Global Information Assurance Certification Paper

GIAC Practical Assignment

Intrusion Detection in Depth

by

William A. Shaffer

Version 3.0 (August 2001)

Table of Contents

Assignment 1: Intrusion Detection Needs to Be Part of Operating Systems and Services Software

Introduction

Intrusion detection is an important part of effective system security. However, there are several trends that could lessen the effectiveness of current knowledge-based intrusion detection systems. The trends include the increased complexity of networks and systems, a number of trends that make pattern recognition difficult, and difficulties in organizations adopting effective intrusion detection.

To partially counteract these problems, intrusion detection systems need to evolve in three ways:

- Intrusion detection needs to be moved into operating systems, services, and other infrastructure software.
- Intrusion detection needs to be specification-based rather than monitoring signatures of past attacks.
- Intrusion detection needs to communicate with analyst stations using standardized intrusion detection communication messages.

Two Approaches to Intrusion Detection

Security experts sometimes differentiate between two approaches to intrusion detection: knowledge-based and behavior-based. These are sometimes referred to as misuse detection and anomaly detection. Each approaches has strengths and limitations.

Knowledge-Based Intrusion Detection

Knowledge-based approaches attempt to recognize one or more characteristics of the attack and signal an alert. Most intrusion detection tools use this approach. The fundamental feature is a "knowledge base" or set of rules for recognizing the attack. Therefore, many tools using this approach are called rules-based. Using these rules, they recognize patterns in the behavior of the computer by matching them against one or more rules in the knowledge base. The intrusion detection systems Snort and several other systems use this approach. Debar [DEBAR2000] discusses this approach in more detail.

Behavior-Based Intrusion Detection

Behavior-based approaches compare computer behavior with a reference behavior. Denning [DENNING1987] outlined the basic principles of this approach in 1987.

There are at least three techniques used: statistical approaches, predictive pattern generation, and neural networks. In statistical approaches, the system determines profiles for users and other

subjects in the system. It then identifies anomalies that deviate substantially from the profile. A particular issue with this approach is the metrics chosen to profile. In predictive pattern generation, the system tries to predict future events based on past events. In the neural network approach, the system trains a neural network to recognize typical behavior and predict future behavior. Deviations from this behavior represent possible anomalies. (See [SUNDARAM1996] for a more detailed description of these approaches.)

Limitation of Intrusion Detection Systems

As Allen et. al. describe [ALLEN2001], there are a number of limitations of intrusion detections systems that may prove to pose a number of problems in the future. These limitations include:

- The complexity of systems, networks, and of attacks has increased and is likely to increase.
- A number of trends make pattern recognition more difficult.
- There are several factors that prevent organizations from selecting and effectively deploying intrusion detection systems.

Increased Complexity of Systems, Networks, and Attacks

The environment in which intrusion detection systems must operate is increasingly complex. Many networks continue to expand both in terms of the number of servers and workstations and the number of services provided on the network. In prior years, a site may have provided electronic mail (e-mail) and file transfer (FTP), and later a Web site, it now may provide Web conferencing, instanct messaging, and remote administration. In the future, it may need to provide telephony, radio broadcasting, and video broadcasting. Trends in supply chain integration and other e-Business activities require an increasing number of services be exposed to the outside Internet. As the amount of traffic grows on the network, network intrusion detection systems may have difficulty keeping up with this traffic.

A large and growing software industry churns out an increasing number of software products, many of which are network enabled. This increases the number of ports used and the number of application protocols. It also increases the number of vulnerabilities. Due to poor software development practices, many software products suffer from buffer overflow. This defect allows an attacker to insert and execute code in a product by transmitting to it more data than the programmer provided space for. (See [DILDOG] for a detailed discussion of buffer overflow exploitation.) The CodeRed Worm, discovered in July, 2001, demonstrated that Year 2000 date problem was not the only poor coding practice that threatened the computing infrastructure throughout the world. (See [SYMANTEC2001]). As the number of software products grows, the number of buffer overflow vulnerabilities will increase.

In many instances, there is a trend toward decreased security in application software. The trend to equip every application program from word processors to e-mail clients with powerful programming language interpreters. Thus, attacks can be launched from documents, spreadsheets, and electronic mail.

Adding to the problem is the insecurity provided by mainstream operating systems. Loscocco et al. make the point current operating systems are unable to provide adequate security for the applications run on them. [LOSCOCCO1998] Given the effort by SANS and other organizations to increase the security of Windows and UNIX, few would dispute Loscocco's statement.

