their initial request, thus defeating the purpose of the probe.

4. Description of Attack

As noted above, this detect logged an attempt to determine if a particular server (192.168.33.0) was running WinGate software. Depending on the version and configuration of the WinGate server, it may return valuable reconnaissance information. The next step would likely be an attempt to login without password

authentication. There are other possible attacks against WinGate servers, as noted in the correlations below.

5. Attack Mechanism

This particular detect relates to a straightforward probe to determine if a WinGate server is available at a particular IP. Assuming this connection came from a site not authorized to connect to the server, the reconnaissance could be for a number of reasons; most likely to determine if the server will allow unauthorized access. However, earlier versions of WinGate were susceptible to Denial-of-Service and password stealing attacks (Source: ntbugtraq/1999/April1999/0006.html)

6. Correlations

Below is a listing of the Common Vulnerabilities and Exposures related to WinGate Servers. (Source: cve.mitre.org)

Name	Description
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-1999-0291	The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.
CVE-1999-0441	Remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service.
CAN-1999-0657	** CANDIDATE (under review) ** WinGate is being used.
CAN-2000-1048	** CANDIDATE (under review) ** Directory traversal vulnerability in the logfile service of Wingate 4.1 Beta A and earlier allows remote attackers to read arbitrary files via a .. (dot dot) attack via an HTTP GET request that uses encoded characters in the URL.

7. Evidence of Active Targeting

If the destination site hosts a WinGate server, this could be indicative of active targeting. Conversely, if the organization using the targeted IP does not intentionally offer this service, it would most likely be part of a larger scan looking for WinGate servers. The fact that two consecutive connection attempts are made (from source ports 52826 and 52827) tends to support the hypothesis that this was part of a larger scan. Scanning utilities often use sequentially higher ports during their scanning routines.

8. Severity

Severity of an event can be calculated by the equation (Criticality + Lethality) – (Network + Host Countermeasures) -- as developed by Northcutt. The potential severity of this event is directly tied to whether the affected host is running a WinGate server. In this instance, we were not. So, criticality and lethality earn very low scores (0 + 1). Since our network IDS (Snort) detected this activity, we come up with a very low score for severity of the event (0 + 1) – (5 + 0) = -4

9. Defensive Recommendation

Given the fact we're not running a WinGate server, we do not need to take any additional steps. However, in the interest of improving security conditions for the Internet community, we should continue to monitor for this type of activity. If, for example, we see recurring suspicious activity from a particular group of IPs, we should contact the source ISP to report the matter.

10. Multiple Choice Test Question

```
03/16-11:23:27.992286 198.25.136.67: 8080-> 212.68.123.1:23
TCP TTL:52 TOS:0x0 ID:20287 IpLen:20 DgmLen:40
******S* Seq: 0x7668FACF Ack: 0x0 Win: 0x800 TcpLen:20
```

Q: Given the packet above, select the best combination of statements from the selection below.

1 - The source of this packet is likely a WinGate server.
2 - The destination IP is likely assigned to a router.
3 - This packet is part of the tear-down (closing) of an established TCP connection.
4 - The capture shows clear signs of packet craft (variables clearly outside established RFCs)

a. 1 & 2 above
b. 1 & 3 above
c. 2 & 3 above

d. 1, 2 & 4 above
e. all of the above

A. Answer is a) 1 & 2 above. 3 cannot be true, as there are no FIN or FIN ACK flags set. There is not enough information in the capture to determine if the packet is crafted, so number 4 is false as well. The source port (8080) is likely a WinGate server. Lastly, the combination of dest IP (routers often end in .1 in an IP range), and the dest port for telnet makes it likely the dest IP is assigned to an interface on a router.

Trace# 2

```
ZoneAlarm Basic Logging Client v2.1.44
Windows 98-4.90.3000- -SP

type   date     time              source      destination  service

FWIN,2001/02/03,11:52:28 -6:00 GMT,12.7.130.12:0,209.99.118.XX:0,ICMP
FWIN,2001/02/03,11:52:28 -6:00 GMT,12.7.130.12:2434,209.99.118.XX:80,TCP
FWIN,2001/02/03,11:52:30 -6:00 GMT,12.7.130.12:137,209.99.118.XX:137,UDP
FWIN,2001/02/03,11:52:50 -6:00 GMT,12.7.130.12:2453,209.99.118.XX:80,TCP

               *Note: last octet of source and dest IPs obfuscated.
```

1. Source of Trace

The logs for this trace were generated by ZoneAlarm v2.1.44 personal firewall running on my home system. This system was running Windows ME during these detects.

2. Detect was Generated By

Detects were generated by ZoneAlarm v2.1.44. ZoneAlarm's approach to host-based intrusion detection seems very effective. It alerts on both inbound connection attempts, as well as on outbound connection attempts. ZoneAlarm's ability to alert on outbound connection attempts can be particularly effective in augmenting network-based intrusion detection (see "Assignment 2 – State of Intrusion Detection" below). For the purpose of reading the logs generated by ZoneAlarm, it's important to know that inbound detects are prefaced with FWIN. Outbound detects are prefaced with PE. PE log entries reflect the name of the process running on the machine that requested network connectivity. Although there is very little information available in the form of a readme file for ZoneAlarm (at least for the shareware version), ZoneAlarm appears to intercept WinSock calls sent from processes running on the host and logs the activity. Much of what I've learned about the behavior of

ZoneAlarm was gleaned through working with it on a test network for the purposes of completing this practical. Security Portal offers an evaluation of ZoneAlarm.

3. Probability the Source Address was Spoofed

Given the sequencing and variety of the traffic, it is unlikely the initiating IP is spoofed.

4. Description of Attack

Tracing the logs above, my system appears to have been the target of a reconnaissance (ICMP echo request), followed immediately by TCP traffic coming from a high port on 12.7.130.12 to port 80 on my system. Since ZoneAlarm does not provide FWIN alerts on http traffic initiated by the host it is protecting, it very likely alerted on a "GET" request initiated from IP (12.7.130.12) – an apparent attempt to see if I were running a Web server. Two seconds later, my system is sent a UDP packet from port 137 to port 137 (NetBIOS Name Service). 20 seconds after that, the same initiating IP sends another TCP packet to port 80 on my system.

