conference

proceedings

Workshop on Intrusion Detection and Network Monitoring (ID '99) Proceedings

Santa Clara, California
April 9–12, 1999

Sponsored by
The USENIX Association

USENIX®

The Advanced Computing
Systems Association

USENIX Association

Proceedings of th

Workshop on Intrusion I

and Network Monit

(ID '99)

April 9–12, 1999
Santa Clara, California, USA

Program Committee

Program Chair

Marcus J. Ranum, *Network Flight Recorder, Inc.*

Program Committee

Charles Antonelli, *University of Michigan*
Frederick Avolio, *Avolio Consulting*
Tina Darmohray, *SystemExperts Corp.*
Rik Farrow, *Consultant*
Dan Geer, *CERTCO*
Norm Laudermilch, *UUNet/Worldcom*

External Reviewers

Travis Corcoran
Rich Salz

USENIX Association

Ellie Young, *Executive Director*
Judith F. DesHarnais, *Conference Coordinator*
Toni Veglia, *Member Services Coordinator*
Cynthia Deno, *Marketing Director*
Linda Barnett, *Marketing and Communications Coordinator*
Jane-Ellen Long, *Publications Director*
Daniel V. Klein, *Tutorials Coordinator*

Message from the Program Chair

Welcome!

A few years ago, a wise old programmer complained that in "Internet Time" technologies go from "Interesting research" to "overhyped" before the researchers get a chance to actually make any progress. Hopefully this workshop will help prove that his complaint is wrong—a lot of smart people are doing research in intrusion detection (ID) today, and are willing to share their ideas with their peers. We have a line-up of papers that we think represent some of the most interesting work being done in the field; we're happy to be able to share them with you.

As program chair, it's an honor to help organize such a workshop: the real work is done by the authors and the USENIX staff. I'd like to thank our authors for submitting their work, and taking on the additional effort of sharing their thoughts with the community. Thanks also to the program committee for reviewing papers, choosing among them, and helping the authors edit their final drafts. Lastly, deep thanks to the USENIX staff, who kept an eye on the schedules, locations, publications, formats, and made sure the "i"s were dotted and "t"s were crossed. Without USENIX's ongoing commitment to spreading knowledge, the computing world would be a little darker. Thank you all.

I welcome you, and hope you enjoy and learn from the workshop.

Marcus J. Ranum
Program Chair

Contents

Workshop on Intrusion Detection and Network Monitoring

April 9–12, 1999
Santa Clara, California, USA

Sunday, April 11

Analysis and Large Networks
Session Chair: Fred Avolio, Avolio Consulting

Software and Processes
Session Chair: Tina Darmohray, SystemExperts Corp.

Monday, April 12

IDS Systems
Session Chair: Charles Antonelli, University of Michigan

Network Data Processing and Storage
Session Chair: Dan Geer, CERTCO

Statistics and Anomalies
Session Chair: Marcus Ranum, Network Flight Recorder, Inc.

Analysis Techniques for Detecting Coordinated Attacks and Probes

John Green
Naval Surface Warfare Center
David Marchette
Naval Surface Warfare Center
Stephen Northcutt
Ballistic Missile Defense Organization
Bill Ralph
ATR Corporation

Abstract

Coordinated attacks and probes have been observed against several networks that we protect. We describe some of these attacks and provide insight into how and why they are carried out. We also suggest hypotheses for some of the more puzzling probes. Methods for detecting these coordinated attacks are provided.

1. Introduction

For approximately the last year, SHADOW analysts have been detecting a new class of network traffic. Although the probes and attacks embedded within this traffic consist mostly of known exploits, subsequent analysis reveals that multiple IP addresses are working together toward a common goal. Therefore, we have coined the phrase "coordinated attacks" to describe the activity that has been observed.

Indications of these concerted efforts appear in network traffic logs as multiple external IP addresses targeting a single address of the protected network. Similarly, a coordinated attack can also look as though multiple attackers are working together to execute a distributed scan on many internal addresses or services. It is believed that probes of this nature have been developed in an attempt to elude the scan detection code present in many intrusion detection systems.

In most of the cases observed, the number of cooperating IP addresses is rather small; four or five is common. However, as many as fifteen different coactive, scanning hosts have been uncovered by SHADOW analysts. Due to their distributed nature, these attacks were well below the threshold for a structured attack in terms of targeting, lethality and scope.

> *"We distinguish two fundamental types of threat. The unstructured threat is random and relatively limited. It consists of adversaries with limited funds and organization and short-term goals. While it poses a threat to system operations, national security is not targeted. This is the most obvious threat today. The structured threat is considerably more methodical and well-supported. While the unstructured threat is the most obvious threat today, for national security purposes we are concerned primarily with the structured threat, since that poses the most significant risk."*
>
> Air Force Lt. Gen. Kenneth A Minihan, director of the National Security Agency - brief to the Senate Government Affairs Committee, June 24 1998.

What is a structured attack? Interviews with premier intrusion detection researchers revealed that they consider structured attacks to be on the order of thousands of related exploits, probes, viruses, scans, denials of service, and ruses over a short period of time. Even though this definition doesn't accurately describe the patterns discussed earlier, we cannot call this activity unstructured. It definitely has structure!