As the number of attacks grows, it places an increasing burden on many types of intrusion detection systems, particularly network intrusion detection systems that function by matching the patterns or signatures in network traffic. For example, the number of rules in the Snort rule set [SNORT2001] continues to grow. Similarly, the number of virus definitions in Norton AntiVirus 2001™ is approximately 58,000. Vendors may find it difficult to weed out old rules and definitions, even though the attacks are unlikely to be active. Thus, the sizes of these rule sets and definition sets will likely continue to grow. At some point they are likely to bog down the recognition process and create a sever knowledge burden on support staff. How does a security analyst make intelligent selection of rules from a rule set of 100,000 rules?

Making Pattern Recognition More Difficult

Several trends are making pattern recognition, particularly for network intrusion detection systems, much more difficult. The first of these is the increased use of encryption. Although encryption can be used to make authentication more secure and to increase the confidentiality of information, encrypted payloads makes recognition of patterns on the network extremely difficult. The cleartext can be analyzed only after the host has decrypted the payload. However, encryption does not necessarily prevent an attacker from attacking a service. For example, an attacker can attempt a buffer overflow over a secure socket layer (SSL) connection, just as he can over a regular HTTP connection.

Second, there are a number of ways of evading network intrusion detection. Ptacek and Nesham described in 1998 identify several ways in which the pattern matching can be evaded by clever manipulation of the TCP/IP packets. They outline a number of techniques in which the intrusion detection system can be made to see information that the receiving host does not see (insertion) or not to see information that the receiving host does see (evasion):

- The time to live (TTL) value may not be large enough for the number of hops to the destination host.
- A packet may be too large for a downstream link to handle without fragmentation.
- The destination host may be configured to drop source-routed packets.
- The destination host may time partially received fragments out differently from the intrusion detection systems.
- The destination host may reassemble overlapping fragments differently from the intrusion detection system.

- The destination host may not accept TCP packets bearing certain options.
- The destination host may silently drop packets with old timestamps.
- The destination host may resolve conflicting TCP segments differently from the intrusion detection system.
- The destination host may not check sequence numbers on RST (reset flag set) messages. (Items taken from Figure 7, [PTACEK1998]).

The fundamental problem is that different systems handle unusual streams of IP and TCP packets in different ways. Sufficiently complex sets of packets can be transmitted so that intrusion detection systems are fooled.

Ptacek and Nesham applied a series of tests to four popular (in 1998) network intrusion detection systems. In the case of each system, they were able to bypass the recognition capabilities and perform simulated attacks without recognition. Their work is now over four years old. No doubt, vendors have addressed some of the problems outlined in the paper. But the complexity of TCP/IP still makes it difficult to handle complex configuration of packets.

A third trend that makes intrusion detection more difficult is increased use of mobile code. The increasing use of Java applets, Javascript, and, in particular ActiveX, makes intrusion detection more difficult. Malicious code or malware can be downloaded from a Web site or sent as an attachment in an e-mail. ActiveX poses particular problems, since an ActiveX component can perform a wide-range of functions on a Windows-based computer. ActiveX uses a code-signing defense involving a digital signature. When the Internet Explore browser identifies the component, the browser displays a dialog box showing the signer of the control. The user can then accept the control, and the browser downloads the component and executes it. While in theory offering protection, these components are so common on so many Web sites, that the user often is constantly authorizing the use of code about which he knows nothing. Monitoring such code is similar to monitoring viruses. The Nimba worm propagates itself via electronic mail. Due to a vulnerability in Microsoft Outlook, the worm can be executed merely by having the user open the e-mail. [SYMANTEC2001a]

A fourth trend is the increased use of network switches instead of hubs. When computer systems are connected via hubs, traffic to all computers on the segment can be "sniffed" and monitored by a network intrusion detection system. With a switch, the traffic for each computer is transmitted only on the cable going to that computer. The intrusion detection system must make use of the monitoring port on the switch. But there may not be anywhere where a network intrusion detection system can view both internal and external traffic [ALLEN2001]. Without an integrated set of detectors, it may be difficult to identify host resonance and other surveillance and attacks involving multiple computers.

A fifth factor is that knowledge-based intrusion detection systems have limited ability to identify new types of attacks. This is a common failing in security. It is hard to guard against those things you cannot imagine. Even the best intrusion detection systems are of no help if someone flies a plane into your server farm. Donn Parker argues that one reason that computer crime is

unpredictable is that computer criminals are unpredictable. [BRINEY1999] Metrics on past attacks and threats may unreliable indicator of future threats.