5. Attack Mechanism

What looked like clear evidence of a structured and focused attack on my system proved to be automated activity from a server used to host a traceroute-like utility called VisualRoute. A whois on 12.7.130.12 resulted in the results below.

```
                Whois for visualroute.zitel.com
        Registrant:
        FORTEL Inc (ZITEL3-DOM)
           46832 Lakeview Blvd.
           Fremont, CA 94538-6543
           US
           Domain Name: ZITEL.COM
           Administrative Contact, Technical Contact, Billing
        Contact:
              FORTEL Inc  (FI1657-ORG) admin.poc@FORTEL.COM
           FORTEL Inc
           46832 Lakeview Blvd.
           Fremont, CA 94538-6543 US
```

After seeing the results of the whois, it dawned on me I had used VisualRoute during the same period out of curiosity. As evidenced by my logs, the VisualRoute server I used must have performed an unadvertised lookup on my IP while it was showing me the results of the lookup I requested. I sent an email to the address above enquiring if this was normal behavior for the VisualRoute server, but received no reply.

In essence, VisualRoute is an adaptation of traceroute that attempts to trace the hops from the originating site to a specific IP using ping, port 80 connection attempts, etc. The software provides an approximate geographic tracing of the hop on a world map overlay. The site www.visualroute.com has links to several Web sites. You can launch your query from one of these sites (see Live Demo link below). Apparently, these sites also perform an unadvertised VisualRoute query on the IP connecting to these servers. Below is a description of VisualRoute found on their homepage.

VisualRoute -- is a visual, fast, and integrated ping, whois, and traceroute program that automatically analyzes connectivity problems, *displaying the results on a World map*. When configured as a Server, VisualRoute provides visual trace route services to clients. Live Demo

6. Correlations

Searches using google.com and dogpile.com disclosed no information related to this type of activity.

7. Evidence of Active Targeting

Although my research revealed this was not an attack attempt, I believe it is interesting that VisualRoute (at least the server I used at the time) launched an unadvertised VisualRoute trace on my IP. Tracing the logs again, it is now clear that these alerts were generated by a VisualRoute trace directed at my system. I didn't initially understand the UDP connection attempt to port 137 of my system, but information found on www.networkice.com offers one explanation: Firewalls will often register a significant number of inbound NetBIOS requests. This is due to the behavior of MS Windows servers that use NetBIOS, as well as DNS to resolve IP addresses. Another explanation, given the purpose of VisualRoute, the NetBIOS packet could have been a deliberate part of the information gathering process directed at the targeted IP.

8. Severity

Given the fact this was -- at worst -- an unadvertised reconnaissance of a system by a VisualRoute server, the severity is zero.

9. Defensive Recommendation

No defensive recommendation is necessary.

10. Multiple Choice Test Question

```
03/16-11:25:38.415534 167.34.56.12:4567-> 224.67.134.1:80
TCP TTL:52 TOS:0x0 ID:53896 IpLen:20 DgmLen:50
***A**** Seq: 0xF3331A8E Ack: 0x0 Win: 0x800 TcpLen: 40
```

Q: Given the packet above, what field value must be invalid?

a. Type Of Service (TOS) cannot hold a value of 0x0.
b. Datagram (DgmLen: 50) length is invalid.
c. Window size (Win: 0x800) is outside the range of available values.
d. The source port (4567) is not within a valid range.

A: Answer is b). Given an IpLen: 20 and a TcpLen:40, the datagram length can't be 50.

Trace# 3

```
[**] IDS28 - PING NMAP TCP [**]
03/16-11:25:38.415534 10.0.0.154:52838-> 192.168.33.0:80
TCP TTL:52 TOS:0x0 ID:53896 IpLen:20 DgmLen:60
***A**** Seq: 0xF3331A8E Ack: 0x0 Win: 0x800 TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL
[**] IDS28 - PING NMAP TCP [**]
03/16-11:25:44.092562 10.0.0.154:52838-> 192.168.33.0:80
TCP TTL:52 TOS:0x0 ID:49984 IpLen:20 DgmLen:60
***A**** Seq: 0x95EDAA84 Ack: 0x0 Win: 0x800 TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL
[**] IDS28 - PING NMAP TCP [**]
03/16-11:25:49.724320 10.0.0.154:52838-> 192.168.33.0:80
TCP TTL:52 TOS:0x0 ID:40025 IpLen:20 DgmLen:60
***A**** Seq: 0xDC3A9891 Ack: 0x0 Win: 0x800 TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL
```

1. Source of Trace

These logs were generated in a test network running multiple OSes (MS Win2000, MS WinNT4.0, Linux, and Solaris).

2. Detect was Generated By

Detects were generated by SNORT-1.7. HTML formatted output obtained from SnortSnarf.

3. Probability the Source Address Was Spoofed

In this particular instance I know the source IP address was not spoofed, as the captured traffic was generated on a test network. However, the NMAP utility used to generate this detect makes it very easy to spoof source IPs. The purpose of this is to obfuscate the true point of origin. So, a "real world" capture similar to the one above could have easily been spoofed.

4. Description of Attack

Below is a listing of the essential information regarding this attack and the Snort rule that alerted on it (Source is Marty Roesch, as posted by www.whitehats.com). Note: this particular scanning attack was carried out by an earlier version of NMAP.

IDSKEY	IDS28
EVENT NAME	probe-nmap_tcp_ping
EVENT DESCRIPTION	A remote user has used the NMAP portscanning tool to probe the server. This alert indicates that an NMAP TCP ping was sent to detemine if a host is reachable.
	Dynamically Generated Signatures
SNORT SIGNATURE	alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS28/probe-nmap_tcp_ping"; ack: 0; flags: A;)
HAS SIGNATURE	YES
	IP Layer
PROTOCOL	TCP

SOURCE IP	$EXTERNAL
SOURCE PORT	any
DIRECTION	->
DESTINATION IP	$INTERNAL
DESTINATION PORT	any
ACK	0
	Protocol Layer
FLAGS	ACK
	Subjective Qualities
CATEGORIES	Pre-Attack_Probe
ATTACKER NEEDS RESPONSE	YES
EASILY SPOOFED	YES
BACKGROUND	This signature will only detect older versions of nmap that set the tcp ack to zero.

PACKET TRACES	`12/22-13:35:44.910929 source:47212 -> target:80`
	`TCP TTL:39 TOS:0x10 ID:54841`
	`******A* Seq: 0xF7D00003  Ack: 0x00000000  Win: 0x1000`

	Indexing
CVE	CAN-1999-0523
	Credit
CREDITS	`Marty Roesch: developed stealth portscan detection in Snort.` `lanl folks: pointed out incorrect packet trace.`
CONTRIBUTOR	roesch@clark.net

5. Attack Mechanism

In this instance, NMAP was used to conduct a probe of port 80 on 192.168.33.0. The SnortSnarf readout of the Snort alert above shows the initial probe of the 192.168.33.XX subnet on the test network. This is very characteristic of an NMAP probe, as it can easily generate a great deal of traffic over a particular subnet in order to learn what systems are alive and what services are running on those systems.