This paper will examine various coordinated attacks and probes, including coordinated traceroutes, NetBIOS scans, Reset scans, SFRP scans and coordinated DNS server exploit attempts. Some of these probes are certainly the work of multiple computers working together; others appear to be fraudulent or decoy mechanisms.

2. Coordinated traceroutes

Coordinated traceroutes serve as a reminder that sites are always vulnerable, even if their firewalls are impenetrable. Information gleaned from this technique can be used to direct a denial of service attack against a site's external connectivity, effectively isolating the facility. Detection of coordinated traceroutes is simple; look for about five traceroutes within two seconds of one another, often with similar names.

Figure 1 shows an example of this activity. Here, five different sources, each from a different backbone network, are shown probing the same target. Most often, the target is a DNS server, or DNS serving firewall, and packet arrival is usually within tenths or hundredths of seconds of each other.

```
12:29:30.01  proberA.39964  >  target.33500:  udp
12 [ttl 1]
12:29:30.13  proberA.39964  >  target.33501:  udp
12 [ttl 1]
12:29:30.25  proberA.39964  >  target.33502:  udp
12 [ttl 1]
12:29:30.35  proberA.39964  >  target.33503:  udp
12 [ttl 1]

12:27:55.10  proberB.46164  >  target.33485:  udp
12 [ttl 1]
12:27:55.12  proberB.46164  >  target.33487:  udp
12 [ttl 1]
12:27:55.16  proberB.46164  >  target.33488:  udp
12 [ttl 1]
12:27:55.18  proberB.46164  >  target.33489:  udp
12 [ttl 1]

12:27:26.13  proberC.43327  >  target.33491:  udp
12 [ttl 1]
12:27:26.24  proberC.43327  >  target.33492:  udp
12 [ttl 1]
12:27:26.37  proberC.43327  >  target.33493:  udp
12 [ttl 1]
12:27:26.48  proberC.43327  >  target.33494:  udp
12 [ttl 1]

12:27:32.96  proberD.55528  >  target.33485:  udp
12 [ttl 1]
12:27:33.07  proberD.55528  >  target.33486:  udp
12 [ttl 1]
12:27:33.17  proberD.55528  >  target.33487:  udp
12 [ttl 1]
12:27:33.29  proberD.55528  >  target.33488:  udp
12 [ttl 1]

12:27:30.55  proberE.21337  >  target.33475:  udp
12 [ttl 1]
12:27:30.56  proberE.21337  >  target.33476:  udp
12 [ttl 1]
12:27:30.58  proberE.21337  >  target.33477:  udp
12 [ttl 1]
12:27:30.59  proberE.21337  >  target.33478:  udp
12 [ttl 1]
```

Figure 1. Coordinated traceroute example.

Coordinated traceroutes do have a commercial use. Some Internet Service Providers (ISP) use them to calculate the best routes back to clients in an attempt to provide optimum web response. The stimulus for this type activity is a host from the protected network visiting a web server supported by an ISP that uses coordinated traceroutes.

As mentioned above, coordinated traceroutes can be a benign effort to improve the performance of a server. As such they can be viewed as providing a useful service. However, they can also be used to determine all the routes into your protected network. Thus it should be of interest to determine who is performing these traceroutes and why.

3. NetBIOS deception

One of the first things that a network analyst learns is that network traffic is not always what it appears to be. Source address spoofing is a classic example of this. Many commonly available exploit tools include this capability. In fact, the latest version of nmap takes spoofing a step further; it includes a decoy option. This option allows the hacker to make an attack appear as though it is coming from multiple sources. Even if an analyst at the targeted site detects the attack, it is very difficult to determine which of the IP addresses were spoofed, and which one was real.

The trace in figure 2 is from a site that receives very few NetBIOS session connection attempts. The traffic shown was detected over a single twenty-four hour period. These source addresses correlate with NetBIOS session connection attempts seen at other sites over several days. The signature of this massive scan is: four connect attempts for each address, the do not fragment option is set, a window size of 8192, and the TTL fields cluster. Perhaps the most interesting signature of this coordinated activity is that the traffic is destined only for IP addresses that are not populated by hosts. The first two traces show all four attempts, the rest have been edited for space.

The source addresses spanned several countries, but certainly could have been spoofed. The scan rate is slow enough that the entire probe could have been generated from a single computer. The fact that the Time To Live (TTL) field is within three hops for all packets is also interesting and points to a single computer. Different operating systems have different TTL defaults. The probes were sent to hosts that do not exist, therefore the TCP three-way handshake was never completed. That would be evidence this was actually a probe. Could this be a hoax?

If this is a hoax, what is the purpose? One possibility is that fake attacks may create fear, uncertainty, and doubt in the same vein as virus hoaxes. Another possibility is an "attacker honey pot". Even a mediocre fake attack will tie up analyst and CIRT resources, and possibly serve as a distraction so that a much lower signal precision attack can get through undetected.

