Global Information Assurance Certification Paper

Behzad Torabi

GIAC Certification Assignment V3.0

SANS, Parliament Hill
Ottawa, CANADA
October, 2001

Table of contents: **Page:**

Assignment 1- Describe the State of Intrusion Detection:

The ever evolving attacks and IDS,
How to teach the IDS some new tricks.

Introduction:

As the average speed of computing and consumer Internet connections increases, we are facing an unprecedented number of ever evolving malicious code spreading over the Internet. These new programs no longer fall in the clear definitions of Virus, Trojan or Worm (1). In this paper, I would use the term "worm" for the ones that are network aware and self- propagating. By definition, a worm is a self–reliant piece of software code that would infest a computer and, by using it's resources, propagate itself to other systems. Any successful worm shows at least four out of these five components:

- Probing.
- Remote Control interface.
- Network readiness.
- Some level of Intelligence.
- Propagation including, and not restricted to, Replication.

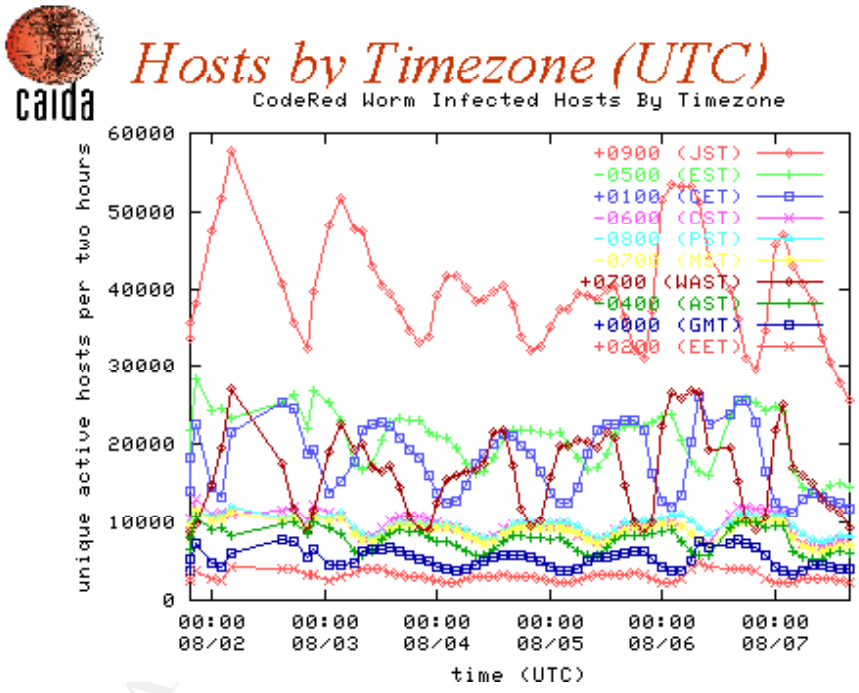
The new worms are smarter, more sophisticated and are getting more resilient. Also, worms these days are not merely there to serve a lonely programmer's pride or to do some damage to personal computers and hunt for personal information. The new worms are carefully written, cautiously deployed and are ever increasingly used as a platform to launch DDOS attacks against specific targets (2).

Although the presence of a large infestation of worms used to make headlines only when they were used as a launching pad for coordinated attacks on highly visible targets, the Internet community and law enforcement are much more aware and prepared these days. With every new reported spread of worms, there's almost and immediate attention and advanced warning which proves not only the awareness, but the effectiveness of current signature based IDS systems being used to detect the suspicious activities.
While this is the case today, it almost certainly would not be the case tomorrow. I would try to show that even if we may feel a bit more aware and secure these days, it's not hard to predict the next steps in evolution of Internet worms and how we may all be at risk if our IDS and know-how won't evolve in the same way. In that regard, I will examine the most recent worm infestations and try to draw a picture of a "perfect-worm". By examining how a next generation worm may be designed to behave and survive "in the wild", we will gain a better understand of what we should prepare ourselves for.

Case study – The evolution of Code Red Worm:

The first release of Code Red worm was introduced on July 12th, 2001. The characteristics of this worm are very well documented (3), so I would focus on its evolution from Version 1 to Version 2 and subsequently to Code Red II.

> *A look at the graph below shows how one of the most successful worms (Code Red) infected systems globally over the first six days of September 2001 (Thanks to Caida.org http://www.caida.org/outreach/presentations/usenix0108/wips/)*



Exploiting a well known vulnerability (4) in Microsoft IIS server, the first version of this worm used a randomly generated list of IP addresses to probe and eventually spread to. Since this version of the worm used a static seed (10) to generate the address list, this list would have been identical on all infected machines. This, in part, resulted in limited spread of the worm over time because after a while, most of the machines in the random list were either impregnable to the attack or already infected. Although this version of Code Red created problems such as increased use of system and network resources, launching DDOS attacks against www1.whitehouse.gov and defacing some websites, its effects were negligible compared to the compound growth effects of Code Red, version 2.

On July 19th 2001, version 2 of Code Red worm started to infect the un-patched machines running Microsoft IIS server. This version of Code Red was identical to version 1 except a single difference. Since version 2 was using a random seed to generate the list of IP addresses, the lists being compiled on infected machines were completely different. This simple change produced a much larger effect. Only after 14 hours in the wild, Code Red version 2 infected more than 350,000 computers world wide. The traffic generated by its probing brought a lot of websites, networks and routers down and created severe bandwidth shortages for the others.

On August 4th, a completely new worm called Code Red II (CRII) began to exploit the same buffer overflow vulnerability in IIS. Although this worm had a brand new code, it had the string "Code Red II" programmed in its source file. CRII was a much more sophisticated worm, it not only used a random seed as a basis to generate the list of IP addresses to probe, but it also used the host's network mask to decide on what IP addresses to probe. Also, CRII was not memory resident like Code Red so rebooting the infected machine won't remove the worm. One of the more interesting behaviors of CRII was that it used a limited number of threads (600 for machines with Chinese characters enabled and 300 for the others) to probe other machines. At last, CRII didn't deface websites or do harm to it's host machine. Instead, it installed backdoors so the host may be used as a Zombie in the future.

As is shown in the evolution of this worm, Code Red evolved from a rather insignificant worm to a global problem in a matter of days and then evolved to something much more sophisticated with long lasting effects in less than 20 days. Although Code Red is in the process of being wiped out, the residual effects of it are still being felt as increased network traffic, use of additional resources and partially overwhelmed boundary firewalls.

New threats and signature based IDS systems:

A signature based IDS systems is only as good as it's database. It's a well known fact that although we may be covered from a previously detected and described threat, given that we updated our IDS database accordingly, we may still be vulnerable to any new exploits which may exist out there. Although keeping up to date with latest patches may look like the obvious path, in practice even a good system administrator would always be a few days (if not a few weeks) behind in distributing the critical patches to his/her systems. This situation is much worse with high-maintenance operating systems because they simply need more time and attention to keep them up to date. This is a contributing factor to why these systems are being targeted more often.

Since we would always be trying to catch up with the latest treats, the time becomes a major factor in stopping any future malicious code in it's steps. With ever more sophisticated worms being found in the wild these days, it's not hard to imagine a piece of code replicating itself in a secretive way for a while. For example, a malicious code may be released in the wild that mimics the normal traffic patterns of hosts around it to avoid detection. If the code is successful in evading detection while gradually populating itself, there won't be that much time for the security community to react when it becomes active. Furthermore, if this time is long enough, by the time the efforts of security community bares fruit, its too late to neutralize the code in time. In the meanwhile, the worm would infect enough nodes to become a real crisis. In a Scenario like this, a detect and kill agent will not be available until it's too late. Although we haven't seen any code that resembles such a threat just yet, bits and components of it are already taking shape in much simpler worms(9).

Designing a "super-worm":

To design a super-worm, we would have to see what may buy it enough time so it can conquer a large mass of nodes on the Internet. Learning from nature as the master designer of highly infectious and resilient entities, I chose the following characteristics for our super-worm:

- ***Adaptive:*** An adaptive worm would learn about the new weaknesses in it's current and future hosts and will adapt itself to exploit them as the means to propagate itself. The super-worm may learn about these new vulnerabilities either by receiving updates from a master (now) or by discovering them by some built-in intelligence (future). An adaptive worm may be released targeted for only one platform but by receiving new components, it may evolve to attack a different platform as well.
- ***Secretive:*** A super-worm would hide it's intentions, signature and communications as long as possible. One of the ways a worm can hide its traffic is to merge them into similar traffic so the IDS would be more likely to miss it. For example, if a super-worm has infects an ftp server, it may use the tcp port 21 as the source port so its traffic would blend in with other ftp traffic. Also, by examining the traffic patterns on the server, the super-worm may decide to use a time-window when the communication is at it's peak so it can easily evade detection. Although encrypted traffic would be one of the best ways for the worm to communicate with its master, an IDS would have a much better chance of spotting these types of traffic where there should be none. To better cover its tracks. our super-worm may change the status of it's hosts interface to "sniff" the local network for the communications coming from its master. This way, the master can send information destined for an address "near" the infected host

without revealing the identity of the infected node(6).

- ***Resistant:*** Even if our super-worm changes its components, it would still have a core that will not change and can be used as a target in signature based IDS and virus detection software. To combat this, the super-worm may be designed to be resistant. This would not only mean that the worm would leave traces of itself that remain dormant for sometime before becoming active again, but it may also mean that the worm would actively fight the attempts to patch the vulnerability which allowed the worm to infect the node in the first place. A resistant worm may be really hard to combat because it may sense the removal attempts, pretend to be removed, and then re-open the vulnerability after a random period of being inactive.