6. Correlations

The IDS Key 28 is the ref number for this signature. CVE Candidate 0523 is cross-referenced to IDS Key 28 on the arachNIDS portion of the whitehats.com site. However, the description found on the CVE reference site (below) describes this exposure as "ICMP echo (ping) is allowed from arbitrary hosts." The Snort rule that alerted on the NMAP probes looks for TCP traffic, not ICMP. I assume this is a typo.

Name	CAN-1999-0523 (under review)
Description	ICMP echo (ping) is allowed from arbitrary hosts.
References	
Phase	Proposed (19990726)
Votes	REJECT(1) Northcutt REVIEWING(1) Frech
Comments	Northcutt> (Though I sympathize with this one :)

7. Evidence of Active Targeting

An NMAP scan of a particular network is indicative of active targeting.

8. Severity

Using the equation (Criticality + Lethality) – (Network + Host Countermeasures) -- as developed by Northcutt, the potential severity of this event is relatively low. The criticality of the network low, as it is a test network. The lethality of the NMAP probe is relatively low as well. In this instance, our IDS (Snort) easily detected the activity. I provide the event a rating of (3 + 1) – (5 + 2) = -3.

9. Defensive Recommendation

A proven defense against this type of probing activity is to conduct periodic audits of your networks – these audits often include the use of NMAP to determine what a

prospective intruder would see, if they scanned your network. By ensuring unnecessary services are disconnected, patches are updated, and basic computer security procedures are in place, you go a long way in defending yourself against this form of reconnaissance.

10. Multiple Choice Test Question

```
03/16-11:25:38.415534 167.34.56.12:4567-> 224.67.134.1:80
TCP TTL:52 TOS:0x0 ID:53896 IpLen:20 DgmLen:60
***A**** Seq: 0xF3331A8E Ack: 0x0 Win: 0x800 TcpLen: 40
```

Q: What fields in the packet capture above can give an indication whether or not source routing is being attempted. Select the best answer.

a. TTL – Time To Live
b. Seq & TcpLen fields -- Sequencing and TCP length
c. Win & Seq fields – Window & Sequencing fields
d. IpLen – IP Header Length
e. DgmLen – IP Datagram Length

A: Answer is d) IpLen. In source routing, the IP addresses the sender is explicitly requesting for intermediate routing points are included in the IP header. An IP header that is larger than normal could give be an indication of source routing.

Trace# 4

```
 ZoneAlarm Basic Logging Client v2.1.44
 Windows 98-4.90.3000- -SP

type   date     time                source          destination   service
FWIN,2001/02/20,18:36:48 -6:00 GMT,24.160.144.XXX:0,209.99.125.XXX:0,ICMP
FWIN,2001/02/20,18:36:48 -6:00 GMT,24.160.144.XXX:2894,209.99.125.XXX:1,TCP
FWIN,2001/02/20,18:36:48 -6:00 GMT,24.160.144.XXX:2895,209.99.125.XXX:2,TCP
FWIN,2001/02/20,18:36:48 -6:00 GMT,24.160.144.XXX:2896,209.99.125.XXX:3,TCP
FWIN,2001/02/20,18:36:48 -6:00 GMT,24.160.144.XXX:2897,209.99.125.XXX:4,TCP
FWIN,2001/02/20,18:36:48 -6:00 GMT,24.160.144.XXX:2898,209.99.125.XXX:5,TCP
FWIN,2001/02/20,18:36:48 -6:00 GMT,24.160.144.XXX:2899,209.99.125.XXX:6,TCP

**Note: log entries summarized for brevity... all entries reflected a
 consistent progression from destination port 7 - 1126**

FWIN,2001/02/20,18:46:44 -6:00 GMT,24.160.144.XXX:2050,209.99.125.XXX:1127,TCP
FWIN,2001/02/20,18:46:44 -6:00 GMT,24.160.144.XXX:2051,209.99.125.XXX:1128,TCP
FWIN,2001/02/20,18:46:44 -6:00 GMT,24.160.144.XXX:2052,209.99.125.XXX:1129,TCP
FWIN,2001/02/20,18:46:44 -6:00 GMT,24.160.144.XXX:2053,209.99.125.XXX:1130,TCP
```

```
FWIN,2001/02/20,18:46:44 -6:00 GMT,24.160.144.XXX:2054,209.99.125.XXX:1131,TCP
FWIN,2001/02/20,18:46:44 -6:00 GMT,24.160.144.XXX:2055,209.99.125.XXX:1132,TCP
FWIN,2001/02/20,18:46:44 -6:00 GMT,24.160.144.XXX:2056,209.99.125.XXX:1133,TCP
FWIN,2001/02/20,18:46:46 -6:00 GMT,24.160.144.XXX:2057,209.99.125.XXX:1134,TCP
FWIN,2001/02/20,18:46:46 -6:00 GMT,24.160.144.XXX:2058,209.99.125.XXX:1135,TCP
```

*Note: last octet of source and dest IPs obfuscated.

1. Source of Trace

The logs for this trace were generated by ZoneAlarm v2.1.44 personal firewall running on my home system. This system was running Windows ME during these detects.

2. Detect was Generated By

Detects were generated by ZoneAlarm v2.1.44. ZoneAlarm's approach to host-based intrusion detection is summarized in Trace #2 (para 2) above.

3. Probability the Source Address was Spoofed

The alerts shown in the log above are indicative of scanning activity. This particular scan was conduct at my request by a co-worker to test ZoneAlarm running on my personal system. Although typically there's a high probability that at least some of the source IPs in a large scan such as this were spoofed, this particular scan was conducted using SuperScan (www.foundstone.com). SuperScan, a freeware security scanner, does not allow for forged source IPs.

4. Description of Attack

Tracing the logs above, the initiating host (IP 24.160.144.XXX) is conducting a TCP scan of my computer starting at port 0 and going sequentially higher. The ports of the initiating host also grow in an ordered, sequential manner.

5. Attack Mechanism

As noted above, the port scanner SuperScan v3.0 was directed at the host being monitored by ZoneAlarm. A range of target ports (0 to 1135) was selected and launched at a specific IP. With an Internet setting of "High," ZoneAlarm logged all the TCP packets sent to my system. Conferring with my co-worker during the scan, he stated he did not receive any returned information during the scan. A netstat –a on my system prior to the scan confirmed that ports 137, 138, and 139 were all in a listening state. As advertised, ZoneAlarm prevented data from being returned to the scanner.

6. Correlations

I conducted various searches using google.com and dogpile.com. There's a significant body of information available regarding network scanning – particularly using NMAP. However, I didn't discover anything particularly relevant to this trace. Actually, I chose to examine the logs generated by ZoneAlarm because of the growing popularity of personal firewalls. As I hope I demonstrated in my white paper below, a layered defense using host-based and network-base IDS can be rather effective in detecting activity by certain Trojan programs.