Another hypothesis is that the attacker is only interested in "brand new" systems that are brought online. Since most security patches are available from vendor websites, many administrators bring up vulnerable systems with the intent of downloading and installing the patches at a later date. This process may take several days, leaving new systems and the networks that they reside on vulnerable to attack.

The fact that only non-existent systems are targeted makes these probes particularly puzzling. It also makes them worthy of concern, since it implies that the attackers have a precise map of the protected network.

Trace 1:

```
00:56:22.78 proberD.3506 > 172.20.124.23.139:
S 14300153:14300153(0) win 8192  (DF)
00:56:25.69 proberD.3506 > 172.20.124.23.139:
S 14300153:14300153(0) win 8192  (DF)
00:56:31.70 proberD.3506 > 172.20.124.23.139:
S 14300153:14300153(0) win 8192  (DF)
00:56:43.69 proberD.3506 > 172.20.124.23.139:
S 14300153:14300153(0) win 8192  (DF)
```

Trace 2:

```
06:49:55.47 proberA.4197 > 172.20.139.137.139:
S 596843772:596843772(0) win 8192 (DF)
06:49:58.44 proberA.4197 > 172.20.139.137.139:
S 596843772:596843772(0) win 8192  (DF)
06:50:04.44 proberA.4197 > 172.20.139.137.139:
S 596843772:596843772(0) win 8192  (DF)
06:50:16.43 proberA.4197 > 172.20.139.137.139:
S 596843772:596843772(0) win 8192  (DF)
```

Additional traces, only the first packet is shown:

```
12:57:56.94 proberE.2038 > 172.20.216.29.139:
S 294167370:294167370(0) win 8192  (DF)
13:37:51.75 proberI.4186 > 172.20.215.205.139:
S 22881687:22881687(0) win 8192  (DF)
13:50:23.64 proberB.3293 > 172.20.53.123.139:
S 355997160:355997160(0) win 8192  (DF)
14:11:01.95 proberC.3491 > 172.20.245.182.139:
S 57370977:57370977(0) win 8192  (DF)
15:41:59.50 proberG.3278 > 172.20.252.141.139:
S 266305199:266305199(0) win 8192  (DF)
22:49:15.39 proberH.3658 > 172.20.124.23.139:
S 14035939:14035939(0) win 8192  (DF)
```

Figure 2. Netbios deception example.

4. Reset scans

If you examine the Internet traffic to your site, there is a very good chance you will find a large number of inbound Resets and SYN/ACKs for which there is no corresponding SYN packet. Generally, these scans originate from a multitude of source addresses and often appear to be coordinated due to their concurrency. Several questions come to mind: "What is going on?" and furthermore, "What are some of the events that cause Reset generation?" Section 4 will explore some of the events that can generate Reset scans, the motivations for Reset scanning, and will also discuss the methods that SHADOW uses to detect them.

4.1. Natural function of TCP/IP

Resets are a normal part of TCP/IP communications. If something goes wrong with a TCP connection, a reset may be generated. Typically, in this case only one would be observed between the server and client. If a connection is attempted to a service that does not exist, a reset may be generated. A single SYN attempt/Reset response from a mscan probe is shown in figure 3. Note that the acknowledgement number is the sequence number incremented by one.

```
13:13:10.670000 www.1880 > mailrelay.6000: S
1393635005:1393635005(0) win 512
13:13:10.680000 mailrelay.6000 > www.1880: R
0:0(0) ack 1393635006 win 0
```

Figure 3. SYN attempt/Reset response example.

Client systems generally attempt to establish connections multiple times. Four SYN "active open" attempts to the same destination address and source port is commonly seen for most services. Electronic mail and web (TCP port 25 and TCP port 80) active opens often try larger numbers of attempts ranging from twelve to twenty five.

From an intrusion detection standpoint, we generally expect to see outbound Resets as a result of activity caused by inbound traffic. Examining the trace in figure 3, we see that the inbound traffic from www to Mailrelay attempts to initiate an X Windows connection. Mailrelay wants no part of this; so an outbound and Reset to www is generated.

If we detect inbound Resets we expect that these were caused by outbound connections from our systems. In the next two possible causes for the generation of Resets, we will look at situations where we observe a medium to large number of Resets, (or SYN/ACKS) inbound. In these cases, there is no corresponding SYN packet.

4.2. Second order effect

The inbound Resets (or Syn/Acks) could also be explained as a "second order effect" of a denial of service attack, or scan on another site. For this to be a second order effect, we must not have initiated the connections with SYN packets and our IP addresses are used (spoofed) to attack someone else. This last case is a dominant factor in the generation of large numbers of inexplicable Resets. IRC servers seem to be the primary targets in a large number of these cases.

A wide variety of Internet addresses have been used for this sport; we have received traces of excessive Resets from all over the globe. Figure 4 illustrates example traffic at two sensor locations: SITE_A and SITE_B. It shows the activity that each site detected on the same day. The time stamps indicate concurrent activity from Irc_victim to multiple destination hosts. This denial of service attack generates Resets from Irc_victim, since the attack was to Irc_victim's inactive ports.