The DARPA Off-Line Intrusion Detection Evaluation provides a measure of the ability of intrusion detection systems to recognize new attacks. Lippmann reports:

> Detection accuracy was poor for previously unseen new, stealthy, and Windows NT attacks. Ten of the 58 attack types were completely missed by all systems. Systems missed attacks because protocols and TCP servicers were not analyzed at all or to the depth required, because signatures for old attacks did not generalize to new attacks, and because auditing was not available on all hosts. [LIPPMANN2000, Abstract]

Difficulties in Deploying Intrusion Detection

Several practical factors inhibit the ability of organizations to select and effectively deploy intrusion detection systems. First, intrusion detection products are constantly changing and new ones are entering the market. There are limited standards for evaluating such systems and limited data on their effectiveness. McHugh notes that:

> Reviews and comparisons of commercial IDS systems appear from time to time, usually at the Web sites of on-line publications. The reviews are generally superficial and lack details concerning the test methods used. The rapid rate at which new products are introduced and existing products modified gives these reviews a limited window of utility. [MCHUGH2000, p. 266]

Vendors seldom describe the time and effort to maintain rule set or other information needed to keep the system running effectively. These factors combine to make the selection of an intrusion detection tool difficult.

There is a lack of data and systems for testing intrusion detection system. As the number of rules increases, it is increasingly unlikely that that all the rules work properly. Experience shows that system configurations do not always yield the desired security without confirmation. As the size of rule sets increase, the likelihood of defects increases.

Due to the relative newness of the field and the shortage of information technology professions in general and information security professionals in particular, there is a lack of qualified personnel to perform analysis.

The Advantages of Intrusion Detection Systems

The limitations discussed so far are not meant to suggest that Intrusion Detection Systems do not have value. Many types of attacks can be detected. Many do not involve a high level of sophistication that could evade detection systems. An intrusion detection can be used to test the effectiveness of firewalls in blocking malicious traffic.

Other trends are likely to improve the ability of intrusion detection systems. Hardware is becoming faster and cheaper, permitting more computing power to be applied to pattern matching. The methodology of intrusion detection is likely to improve. Moreover, new

techniques are being researched and attempted. (See [ALLEN2000, Appendix D] for examples of systems being researched.)

Recommendations for Future Directions

Predicting the future of intrusion detection is risky, but given the limitations described above, it seems possible that intrusion detection as commonly seen today is losing the battle. These are some recommendations for improving the situation:

- Intrusion detection needs to be pushed down into the operating systems and network services software.
- Intrusion detection needs to use specification-based rules as well as other techniques.
- Intrusion detection need to communicate with analysis workstations using standard intrusion detection messages.

Moving Intrusion Detection into the Operating System and Services Software

Intrusion detection needs to be incorporated into the operating systems and application programs as an addition to current intrusion detection practices. Placing intrusion detection at this level allows monitoring of critical operating system primitives like:

- spawning taks
- deleting files
- modifying files.

Much current intrusion detection monitors symptoms, but not the results of actions. By placing inrusion detection in the operating system, the results of attacks can be monitored more accurately.

For example, the TCP/IP stack could contain rules to check for out of specification packets or other unusual packets like:

- source and destination addresses are the same
- meaningless combinations of IP flags like the SYN and RST flags both set on a packet
- Unusual or unnecessary fragmentation of packets.

The operating system could monitor the execution of new tasks and issue alerts when tasks are spawned for users with super user privileges who are accessing the system from a remote location.

Services software like Domain Name Servers (DNS) and Web servers can similarly be instrumented to alert on unusual activity. For example, Microsoft has introduced the

URLSCAN.DLL in Internet Information Server that checks URLs for various problems. Detects are logged in a special log file for this capability.

Operating systems and other software perform some of these types of detection, but enhancements to current logging need to be made.

- Alerting would direct machine readable messages to an analyst workstation. Although logging to log files is useful, it can be difficult to manually scan many log files. Log files can also be difficult to process by computer.

- There would be a modifiable set of rules that could be customized by the system administrator.

- The rule set would check more types of events than are typically logged.

Monitoring at the operating system and service software level avoids some, but not all, of the limitations described earlier in this paper. Evasion techniques will not be as effective, since the intrusion detection is seeing the same packets assembly from the TCP/IP stack as other system components. The stack will perform whatever fragmentation assembly is needed. The stack will also have an accurate picture of the state of TCP sessions. The intrusion detection will not fall out of synchronization with the host.