7. Evidence of Active Targeting

A trace of the logs will quickly reveal this was a focused probe of the host using the destination IP.

8. Severity

Using the equation (Criticality + Lethality) – (Network + Host Countermeasures), the potential severity of this event is low. The criticality of the targeted system, because it is my own, I believe is rather high. The lethality of the SuperScan probe is low to medium. In this instance, our IDS (ZoneAlarm) easily detected the activity and prevented information from being returned. I provide the event a rating of (5 + 1) – (1 + 5) = 0.

9. Defensive Recommendation

No defensive recommendations are necessary.

10. Multiple Choice Test Question

```
03/16-06:29:18.327830 192.168.1.11:1041 -> 192.168.1.1:25
TCP TTL:128 TOS:0x0 ID:42755 IpLen:20 DgmLen:45 DF
***AP*** Seq: 0x455E58  Ack: 0x50AC21  Win: 0x21D3 TcpLen: 20
48 45 4C 4F 20                                      HELO
```

Q: Given the packet above, what answer below best describes the activity?

a. Telnet login attempt.
b. Response to a login attempt from an FTP server.
c. Response from a SMTP mail server to a login.
d. Response from a telnet server to a login attempt

A: Answer c) is correct. The destination port of 25 tells us it is likely a packet destined for a mail server. The payload of "HELO" is part of the initial handshake that a client initiates with the server prior to sending the server email for delivery.

Trace# 5

```
[**] OVERFLOW - Possible attempt at MS Print Services [**]
03/16-11:24:54.498894 10.0.0.154:52829-> 192.168.33.0:515
TCP TTL:52 TOS:0x0 ID:61844 IpLen:20 DgmLen:40
******S* Seq: 0x7668FACF Ack: 0x0 Win: 0x800 TcpLen: 20

[**] OVERFLOW - Possible attempt at MS Print Services [**]
03/16-11:24:54.817950 10.0.0.154:52830-> 192.168.33.0:515
TCP TTL:52 TOS:0x0 ID:6030 IpLen:20 DgmLen:40
******S* Seq: 0x855C2D85 Ack: 0x0 Win: 0x800 TcpLen: 20
```

1. Source of Trace

These logs were generated in a test network running multiple OSes (MS Win2000, MS WinNT4.0, Linux, and Solaris).

2. Detect was Generated By

Detects were generated by SNORT-1.7. HTML output obtained from SnortSnarf.

3. Probability the Source Address was Spoofed

This traffic was generated on a test network, but it is unlikely that "real world" attempts to exploit the Windows NT Spooler service (Spoolss.exe) would be spoofed. In their detailed overview of the exploit, www.eeye.com notes the exploits of this vulnerability can be accomplished remotely, but a more likely scenario would be use by someone with network access. Running the exploit remotely is more difficult -- sending a buffer overflow to the victim machine, as part of spoofed packets, would make it even more difficult. Without a TCP connection, the attacker would essentially be flying blind -- possible, but not likely.

4. Description of Attack

The Windows NT Spooler service (Spoolss.exe), used for various printing activities, contains a number of security holes that allow for data overflows. These vulnerabilities are evident when someone passes data to various spooler service API's and spoolss.exe does not check the size of the receiving buffer to make sure it can hold the incoming data.

5. Attack Mechanism

This particular detect relates to an attempt to perform a remote buffer overflow against the Spools.exe. As noted in their write up of the exploit, this particular exploit can be executed only if you are a "Power User". This particular detect relates to a remote buffer overflow attack that does not require you to be at the power user level.

6. Correlations

Below is a listing of the Common Vulnerabilities and Exposures related to Win NT via print services (Source: cve.mitre.org)

CVE-1999-0898	Buffer overflows in Windows NT 4.0 print spooler allow remote attackers to gain privileges or cause a denial of service via a malformed spooler request.

7. Evidence of Active Targeting

Since this attack is particularly focused (eg. It is only effective against an unpatched NT system with a network printer), this would be fairly clear sign of active targeting. If it were, say a blind shotgun attempt at a network, then that would be evident in a review of logs for the period.

8. Severity

Severity of an event can be calculated by the equation (Criticality + Lethality) – (Network + Host Countermeasures) -- as developed by Northcutt. The potential

severity of this event is relatively low. Most remote buffer flow exploits targeting this vulnerability result in a denial of service to that particular printer. Unless the printer were a high capacity machine critical to business operations, its criticality isn't particularly high. The lethality of this attack to an unpatched system is fairly high. This event, thanks to network monitoring, earns a zero for severity: (1 + 4) – (5 + 0) = 0

9. Defensive Recommendation

The best defensive measures are to ensure the latest patches are in place for all Win NT systems. Secondly, if we see persistent malicious activity from a particular set of IPs, we can develop a watchlist and block these addresses at our routers.

10. Multiple Choice Test Question

```
01/11-16:39:45.074001 source -> target
ICMP TTL:254 TOS:0x0 ID:13170
ADDRESS REQUEST
F3 2B 5E 9C 00 00 00 00                    .+^.....
```

Q: Given the trace above, what answer below best describes the threat posed by this activity?

a. ICMP redirect for a specific network – denial of service.
b. ICMP redirect for a specific host - denial of service.
c. ICMP subnet mask request – reconnaissance.
d. ICMP information request- reconnaissance.

A: Answer c) is correct. The trace above is an attempt to obtain the netmask of a particular network from a device connected to that network CAN-1999-0524. The threat comes in the form of reconnaissance.

Assignment 2 – The State of Intrusion Detection

Detecting Trojan Programs that Use Email to Remotely Monitor Victim Systems

Premise

There are a number of Trojan programs designed to covertly monitor activity on a victim host – typically employing keystroke and screen capture, or simple password stealing on Win95/98/NT OSes. The results are then emailed from the victim host by the Trojan to a specific email account at various intervals. The use of "legitimate," outbound high-volume traffic (in this instance email) to send out data from the victim host, can represent quite a challenge to traditional network-based intrusion detection. To address this type of attack, a layered approach --integrating host-based and network-based intrusion detection systems – offers the best solution for detection.

Review of Three Covert Monitoring Programs

The following is an overview of three programs that use email to surreptitiously extract information from victim hosts. A brief description of the programs, and sample output (sniffer and email) are provided below. All three programs are written to exploit MS Windows 95/98. The traffic was generated on a test network using an Infradig Mailserver (POP3) for delivery, with no DNS support. Traffic was captured by Snort in sniffer mode (-v & -d options). Additionally, recommended Snort rule sets are provided to detect on specific signatures found in traffic generated by these programs. Depending on the traffic load and positioning of the Snort sensor, monitoring port 25 may prove impractical. This fact lends support to the premise of this paper.