Excerpts from SITE_A tcpdump at ~02:00:

```
02:13:23.55 Irc_victim.37762 >
192.168.129.191.18602: R 0:0(0) ack 1940197743
win 0
02:14:00.07 Irc_victim.25013 >
192.168.251.67.26831: R 0:0(0) ack 397924438
win 0
02:14:20.68 Irc_victim.32824 >
192.168.123.30.17807: R 0:0(0) ack 1747849368
win 0
```

Excerpts from SITE_B tcpdump at ~02:00:

```
02:13:21.54 Irc_victim.4723 >
172.20.96.61.7790: R 0:0(0) ack 172384509 win
0
02:14:09.39 Irc_victim.45991 >
172.20.72.145.18363: R 0:0(0) ack 578682865
win 0
02:14:12.35 Irc_victim.58839 >
172.20.46.51.51347: R 0:0(0) ack 1901339874
win 0
```

Figure 4. Expected behavior for inactive ports.

In contrast, figure 5 depicts the network traffic pattern for active ports. Note the change in the 12:00 hour activity to the active port 6667 with the expected SYN/ACK response from Irc_victim, followed by the RST/ACK segment indicating an aborted connection.

For the truly paranoid, we offer an alternate interpretation of the traces shown in figure 5. A "man in the middle" scan could create this signature. In this case, the attackers must compromise our site, or a node on the route to our site. They must place a sniffer that is tuned to collect Resets and Syn/Acks on a compromised site. They then port scan the target from another location spoofing our address space. The sensor located on the compromised host collects the results and sends them to the attacker. This is unlikely to be the case in attacks against multiple sites.

It is important to point out that this kind of secondary effect will appear as a coordinated attack against the protected network if the attacker targets multiple hosts or networks, and always spoofs our IP addresses in the attack.

Sample trace from SITE_A at ~12:00:

```
12:47:03.65 Irc_victim.6667 >
192.168.140.187.10496: S 157348803:157348803(0)
ack 687865857 win 16384 <mss 1460> (DF)

12:47:03.87 Irc_victim.6667 >
192.168.140.187.10496: R 1:1(0) ack 1 win 16384
(DF)

12:48:38.57 Irc_victim.6667 >
192.168.246.165.33026: S 2670541452:2670541452(0)
ack 2164391937 win 16384 <mss 1460> (DF)

12:48:39.07 Irc_victim.6667 >
192.168.246.165.33026: R 1:1(0) ack 1 win 16384
(DF)
```

Sample trace from SITE_B at ~12:00:

```
12:47:07.43 Irc_victim.irc >
172.20.246.181.36126: S
1105399373:1105399373(0) ack 2367553537 win
16384  (DF)

12:47:07.56 Irc_victim.irc >
172.20.246.181.36126: R 1:1(0) ack 1 win 16384
(DF)
12:47:20.35 Irc_victim.irc >
172.20.64.221.18178: S
1443077754:1443077754(0) ack 1191313409 win
16384  (DF)

12:47:20.35 Irc_victim.irc >
172.20.64.221.18178: R 1:1(0) ack 1 win 16384
(DF)
```

Figure 5. Expected behavior for active ports.

4.3. Resets for intelligence gathering

Reset scanning works like any other inverse mapping method. This is because the routers are thinking IP, not TCP and the IP address is in the IP layer. When destination IPs or ports are inactive, the routers simply want to be helpful and return an address unreachable message. There are a variety of techniques (including Reset scanning) to locate the hosts, nets, and active service ports that do not exist. The attacker simply has to take the converse of the map to get a first order understanding of what does exist.

Figure 6 is an example from the point of view of the Reset scanner. They know the address of the system(s) they have scanned, so they wait for icmp error messages from the destination network's router. The results of interest could look like net (or host) unreachable or time exceeded.

```
20:38:11.783596 router > 192.168.32.192: icmp:
time exceeded in-transit [tos 0xc0]
20:38:55.597130 router > 192.168.31.15: icmp:
time exceeded in-transit [tos 0xc0]
20:41:41.824191 router > 192.168.52.99: icmp:
time exceeded in-transit [tos 0xc0]
20:43:50.750498 router > 192.168.52.99: icmp:
time exceeded in-transit [tos 0xc0]
20:44:01.280339 router > 192.168.61.209: icmp:
time exceeded in-transit [tos 0xc0]
20:44:27.790505 router > 192.168.59.164: icmp:
time exceeded in-transit [tos 0xc0]
```

Figure 6. Results from a reset scan.