The intrusion detection will not be confused by encryption, since the packet payload will already be decrypted. By focusing on basic operating system and application functions, the analysis is simplified.

Specification-Based Monitoring

Specification-based monitoring uses a set of logical statements to determine unusual events occurring in the system. While this sounds like the knowledge-based approach, the focus is different. In the knowledge-based approach, the focus is on constructing rules to recognize attacks. The rules are developed based on experience with attacks. In the specification approach, rules are developed based on an analysis of the underlying system components.

In the example of the rules for the TCP/IP stack, the logical requirements for the stack would be analyzed and anomalies identified. Hopefully, anomalies would be checked even if there were no history of an attack.

Standard Intrusion Detection Messaging

For this approach to work, it is important that operating systems, service software, and other software can communicate detects to one or more analyst workstations, so that results can be correlated, filtered, and presented to the analyst in a meaningful way. The current system of log files from different vendors does not provide the compatibility and ease of analysis that is needed. A common, standard format, like the one being worked on by the Intrusion Detection Working Group, would permit accumulating alerts from the multiple software components and would allow analysis of the raw data.

Summary

The future of intrusion detection is clouded by trends in networks and systems. Three ways to help alleviate some of the problems are to move intrusion detection into the operating system, use specification-based detection, and use standard intrusion detection messaging to accumulate alerts at an analyst workstation.

There are some practical limitations to these approaches. As Loscocco [LOSCOCCO1998] describes, there are many suggested ways of improving operating system security. The industry has been slow to adopt even some basic measures. Why would it adopt intrusion detection? However, the growth of network appliances gives hope for improvements in operating system security. A network appliance is a server with a dedicated function, like a Web server, DNS server, or e-mail server. The growth of network appliances is being fueled by less expensive hardware. Simplifying the configuration of servers to run just a few services can enhance computer security. Since the hardware is not as expensive now as it was, several computers can be dedicated to particular functions. A high functionality operating system is unnecessary in this environment, since it does not have to support the wide range of capabilities required by multiple software packages. The operating system can focus on becoming simpler, more reliable, and more secure.

Assignment 2: Network Detects

Detect 1: HTTP Port Probe on Private Network

The Detect

Attack list from Black ICE Defender

Severity	Timestamp (GMT)	IssueId	IssueName	Intruder IP	Intruder Name	Victim Ip	Parameters	Count
39	2001-09-17 14:48:06	2003001	HTTP port probe	198.95.227.130	UCLA	192.168.72.199	port=80& reason=Firewalled	6
39	2001-09-19 13:59:16	2003001	HTTP port probe	155.135.16.62	SUVA	192.168.72.199	port=80& reason=Firewalled	6
39	2001-09-19 14:02:06	2003001	HTTP port probe	198.143.0.155	tv1.intercom. com	192.168.72.199	port=80& reason=Firewalled	4
39	2001-09-19 14:06:38	2003001	HTTP port probe	198.181.133.11		192.168.72.199	port=80& reason=Firewalled	4
39	2001-09-19 14:13:22	2003001	HTTP port probe	198.143.0.155	tv1.intercom. com	192.168.72.199	port=80& reason=Firewalled	4
39	2001-09-19 14:14:02	2003001	HTTP port probe	198.181.133.11		192.168.72.199	port=80& reason=Firewalled	4
39	2001-09-19 14:17:35	2003001	HTTP port probe	198.3.183.73	faiis2.interliant. com	192.168.72.199	port=80& reason=Firewalled	4
39	2001-09-19 14:20:47	2003001	HTTP port probe	98.143.204.13	RUACH	192.168.72.199	port=80& reason=Firewalled	4

Snort log

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

09/19-10:13:18.501228 ARP who-has 0.0.0.0 tell 0.0.0.0

09/19-10:13:18.509816 ARP who-has 0.0.0.0 tell 0.0.0.0

09/19-10:13:19.617680 198.143.0.155:2978 -> 192.168.72.199:80
TCP TTL:111 TOS:0x0 ID:13423 IpLen:20 DgmLen:48 DF
******S* Seq: 0xCB3532E1  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:19.622591 192.168.72.199:1132 -> 198.143.0.155:137
UDP TTL:128 TOS:0x0 ID:1675 IpLen:20 DgmLen:78
Len: 58
80 B0 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:22.933711 198.143.0.155:2978 -> 192.168.72.199:80
TCP TTL:111 TOS:0x0 ID:13679 IpLen:20 DgmLen:48 DF
******S* Seq: 0xCB3532E1  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:23.830293 192.168.72.199:1132 -> 198.143.0.155:137
UDP TTL:128 TOS:0x0 ID:1676 IpLen:20 DgmLen:78
Len: 58
80 B0 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
```