1) Barok v.1.0

As outlined in the terse readme.txt file that comes with the download (below) I found on antionline.com, the author "Spyder" claims the program can copy various cached passwords, as well as other information.

```
barok v.1.0
email password sender
(ras and cache) passwords
includes  phone number, ip address, dns address, win
address, etc...

files:
  server.exe ---->> server (trojan)
  setup.exe ---->> configuration (client)(setup)


copyright (c) 2000 GRAMMERSoft Group
                by: spyder
   email: spyder@super.net.ph
```

R … " is using a m…

```
Querying Mail routing information (mx) for
super.net.ph  -  Mar 17, 2001
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43343
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 0
;;       super.net.ph, type = MX, class = IN
super.net.ph.           0S IN MX       10
casper.super.net.ph.
```

Below is an email sent by the Barok Trojan and delivered to its destination email address. The Trojan successfully copied and transmitted hostname, username, and IP address of the victim host – no RAS or cached passwords were available on the victim host for retrieval. For the purpose of developing a Snort rule set to detect this traffic, we'll key on the "hard-coded" subject line: "PSWRD Sender Trojan."

```
Return-Path: <cmorgan@192.168.1.1>
Received: from preferred.192.168.1.1 ([192.168.1.11]) by
192.168.1.1 with id 3AB2CCF8.00000135@192.168.1.1; Sat,
17 Mar 2001 02:33:28 GMT
From: preferred-user@192.168.1.11
To: cmorgan@192.168.1.1
Subject: Barok.... PSWRD Sender Trojan
X-Mailer: Barok... email PSWRD sender--- by: spyder
Message-ID: <3AB2CCF8.00000135@192.168.1.1>
Date: Sat, 17 Mar 2001 02:33:28 GMT

Host: preferred-user
Username: jg
IP Address: 192.168.1.11
```

Snort (in sniffer mode) capture of email traffic generated by Barok (see email above).

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

03/16-06:29:18.655601 192.168.1.11:1041 -> 192.168.1.1:25
TCP TTL:128 TOS:0x0 ID:44291 IpLen:20 DgmLen:275 DF
***AP*** Seq: 0x455ED3  Ack: 0x50ACDD  Win: 0x2117  TcpLen: 20
54 6F 3A 20 63 6D 6F 72 67 61 6E 40 31 39 32 2E     To: cmorgan@192.
31 36 38 2E 31 2E 31 0D 0A 53 75 62 6A 65 63 74     168.1.1..Subject
3A 20 42 61 72 6F 6B 2E 2E 2E 2E 20 50 53 57 52  :    Barok.... PSWR
44 20 53 65 6E 64 65 72 20 54 72 6F 6A 61 6E 0D  D    Sender Trojan.
0A 58 2D 4D 61 69 6C 65 72 3A 20 42 61 72 6F 6B     .X-Mailer: Barok
2E 2E 2E 20 65 6D 61 69 6C 20 50 53 57 52 44 20     ... email PSWRD
73 65 6E 64 65 72 2D 2D 2D 20 62 79 3A 20 73 70     sender--- by: sp
79 64 65 72 0D 0A 0D 0A 48 6F 73 74 3A 20 70 72     yder....Host: pr
65 66 65 72 72 65 64 2D 75 73 65 72 0D 0A 55 73     eferred-user..Us
65 72 6E 61 6D 65 3A 20 44 65 66 61 75 6C 74 0D     ername: Default.
0A 49 50 20 41 64 64 72 65 73 73 3A 20 31 39 32     .IP Address: 192
2E 31 36 38 2E 31 2E 31 31 0D 0A 0A 52 41 53 20   .168.1.11...RAS
50 61 73 73 77 6F 72 64 73 3A 20 0D 0A 0A 0D 0A   Passwords: .....
43 61 63 68 65 20 50 61 73 73 77 6F 72 64 73 3A      Cache Passwords:
20 0D 0A 0A 0D 0A 0D 0A 2E 0D 0A                  ..........


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

The following is a recommended Snort content rule for detecting this activity. As the author (Marty Roesch) of Snort points out in his HowTo page for writing rules, content detection is

computationally expensive, so we key on the string: "PSWR Sender." Intentionally brief to reduce CPU load, but unique enough to limit the number of false alarms.

```
$MYHOST.NET 25 -> alert tcp any any  (content: "PSWR Sender"; msg: "Barok Email Trojan!";)
```

2) **Kuang2 pSender Full v0.34**

This program has a lighter weight companion called Kuang2 pSender v0.21; but I opted to analyze the "Full" version available at www.11th.co.uk. The author "Weird" claims the program performs keystroke and screen capture and mails the results to a user defined email address. It uses a setup program to define a number of variables, to include the size of the keyboard buffer that triggers the results to be sent via email from the victim host. Excerpts from the author's ReadMe file are found in Attachment 1.

Below is an email sent by the Kuang2 Full Trojan and successfully delivered to its destination email address. The Trojan conducted a combination keystroke and screen capture and transmitted the information via this email. The payload begins with "c:\Trojans\sesame"… and ends with "[Welcome to the SESAME Control Center V1.02]." This email captures part of my keystroke activity, while I was configuring another Trojan named Sesame (addressed in para 3 below). For the purpose of developing a Snort rule set to detect this traffic, we'll key on the "hard-coded" subject line: "Kuang2 report." Note: TCPDump display of the same information omitted for brevity.

```
Return-Path: <victim@192.168.1.2>
Received: from preferred.192.168.1.1 ([192.168.1.11]) by
192.168.1.1 with id 3AB2C615.00000084@192.168.1.1; Sat,
17 Mar 2001 02:04:05 GMT

SUBJECT: Kuang2 report
FROM: ku@ng.pSender
Message-ID: <3AB2C615.00000084@192.168.1.1>
Date: Sat, 17 Mar 2001 02:04:05 GMT

-----
c:\Trojans\sesame
No new directory defined
Win 95/98 detected
15000
c:\Trojans\sesame\history.txt
cmorgan@192.168.1.1
spy@bogus.com
3
OFF
>password<
[Welcome to the SESAME Control Center V1.02]
===

# PREFERRED USER
```

The following is a Snort content rule that will detect the signature string in Kuang2 on outbound email from an infected system.

```
$MYHOST.NET 25 -> alert tcp any any  (content: "Kuang2"; msg: "Kuang2 Email Trojan!";)
```

3) **Sesame v1.02**

Sesame is an interesting program since it does not appear to be innately malicious. However, like many security applications, it can be easily used in a malicious fashion. Since this program monitors changes in a targeted file on the host computer, it could be used to alert a system administrator of changes in key files. The author's ReadMe.txt file describes this program as a "Stealth Email SMTP Autosender ModulE" (sic) – full text is in Attachment 2. It's also worth noting Sesame v1.02 does not claim (nor appear to) perform keystroke or screen capture. However, it could very easily be packaged with a small keystroke capture program. If not being used as part of an organization's security policy, it would be an obvious threat.