In the early days of this technique, Reset scans were easy to detect due to common "signature acknowledgement numbers"; the TCP header ACK field was always a fixed number, usually 674719802 or 674711610. Figure 7 shows a Reset probe from two attackers that can trivially be detected due to the signature Ack number.

```
17:40:45.87 hook.24408 > target1.1457: R
0:0(0) ack 674719802 win 0
17:40:53.03 hook.33174 > target2.1457: R
0:0(0) ack 674719802 win 0
17:41:12.16 hook.36250 > target3.1979: R
0:0(0) ack 674719802 win 0
17:43:37.61 router > hook: icmp: time exceeded
in-transit
17:43:43.14 hook.44922 > target4.1496: R
0:0(0) ack 674719802 win 0
17:42:30.40 grin.3532 > target1a.1167: R
0:0(0) ack 674719802 win 0
17:42:40.58 grin.33233 > target2a.1797: R
0:0(0) ack 674719802 win 0
17:44:28.84 grin.52504 > target3a.1634: R
0:0(0) ack 674719802 win 0
17:47:52.58 grin.46657 > target4a.2121: R
0:0(0) ack 674719802 win 0
17:47:52.70 router > grin: icmp: time exceeded
in-transit
```

Figure 7. Example of "signature acknowledgement numbers".

Unfortunately, some of the more recent probes have random acknowledgement numbers. Probes of this type have been observed from at least fourteen different cooperating Internet addresses, primarily ISPs, all within a twenty-four hour period. And of course, how do you sort between the scans and second order effects?

Many people want to label all Resets as a second order effect and just not deal with it. This is foolish; when there is this much smoke, find the fire. These probing systems are working together to map multiple target sites. Reset traces from all over the world provide strong evidence that this activity is a long-term, Internet wide effort. The scan rate from some attacks is as low as 2 packets per day per target site, well below commonly set thresholds for scan detectors.

This begs the question: "Without a signature Ack how can we detect Reset scans?" In this case, the primary signature is the Reset code bit set with no other activity from that source, (such as an active open [SYN] from the source or the target). An obvious solution is to keep track of the state of each TCP connection and alarm: if a Reset, Syn/Ack, or Fin is detected without the active open. However, this solution can be compute intensive for large networks. The answer lies in less expensive mechanisms; namely scan detectors.

Inverse mapping is best detected over a longer time window, such as an hour, or even a day. In this case, we can test for an external host making connections to n internal hosts where n is a small value, (Shadow systems default to 7, but this is configurable). This technique will detect any scan that meets or exceeds the tally trigger over the time window. Figure 8 shows how SHADOW displays detected scans.

```
Hourly Tally Counter

8       192.168.2.1     hook
7       172.20.20.20    grin
7       10.32.21.12     false_positive.net
```

Figure 8. SHADOW scan detection output.

The advantage of such a technique is that it will detect any scan, so it will detect scans for which there is no signature. The disadvantages of this approach are threefold: scans below the tally trigger point will be missed, the scan detector has no provision for a focusing filter, collecting low and slow probes on an hourly basis is a manual technique and therefore prone to error.

Some attackers are patient enough to scan at rates as low as two packets per day; in these cases an hour clearly is not a reasonable time window. Figure 9 illustrates example output from a 24 hour scan detection tool called look4scans.pl. This tool was part of the version 1.5 Shadow software release.

```
10.9.8.7 :   Reset.host.net
10.9.8.7  > 192.168.103.90 : R
10.9.8.7  > 192.168.114.15 : R
10.9.8.7  > 192.168.122.80 : R
10.9.8.7  > 192.168.137.149 : R
10.9.8.7  > 192.168.157.224 : R
10.9.8.7  > 192.168.164.44 : R
10.9.8.7  > 192.168.174.161 : R
10.9.8.7  > 192.168.201.148 : R
10.9.8.7  > 192.168.202.85 : R
10.9.8.7  > 192.168.204.79 : R
10.9.8.7  > 192.168.213.156 : R
10.9.8.7  > 192.168.29.38 : R
10.9.8.7  > 192.168.41.157 : R
10.9.8.7  > 192.168.43.145 : R
10.9.8.7  > 192.168.45.174 : R
10.9.8.7  > 192.168.85.28 : R
10.9.8.7  > 172.20.107.109 : R
10.9.8.7  > 172.20.113.214 : R
10.9.8.7  > 172.20.115.6 : R
10.9.8.7  > 172.20.13.168 : R
10.9.8.7  > 172.20.140.69 : R
10.9.8.7  > 172.20.145.25 : R
10.9.8.7  > 172.20.191.30 : R
10.9.8.7  > 172.20.205.137 : R
10.9.8.7  > 172.20.207.56 : R
10.9.8.7  > 172.20.224.98 : R
10.9.8.7  > 172.20.23.185 : R
10.9.8.7  > 172.20.31.98 : R
10.9.8.7  > 172.20.41.248 : R
10.9.8.7  > 172.20.42.114 : R
10.9.8.7  > 172.20.62.140 : R
10.9.8.7  > 172.20.71.217 : R
10.9.8.7  > 172.20.84.178 : R
```

Figure 9. Example 'look4scans.pl' output.

Slow coordinated attacks are particularly difficult to detect using these methods. If the attackers can guess your detection threshold they can ensure that no single IP address sends enough packets to trip that threshold. Unless the attackers are foolish enough to include some other signature in the scan, these will be particularly difficult to detect.

4.4. Resets as an indicator of TCP session hijacking

The nmap scanning tool released December 1998 has a sequence number evaluator as part of its most basic functionality, so hijack will be with us for a while yet! The idea is to find an active connection, and predict the sequence numbers on both sides of the connection. Hit the side you *don't* want to penetrate with a Reset to break off the connection from their point of view. Assume the connection and attack the other side. The signature for this attack is the correct sequence number and wrong IP address.