```
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
. . . (part of log omitted)
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:50.119654 ARP who-has 0.0.0.0 tell 0.0.0.0

09/19-10:13:51.891841 ARP who-has 0.0.0.0 tell 0.0.0.0

09/19-10:13:59.607738 198.181.133.11:1692 -> 192.168.72.199:80
TCP TTL:115 TOS:0x0 ID:64173 IpLen:20 DgmLen:44 DF
******S* Seq: 0x9DF210D  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:59.608044 192.168.72.199:1132 -> 198.181.133.11:137
UDP TTL:128 TOS:0x0 ID:1776 IpLen:20 DgmLen:78
Len: 58
80 B0 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:59.809892 192.168.72.199:1137 -> 208.216.229.253:53
UDP TTL:128 TOS:0x0 ID:1777 IpLen:20 DgmLen:73
Len: 53
00 19 01 00 00 01 00 00 00 00 00 00 02 31 31 03  .............11.
31 33 33 03 31 38 31 03 31 39 38 07 69 6E 2D 61  133.181.198.in-a
64 64 72 04 61 72 70 61 00 00 0C 00 01           ddr.arpa.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:59.824788 208.216.229.253:53 -> 192.168.72.199:1137
UDP TTL:252 TOS:0x0 ID:63221 IpLen:20 DgmLen:148 DF
Len: 128
00 19 81 83 00 01 00 00 00 01 00 00 02 31 31 03  .............11.
31 33 33 03 31 38 31 03 31 39 38 07 69 6E 2D 61  133.181.198.in-a
64 64 72 04 61 72 70 61 00 00 0C 00 01 03 31 39  ddr.arpa......19
38 07 69 6E 2D 61 64 64 72 04 61 72 70 61 00 00  8.in-addr.arpa..
06 00 01 00 00 28 74 00 2F 09 41 52 52 4F 57 52  .....(t./.ARROWR
4F 4F 54 04 41 52 49 4E 03 4E 45 54 00 04 62 69  OOT.ARIN.NET..bi
6E 64 C0 53 77 46 3D 40 00 00 07 08 00 00 03 84  nd.SwF=@........
00 0A 8C 00 00 00 2A 30                          ......*0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:13:59.826446 192.168.72.199:137 -> 198.181.133.11:137
UDP TTL:128 TOS:0x0 ID:1778 IpLen:20 DgmLen:78
Len: 58
80 76 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  .v.......... CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:14:01.323972 192.168.72.199:137 -> 198.181.133.11:137
UDP TTL:128 TOS:0x0 ID:1779 IpLen:20 DgmLen:78
Len: 58
80 78 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  .x.......... CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