Fortunately for us, as with the examples above, this program (at least the unregistered version) uses a "hard-coded" subject line string in the email it sends. In this instance, the string is "SESAME Email." The payload is always an attachment; specifically the file that you configured it to monitor prior to installation. The Sesame v1.02 setup program allows a user to configure it to send out the targeted file based on a system clock setting, after the file is altered, or after the file grows to a certain size. Our primary concern would be that it could be configured to send out a keystroke log or password file after it reaches a certain size or is altered. The email capture below depicts the transmission of the targeted file "Sensitive.txt" on the victim system.

```
X-Registered-To: Peter T. Schmidt Software(PTS)
Date: Sat, 17 Mar 2001 0:24 -0600
To: <cmorgan@192.168.1.1>
From: <spy@bogus.com>
Subject: < SESAME Email (2) UNREGISTERED >
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="=====_4206312_====="
Message-ID: <3AB2C9FC.000000E1@192.168.1.1>

--=====_4206312_=====
Content-Type: text/plain


Please see attachment for the file.

--=====_4206312_=====
Content-Type: application/octet-stream; name="Sensitive.txt "
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="Sensitive.txt "

dGhpcyBpcyBhIHRlc3QgdG8gc2VlIGl0IHNlc2FtZSBpcyBjYXB0dXJlaW5nbiBte
SBzZWNyZXQg …..

--=====_4206312_=====--
```

Snort (in sniffer mode) capture of email traffic generated by Sesame (see email above).

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

03/16-06:23:38.021301 192.168.1.11:1040 -> 192.168.1.1:25
TCP TTL:128 TOS:0x0 ID:32515 IpLen:20 DgmLen:82 DF
***AP*** Seq: 0x40097B  Ack: 0x4B56CF  Win: 0x211D  TcpLen: 20
53 75 62 6A 65 63 74 3A 20 3C 20 53 45 53 41 4D     Subject: < SESAM
45 20 45 6D 61 69 6C 20 28 32 29 20 55 4E 52 45     E Email (2) UNRE
47 49 53 54 45 52 45 44 20 3E                       GISTERED >
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The following is a Snort content rule that will detect the signature string in Sesame v1.02.

$MYHOST.NET 25 -> alert tcp any any (content: “SESAME Email"; msg: "Sesame Stealth Emailer";)

Conclusion

The use of email to transmit the covert monitoring of individual computers continues to present a challenge to traditional network-based intrusion detection systems, particularly those deployed in a medium to large enterprise. I could find no specific CVE for this form of attack. The closest was a candidate CVE: CAN-1999-0660 "A hacker utility or Trojan Horse installed on a system…" Also, SANS published a paper regarding the ports often associated with Trojan programs. Although the Trojans Barok and Sesame are not listed in this paper, port 25 (SMTP) is listed as used by Kuang2 and a few other Trojans.

Given the difficulty of detecting this activity using conventional intrusion detection means, the most logical solution seems to be a layered approach that uses network-based and host-based (more specifically, workstation-based) intrusion detection. Fortunately, anti-virus software can detect most of these freely available Trojans; however, neither McAfee, nor Norton (at least the 2000 versions I used) detected the Sesame Stealth Emailer. This could be intentional, as Sesame can be used for legitimate security purposes.

Looking specifically at intrusion detection for the individual PC, there are a series of products that provide effective host-based intrusion detection. Those products include BlackIce, ZoneAlarm, and TinyFirewall, to name the more popular ones. For the purpose of examining the effectiveness of this host-based approach, I installed ZoneAlarm on the victim host used in the traces of the three programs above. ZoneAlarm detected the fact that all three programs requested WinSock access on the victim computer when they attempted to mail out their payloads (These detects were made with a ZoneAlarm Internet setting of "High"). Below is an excerpt from a log generated by ZoneAlarm -- detects are in bold print. These detects, as indicated by the type of PE, were requests by processes for WinSock access on the host (victim) OS. SPOOL.EXE is the Barok Trojan. The process "beta" is the Sesame v1.02 program.

```
ZoneAlarm Basic Logging Client v2.1.44
Windows 98-4.10.1998- -SP

type   date   time              source          destination    transport
PE,2001/03/15,22:54:27 -6:00 GMT,Outlook Express,192.168.1.1:25,N/A
FWIN,2001/03/24,22:33:54 -6:00 GMT,192.168.1.1:1153,192.168.1.11:23,TCP
FWIN,2001/03/24,22:35:36 -6:00 GMT,192.168.1.1:1165,192.168.1.11:21,TCP
FWIN,2001/03/24,22:36:52 -6:00 GMT,192.168.1.1:1172,192.168.1.11:23,TCP
PE,2001/03/24,22:52:20 -6:00 GMT,Windows Explorer,127.0.0.1:1027,N/A
PE,2001/03/26,00:03:48 -6:00 GMT,SPOOL64.EXE,192.168.1.1:25,N/A
PE,2001/03/26,00:06:57 -6:00 GMT,beta,192.168.1.1:25,N/A
PE,2001/03/26,00:18:14 -6:00 GMT,beta,192.168.1.1:25,N/A
PE,2001/03/26,00:20:12 -6:00 GMT,SPOOL64.EXE,192.168.1.1:25,N/A
PE,2001/03/26,00:38:40 -6:00 GMT,SPOOL64.EXE,192.168.1.1:25,N/A
```

All these programs had unique signatures that make it possible to detect through content monitoring of outbound network traffic. However, monitoring on a very active port, such as 25, may outstrip the capabilities of many network-based intrusion detection systems. Additionally, subsequent versions of these or similar Trojan programs may allow the user to configure all aspects of the email, thus eliminating the static signatures necessary for traditional network-based intrusion detection.

Analyze This

The following is an analysis of suspicious traffic affecting your network. The analysis draws upon data recently collected from you Snort sensors. To provide you a more meaningful context, I drew from Mr. Marc Bayerkohler's earlier excellent work to address changes in the threats to your networks. An overview of the suspicious activity captured by Snort is presented below. Emphasis is placed on that activity that is most relevant to the security of your networked systems.