4.5. ISS RealSecure kill

We have seen this only twice. If an ISS RealSecure thinks the site it is protecting is under attack, it may generate a connection Reset. In this case, the packet contains the ID Number of the RealSecure engine.

4.6. Deception

As stated earlier, several freely available scanners can generate Resets with spoofed addresses simply as a smokescreen. They accomplish no purpose except possibly to consume analyst and CIRT resources.

How big of a problem is this? There are a few areas of concern:

A. If some portion of the inexplicable Resets is related to mapping attempts, then external actors are gaining intelligence about the networks that we are supposed to defend. In this case, the solution is to implement a firewall that can drop these packets.

B. Though we aren't particularly bothered by the second order effect problem, it is bad from a public perception standpoint if it is widely thought that our sites are attacking other sites, since our address space is being used.

5. SFRP scans

In the previous scan examples, the attackers came to us. This is not always the case. Scanning can happen when we visit the attacker. In this case, malformed packets with SYN, Reset, FIN and Urgent are detected coming from web servers to the browsing client. The most common pattern is one SFRP (SYN/FIN/Reset/PUSH) packet sent to each browsing client per session. Sometimes SRP's are also sent, Figure 10 illustrates the pattern.

```
10:47:36.61 media.com.2048 > target.48579: SFR
2842082:2842590(508) ack 2642669109 win 768
urg 2571  (DF)
11:23:42.97 media.com.2048 > target.47720: SFP
4820865:4821409(544) win 3840 urg 2571 (DF)
13:49:44.33 gm.com.49608 > target.49606: SFP
7051:7607(556) ack 2147789506 win 7768 (DF)
13:49:44.72 gm.com.22450 > target.1591: SFRP
2038:2074(36) ack 116065792 win 0 urg 0 (DF)
```

Figure 10. TCP stack analysis.

Figure 11 shows related activity that is not from the original site but is within the same general timeframe. The stimulus here is the client visiting the web server. These are examples of what comes back. Each client gets at least one packet and as many as four, (with different combinations), during a visit to a web server.

```
12:18:46.25 im.com.5500 > target.1137: SFP
3241821:3242365(544) win 13234 urg 55134 (DF)
13:37:30.33 im.com.22555 > target.22555: SF
8440982:8441538(556) win 10240 (DF)
14:52:57.45 scannernet.30975 > target.16940:
SFRP 2029994540:2029995068(528) ack 2029994540
win 16940 urg 16940 <[bad opt]> (DF)
14:53:01.63 scannernet.30975 > target.556:
SFRP 2029978156:2029978684(528) ack 2029978156
win 556 urg 556 <[bad opt]> (DF)
```

Figure 11. Cooperative tcp stack analysis example.

We have a pattern we have never seen before and it occurs during transactions with multiple web servers from multiple domains. During the height of this technique, in October 1998, over twenty web-servers from a very large ISP were exhibiting this behavior.

After tracking this for several weeks, we were still leaning toward considering this benign, perhaps some error in the web-server code. However, two weeks later, probes were observed from the same address family that did not have any stimulus (no one visited a web page). These non-stimulus caused probes were targeting DNS and mail servers. At this point, the activity was considered hostile. Since multiple web-servers were performing these probes in concert, this was also considered a coordinated attack.

6. Target based analysis

Until now, every example has shown multiple attackers, multiple targets, and we have focused on the activity of the inbound packets and the analysis of that activity. Now let's consider a different analysis technique: examining the targets. One of the factors that helped us understand the fact that Reset scanners were working together was that they did not duplicate targets; each system probed was unique. Furthermore, many of the attackers would scan three hosts from one site and twelve from a second and this pattern would continue day after day.

Infrastructure systems such as DNS and email servers are a good starting place for target analysis. In a given week, a large number of the total attacks are usually against these types of systems. In figure 12, the traces show attacks that come from vastly different IP addresses. These IP addresses originate from Australia, Asia and the USA, but all include the same targets, and occurred over a single weekend. "Whoops" isn't really a name server or email server, though it was erroneously listed as one in a DNS table.

Also, please note that SourceA and SourceB have different IP address numbers. Since this is TCP, the exploit cannot work if they spoof the source address. One of the probe sets could be a decoy; it could be a multihomed host, or it could be two systems working together. Please note the packet arrival times to see how related the first two scans appear to be and also the static source port. The third trace has a significant difference from the first two; the source port pattern indicates two processes. In the following example, we would assume that the first two traces are related and the third trace is a different actor.

One of the themes of this story is that the events of interest we classify as coordinated attacks are often detects that we had never seen before. Suddenly, we see it from (or to) multiple locations. To detect and classify a coordinated attack, it really helps to have a database of all traffic and techniques to complement your signatures. Without a database of traffic that covers a time window of at least a couple months, there is no way to determine whether this activity has been going on and simply hasn't been detected, or if it is a new pattern. Recently, we tested a pattern that had been detected by an analyst at another site. We were sure we had never seen it before. Wrong! What really stung us was that one of the attackers had spun this attack off of source port 7 (echo), something a good analyst should never miss. Oh well.