Fighting a super-worm:

As we shown, fighting a super-worm would be extremely difficult. Since this type of worm would change, hide and resist, the best way to combat it would be to try to defuse it in different ways at a same time. One of the biggest contributing factors to the success of a worm would be the effect of its compound growth and it would probably be most effective to fight the worms using the same technique. A multi-layer defense to combat our super- worm may consist of the following components:

- ***Active IDS:*** Using "behavior based" intrusion detection, it is possible to look for irregularities in the pattern of network communications. Although these patterns would be extremely hard to identify in smaller networks, they will stand out in much larger scale which would be an indication of worm activity.
- ***Probing component:*** This would be used to identify the infected networks. Although the super-worm would try to stay hidden, sooner or later there would be a distinct anomaly in usual traffic patterns in large scale due to the effect of compound growth.
- ***False commander:*** Once the method of communications between worm and commander are identified, a set of false commanders may be utilized to make infected nodes to identify themselves so they can be cleaned.
- ***Defense agents:*** These types of "friendly worms" may be utilized under extreme circumstances to defuse a super-worm threat. To make sure these worms won't have a long lasting effect on Internet traffic, they may be designed to infect the core component of the super-worm using the command interface, propagate to secondary infected systems using the same algorithm of the host worm and finally destroy itself as well as the host worm (7).

Conclusion:

Although we are much less vulnerable to threats from traditional malicious code these days, there are compelling evidence out there that the new generation of worms and viruses are not far away. While IDS, firewall and virus detection technologies come a long

way since the days of W.com (8) worm, the development of new software would always leave some level of exploit open that may be used by malicious code.

This paper tried to show an emerging pattern in evolution of Internet worms and the new challenges that security experts should brace themselves for. I tried to show how smarter components are emerging in the design of new worms and attempted to picture of where we may be heading to. We shown the challenges that IDS needs to overcome and made an effort to design a multi-layered defense mechanism.

Internet today is much more than just a media for news and email to flow through. For the professionals that launch the attacks these days, there are goals to be accomplished and money to be made. Attacks are also becoming more common as tools to make political statements or even open new fronts in a war. This would mean that the pattern of curious teenage hackers releasing worms in to the wild out of boredom is gradually being replaced by well funded professionals on a mission. To face this new threat, a much more collaborative approach is needed between security organizations and experts, if we are to keep up with what is awaiting us. Obviously such a collaboration will not happen overnight but we have already taken the first baby steps when law enforcement agencies and security experts started to work hand in hand to apprehend the hackers. Probably what we all need to understand here is that working together is not an option but a necessity if we are to stand a chance in the shadow of threats to come.

References:

1) http://www.sans.org/infosecFAQ/malicious/threats.htm
2) http://www.zdnet.com/zdnn/stories/news/0,4586,2435149,00.html?chkpt=zdhpnews01
3) http://www.eeye.com/html/Research/Advisories/AL20010717.html
http://www.incidents.org/react/code_redII.php
http://www.caida.org/analysis/security/code-red/
4)
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/security/topics/codeptch.asp
5) http://www.sophos.com/virusinfo/analyses/w32hybrisb.html
6) http://www.crimelabs.net/docs/worms/worm.pdf Page 16
7) For more information on "Cheese worm" visit:
http://www.idg.net/idgns/2001/05/18/NewWormTriesToFixInfected.shtml
8) http://www.ciac.org/ciac/bulletins/a-02.shtml
9) For more info on worms with upgradeable components:
http://news.cnet.com/news/0-1005-200-3726171.html?tag=st.ne.1002.thed.ni
10) For a definition of random seed:
http://depts.washington.edu/smccweb/courses/401-3/randseed.html

Assignment 2 – Network Detects:

Checkpoint Firewall-1 Log file format (http://www.chekpoint.com)

The log file format is as follows:

[date] [time] [actions] [firewall] [interface] [protocol] [source IP] [destination IP] [destination port/ service] [source port] [length] [firewall rule number]

For example, the following line indicates a typical log entry for Firewall-1.
12Oct2001 3:9:34 drop *firewall* >*interface* proto udp src 172.21.79.145 dst x.y.z.150 service udp-500 s_port 30819 len 84 rule *1*

Solaris Snoop command (http://www.sun.com)

Just like tcpdump, snoop command in Solaris has many different options that would result in different representation on data. The most common use for snoop for summary output is "Snoop –d *interface*" which would produce a result with the following format: [source IP] -> [destination IP] [Service] [source port]

Snort alert log (http://www.snort.org)

"Syslog" can be used as logging server for snort, if it is configured to use it:
[Date] [time] [source IP: port number] [destination IP: port number] [tcp flags or udp]

Zone Alarm personal firewall (http://www.zonelabs.com)

ZoneAlarm is a small application firewall which is free for personal use at home. The firewall creates a decent enough logfile with the following format:

[Direction of traffic] [date] [time] [source IP: source port] [destination IP: destination port] [protocol | type/ flags]

Sources of detects:

I have use three major sources for my detects:

1- Core network of a consumer DSL service: This location has about 150,000 customers connected to this core network for their IP services and in the past has proven to be a source of "interesting" traffic. The logs have been collected on the boundary firewall (Checkpoint Firewall-1 running on Solaris) that guards this core network from the Internet. "alert.sh" by Lance Spitzner (http://www.enteract.com/~lspitz/intrusion.html) was used as Firewall-1 IDS.
2- Boundary firewalls between a corporate network and the Internet. Again, the firewalls are Checkpoint running on Solaris.
3- ADSL Broadband network at home running ZoneAlarm personal firewall on

two WIN2K machines.

Detect#1:

1-Source of trace:

Trace is from a Checkpoint Firewall-1 deployed at the boundary of DSL core network and the customer broadband networks.

2- Detect was generated by:

This detect is from a Firewall-1 log processed by alert.sh IDS for FW1 to raise awareness of these types of activity. The number of events are being matched against a threshold and an alert is being sent once the events exceeds that.

3- Probability of source address being spoofed:

By just examining the firewall logs, this looks just like a legitimate Adobe Server 2 (tcp port 1103) replying to connection requests by a source that spoofed our IP address. By looking closer at the actual packets though (this is an ongoing attack), we can see the TCP sequence numbers coming from the source would change after four packets, which is an evidence of active retry. Based on this, the source IP address most likely is not spoofed. Also, the packet is coming from only 3 hops away from us (unless the IP header is crafted), which is the correct number of hops for a host in our consumer IP address space.

4- Description of attack:

The firewall logs indicate numerous attempts by 10.11.70.148 (broadband DSL customer) to connect to two of our DNS servers on port 60000:

```
13Oct2001 13:57:11 drop   dsl_fw >int0 proto tcp src 10.11.70.148 dst 1.1.1.150 service 60000 s_port
1103 len 44 rule 22
13Oct2001 13:57:43 drop   dsl_fw >int0 proto tcp src 10.11.70.148 dst 1.1.1.130 service 60000 s_port
1103 len 44 rule 22
13Oct2001 13:58:49 drop   dsl_fw >int0 proto tcp src 10.11.70.148 dst 1.1.1.130 service 60000 s_port
1103 len 44 rule 22
13Oct2001 14:04:22 drop   dsl_fw >int0 proto tcp src 10.11.70.148 dst 1.1.1.150 service 60000 s_port
1103 len 44 rule 22
```

Looking inside the packets (this is an ongoing attack), an interesting pattern emerges. The source IP address tries the connection with the same TCP sequence number four times before selecting a different sequence number:

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 1 arrived at 13:57:11
ETHER:  Packet size = 60 bytes
ETHER:  Destination = 8:0:20:a2:af:9b, Sun
```

```
ETHER:  Source      = 0:90:86:d4:78:38,
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 44 bytes
IP:   Identification = 5452
IP:   Flags = 0x0
IP:         .0.. .... = may fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 252 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = 4c6c
IP:   Source address = 10.11.70.148,
IP:   Destination address = 1.1.1.150,
IP:   No options
IP:

TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 1103
TCP:  Destination port = 60000
TCP:  Sequence number = 524578816
TCP:  Acknowledgement number = 0
TCP:  Data offset = 24 bytes
TCP:  Flags = 0x02
TCP:        ..0. .... = No urgent pointer
TCP:        ...0 .... = No acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..1. = Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 1024
TCP:  Checksum = 0xbcd2
TCP:  Urgent pointer = 0
TCP:  Options: (4 bytes)
TCP:    - Maximum segment size = 512 bytes
TCP:
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 2 arrived at 13:57:43
ETHER:  Packet size = 60 bytes
ETHER:  Destination = 8:0:20:a2:af:9b, Sun
ETHER:  Source      = 0:90:86:d4:78:38,
ETHER:  Ethertype = 0800 (IP)
```

```
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 44 bytes
IP:   Identification = 5453
IP:   Flags = 0x0
IP:         .0.. .... = may fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 252 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = 4c6b
IP:   Source address = 10.11.70.148,
IP:   Destination address = 1.1.1.150,
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 1103
TCP:  Destination port = 60000
TCP:  Sequence number = 524578816
TCP:  Acknowledgement number = 0
TCP:  Data offset = 24 bytes
TCP:  Flags = 0x02
TCP:        ..0. .... = No urgent pointer
TCP:        ...0 .... = No acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..1. = Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 1024
TCP:  Checksum = 0xbcd2
TCP:  Urgent pointer = 0
TCP:  Options: (4 bytes)
TCP:    - Maximum segment size = 512 bytes
TCP:
```

The attacker attempts a connection to port 60000 on two of our primary DNS servers roughly once every minute. This has been ongoing for the past week or so. Port 60000 is the source port that DeepThroat Trojan uses to connect to its master, usually on port 2140.

5- Attack mechanism:

We've been seeing these connection attempts for a while. I was unable to find a known

service or Trojan that uses tcp-60000 as a destination port. The closest that I came to was a DeepThroat server (version 2 or version 3) attempting to connect to an infected host but the usual pattern for that is the infected host trying to connect to the server (using UDP) from source port 60000 to destination port 2140. Even if the connection attempt was via UDP (which is not) it seems highly unlikely that a master would try to connect to our DNS servers thinking they are infected because our we are not running a Windows platform. As this is a clear violation of our ADSL usage policy, we have sent a warning to the user of this IP address (our own Broad Band customer) about this but this is either an attempted connection for a backdoor, or more likely a misconfigured application on user's machine.

6- Correlations:

I have not been able to find any correlation to this looking at incidents.org and searching the web using several search engines.

7- Evidence of active targeting:

These are primary DNS servers for our the DSL network. Since there are no other servers being probed in that address range, at first this can be categorized as active targeting but since we have not been able to establish an attack pattern here, this more likely is just an incidental targeting.

8- Severity:

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

Criticality - These are two out our three primary DNS servers which provide service for a lot of customers, so criticality is 4.

Lethality – We have no process listening on port 60000 on any of those servers, so lethality is 1.

System Countermeasures – Two of the systems are recently patched and the third one is new with latest releases of software and operating system. A probe of open ports on those servers revealed no open ports other than the required ones, so I give this a 4.

Network Cuntermeasures – Since the firewall stopped the attempted connections, the score is a 4.

Severity = (4+1) – (4+4) = -3

9- Defensive recommendations:

This has been going on for almost a week and was investigated a few days ago. We were unable to find anything listening to this port on the targets but because of the sheer annoyance that it creates, we have sent the customer a warning to stop these connection attempts. Firewall dropped the packets and defenses are good.

10- Multiple choice test question:

What known Trojan uses port 60000 for connection to its master?

a. Back Orffice, node uses source port 60000 to connect to master on port 31337
b. Sub Seven 2.1, node uses source port 27374 to port 60000 on master.
c. DeepThroat v2, node use source port 60000 to port 2140 on master.
d. Port 60000 is used by Adobe Server and is not used by a Trojan.

Answer: c

Detect#2:

1-Source of trace:

Trace is from a Checkpoint Firewall-1 deployed at the boundary of corporate network and the customer broadband networks with logs processed by *alert.sh* IDS to raise awareness.

2- Detect was generated by:

This detect is from a Firewall-1 log processed by alert.sh IDS for FW1 to raise awareness of this type of activity.

3- Probability of source address being spoofed:

The TCP connections on port 53 (DNS) are primarily used for DNS zone transfers. The possibility of the source address being spoofed is slim because otherwise the attacker would not receive the zone transfer data. At the same time, our corporate DNS servers don't transfer zones with any outside servers and a spoofed source would serve no purpose for the attacker.

4- Description of attack:

There were two attempts by 193.107.50.2 to connect on TCP port 53 on our corporate DNS servers which serve both our internal and external customers.