Overview of Suspicious Network Traffic

Description of Alert	No. Alerts
STATDX UDP attack	1
Happy 99 Virus	1
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	1
site exec - Possible wu-ftpd exploit - GIAC000623	2
Probable NMAP fingerprint attempt	8
External RPC call	59
Back Orifice	77
TCP SMTP Source Port traffic	100
Broadcast Ping to subnet 70	154
connect to 515 from inside	159
SUNRPC highport access!	204
SMB Name Wildcard	515
Russia Dynamo - SANS Flash 28-jul-00	546

Description of Alert	No. Alerts
NMAP TCP ping!	558
SNMP public access	591
Queso fingerprint	710
Null scan!	826
Attempted Sun RPC high port access	2053
WinGate 1080 Attempt	2239
Watchlist 000222 NET-NCFC	2401
connect to 515 from outside	4238
Tiny Fragments - Possible Hostile Activity	5340
DNS udp DoS attack described on unisog	16146
SYN-FIN scan!	51192
Watchlist 000220 IL-ISDNNET-990517	105918

The tables above summarize the alerts generated by Snort for the period evaluated. The left column of each table offers a brief description of the alert and the right column indicates the number of times that particular alert occurred. As you can see, the largest number of alerts (105,918) are associated with traffic coming from or destined for an IP range on Watchlist 000220. Conversely, only one occurrence of a STATDX UDP attack was detected. We'll compare this activity with that analyzed by Mr. Bayerkohler. Since his work was built upon that of Mr. Lenny Zeltser, we're able to provide you a valuable trend analysis tool regarding the security of your networks. In addition to providing you trend data, we provide recommendations for assessing the significance of this activity to your enterprise. Information regarding the most actively targeted hosts is provided after the applicable tables.

Top Alert Destination Hosts (Your hosts receiving the most suspicious traffic)

Tend Analyses of Previous Activity

Host	1st Set of Alerts	2nd Set of Alerts	Current # of Alerts	Updated Status
MY.NET.253.105	22118	47	8	No Significant Change
MY.NET.217.2	4197	6	153	Slight Increase
MY.NET.253.41	4176	4387	296	Significant Decrease
MY.NET.100.230	3462	749	808	No Significant Change

New Activity

Host	Number of Alerts	Status
MY.NET.201.222	37,609	**Immediate Attention**
MY.NET.202.30	2,292	**Immediate Attention**
MY.NET.209.154	859	Deserves Attention
MY.NET.6.7	569	Deserves Attention
MY.NET.213.158	663	Deserves Attention

MY.NET.253.105: As noted in the first table above, those machines which were most active during the last two analyses, have shown little change that should trigger a great deal of concern. During this period, MY.NET.253.105 received a number of null scans from 216.51.104.65.

MY.NET.217.2: The majority of the activity directed at this machine was various forms of scanning, particularly spp_portscans.

MY.NET.253.41: The number of alerts for this site decreased slightly. As was true during the last analysis, the majority of the alerts were triggered by traffic from Chinese and Israeli sites on Watchlists 000220 & 000222.

MY.NET.100.230: Alerts related to traffic destined to this system remained about the same. However, the number of scans originating from this machine is cause for concern (see next section re Top Alert Source Hosts).

MY.NET.201.222, MY.NET.209.154 & **MY.NET.202.30**: These systems have apparently drawn the sustained attention of the Israeli site on Watchlist 000220. Given the volume of traffic destined for these machines, they both deserve immediate attention to fully assess their security status.

MY.NET.6.7: This machine comes to our attention because of the amount of traffic produced by a Chinese site on Watchlist 000222 destined for it.

MY.NET.213.158: The vast majority of the alerts associated with this machine relate to SUN RPC access with machines using an IP address range of 205.188.153.XXX (registered to AOL.com). If you have a data sharing arrangement with a company using this IP range, and that arrangement uses RPC services, then these alerts are of no concern. Conversely, if you do not have a data sharing arrangement, then this system deserves immediate attention.

Top Alert Source Hosts (Those hosts generating suspicious traffic)

Host	1st Set of Alerts	2nd Set of Alerts	Current # of Alerts	Updated Status
202.38.128.188	22338	0	0	**No Change**
MY.NET.253.12	18869	0	3	Nominal Increase
204.60.176.2	13619	0	0	**No Change**
159.226.45.3	5066	1558	0	Significant Decrease
142.150.225.137	4594	0	0	**No Change**

New Activity

Host	Number of Alerts	Status
MY.NET.217.182	4052	**Immediate Attention**
MY.NET.217.126	4847	**Immediate Attention**
MY.NET.217.150	22513	**Immediate Attention**
MY.NET.217.158	15918	**Immediate Attention**
MY.NET.6.7	187	Deserves Attention
212.179.27.111	39015	**Immediate Attention**
147.8.182.157	8460	Deserves Attention

202.38.128.188: No traffic from this address was observed. Awarded "green" for two consecutive periods of no suspicious activity

MY.NET.253.12: Curiously, the only traffic originating from this machine was a narrow port scan. This could be the result of legitimate activity by one of your system administrators

204.60.176.2: No additional traffic from this address has been observed.

159.226.45.3: No activity from this site was observed for this period; however, given the history of activity from the Chinese Watchlisted site, it should be observed through another evaluation period.

142.150.225.137: No additional traffic from this address has been observed.

MY.NET.217.182, MY.NET.217.126, MY.NET.217.150 & MY.NET.217.158: All of these hosts on your network require immediate attention. Given the large number of scans originating

from these machines, they have either been compromised or are being used well outside the scope of most site policies.

MY.NET.6.7: Although there were a relatively small number of suspicious connections coming from this machine, it still deserves some attention. This is true given its prior history as the destination of suspicious telnet activity from 159.226.45.3.

212.179.27.111: This traffic is associated with the Watchlist site of 000220 (Israel). The volume of traffic associated with this site makes it a good candidate for blocking at your network routers.

147.8.182.157: As is true with the traffic from Israel, traffic coming from this site (an ISP in Hong Kong) should be evaluated for blocking. This particular site conducting large scale scans of your network for POP2 service. This is likely an attempt to discover systems vulnerable to exploits of this older mail delivery protocol.

Methodology

I used a number of methods to analyze the traffic generated by Snort. Since the captured data was either comprised of alerts or scans, I appended log files of like data together into one file (eg. cat SnortA*.txt >> all_snortA.txt). To get a better overall view of both scan and alerts combined, I also created a mega file that consisted of SnortA*, SnortS* and OOSche* data. Running searches on this large file was a bit time consuming, but provided some valuable insight.