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

```
09/19-10:14:02.826124 192.168.72.199:137 -> 198.181.133.11:137
UDP TTL:128 TOS:0x0 ID:1780 IpLen:20 DgmLen:78
Len: 58
80 7A 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  .z.......... CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:14:02.887653 198.181.133.11:1692 -> 192.168.72.199:80
TCP TTL:115 TOS:0x0 ID:10160 IpLen:20 DgmLen:44 DF
******S* Seq: 0x9DF210D  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:14:03.487115 192.168.72.199:1132 -> 198.143.0.155:137
UDP TTL:128 TOS:0x0 ID:1781 IpLen:20 DgmLen:78
Len: 58
80 B0 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/19-10:14:03.487253 192.168.72.199:1132 -> 198.181.133.11:137
UDP TTL:128 TOS:0x0 ID:1782 IpLen:20 DgmLen:78
Len: 58
80 B0 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

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

1. Source of Trace

The traces came from the author's employer's internal network. This network is a switched network connected to the Internet via a T1 line controlled by a CISCO router. The router provides a firewall and network address translation. The targeted computer is the author's laptop computer connected to the network. The intrusion detection software is running on that laptop.

2. Detect was generated by

The first trace is the attack list from the Black ICE Defender. The meaning of the fields according to Network ICE is:

The columns are, from left to right:

severity

This is a number from 1-99 that indicates the severity of an attack, where 1 is not very severe, and 99 is the most severe attack. Unfortunately, these levels do not have any precise meaning. Even an attack at level 1 may result in a compromise

of the machine, whereas an attack at level 99 could be harmless. The assigned level is just a best-guess.

timestamp

This indicates the time and date of the **last** time the attack occurred. Attacks are "coalesced", meaning that if the same attack occurs multiple times, earlier attacks are sometimes removed from the list and simply merged with the latest one. A count of the number of times an attack has occurred is kept in another column. *This timestamp is kept in GMT (aka UTC), and is probably several hours off from the time you see in the user interface.* The ISP will want the time in this format so they don't have to worry about what timezone you are in.

"issueId"

A numeric identifier for this attack type. Each of the more than 300 attacks that the intrusion-detection component detects is assigned a unique number. This number is used for all internal processing of events. This number may also be pasted at the end of the URL http://advice.networkice.com/advice/intrusions/ in order to get help on the event.

"issueName"

The name of the attack. Each of the unique "issueId" numbers has a name associated with it.

intruder's IP address

The IP address of the attacker. Remember that IP addresses can sometimes be "spoofed" (forged), or that an intrusion may be a "false-positive", so there isn't a 100% chance that this is actually a hostile person.

intruder's name

The name of the intruder. We scan both Internet databases like DNS as well as the attacker itself in order to find the "best-name" of the machine, then display it here.

victim's IP address

This is the IP address of the host the intruder was attacking. For example, if a user is running the product and gets attacked on a dial-up, then this will be the IP address assigned to that machine during that dialup session.

"parameters"

This contains some detailed information about the attack. For example, in a "TCP port probe" scan, this will contain a list of "ports" the attacker was scanning. The meaning of this information is documented in the "advICE" database.

count

The number of times this attack was seen. [NETWORKICE1999]

The second trace is the log from Snort. Snort was configured to capture all traffic on the network segment and dump the values in hexadecimal and ASCII. The fields are:

Line #	Fields
1	`Date-Time Source Address:Port Number ->Destination Address:Port Number`
2	Protocol, Time to Live(TTL), Type of Service (TOS), IP identifier, IP Header Length, Datagram Length, Flags
For TCP	
3	Flags, Sequence, Acknowledgement #, Window Size, TCP Header Length
4	TCP Options

3. Probability the source address was spoofed

The attackers' addresses are probably not spoofed. This activity is probably an attempt to scan the network. The information from the scan will not return to the attacker if the address is spoofed. The attacker's system retransmits the original SYN TCP packet when it does not receive an acknowledgement from the original packet.

4. Description of attack

This type of attach was originally reported by employees whose computers were equipped with Black ICE Defender. The presence of these probes surprised employees, since the internal network uses private IP addresses which are not supposed to be routed on the external Internet. The question arose about how these packets could reach the firewall in the first place.

Following reports of these probes, the author ran Black ICE defender. Black ICE Defender signaled six separate probes. When the author started to observe probes, he activated Snort to record all network packets. The Snort listing above is a selection from this log. During the recording, the author was also accessing the World Wide Web. The lengthy listing of these accesses have been deleted.