```
11Oct2001  7:47:48 drop   corp_fw>int3 proto tcp src 193.107.50.2 dst xxx.yyy.zzz.64 service domain-tcp
s_port 4101 len 40 rule x
11Oct2001  7:49:00 drop   corp_fw>int3 proto tcp src 193.107.50.2 dst xxx.yyy.zzz.64 service domain-tcp
s_port 4109 len 40 rule x
```

The firewall has dropped both connection attempts.

5- Attack mechanism:

This may well be an innocent attempt by a user in Europe to do a DNS query, but those queries are usually UDP not TCP. It is hard to judge what this user was attempting to do without having the actual connection dumps but this is either a crude attempt to try a zone transfer against our DNS server or the action of an automated agent, like Adore Worm, to exploit a vulnerability in BIND (http://www.sans.org/y2k/adore.htm). Once the TCP session is established, crafted UDP packets are used to exploit the BIND information leak vulnerability (http://www.cert.org/advisories/CA-2001-02.html)
Adore worm is native to RedHat 7.0 and it would have been helpful to know the platform of the attacker but subsequent attempts to fingerprint the attacker's OS were unsuccessful.

```
dns:$> newwhois 193.107.50.2
```

European Regional Internet Registry/RIPE NCC (NETBLK-RIPE)
These addresses have been further assigned to European users.
Contact info can be found in the RIPE database, via the
WHOIS and TELNET servers at whois.ripe.net, and at
http://www.ripe.net/db/whois.html
NL

Netname: RIPE-CBLK
Netblock: 193.0.0.0 - 193.255.255.255
Maintainer: RIPE

Coordinator:
Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN)
nicdb@RIPE.NET
+31 20 535 4444

Domain System inverse mapping provided by:

NS.RIPE.NET	193.0.0.193
NS.EU.NET	192.16.202.11
AUTH03.NS.UU.NET	198.6.1.83
NS2.NIC.FR	192.93.0.4
SUNIC.SUNET.SE	192.36.125.2
MUNNARI.OZ.AU	128.250.1.21
NS.APNIC.NET	203.37.255.97

To search on arbitrary strings, see the Database page on
the RIPE NCC web-site at http://www.ripe.net/db/

Record last updated on 16-Oct-1998.
Database last updated on 13-Oct-2001 23:19:15 EDT.

6- Correlations:

http://www.cert.org/incident_notes/IN-2001-03.html
http://www.isc.org/products/BIND/bind-security.html

The Adore worm was very active around April and this may be a residual effects of the worm.

7- Evidence of active targeting:

Although our corporate DNS server is very well known, this seems to be an isolated incident an not an act of active targeting.

8- Severity:

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

Criticality – This is one our primary corporate DNS servers which provide service for a lot

of customers, so criticality is 5.

Lethality – There is no evidence that a connection was allowed. Our bind is the latest release with no known vulnerability so the lethality is 1.

System Countermeasures – Systems is recently patched is running the latest patch level of BIND. I give this a 4.

Network Countermeasures – Since the firewall stopped the attempted connection, the score is a 4.

Severity = (5+1) – (4+4) = -2

9- Defensive recommendations:

Protection seems to be sufficient. We are running the latest patch levels of Solaris and the latest release of bind 9.x. At this time, there are no known vulnerabilities with either of them.

10- Multiple choice test question:

An evidence of active Adore worm would be:

a. A lot of attempted connections on TCP port 53 in a very short time.
b. A crafted TCP packet destined for TCP port 53.
c. A RedHat Linux machine attempting UDP connections on port 53.
d. A RedHat Linux machine attempting TCP connections on port 53.

Answer: d

Detect#3:

1-Source of trace:

Trace is from my home WIN2K machine running ZoneAlarm personal firewall. This machine is connected to the Internet via Broad Band ADSL.

2- Detect was generated by:

This detect is from ZoneAlarm personal firewall processed and sorted using Microsoft Excel.

3- Probability of source address being spoofed:

This looks like an OS fingerprinting attempt so the attacker would need to get an answer back and because of this, the source IP address is probably not spoofed. There is a slim possibility that the source address may be spoofed and attacker has access to the same network segment as the spoofed address and using promiscuous mode of the interface, is listening to the responses. In this case, this is unlikely because this looks like a one time scan.

4- Description of attack:

FWIN,2001/10/07,22:16:12 -7:00 GMT,210.255.48.82:0,my.home:23,TCP (flags:SF)

There is an attempt to connect to port 23 (telnet) on my home machine which may look like a simple port scan. Looking closer at the firewall log reveals this been done with SYN/FIN flags set and the connection originated from port number 0. These are both evidence of a crafted packet because the source port 0 is not a legitimate port number and standard TCP hand shake won't allow the SYN/FIN flags set simultaneously. This raises the possibility of OS fingerprinting (1) rather than port scan. The source IP address is from UNIMAC Co. In Japan.

Net1:$ > whois -h whois.apnic.net 210.255.48.82

% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html
% (whois6.apnic.net)

inetnum: 210.248.0.0 - 210.255.255.255
netname: JPNIC-NET-JP
descr: Japan Network Information Center
country: JP
admin-c: JNIC1-AP
tech-c: JNIC1-AP

```
remarks:    JPNIC Allocation Block
remarks:    Authoritative information regarding assignments and
remarks:    allocations made from within this block can also be
remarks:    queried at whois.nic.ad.jp. To obtain an English
remarks:    output query whois -h whois.nic.ad.jp x.x.x.x/e
mnt-by:     APNIC-HM
mnt-lower:  MAINT-JPNIC
changed:    apnic-ftp@nic.ad.jp 19991115
source:     APNIC

role:       Japan Network Information Center
address:    Kokusai-Kougyou-Kanda Bldg 6F, 2-3-4 Uchi-Kanda
address:    Chiyoda-ku, Tokyo 101-0047, Japan
country:    JP
phone:      +81-3-5297-2311
fax-no:     +81-3-5297-2312
e-mail:     hostmaster@nic.ad.jp
admin-c:    NM6-AP
tech-c:     YM15-AP
tech-c:     IK6-AP
tech-c:     KM19-AP
nic-hdl:    JNIC1-AP
mnt-by:     MAINT-JPNIC
changed:    apnic-ftp@nic.ad.jp 19990629
changed:    hostmaster@apnic.net 20011011
source:     APNIC

inetnum:    210.255.48.80 - 210.255.48.87
netname:    UNIMAC-NET
descr:      UNIMAC Co.,Ltd.
country:    JP
admin-c:    TN2056JP
tech-c:     TN2056JP
remarks:    This information has been partially mirrored by APNIC from
remarks:    JPNIC. To obtain more specific information, please use the
remarks:    JPNIC whois server at whois.nic.ad.jp. (This defaults to
remarks:    Japanese output, use the /e switch for English output)
remarks:    This information has been partially mirrored by APNIC from
remarks:    JPNIC. To obtain more specific information, please use the
remarks:    JPNIC whois server at whois.nic.ad.jp. (This defaults to
remarks:    Japanese output, use the /e switch for English output)
changed:    apnic-ftp@nic.ad.jp 19990811
changed:    apnic-ftp@nic.ad.jp 20011010
source:     JPNIC
```

5- Attack mechanism:

This is the reconnaissance stage of an attack. At this stage, the attacker is looking for the desired platform to launch an attack. The other piece of information that is suspicious in

here is the source port 0 which is not used in standard TCP communications and is not likely to be the source port for an attempted connection to telnet daemon. It would have been more helpful if I had a tcpdump capture of communication attempt as the current evidence implies, this is a crafted packet.

6- Correlations:

There are a lot of incidents like this reported at incidents.org like:
http://www.incidents.org/archives/intrusions/msg01725.html
http://www.incidents.org/archives/y2k/050400-0930.htm
http://www.incidents.org/archives/intrusions/msg00876.html

These incident reports have a lot in common with my scan.

7- Evidence of active targeting:

This was a one time scan and was not repeated again so there is no evidence that my machine was actively targeted.

8- Severity:

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

Criticality – Although rebuilding this machine would take a good day if its ever compromise, it is a mission critical machine so criticality is 3.

Lethality – I have no telnet daemon running on this machine so I give it a 1.
System Countermeasures – System is running the latest security patches so it's a 4.
Network Countermeasures – The firewall stopped the probing in its tracks. My Firewall is configured to allow only outbound connections with no exceptions and I have the latest release of it running so the score is a 4.

Severity = (3+1) – (4+4) = -4

9- Defensive recommendations:

This was a single probe. I am running the latest security patches from Microsoft and my firewall is the latest release and successfully dropped the packet. No extra defensive measures are required.

10- Multiple choice test question:

A TCP packet with SYN/FIN flags set is:

a. An OS fingerprinting attempt.
b. Part of a three way handshake of a TCP connection.
c. Part of a SYN flood.
d. an indication that the host is infected with Adore worm.

Answer: a

1) Operating System Finger Printing:

One of the first stages of a targeted attack is reconnaissance and is mostly done via a process known as OS finger printing. At this stage, the attacker may choose to craft a packet that contains values that are not permitted in the RFC. Since there is no standard way for operating systems to react to such crafted packets, each OS may behave differently when it encounters such a packet. By analyzing the response from the host machine, the attacker may be able to identify the victim's operating system and target the next attacks more specifically for that.

Detect#4:

1-Source of trace:

Trace is from a Checkpoint Firewall-1 deployed at the boundary of DSL core network and the customer broadband networks with logs processed by *alert.sh* IDS to raise awareness.

2- Detect was generated by:

This detect is from a Firewall-1 log processed by alert.sh IDS for FW1 to raise awareness of this type of activity. The number of events are being matched against a threshold and an alert is being sent once the events exceeds that.

3- Probability of source address being spoofed:

The source address doesn't seem to be spoofed. As I would explain in the "Attack mechanism" section, this is an active scanning and the source needs to get the reply back, so the chances of source being spoofed are slim.

4- Description of attack:

The firewall logs indicate numerous attempts by 172.21.79.145 to locate servers who respond to UDP port 500 (IPSEC VPN):