I, like my predecessor, attempted to use SnortSnarf to parse through all the alert logs at once. However, I found that SnortSnarf had difficulty working the size files I created. In every instance, SnortSnarf was able to provide an accurate count of all the various alerts, but it had problems building and linking the individual html files that provide detail regarding specific source and destination addresses. The work-around I attempted consisted of analyzing five to six alert logs at a time; but that proved too time consuming. Perhaps my problems related to the fact I was using SnortSnarf in a Windows 2000 environment. Although I changed the environment variables in the body of the SnortSnarf program to account for a Windows OS, I couldn't use SnortSnarf to its full potential in analyzing the alert and scan logs in their entirety. Like my predecessor, I used grep and egrep scripts to parse through the combined Snort alert and scan logs:

Initially, I tried the same approach as my predecessor, but ended up making some slight modifications. For example, to select alerts related to traffic destined for a particular address, I used the following:

```
egrep –e '-> MY.NET.253.41' all_snort.txt | wc –l
296
```

To search of alert log entries related to traffic originating form a particular address, I used the following... additionally I would periodically do a global search for the same address to check my methodology.

```
egrep –e ' from MY.NET.253.41' all_snort.txt | wc –l
52
egrep –e 'MY.NET.253.41' all_snort.txt | wc –l
348
```

Instead of the default output for wc, I chose to use the –l switch. This gives a count of how many lines in a file a particular string was found. Given the construction of Snort logs, this gave me an accurate accounting of the number of alerts, and it reduced the chance for me to make an error based on the default output of wc. I worked using both Linux and Windows 2000 systems. I often used simple find commands in a windows environment to double check the formats of the various log types (eg SnortA, SnortS, and OOSche) to ensure I constructed the egrep strings correctly.

Attachment 1

Kuang2 pSender v0.21
Kuang2 pSender FULL v0.34

Dedicated to the peace in Yugoslavia!

By using this program you support Anti-NATO campaign.
Stop killing! Stop lies! Stop bombs! Stop war! Stop deaths!

First of all, forgive me for my poor English.
Note: this progie was made just for education purposes. You use it on your own risk! I am not responsible for any damage.

What is this?

This is part of my **Kuang2** project.

Kuang2 pSender is a small Trojan horse, that will send on your e-mail somebodies internet passwords. There is no way to hide passwords from it: even if victim change its passwords, or do not save the password in 'Connect to' dialog, even if victim use alternate way of connection - **Kuang2 pSender** will always update new passwords to your e-mail. Also, this program could be used like universal plug-in for any Trojan horse.

Kuang2 pSender FULL is much powerfull. It send also any typed password wherever and whenever it was typed (not only for internet provider). So you can get hotmail passwords, web sites passwords, some personal passwords for personal programs, etc. Everywhere a victim types a password it will be send. New version **v0.34**

has also special care for the Internet passwords.

How to use it?

First you *must* to setup **Kuang2 pSender** or **Kuang2 pSender FULL** .exe file. You need to enter SMTP server address, destination e-mail address on that server, and, optionally, source address (only in cases when SMTP server need existing domen for the source address). After that, you can change the name of .exe file and you are ready. You can infect somebody in 3 different ways:
1) send .exe to him and ask him to start it, or go to your friend and run&delete **Kuang2 pSender**.
2) use any kind of loader, so you can send him a joke program that first run **Kuang2 pSender**. You also can use my **Kuang2 tLoader** programs for this, if you dont want to make your own loaders.
3) if a victim is infect with some Trojan horse, you can upload .exe to victim computer, run it and delete it.

Weird
[ThuNderSoft]
<weird173@yahoo.com>
http://members.tripod.com/~weird173
http://move.to/weird

Attachment 2

<< Stealth Email SMTP Autosender ModulE (SESAME), Version 1.02 <<

for Win 95, Win 98 and Win NT.

(c) IOPUS Software http://www.iopus.com

About SESAME:

SESAME is a unique tool that allows you to supervise every
kind of file on your PC and email it automatically to your
SMTP / POP3 email account. For use with security applications
like the well-known PC#Protect Access Control SESAME can be

run invisibly in the background (STEALTH mode), i. e. it does
not show up in the task bar, system tray or task list.

SESAME automatically sends any kind of program output (logfiles,
measurement results, configuration files, text files, WORD documents...)
from your office, offsite or customer PC to your email account.
System administrators can use it to get a message whenever a specific file
is changed on one of the administrated PCs. This automatic mailing
can be triggered by various criteria like a fixed time interval,
a change in the supervised file or in its file size.

SESAME encodes the file attachments in a way which is automatically
decoded by all popular email clients like MS Outlook, Eudora,
Netscape Messenger and many others. SESAME can be used with email accounts
that require POP3 before SMTP for user authentication purposes.

For more information, please see:

=> license.txt for the shareware license and legal disclaimer
=> register.txt for registration details
=> history.txt for the list of changes
=> user_feedback.txt for user feedback from previous releases
=> help.htm for instructions (HTML based help)
=> http://www.iopus.com for information, reported bugs, FAQ and FREE updates
=> support@iopus.com send an email to IOPUS

This package will install the
following files on your PC:

****
File list:
****
SESAMEctrl.exe SESAME Control Center
SESAMEsys.exe SESAME Email Robot
see32.dll The SESAME program library
(used by SESAMEctrl.exe and SESAMEsys.exe)

help.htm HTML based documentation main file
h1.htm, These files are also part of
h2.htm, the HTML documentation
h3.htm,
hg1,
hg2,

hg3

uninst.exe	SESAME uninstallation
uninst.dll	Program library for uninstallation

readme.txt	this text
license.txt	License and redistribution information
register.txt	Registration information
history.txt	Version history (changes between the different releases)

Note: No system files are overwritten and no
modifications in your PC settings are made
during setup. The SESAME Uninstall can
completely erase SESAME from your PC without
any trace should you ever require to do so.

****
****
Thanks to all the user's of the previous version for their helpful feedback.
Please keep the suggestions coming in for this release, too. The frequently
asked feature to check and email multiple files will come with SESAME 2.0 .

Anything else YOU want to see in 2.0 ?

As usual with all IOPUS Software, lifetime FREE UPGRADES for registered users.
****
****

(c) IOPUS 1998, 1999

Last Updated: May 23rd, 2017

Upcoming Training

Mentor Session - SEC503	**Oceanside, CA**	**May 29, 2017 - Jun 29, 2017**	**Mentor**
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**
SANS Baltimore Fall 2017	**Baltimore, MD**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
Baltimore September 2017 - SEC503: Intrusion Detection In-Depth	**Baltimore, MD**	**Sep 25, 2017 - Sep 30, 2017**	**vLive**
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**
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**
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**
SANS San Diego 2017	**San Diego, CA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SIEM & Tactical Analytics Summit & Training	**Scottsdale, AZ**	**Nov 28, 2017 - Dec 05, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**