> *If you can only detect and examine traffic that matches your signature set, then how can you detect a new, or novel attack?*

```
06:10:56.53 SourceA.10053 > NS1.111: S
1935318310:1935318310(0) win 242
06:32:42.15 SourceA.10053 > NS2.111: S
552822870:552822870(0) win 242
06:54:27.32 SourceA.10053 > MAIL1.111: S
944974642:944974642(0) win 242
07:16:12.73 SourceA.10053 > MAIL2.111: S
3045099303:3045099303(0) win 242
07:37:58.16 SourceA.10053 > Whoops.111: S
323776127:323776127(0) win 242

06:12:33.28 SourceB.10053 > NS1.domain: S
992750649:992750649(0) win 242
06:34:18.66 SourceB.10053 > NS2.domain: S
3455530061:3455530061(0) win 242
06:56:04.046 SourceB.10053 > MAIL1.domain: S
1895963699:1895963699(0) win 242
07:17:49.44 SourceB.10053 > MAIL2.domain: S
2485794595:2485794595(0) win 242
07:39:34.811723 SourceB.10053 > Whoops.domain:
S 3785701160:3785701160(0) win 242

08:01:20.23 SourceB.1025 > NS1.imap: S
1471781129:1471781129(0) win 512
08:23:05.64 SourceB.21053 > NS2.imap: S
4110489384:4110489384(0) win 512
08:24:50.96 SourceB.1026 > MAIL1.imap: S
1486592867:1486592867(0) win 512
08:23:05.64 SourceB.21055 > MAIL2.imap: S
1112489384:1112489384(0) win 512
08:44:50.96 SourceB.1028 > Whoops.imap: S
0486592777:0486592777(0) win 512
```

Figure 12. Target based analysis.

```
AttackerB.6667 -> 192.168.229.72.1437, 1
packet
AttackerB.6667 -> 192.168.229.72.1437, 2
packets
AttackerB.6667 -> 192.168.229.82.1437, 1
packet
AttackerB.6667 -> 192.168.229.82.1437, 2
packets
AttackerB.6667 -> 192.168.229.95.1437, 1
packet
AttackerB.6667 -> 192.168.229.95.1437, 2
packets
AttackerB.6667 -> 192.168.229.6.1437, 1
packet
AttackerB.6667 -> 192.168.229.6.1437, 1
packet
AttackerB.6667 -> 192.168.229.79.1437, 1
packet
AttackerB.6667 -> 192.168.229.79.1437, 2
packets
AttackerB.6667 -> 192.168.229.45.1437, 1
packet
AttackerB.6667 -> 192.168.229.45.1437, 2
packets

AttackerC.139 -> 192.168.229.28.1437, 1
packet
AttackerC.139 -> 192.168.229.28.1437, 1
packet
AttackerC.139 -> 192.168.229.28.1437, 1
packet
AttackerC.139 -> 192.168.229.122.1437, 1
packet
AttackerC.139 -> 192.168.229.122.1437, 1
packet
AttackerC.139 -> 192.168.229.122.1437, 1
packet
AttackerC.139 -> 192.168.229.122.1437, 1
packet
AttackerC.139 -> 192.168.229.28.1437, 1
packet
AttackerC.139 -> 192.168.229.28.1437, 1
packet
AttackerC.139 -> 192.168.229.28.1437, 1
packet
AttackerC.139 -> 192.168.229.75.1437, 1
packet
AttackerC.139 -> 192.168.229.75.1437, 1
packet
AttackerC.139 -> 192.168.229.75.1437, 1
packet
```

Figure 13.

Figure 13 shows the traffic from an event that took place over a four-day weekend. In this case, multiple addresses began to target a specific destination port. In the first trace, notice the one packet two packet pattern and the source port of 6667 (IRC). Attacker C has a different pattern or their IDS interprets it differently. For two months different IP addresses were probing this site on the same destination port. No other sites with which we share information have detected this activity.

7. Conclusions

The examples shown in this paper represent a change in the kinds of attacks and probes we track. Previously, it had been common for a single attacker to target multiple sites. Now we see indications of multiple attackers working together to target either single sites or multiple sites. We can use all of the analysis techniques we have learned to find differences or similarities in delivery mechanisms. These may help provide clues as to the number of discrete attackers involved, especially when we have data across a fairly large time window, such as a week or longer.

It should be noted that these techniques are starting to be widely used and the attacker community is building decoy techniques into commonly available tools. However, we are not aware of a widely available distributed scanner, or exploit delivery system. Additionally, these coordinated attacks display a significant amount of variability making them difficult to detect with signature-based algorithms.

There are three obvious purposes for coordinated attacks and probes: stealth, firepower, and intelligence gathering.

7.1. Stealth

By working from multiple IP addresses the attackers achieve a smaller per-IP signature and are more difficult to detect through conventional means. In addition, stealth is enhanced by the development of new hard-to-detect probing techniques such as Reset scans.