```
13Oct2001 13:26:22 drop   dsl_core_fw >int0 proto udp src 172.21.79.145 dst 1.1.1.43 service udp-500
s_port 30261 len 84 rule x
13Oct2001 13:27:24 drop   dsl_core_fw >int0 proto udp src 172.21.79.145 dst 1.1.1.45 service udp-500
s_port 30260 len 84 rule x
13Oct2001 13:27:25 drop   dsl_core_fw >int0 proto udp src 172.21.79.145 dst 1.1.1.49 service udp-500
s_port 30261 len 84 rule x
```

It looks like that the attacker is actively looking for VPN servers with Internet Key Exchange (IKE). As I was able to find at least one weak implementations of this protocol (http://www.securiteam.com/securitynews/5MP060A3QW.html) regarding the Nortel Networks CES (both DES* and 3DES versions), this may be an active probe. Nortel VPN servers with versions prior to 3.50 use only DES during the Security Association phase 1 (ISAKMP SA), regardless of the actual security settings on the VPN box. Since a single DES is crackable in 24 hours, the attacker would be able to gain access to information being sent over a secure channel. Having said that, the chance of such an attack is remote. Another possibility of such connection attempts in a misconfigured WIN2K machine with wrong VPN setup although the usual connection pattern for such clients is from UDP port 500 (which is not the case here).

**DES:* DES applies a 56-bit key to each 64-bit block of data. The process can run in several modes and involves 16 rounds or operations. Although this is considered "strong" encryption, many companies use "triple DES" or 3DES, which applies three keys in succession. DES originated in IBM in 1977. For more information on DES, refer to: http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213893,00.html

5- Attack mechanism:

If this indeed is an attack, this would be the probing stage of such attack. Once a VPN server is found, additional fingerprinting via a more targeted attack is required to identify the VPN severs as a vulnerable one. At this stage, the attacker seems to be only probing for such VPN servers and since we don't have any, we possibly won't see any advanced phases of this attack. Although this pattern of attack is new to our core DSL, we had a lot more instances of the same pattern regarding other DNS servers on our corporate side. This opens the possibility of a misinformed user (this is a consumer network after all) with a poorly configured VPN client trying to connect to a server. Although probing all three DNS servers is an evidence of active targeting (by default, the user should only probe two out of three DNS servers), it may well be nothing but a false alarm.

6- Correlations:

The only correlation I was able to find was at http://www.securiteam.com/securitynews/5MP060A3QW.html which does not show the name of the person who posted it. As of February 2001, Nortel Networks has released a new version of code for both client and the server which fixes the problem.

7- Evidence of active targeting:

The three servers being scanned are DNS servers serving the DSL network. There is a chance that this is just a random scan but there are no other servers being probed in that address range, this can be categorized as active targeting.

8- Severity:

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

Criticality - These three are our primary DNS servers which provide service for a lot of customers, so criticality is 5.

Lethality – We have no VPN servers running in the targeted network (I hope), so lethality is 1.

System Countermeasures – Two of the systems are recently patched and the third one is new with latest releases of software and operating system, so I give this a 4.

Network Countermeasures – Since the firewall stopped the probing in its tracks, the score is a 5.

Severity = (5+1) – (4+5) = -3

9- Defensive recommendations:

Since this was a new probe, we did not get an early warning from our alert.sh script. It remains to be seen if the attacker would continue his/ her probes or that was a one time incident. I would certainly keep an eye on this user for a while and since we manage this network, we would terminate his/ her service is we see a pattern of suspicious activities.

10- Multiple choice test question:

3DES encryption if far more secure than DES because:

a. Three servers are used in 3DES, so the security is improved.
b. Client has to send three different encrypted sessions before authenticating.
c. 3DES uses three DES encryptions on a single data block.
d. 3DES uses SSL 128 bit keys compared to 40 bits key of DES.

Answer: c

Detect#5:

1-Source of trace:

Trace is from my second machine (laptop) at home running WIN2K. This machine is connected to the Internet via ADSL broadband and is running ZoneAlarm personal firewall.

2- Detect was generated by:

This detect is from a Zone Alarm personal firewall log file processed and sorted using Microsoft Excel.

3- Probability of source address being spoofed:

This is a tcp connection attempt to address an exploit in Unix LPR. Since the attacker needs the information back in order to collect information for launching an attack, the possibility of the source address being spoofed is slim. Also, the probes are repeated roughly every 24 hours from a different IP in (possibly) the same subnet which makes this look a lot less accidental.

4- Description of attack:

The firewall logs indicate numerous attempts to connect to tcp port 515:

FWIN,2001/10/04,21:01:49 -7:00 GMT,202.110.195.88:3762,my.home:515,TCP (flags:S)
FWIN,2001/10/05,23:18:27 -7:00 GMT,202.110.195.82:2715,my.home:515,TCP (flags:S)
FWIN,2001/10/06,21:55:54 -7:00 GMT,202.110.195.89:1613,my.home:515,TCP (flags:S)

This is a probe to find Linux/ Unix machine with LPR port 515 (printer spooler) open. Once the attacker finds the target, he/she would try to exploit a vulnerability that exists in Red Hat Linux 7.0 LPR and gain command access to the system.

server:$ > whois -h whois.apnic.net 202.110.195.88

% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html
% (whois6.apnic.net)

inetnum: 202.110.195.80 - 202.110.195.95
netname: QDFRTCX-COM
descr: shandong qingdao furuitai chenxi business company
country: CN
admin-c: DS95-AP
tech-c: DS95-AP
mnt-by: MAINT-ZXF
changed: zxf@sdinfo.net 20000727
source: APNIC

person: Data Communication Bureau Shandong
address: No.77 Jingsan Road,Jinan,Shandong,P.R.China
country: CN
phone: +86-531-6052163
fax-no: +86-531-6052245
e-mail: ip@sdinfo.net
nic-hdl: DS95-AP
mnt-by: MAINT-ZXF
changed: zxf@sdinfo.net 20010206
source: APNIC

Although the IP addresses are different in each probe, all of them were initiated from the same company in China. This raises the possibility of the attacker using different, but valid, IP addresses in the same subnet to evade attention and make this probe look more accidental.

5- Attack mechanism:

This is only the first stage of an attack. Once the attacker receives an ACK to this connection attempt, he/she would launch the attack targeting Linux/ Unix systems with vulnerable LPR code (print spooler version 3.6.24 or older). Once he/ she gains access to such Unix server he/she would try to send format strings to make the spooler crash or execute the attacker's commands. On most Linux/Unix systems, LPR daemon runs as root so this may give the attacker further ability to setup a backdoor into the system.

6- Correlations:

Here is a description of this Vulnerability:
http://www.sans.org/newlook/alerts/port515.htm
There are several reports of such probes at sans.org like the one below:
http://www.sans.org/y2k/041301.htm

7- Evidence of active targeting:

Since the target machine is a WIN2K system and the fact that my firewall dropped all connection attempts would imply that I should not be an active target. On the other hand, the fact that all the connection attempts were initiated from the same subnet and within 24 hours intervals implies an order that's hardly coincidental. I have examined the logs on other PC at home (same subnet) but didn't find a trace of same connection attempt. The most plausible explanation at this moment is that the attacker is running some sort of script on a single machine and my machine's IP address is in the probe list. The changing of the source address is either because attacker is getting IPs of a DHCP server or he/she is changing them everyday to make this look more random.

8- Severity:

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

Criticality – This is a WIN2K laptop with no essential application that I can't reinstall in an few hours. criticality is 3.

Lethality – This is not a Unix machine and WIN2K TCP/IP LPR is not installed so I'm giving lethality a score of 1.

System Countermeasures – System was patched almost a month ago with latest security patches so I give it a 3.

Network Countermeasures – The firewall stopped the probing in its tracks with no evidence of successful connection attempts so the score is a 4.

Severity = (3+1) – (3+4) = -3

9- Defensive recommendations:

The security provided by the LPR service not installed and the Firewall looks sufficient, no other defensive measures required.

10- Multiple choice test question:

Unix LPR service is used for:

a. Looking up printers in the local Ethernet segment.
b. Spooling the print jobs and sending them to the printer.
c. Loading the port rules in a Unix firewall.
d. Formatting the text before handing the jobs to the printer spooler daemon.

Answer: b

Assignment 3- Analyze this:

Summary:

I have been tasked to provide a security audit for an imaginary university, let's call it University Of South Pole (USP), based on Snort (http://www.snort.com) alert and Scan files generated for five consecutive days from May 4th to May 8th 2001. The original scans were taken from http://www.research.umbc.edu/~andy.

List of files used for this analysis:

- scans.010504
- alert.010504
- scans.010505
- alert.010505
- scans.010506
- alert.010506
- scans.010507
- alert.010507
- scans.010508
- alert.010508
- oos_May.4.2001
- oos_May.5.2001
- oos_May.6.2001
- oos_May.7.2001
- oos_May.8.2001

Analysis Tools and method:

I have used "SnortSnarf" from Silicon Defense (http://www.silicondefense.com) as the analysis tool to process the Snort Alert and Scan data. SnortSnarf simplifies the processing of scan/ alert files by correlating the information together and providing HTML output files in a directory structure that is easier to understand.

***Note**: Since SnortSnarf is unable to understand the MY.NET.xxx.yyy address scheme used in the scan/ alert files to hide the actual localnets, I have changed all instances of MY.NET.xxx.yyy to 1.1.xxx.yyy. This way, the localnet address remains hidden and we would be able to correlate them correctly.*

Earliest alert at **00:02:00**.982598 *on 05/04/2001*
Latest alert at **23:55:33** *on 5/8/2001*

Signature	***# Alerts***	***# Sources***	***# Destinations***
TCP **SFRPA* scan	1	1	1
connect to 515 from inside	1	1	1
ICMP SRC and DST outside network	3	3	3
SUNRPC highport access!	5	2	2
Tiny Fragments - Possible Hostile Activity	10	1	8
High port 65535 tcp - possible Red Worm - traffic	11	5	5
Back Orifice	23	2	23
Port 55850 tcp - Possible myserver activity - ref. 010313-1	24	6	7
Null scan!	24	18	19
TCP SRC and DST outside network	27	11	21
SNMP public access	43	3	43
WinGate 1080 Attempt	82	38	52
Watchlist 000222 NET-NCFC	94	8	10
Queso fingerprint	198	18	50
connect to 515 from outside	234	1	213
SMB Name Wildcard	263	96	199
High port 65535 udp - possible Red Worm - traffic	943	12	12
SYN-FIN scan!	1305	1	1305
External RPC call	1356	12	853
Watchlist 000220 IL-ISDNNET-990517	2344	42	48
UDP SRC and DST outside network	5791	32	213
Attempted Sun RPC high port access	5871	3	3
TCP **S***** scan	29169	75	18348
UDP scan	97351	200	12994

Analysis of attacks with less than 300 alerts (in descending order):

Although these alerts are not very significant, I have picked a few of them that looked interesting for further analysis. This would provide a snapshot of attacks techniques being used to target your internal machines.

1- TCP **SFRPA* scan

This scan is based on what is usually called a “Christmas Tree” packet. The scanner would try to probe a target with a TCP packet that has all TCP options enabled. Since packet type is not permitted within the definitions of RFC, every individual operating system may decide to behave differently in response to this, hence making it an OS finger

printing tool. Also, some old operating systems may get confused by this packet format and stop functioning.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
202.229.228.180	1	1	1	1

This scan was initiated from Japan from the above mentioned IP address and was destined to your internal host 1.1.217.22. This internal machine has been the subject of one other scan (SYN scan) from another network block and does not appear to be a target. If required, the administrator for this IP block can be reached at hostmaster@nic.ad.jp.

2- Tiny Fragments - Possible Hostile Activity

This attack is selected because a single attacker can be identified with it, so you may decide to block this source in the future.

Under normal circumstances, packet fragmentation happens when data is too large to be transmitted via a single packet. If fragmentation is allowed into a network, one expects to see them in packets with maximum allowable size. The activity bellow is suspicious because the fragments are too small to justify the fragmentation.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
202.39.78.125	10	10	8	8

Top four destinations:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.222.190	2	6	1	4
1.1.229.74	2	5	1	3
1.1.202.242	1	1	1	1
1.1.205.50	1	2	1	2

The source of these fragments is a network in Taiwan. If this activity happened from 5/4/2001 to 5/8/2001. The followings is the contact information for this network:

```
inetnum:     202.39.78.0 - 202.39.78.127
netname:     HU-JUN-JIA-TN-NET
descr:       Hu, Jun Jia
descr:       No.178, Bao An Rd., Tainan
descr:       Tainan Taiwan
country:     TW
admin-c:     JJH19-TW
tech-c:      JJH19-TW
remarks:     This information has been partially mirrored by APNIC from
remarks:     TWNIC. To obtain more specific information, please use the
remarks:     TWNIC whois server at whois.twnic.net.
```

mnt-by: TWNIC-AP
changed: network-adm@hinet.net 20010724
source: TWNIC

3- TCP SRC and DST outside network

I have picked up this attack as interesting because it is an indication of compromised hosts inside your network.

The traffic is originated from inside your network and is mostly destined for America Online port (TCP port 5190) or NetBIOS Name Services port (TCP 137). The most possible explanation for these alerts is that there are compromised hosts in your network using crafted packets to connect to destinations outside MY.NET (1.1). The best way to trace this traffic and find the source is to use a network sniffer to trace the MAC address is being associated to the source IP address. If that MAC address happens to be of a router, this needs to be tried on the next segment until the actual host is found.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
169.254.101.152	14	40	10	34
128.220.63.215	3	4	2	2
172.139.123.189	2	2	1	1
172.152.205.90	1	1	1	1

Top 3 destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
205.188.46.242	3	3	1	1
205.188.45.80	2	2	1	1
205.188.45.9	2	2	1	1

As can be seen the above tables, most of the attacks has been launched by the host(s) "borrowing" the 169.254.101.152 IP address. Since this has been the most active source, I have included a sample of traces originated from this address. This attack has been logged from May 4th to May 8th 2001.

05/04-09:40:15.570273 [**] TCP SRC and DST outside network [**] 169.254.101.152:1249 -> 205.188.47.20:5190
05/04-17:32:30.542846 [**] TCP SRC and DST outside network [**] 169.254.101.152:3278 -> 205.188.45.80:5190
05/04-20:26:00.405167 [**] UDP SRC and DST outside network [**] 169.254.101.152:137 -> 209.153.205.125:137
05/05-11:51:15.790226 [**] TCP SRC and DST outside network [**] 169.254.101.152:2239 -> 205.188.46.242:5190
05/05-13:20:51.759672 [**] TCP SRC and DST outside network [**] 169.254.101.152:2434 -> 205.188.45.83:5190
05/05-21:19:28.752316 [**] TCP SRC and DST outside network [**] 169.254.101.152:2440 -> 205.188.45.148:5190
05/06-13:23:55.364691 [**] TCP SRC and DST outside network [**] 169.254.101.152:4295 -> 205.188.46.242:5190
05/06-18:22:39.652577 [**] TCP SRC and DST outside network [**] 169.254.101.152:1659 -> 205.188.46.242:5190
05/06-21:55:01.585988 [**] TCP SRC and DST outside network [**] 169.254.101.152:1143 -> 205.188.45.9:5190

05/06-22:54:53.835311 [**] TCP SRC and DST outside network [**] 169.254.101.152:1622 -> 205.188.45.11:5190
05/07-16:03:54.456021 [**] UDP SRC and DST outside network [**] 169.254.101.152:137 -> 168.167.15.34:137
05/07-16:07:08.749789 [**] TCP SRC and DST outside network [**] 169.254.101.152:1243 -> 172.168.21.153:2768
05/07-16:10:50.505941 [**] UDP SRC and DST outside network [**] 169.254.101.152:137 -> 168.37.216.217:137
05/07-16:40:00.250639 [**] UDP SRC and DST outside network [**] 169.254.101.152:137 -> 158.83.203.40:137
05/07-17:13:26.981341 [**] UDP SRC and DST outside network [**] 169.254.101.152:137 -> 212.143.173.129:137
05/07-17:32:50.233126 [**] UDP SRC and DST outside network [**] 169.254.101.152:137 -> 137.216.162.14:137

4- Connect to 515 from outside:

This attack happened pretty quick in almost 15 seconds on 8:29AM, May 6th 2001 in the form a rapid scan for port 515 on two internal class C networks (1.1.132.0 and 1.1.137.0).

TCP port 515 is used by LINUX/ UNIX printer daemon (LPR) which is responsible for queuing up and sending the documents for printing. There is a known exploit in some older releases of LPR (print spooler version 3.6.24 or older) that the attacker may use to shutdown the print functions or even gain administrative access to host machine(s).

The fact that many hosts on the above mentioned networks responded with a SYN to these connection attempts indicates that there are many hosts in those networks that may be compromised if defensive action (like blocking the source 216.64.197.68 and installing the latest release of LPR on your machines) is not taken.
Top source initiating this attack:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
216.64.197.68	234	468	213	213

Top 5 destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.132.33	2	5	1	2
1.1.132.4	2	4	1	1
1.1.132.35	2	5	1	2
1.1.132.37	2	6	1	2
1.1.132.21	2	5	1	2

WHOIS Query Result for 216.64.197.68:

Exodus Communications Inc. Chicago-3,EC48 (NETBLK-EC48-1)
2831 Mission College Blvd.
Santa Clara, CA 95054
US

Netname: EC48-1
Netblock: 216.64.192.0 - 216.64.223.255
Maintainer: EC48

Coordinator:
Center, Network Control (NOC44-ARIN) CompServ@Exodus.net
(888) 239-6387 (FAX) (888) 239-6387

Domain System inverse mapping provided by:

NS.EXODUS.NET 206.79.230.10
NS2.EXODUS.NET 207.82.198.150

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
* Rwhois reassignment information for this block is available at:
* rwhois.exodus.net 4321

Record last updated on 31-Jan-2001.
Database last updated on 18-Nov-2001 19:53:59 EDT.

Analysis of attacks with more than 300 alerts (in descending order):

These alerts are far more significant because of the volume of that attack. The attacks that I have selected here present a noticeable volume and at the same time are more specific based on the categories such as source, type of attack and frequency of attacks. In each case, a list of top attackers, top listeners and a sample of traffic that triggered the attack is included. I have also included the "whois " database entries for the most active attackers for further action.

5- High port 65535 UDP – possible Red Worm - traffic

This traffic is mostly initiated from the UDP source port 65535

to destination port 27960 which is an strong indication of communication with an Internet game server called Quake 3 Arena Server (http://lists.suse.com/archive/suse-security/2000-Dec/0281.html).
Based on analysis of this traffic, it looks like that MY.NET.70.242 (1.1.70.242) is hosting a Quake 3 server. This traffic was present from May 5th to May 8th 2001.

Top 4 sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
66.79.18.70	930	930	1	1
64.40.74.155	2	2	2	2
1.1.205.34	1	60	1	37
1.1.219.46	1	1501	1	520

Top 2 destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.70.242	931	934	2	4
64.40.74.173	2	2	2	2

Sample of trace from top talker to top listener.

05/08-16:50:04.255845 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960
05/08-16:50:04.703994 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960
05/08-16:50:04.761899 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960
05/08-16:50:06.612428 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960
05/08-16:50:07.592232 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960
05/08-16:50:08.074484 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960
05/08-16:50:09.124529 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960
05/08-16:50:10.355800 [**] High port 65535 udp - possible Red Worm - traffic [**] 66.79.18.70:65535 -> 1.1.70.242:27960

WHOIS Query Result for 66.79.18.70:

Mebtel Communications (NETBLK-MEBTEL-BLK-3)
103 South Fifth Street
Mebane, NC 27302
US

Netname: MEBTEL-BLK-3
Netblock: 66.79.0.0 - 66.79.95.255
Maintainer: MEBT

Coordinator:
REITER, DENNIS (DR666-ARIN) REITERD@GALLATINRIVER.COM
3093455261

Domain System inverse mapping provided by:

DNS0.MEBTEL.NET 208.241.20.25
DNS1.MEBTEL.NET 208.241.20.26

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 07-Aug-2001.
Database last updated on 18-Nov-2001 19:53:59 EDT.

6- SYN-FIN Scan!

This traffic is an indication of reconnaissance portion of an attack. At this stage, the attacker is collecting information such as open ports and OS fingerprinting. The combination of TCP SYN (initiate connection) and FIN (terminate an already existing connection) is not permitted in RFC. The attacker would use this to see what kind of response he/she would get from the attacked host(s). Based on the information gathered in here, the attacker would plan a more targeted attacks of hosts he/she would deem as