As the Snort listing indicates, the intruder attempts to make a connection to Port 80 on the target machine. The connection is stopped by Black ICE Defender, which, in turn, attempts to query the intruder's computer on Port 137 to see if it will respond to a NetBios query. Black ICE Defender performs this query to obtain more information about the intruder. The listing shows probes from two different sources, and the Black ICE response. The second probe also shows Black Ice performing DNS lookup, also to gain more information about the intruder.

5. Attack mechanism

The intruder appears to attempt a connection to port 80. The issue is how is that possible, since routers on the Internet will not route private addresses. Three hypotheses were considered.

1. The intruder entered through our Windows RAS modem connection rather than through the router. But if this happened, the intruder would have to have an IP address on the employer's network.

2. The second hypothesis is that the packets were misinterpreted packets used to scan the router address. With network address translation, the router takes a packet from the laptop and converts the address to the router's address and changes the port number to a selected, unused port on the router. If a packet coming into the router has that port selected, the router will route it to the laptop. If an intruder were to scan higher ports on the router, he could send a packet to the port, and the router could pass it along to the laptop. However, this hypothesis was rejected because the probe would come into the port the laptop had opened, not port 80.

3. The third hypothesis is that the router was handling a broadcast message sent to port 80 of the employers class C broadcast address in the form of x.x.x.255. The router then passes the packet to computers on the internal network. This seems the most likely possibility.

6. Correlations

David Jones [JONES2001] reports a similar phenomena. Vicki Irwin suggested the following explanation:

> I have seen routers translate the IP address as well as the hardware address when forwarding a broadcast to their locally attached LAN. For example, if your subnet is 10.1.1.x and someone outide your subnet sends a "ping 10.1.1.255", when the router attached to the 10.1.1.x subnet receives the packet destined for 10.1.1.255, the router may change the destination address to 255.255.255.255 before forwarding the packet.
>
> Here's an example using tcpdump (below) and a Cisco router. One odd thing you mentioned though is that you are seeing the packets destined for port 80 ... I'm assuming that is TCP port 80? A broadcast using TCP doesn't make any sense .... I don't know what that might be used for. [IRWIN2001]

Sending the TCP packet to Port 80 seems to be an attempt to detect hosts running Web servers.

7. Evidence of active targeting

The intruder is not targeting a specific host, but is probably trying to locate Web servers at one or more Class C networks.

8. Severity

The table below show the calculation of the severity of the attack.

Aspect	Value	Reason	Guide	Description
Criticality	3	Web servers targeted	5	Firewall, DNS server, core router
			4	E-mail relay/exchange
			2	User UNIX desktop system
			1	MS-DOS 3.11
Lethality	2	Scan may succeed	5	Attacker can gain root across net
			4	Total lockout by denial of service
			4	User password, like a sniffed password
			1	Attack is unlikely to succeed
Countermeasures			5	Modern operating systems, all patches
System	4	Up-to-date op sys	3	Older operating systems, missing patches
Net	3	Network needs to improve	1	No wrappers/allows fixed unencrypted passwords
Severity	**-2**			(criticality+lethality-system-net)

9. Defensive recommendation:

This incident, while not particularly dangerous in itself, underscored the need to improve intrusion detection and network defenses on the employer's network. The following steps are recommended:

- Review the router configuration and firewall configuration to insure that the configuration performs the desired functions. A contractor installed the router, but no longer is available for support. The configuration of the router needs to be reviewed and tested on a regular basis.

- The company should institute network intrusion detection on the internal network, and outside the firewall if possible.

- The company should equip laptop users with a personal firewall product like Black ICE Defender. Not primarily to protect against threats on the internal network, but because many users also dial into an ISP to access the company's e-mail and other computer resources.

10. Multiple choice test question

A router performing network address translation performs which operation on an outgoing packet:

1. The destination address is changed to the router's address, and the destination port number is changed to an unused number.

2. The source address is changed to the router's address, and the source port is changed to an unused port number.

3. The source address is changed to the router's address, and the source port remains unchanged.

4. The destination address is remains unchanged, and the port number is changed to Port 80.

(Answer: 2)

Detect 2: Code Red Attack