7.2. Firepower

By coordinating multiple attacking IP addresses, the attackers will be able to deliver more exploits to destination hosts in a smaller period of time. Furthermore, the defensive technique of blocking an attacker IP, also known as shunning, will be less effective. A single attacking entity can utilize multiple non-related Internet addresses for the attacks. This is especially true for denial of service attacks; most of these do not rely on a connection being made, so the probability of the address being spoofed is very high. Some of these coordinated probes and scans we detect today may be practice runs for future larger scale attacks. After a new exploit is discovered, there is often a limited "window of opportunity" for its use; usually until countermeasures are developed.

7.3. Intelligence gathering

As discussed in the coordinated traceroute example, by working from different IP addresses on different backbones against the same target, it is possible to obtain data that is impossible to obtain from a single source IP scan or probe. These data may include shortest route data, (i.e. packets from source A arrive faster than from source B), or even potential backdoors, (i.e. packets from source A can gain access to hosts that source B can't see). This type of data can be used to optimize future scans, probes, or attacks. It could also be used to isolate a target site by attacking the links it uses to communicate with the outside world.

The SFRP example shows how a network of servers can simply wait for the customer to come to them. The progress in TCP stack analysis is very impressive and we wouldn't be surprised to see this capability become integrated into commercial server software as one more method of gathering intelligence about the systems that visit the server.

8. Final words

Analysis of the network traffic collected by the SHADOW team indicates that new exploit delivery mechanisms are being developed and refined. These techniques employ multiple attackers and decoy hosts in an attempt to increase stealth, firepower, and reconnaissance.

Much of the network traffic discussed in this paper is definitely the result of coordinated attacks. However, a small portion can also be attributed to deception techniques and second order effects. Therefore, it is extremely important for the analyst to differentiate between the two possible causes of suspicious network traffic, and react accordingly. To this end, we have discussed the motivations for coordinating attacks, and also provided examples of each attack type. Methods for detecting this coordinated activity have been illustrated when possible.

Reference:

SHADOW is a freely available, public domain intrusion detection system. Information and software for the SHADOW System can be downloaded from the following website:

http://www.nswc.navy.mil/ISSEC/CID

Intrusion Detection and Intrusion Prevention on a Large Network. A Case Study.

Tom Dunigan, Network Research
Oak Ridge National Laboratory
Greg Hinkel, Computer & Network Security
Oak Ridge National Laboratory

Abstract

This paper describes the general requirements for an Intrusion Prevention and Detection System and the methods used to prevent and detect intrusions into Oak Ridge National Laboratory's network. In this paper we describe actual intrusions, how they were detected, and how they were handled. We also describe the monitoring tools we use for detecting intrusions.

Introduction

At Oak Ridge National Laboratory (ORNL), we have an open environment in which researchers around the world must collaborate with ORNL researchers. These users want and need easy access to each other's data, programs, and correspondence. Furthermore, many of the researchers have been accustomed to unfettered access to and from the Internet. Obviously, we also have data that should not be available to external users.

Our network consists of approximately 18,000 computers running a variety of operating systems, including UNIX, VMS, Windows, and MacOS. Our users abilities range from "untrained" desktop users to highly trained supercomputer programmers.

An open environment like ORNL's poses many security concerns. The dynamic nature of the work performed at ORNL introduces additional security concerns in that new project initiatives, with new users and new computers, begin almost daily. These new projects often create sudden increases in network activity from new and different computer systems, and the sudden increases make it difficult to weed out "new project" traffic from intrusion attempts. Also, many of our "users" are not physically located at ORNL. Trying to determine if a remote user is the "legitimate user" is not an easy task. The question, "Was login information sniffed by a hacker who is now logging in?," is quite difficult to answer.

A security plan is essential. Knowing what to look for takes time, experience, diligence, and a lot of luck. Our plan needed to answer the following questions.

- What is the threat?
- What can happen if an intrusion occurs?
- What should we watch for?
- What should we report?
  - What should our intrusion detection system report to us?
  - Should we report intrusions to someone and if so, to whom?
- What should we do if and when we suspect an intrusion?

Intrusion prevention is our goal. However, it was clear that we would not be able to completely prevent intrusions, so we decided to:

- try to reduce the number of possible intrusions, and
- quickly detect any intrusions that did occur.

A simple solution to intrusion prevention and detection was not possible at ORNL. Trying to reduce the number of intrusions would have to be accomplished by providing secure mechanisms for end users to access their computer systems and then educating those users and their system administrators about the proper use of those secure mechanisms. Additional hardware and software would be required for intrusion detection. Detecting intrusions in real time is preferable and in isolated cases is possible. However, to reduce the likelihood of terminating a legitimate connection and to be more effective at detecting intrusions, it was clear that we would have to log and analyze users' activities. There are commercial packages that satisfy some of our requirements; however, none would satisfy all of our requirements. Therefore, we had to implement a specialized program that used commercial packages in conjunction with solutions developed in-house.

At ORNL, we use a layered approach to network security because multiple layers make penetration