interesting.

The fact that a single host launched this probe of your internal network would stress the possibility of active probe.

Source triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
192.168.0.1	1305	2609	1305	1308

Please note that the source IP address of 192.168.0.1 is what is called a "non-routable IP address". What this means is that no router on the Internet should route a packet to such destination and this is further evidence that the attacker is trying to hide his/ her identity by crafting a packet with a spoofed source address.

Top 5 destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.156.64	1	2	1	1
1.1.156.68	1	2	1	1
1.1.18.62	1	2	1	1
1.1.18.64	1	2	1	1
1.1.26.83	1	3	1	2

This activity started at 00:21:56 AM May 5th 2001 and lasted until 03:59:32 AM the same day.

This is a sample of the traffic initiated from the attacker:

May 5 15:25:39 192.168.0.1:21 -> 1.1.1.11:21 SYNFIN **SF****
05/05-15:25:39.843034 [**] SYN-FIN scan! [**] 192.168.0.1:21 -> 1.1.1.11:21
May 5 15:25:40 192.168.0.1:21 -> 1.1.1.27:21 SYNFIN **SF****
May 5 15:25:40 192.168.0.1:21 -> 1.1.1.43:21 SYNFIN **SF****
May 5 15:25:40 192.168.0.1:21 -> 1.1.1.45:21 SYNFIN **SF****
May 5 15:25:40 192.168.0.1:21 -> 1.1.1.53:21 SYNFIN **SF****
05/05-15:25:40.161474 [**] SYN-FIN scan! [**] 192.168.0.1:21 -> 1.1.1.27:21
05/05-15:25:40.496896 [**] SYN-FIN scan! [**] 192.168.0.1:21 -> 1.1.1.43:21
05/05-15:25:40.529710 [**] SYN-FIN scan! [**] 192.168.0.1:21 -> 1.1.1.45:21
05/05-15:25:40.673374 [**] SYN-FIN scan! [**] 192.168.0.1:21 -> 1.1.1.53:21

7- External RPC Call:

Sun RPC is a network socket mapping application used in Solaris. RPC would allow applications to run on a dynamic high port which in turn is registered with the RPC portmapper. This is known for a high number of vulnerabilities and successful attempts to

establish such a connection from the Internet should be investigated.

This traffic happened on May 7th 2001 from 08:50:36 to 09:07:51.

Top 5 sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
208.233.96.131	327	477	271	272
202.107.231.92	155	155	147	147
202.106.124.28	136	205	129	129
211.173.205.7	126	158	121	121
202.192.240.38	126	188	122	122

Top 5 destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.134.84	6	9	5	6
1.1.134.54	5	7	4	4
1.1.135.172	5	7	4	5
1.1.137.107	5	7	4	5
1.1.133.16	5	7	5	5

Trace of RPC connection attempts from top talker:

05/07-08:50:36.583005 [**] External RPC call [**] 208.233.96.131:41651 -> 1.1.100.130:111
05/07-08:50:39.950143 [**] External RPC call [**] 208.233.96.131:41651 -> 1.1.100.130:111
May 7 09:06:55 208.233.96.131:56058 -> 1.1.132.6:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56062 -> 1.1.132.10:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56059 -> 1.1.132.7:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56074 -> 1.1.132.22:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56060 -> 1.1.132.8:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56078 -> 1.1.132.26:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56086 -> 1.1.132.34:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56088 -> 1.1.132.36:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56090 -> 1.1.132.38:111 SYN **S*****
May 7 09:06:55 208.233.96.131:56089 -> 1.1.132.37:111 SYN **S*****
05/07-09:06:55.785215 [**] External RPC call [**] 208.233.96.131:56058 -> 1.1.132.6:111
05/07-09:06:55.788967 [**] External RPC call [**] 208.233.96.131:56062 -> 1.1.132.10:111
05/07-09:06:55.793568 [**] External RPC call [**] 208.233.96.131:56059 -> 1.1.132.7:111
05/07-09:06:55.807058 [**] External RPC call [**] 208.233.96.131:56074 -> 1.1.132.22:111
05/07-09:06:55.823540 [**] External RPC call [**] 208.233.96.131:56060 -> 1.1.132.8:111

Who is entry for 208.233.96.131 (top talker #1):

Wam Net Enterprises Inc. (NETBLK-UU-208-233-96)
123 NW 13th St
Boca Raton, FL 33432
US

Netname: UU-208-233-96
Netblock: 208.233.96.0 - 208.233.111.255
Maintainer: WAM

Coordinator:
Massias, Andrew (AM646-ARIN) andrew@WAMNETRUNNER.COM
407-392-9422

Record last updated on 05-Dec-1997.
Database last updated on 18-Nov-2001 19:53:59 EDT.

Who is entry for 208.233.96.131 (top talker #2):

inetnum	202.107.231.80 - 202.107.231.95
Origin	JINHUA-ADSL-TEST2
descr	JINHUA ADSL TEST NETWORK2
descr	Jinhua,Zhejiang Province
country	CN
Admin. Contact	CZ61-AP
Tech. Contact	CZ61-AP
mnt-by	MAINT-CHINANET-ZJ
changed	master@dcb.hz.zj.cn 20010207
source	APNIC
person	CHINANET ZJMASTER
address	no 378,yan an road,hangzhou,zhejiang
country	CN
phone	+86-571-7015441
fax-no	+86-571-7027816
e-mail	master@dcb.hz.zj.cn
NIC Handle	CZ61-AP
mnt-by	MAINT-CHINANET-ZJ

8- Watchlist 000220 IL-ISDNNET-990517:

This watch list has been specifically created to monitor activities from 212.179.x.x which apparently is an ISDN network in Israel. There is a variety of traffic from this network into

your internal network but the dominant traffic is for an Internet peer to peer file sharing protocol called Gnutella.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
212.179.83.94	429	429	1	1
212.179.79.2	329	329	13	13
212.179.31.157	306	306	2	2
212.179.95.5	288	288	8	8
212.179.84.7	272	272	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.207.82	429	429	1	1
1.1.229.206	307	314	2	9
1.1.214.138	272	273	1	2
1.1.207.6	257	261	1	3
1.1.203.46	241	241	1	1

Sample of traffic from top talker:

05/05-16:51:50.275716 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.94:1112 -> 1.1.207.82:6346
05/05-16:51:54.181632 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.94:1112 -> 1.1.207.82:6346
05/05-16:51:54.875651 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.94:1112 -> 1.1.207.82:6346
05/05-16:52:03.478839 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.94:1112 -> 1.1.207.82:6346
05/05-16:52:05.384791 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.94:1112 -> 1.1.207.82:6346
05/05-16:52:06.282995 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.94:1112 -> 1.1.207.82:6346
05/05-16:52:07.492098 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.94:1112 -> 1.1.207.82:6346

WHOIS Query Result for 212.179.83.94:

inetnum	212.179.80.0 - 212.179.94.255
Origin	L2TP-PROJECT
descr	2st-pool-Dailup-L2TP-client.
country	IL
Admin. Contact	NP469-RIPE
Tech. Contact	NP469-RIPE
status	ASSIGNED PA
Notify	hostmaster@isdn.net.il
mnt-by	RIPE-NCC-NONE-MNT
changed	hostmaster@isdn.net.il 20000402
source	RIPE
route	212.179.0.0/17
descr	ISDN Net Ltd.

Origin	AS8551
Notify	hostmaster@isdn.net.il
mnt-by	AS8551-MNT
changed	hostmaster@isdn.net.il 19990610
source	RIPE
person	Nati Pinko
address	Bezeq International
address	40 Hashacham St.
address	Petach Tikvah Israel
phone	+972 3 9257761
e-mail	hostmaster@isdn.net.il
NIC Handle	NP469-RIPE
changed	registrar@ns.il 19990902
source	RIPE

9- UDP SRC and DST outside network:

As was mention is the analysis of case 3 (TCP SRC and DST outside network), this traffic is being originated from inside your network, probably from compromised hosts.
In this case, the packets are crafted with snooped IP addresses (IP addresses other than your being "borrowed" by the attacker) so the identity of the attacker is not revealed. The source of the attack can be identified by MAC address. If this address belongs to a router, the next segment needs to be probed until the host initiating this traffic can be found.

Although this attack was persistent from May 4th to May 8th 2001, there some interesting patterns in some of this traffic that I would like to discuss:

Top talkers

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
134.129.71.203	3186	3186	1	1
206.190.36.120	1648	1648	1	1
134.129.125.158	593	593	2	2
169.254.67.123	151	151	133	133
169.254.107.122	44	44	9	9

In this table, the traffic initiated from 134.129.71.203 is rather interesting. The duration of attack in this case was only 7 minutes (from 11:41:31 on 05/04/2001 to 11:47:00 on the same day) which raises the possibility of a Denial of Service (DoS) attack.

The traffic pattern has been initiated from UDP port 4405 to the victim's UDP port 64546. I was unable to find any specific service being associated with that port which raises the possibility of a DoS attack. The snooped IP address belongs to North Dakota State University Computer Center which the attacker is pretending to be from.

```
05/04-11:41:31.358493 [**] UDP SRC and DST outside network [**] 134.129.71.203:4405 -> 233.24.119.155:64546
05/04-11:41:31.359324 [**] UDP SRC and DST outside network [**] 134.129.71.203:4405 -> 233.24.119.155:64546
05/04-11:41:32.077031 [**] UDP SRC and DST outside network [**] 134.129.71.203:4405 -> 233.24.119.155:64546
05/04-11:41:32.085437 [**] UDP SRC and DST outside network [**] 134.129.71.203:4405 -> 233.24.119.155:64546
05/04-11:41:32.277266 [**] UDP SRC and DST outside network [**] 134.129.71.203:4405 -> 233.24.119.155:64546
05/04-11:41:32.278126 [**] UDP SRC and DST outside network [**] 134.129.71.203:4405 -> 233.24.119.155:64546
```

What's more interesting is the destination IP address is a Multicast address. What this means that if the attack is targeted properly, it may easily flood a network given the fact that routers along the way would send this traffic to all hosts in the Multicast address space. The attack initiated from 134.129.125.158 would match the exact same pattern and is probably initiated from the same compromised host.

Top 5 destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
233.24.119.155	3826	3826	5	5

233.28.65.62	1648	1648	1	1
216.188.92.51	24	24	1	1
193.247.221.115	13	13	1	1
164.124.101.2	9	9	3	3

As can be seen here, the top two destinations of these attacks are Multicast IP addresses (224.0.0.0 to 239.255.255.255) which is further proof of DoS attack on Multicast addresses.

10- Attempted Sun RPC high port access:

As we described before, Sun RPC is known for the high number of vulnerabilities associated with it's network socket mapping. This traffic is the indication of attempts from outside your network to connect to hosts with the RPC high ports open.

The bulk of the traffic has been initiated from 24.21.203.64 for the duration of less than two minutes from 20:39:22 on 05/07/2001 to 20:40:59 on the same day.

Top 3 sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
24.21.203.64	5864	5864	1	1
205.188.153.98	5	5	1	1
205.188.153.101	2	2	1	1

Top 3 destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.229.166	5864	5865	1	2
1.1.206.150	5	5	1	1
1.1.225.234	2	5	1	3

As can be seen, the top destination is MY.NET.229.166 (1.1.229.166) which needs to be examined for evidence of compromise.

Sample of traffic:

May 5 09:01:12 213.228.154.156:3953 -> 1.1.229.166:21 SYN **S*****
05/07-20:39:22.302984 [**] Attempted Sun RPC high port access [**] 24.21.203.64:32768 -> 1.1.229.166:32771
05/07-20:39:22.557438 [**] Attempted Sun RPC high port access [**] 24.21.203.64:32768 -> 1.1.229.166:32771
05/07-20:39:23.563789 [**] Attempted Sun RPC high port access [**] 24.21.203.64:32768 -> 1.1.229.166:32771
05/07-20:39:24.572675 [**] Attempted Sun RPC high port access [**] 24.21.203.64:32768 -> 1.1.229.166:32771
05/07-20:39:24.827163 [**] Attempted Sun RPC high port access [**] 24.21.203.64:32768 -> 1.1.229.166:32771
05/07-20:39:25.239569 [**] Attempted Sun RPC high port access [**] 24.21.203.64:32768 -> 1.1.229.166:32771
05/07-20:39:27.819662 [**] Attempted Sun RPC high port access [**] 24.21.203.64:32768 -> 1.1.229.166:32771

As we mentioned before top attacker's IP address is 24.21.23.64 which is probably a cable modem user (part of @home network):

Information for NETBLK-HOOVER1-AL-2

@Home Network (NETBLK-HOOVER1-AL-2)
425 Broadway
Redwood City, CA 94063
US

Netname: HOOVER1-AL-2
Netblock: 24.21.200.0 - 24.21.207.255

Coordinator:
Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

Record last updated on 21-Apr-2000.
Database last updated on 19-Nov-2001 19:55:29 EDT.

11- TCP **S***** scan:

This is basically various connection attempts to different TCP ports inside your network. This type of scan in general is a probe from a source outside your network looking for hosts with well known ports (DNS, FTP, Telnet, …). Once these servers are found, the attacker would focus on more targeted ways to exploit the possible vulnerabilities on those servers.

Top five attacker:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
213.228.154.156	7350	7350	6129	6129
217.0.95.245	6054	6054	4983	4983
193.253.215.169	4544	4544	4280	4280
64.230.90.125	4048	4049	3778	3779
64.123.131.2	3419	3419	3111	3111

Of these five top attackers, 213.228.154.156 was the most active. This attack was in the form of scanning your internal network for ftp servers in duration of almost 2 ½ hours (starting at 7:07 am) on May 5th 2001:

May 5 07:07:04 213.228.154.156:2098 -> 1.1.1.29:21 SYN **S*****
May 5 07:07:05 213.228.154.156:2162 -> 1.1.1.93:21 SYN **S*****
May 5 07:07:05 213.228.154.156:2174 -> 1.1.1.105:21 SYN **S*****
May 5 07:07:05 213.228.154.156:2176 -> 1.1.1.107:21 SYN **S*****
May 5 07:07:05 213.228.154.156:2178 -> 1.1.1.109:21 SYN **S*****
May 5 07:07:05 213.228.154.156:2182 -> 1.1.1.113:21 SYN **S*****

The attack pattern indicates more of an orderly scan when (for example) IP addresses ending up with an even last octets are scanned in one class C address and the ones with odd last octets (above) scanned in the other. This attack (see below) has been originated from Portugal:

WHOIS Query Result for 213.228.154.156:

inetnum	213.228.154.0 - 213.228.154.255
Origin	CLI-PA
descr	Cabovisao, SA
descr	Clients Palmela
country	PT
Admin. Contact	AF3163-RIPE
Tech. Contact	VC1011-RIPE
status	ASSIGNED PA
mnt-by	RIPE-NCC-NONE-MNT
changed	vladimiro.casinha@cabovisao.pt 20010115
source	RIPE
route	213.228.128.0/19
descr	Cabovisao SA - Internet Provider
Origin	AS13156

mnt-by	AS13156-MNT
changed	vladimiro.casinha@cabovisao.pt 20000719
source	RIPE
person	Ahmad Fadami
address	Lugar de Pocos
address	Vale de Touros
address	2950 Palmela
phone	+351 1 2338700
fax-no	+351 1 2333020
NIC Handle	AF3163-RIPE
changed	joaquim.sousa@eastecnica.pt 19990915
source	RIPE
person	Vladimiro Casinha
address	Cabovisaoaddress: Lugar de Pocos
address	2950-425 Palmela
address	Portugal
phone	+351 21 233-8700
fax-no	+351 21 233-3020
e-mail	ripe.net-register@cabovisao.pt
e-mail	vladimiro.casinha@cabovisao.pt
NIC Handle	VC1011-RIPE
changed	ripe.net-register@cabovisao.pt 20000323
source	RIPE

The second top attack originated from 217.0.95.245 is scanning your internal network for ftp servers as well but this one lasted more than 4 hours on May 6th 2001, which may be because the scanner was hoping to avoid detection. This attack was launched from Germany:

May 6 02:17:59 217.0.95.245:1265 -> 1.1.5.99:21 SYN **S*****
May 6 02:17:59 217.0.95.245:1272 -> 1.1.5.106:21 SYN **S*****
May 6 02:17:59 217.0.95.245:1268 -> 1.1.5.102:21 SYN **S*****
May 6 02:17:59 217.0.95.245:1225 -> 1.1.5.59:21 SYN **S*****
May 6 02:18:00 217.0.95.245:1223 -> 1.1.5.57:21 SYN **S*****

WHOIS Query Result for 217.0.95.245:

European Regional Internet Registry/RIPE NCC (NET-217-RIPE)
These addresses have been further assigned
to European users. Contact information can
be found in the RIPE database at whois.ripe.net
NL

Netname: 217-RIPE
Netblock: 217.0.0.0 - 217.255.255.255
Maintainer: RIPE

Coordinator:

Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN)
nicdb@RIPE.NET
+31 20 535 4444

Domain System inverse mapping provided by:

NS.RIPE.NET	193.0.0.193
NS.EU.NET	192.16.202.11
AUTH00.NS.UU.NET	198.6.1.65
NS3.NIC.FR	192.134.0.49
SUNIC.SUNET.SE	192.36.125.2
MUNNARI.OZ.AU	128.250.1.21
NS.APNIC.NET	203.37.255.97
SVC00.APNIC.NET	202.12.28.131

Record last updated on 05-Jun-2000.
Database last updated on 19-Nov-2001 19:55:29 EDT.

inetnum	217.0.0.0 - 217.5.127.255
Origin	DTAG-DIAL13
descr	Deutsche Telekom AG
country	DE
Admin. Contact	RH2086-RIPE
Tech. Contact	ST5359-RIPE
status	ASSIGNED PA
remarks	**********
remarks	* ABUSE CONTACT: abuse@t-ipnet.de IN CASE OF HACK ATTACKS, *
remarks	* ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC. *
remarks	**********
Notify	auftrag@nic.telekom.de
Notify	dbd@nic.dtag.de
mnt-by	DTAG-NIC
changed	auftrag@nic.telekom.de 20010920
source	RIPE
route	217.0.0.0/13
descr	Deutsche Telekom AG, Internet service provider
Origin	AS3320
mnt-by	DTAG-RR
changed	rv@NIC.DTAG.DE 20000728
source	RIPE
person	Reinhard Hausdorf
address	Deutsche Telekom AG
address	Am Kavalleriesand 3
address	D-64295 Darmstadt
address	Germany
phone	+49
NIC Handle	RH2086-RIPE
Notify	auftrag@nic.telekom.de
Notify	dbd@nic.dtag.de
mnt-by	DTAG-NIC
changed	auftrag@nic.telekom.de 20010321
source	RIPE
person	Security Team
address	Deutsche Telekom AG

address	Technikniederlassung Schwaebisch Hall
address	D-89070 Ulm
address	Germany
phone	+49 731 100 84055
fax-no	+49 731 100 84150
e-mail	abuse@t-ipnet.de
NIC Handle	ST5359-RIPE
Notify	auftrag@nic.telekom.de
Notify	dbd@nic.dtag.de
mnt-by	DTAG-NIC
changed	auftrag@nic.telekom.de 20010321
source	RIPE

Top 5 Destinations receiving this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
1.1.101.142	66	66	1	1
193.10.65.125	48	48	1	1
1.1.221.238	46	48	3	3
64.55.147.130	46	72	1	1
24.182.220.102	44	44	1	1

Please note that since these types of attack are basically a form of scanning, the list of destinations in MY.NET is very long.

12- UDP Scan:

These are various types of scanning for different UDP ports inside your network. The attacks are usually in the form of scanning your internal subnets for open UDP ports, usually of well known Services such as DNS (UDP port 53).

Top 5 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
1.1.229.166	11065	11069	147	148
1.1.220.170	9904	9904	602	602
1.1.229.74	8403	8403	1687	1687
205.188.233.185	6631	6631	20	20
205.188.233.121	4117	4117	20	20

Interestingly enough, top three attackers in this case are from within your internal network. The scan from 1.1.229.166 was active from 20:19:18 on 5/7/2001 to 22:29:59 on the same day. The attacker is primarily looking for Sun RPC ports (UDP port 32768 Filenet TMS). As for 1.1.220.170, it is apparently looking for open Microsoft Gaming Zone ports (for more information, please check http://support.microsoft.com/support/kb/articles/Q159/0/31.ASP).

Correlation:

http://archives.neohapsis.com/archives/incidents/2000-09/0008.html

http://www.incidents.org/archives/y2k/093000.htm

As for the external sources of UDP scan, 205.188.233.185 has been the most active one, primarily scanning for port 6970. This attack has been initiated from an AOL IP address:

WHOIS Query Result for 205.188.233.185:

America Online, Inc (NETBLK-AOL-DTC)
22080 Pacific Blvd
Sterling, VA 20166
US

Netname: AOL-DTC
Netblock: 205.188.0.0 - 205.188.255.255

Coordinator:
America Online, Inc. (AOL-NOC-ARIN) domains@AOL.NET
703-265-4670

Domain System inverse mapping provided by:

DNS-01.NS.AOL.COM	152.163.159.232
DNS-02.NS.AOL.COM	205.188.157.232

Record last updated on 27-Apr-1998.
Database last updated on 19-Nov-2001 19:55:29 EDT.

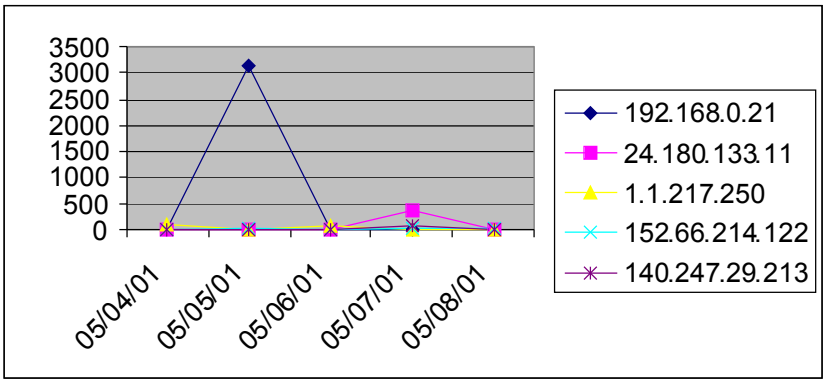
Correlation: http://www.sans.org/y2k/051400.htm

Analysis of Out Of Scope (OOS) files:

In order to analyze this data, I have merged all the OOS files from May 4th to May 8th 2001 into one file. Then using "awk" by sorting based on time, source IP address and destination ports, I created two following charts.

Chart 1- Top 5 talkers.

Distribution of connections over time based on source IP address:



1- 192.168.0.21: This internal IP address was most active (3135 entries) on May 5th 2001 around 3 pm. The activity was scanning the internal IP addresses looking for active ftp servers (tcp port 21). Please look at Chart-2 as well.

2- 24.180.133.11: This IP was most active on May 7th 2001 at around 7 pm. The activity was connection attempts from high port numbers to 1.1.98.88. This may be evidence that someone snooped your internal IP address to attack this server and these are replies from the server being received in your network.

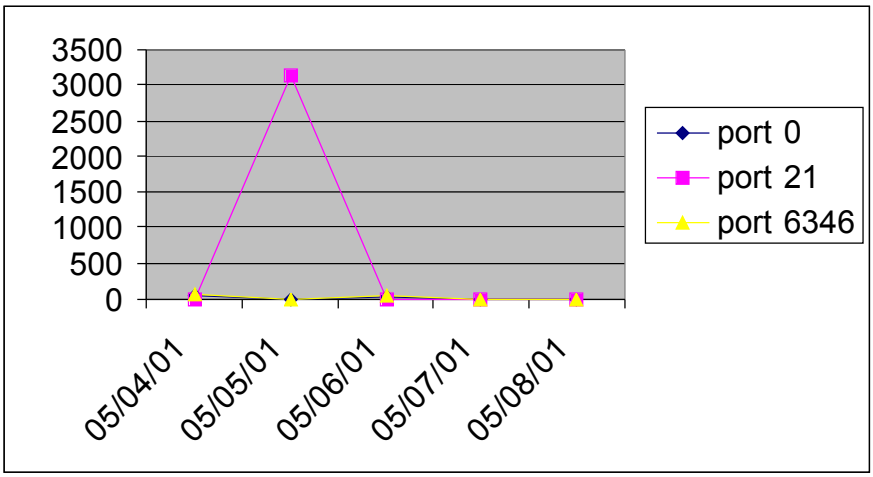
3- 1.1.217.250: The connection attempts from this internal host have evidence of

crafted packets (initiated from ports 0 and 6346) and was targeted toward a variety of hosts inside and outside your network. This host was most active on May 4th (101 attempts) and May 6th (62 attempts) 2001. Please look at Chart-2 as well.

4- 152.66.214.122: This address launched several scanning attempts to connect to TCP port 8080 (commonly used as proxy server port) on several hosts inside your network. The activity continued from May 4th to May 8th 2001.

5- 140.247.29.213: This address was active around 11 am on May 7th 2001. It was actively trying to connect to port 80 (HTTP) on MY.NET.100.165 (55 attempts).

Chart 2- Top 3 destination ports:



Analysis process notes:

At first I tried to select data from October 2001 but none of the servers that I had in my possession were able to process them using SnortSnarf. At first I tried using my Sun Ultra-5 workstation (Solaris 8, 8GB HDD, 256 MGB RAM) but after about an hour it ran out of memory. Then I tried using my laptop (WIN2K, Intel Pentium-3 750MHZ, 512 MB RAM, 20GB HDD) but it didn't take long for that to grind to a halt as well.
Next I got permission for 12 hours use of a Sun Enterprise-220 server (dual UltraSparc 450MHZ cpu, Solaris 2.6, 2x36GB HDD, 1GB RAM). Although this server had no problem with running out of memory or becoming slow, after 11 hours of processing I just made it through 4 out of 5 days of scan + alert files. Running "truss" to trace what is going on, I found SnortSnarf is busy writing (mostly html) files. At this time I had to return the server to it's respective administrator and get back to the drawing board.

Since I none of the hardware in my possession was obviously powerful enough to process the recent scan+ alert files, I decided to go back and find files smaller in size. As the result, I choose scan + alert files from May 4th to May 8th 2001, which I successfully processed on my workstation in 15 hours.

For this process, I first converted all instances of MY.NET.x.x to 1.1.x.x so SnortSnarf would be able to successfully do the correlations of IP addresses. Then I merged all OOS files into one and using a shell script sorted them into different files based on date, source IP addresses and destination ports. I then imported this data to Microsoft Excel which I used to create the charts.

Defensive Recommendations:

1- Investigate and, if required, block 202.229.228.180. This IP address have used crafted packets and should be considered for further action.
2- Investigate and, if required, block 202.39.78.125. Although the traffic from this host may be legitimate, there is a strong possibility that this address tried using tiny fragments to launch an attack into your network.
3- There was evidence of suspicious activity from several internal IP addresses with some of them masking their identity using spoofed IP addresses. These connection attempts need to be investigated by their MAC addresses so the source of these attacks (such as TCP/ UDP SRC and DST outside your network) can be traced and eliminated.
4- Investigate and, if required, block 216.64.197.68. This address attempted to connect to TCP port 515 and be trying to exploit a vulnerability with your internal Linux/ Unix hosts. Also, please keep in mind that all such hosts should have the latest release of LPR and security patches installed.

5- Block well known Internet game ports.
6- Block Internet peer to peer file sharing ports (such as Napster, Gnutella, …).
7- Block public access to high risk ports with well known vulnerabilities such as Sun RPC. Also, please make sure that the internal hosts have the latest security patches installed.
8- The addresses mention in TCP **S***** section need to be investigated and if required, blocked.
9- The following internal addresses were very active in UDP scans and need to be investigated: 1.1.229.166, 1.1.220.170, 1.1.229.74.

Relationship between machines used to generate the logs:

In order to create meaningful scan/ alert logs that can be used for intrusion detection analysis, we usually try to intercept the traffic both before (on Internet or "dirty" side) and after (on Intranet or "clean" side) of perimeter firewalls.
Using this method and conjunction with firewall security policy, meaningful logs with better trace of communication attempts can be collected.

Insight into internal Machines:

As was mentioned in the last section, several internal machines are actively launching scans/ attacks with some of them masking their identities using crafted packets and spoofed source addresses. These machines may either be compromised or being used as an attack platform which in either case they need to be investigated.

Correlation with other student practicals:

The correlation was based on searching the other practicals and trying to make sense of what I see in the logs and scans.