```
209.82.88.140 - - [19/Jul/2001:12:20:32 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207

209.57.74.67 - - [19/Jul/2001:12:34:15 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207

216.34.97.18 - - [19/Jul/2001:12:44:48 -0400] "GET /resume.htm HTTP/1.1" 200 19576
211.172.232.84 - - [19/Jul/2001:12:45:04 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207

207.68.177.17 - - [19/Jul/2001:12:45:17 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207

203.67.42.66 - - [19/Jul/2001:13:09:35 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207

205.247.29.39 - - [19/Jul/2001:13:11:50 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207

161.58.176.75 - - [19/Jul/2001:13:30:37 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207
```

1. Source of Trace

This trace comes from the Web server operated by the author. The Web server is connected to the Internet via cable modem.

2. Detect was generated by

The log above was generated by the iPlanet Web server. The fields in the log are:

Field #	Field Name	Description
1	Ses->client.ip	IP Address of the client (requester)
2	Req->vars.auth-user	Authorized User – Not filled due to anonymous access
3	[%SYSDATE%]	System date and time (Eastern Daylight Savings)
4	Req->reqpb.clf-request	The request
5	Req->srvhdrs.clf-status	Request status
6	Req->srvhdrs.content-length	Server header content length

3. Probability the source address was spoofed

Since the initiator of the attack needs to establish a TCP session, the address is not spoofed, although the attacker is likely operating from a victimized computer whose owner is unaware that it is the source of the attack.

4. Description of attack

The attack is documented in a number of places, including CERT [CERT2001] and Microsoft [MICROSOFT2001]. A detailed analysis is provided by eEye Digital Security [EEYE2001]. The attack is caused by a "worm" that searches out Web servers on Port 80. It exploits a buffer overflow in the Microsoft Index Service. If it is successful in the exploit, it begins executing on the victimized host. In an early variant, if the host has a default language of English, the worm would deface the host's Web page with the message:

HELLO! Welcome to http://www.worm.com! Hacked By Chinese!

The worm's behavior is time sensitive, based on the day of the month. In days 1 through 19, the worm attempts to propagate to other, randomly chosen hosts. In days 20 through 27, it launches a packet-flood denial of service attack at IP address 198.137.240.91.

The logs of the author's server show limited activity on July 19. Then there is no activity until August 1, in which the Code Red activity resumes. This is consistent with the analysis of the worm's behavior. (See below.) After August 1, there is a substantial amount of requests of the Code Red type, including subsequent variants like Code Red.

5. Attack mechanism

The worm exploits a vulnerability in the Microsoft Index Service. In Microsoft Internet Information Server, there are ISAPI extensions that cause certain dynamic linked libraries (DLLs) when the client attempts to retrieve files of certain extensions. Files with the .ida extension cause the Index Service IDL.DLL to run. This DLL has a buffer overflow

defect. It is not necessary for the Index Service to be running, nor does the requested file have to exist. Typically there is no default.ida file as listed in the GET request.

The Web server logs show the signature of a typical buffer overflow. The request consists of the file name defalt.ida, followed by a number of letter n's. After these are the instructions that will be executed when the buffer in the IDL.DLL. The code has a few Intel instruction set NOOPS (0x90).

6. Correlations

Common Vulnerabilities and Exposure #: CAN-2001-0500

The exploit was widely reported, but much of it well after the fact. Microsoft reported the vulnerability and had a patch on June 18, 2001 [MICROSOFT2001]. The report did not reference the Code Red worm itself. eEye Digital Security issued an early, detailed discussion of the Code Red worm on July 17 [EEYE2001]. CERT issued an advisory on July 19 [CERT2001].

7. Evidence of active targeting

The worm is content to propagate to any server running IIS with the Index Service vulnerability.

8. Severity

The table below outlines the severity from the point of view of the author's individual server. The severity of the attack, in terms of use of band width on the Internet and the encouragement to further refinements of this attack, is much higher from the perspective of the Internet community.

Aspect	Value	Reason	Guide	Description
Criticality	3	Web servers targeted	5	Firewall, DNS server, core router
			4	E-mail relay/exchange
			2	User UNIX desktop system
			1	MS-DOS 3.11
Lethality	5	Gains SYSTEM priv	5	Attacker can gain root across net
			4	Total lockout by denial of service
			4	User password, like a sniffed password
			1	Attack is unlikely to succeed
Countermeasures			5	Modern operating systems, all patches
System	5	Up-to-date op sys	3	Older operating systems, missing patches
Net	4	Identified by Snort	1	No wrappers/allows fixed unencrypted passwords
Severity	**-1**			(criticality+lethality-system-net)

9. Defensive recommendation

The author's server was not substantially impacted, because the author runs the iPlanet Web Server, rather than IIS.

IIS can protect against the worm by installing the patch to the Index Server. However, a much more comprehensive approach to IIS is needed. Fossen outlines a program of configuration management that will provide much greater safety for IIS. A particular step that would have protected IIS servers even without the patch is the removal of unneeded ISAPI extensions, including the Index Service, from the IIS configuration.

10. Multiple choice test question

Which statement below is true of a buffer overflow:

1. It disables a computer by filling up the disk.
2. It can only occur if the initiator has superuser or root privileges.
3. It only affects Web servers.
4. It can cause a computer to crash.

(Answer: 4)

Detect 3:

```
[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 3]
11/13-14:48:28.916005 63.117.235.7:80 -> 192.168.1.2:80
TCP TTL:46 TOS:0x0 ID:52004 IpLen:20 DgmLen:40
***A**** Seq: 0xF8  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 3]
11/13-14:48:29.146386 204.95.220.10:80 -> 192.168.1.2:80
TCP TTL:44 TOS:0x0 ID:52010 IpLen:20 DgmLen:40
***A**** Seq: 0xFA  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 3]
11/13-14:48:34.114249 204.95.220.10:80 -> 192.168.1.2:80
TCP TTL:44 TOS:0x0 ID:52110 IpLen:20 DgmLen:40
***A**** Seq: 0x10A  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]
```

1. Source of Trace

This trace comes from the Web server operated by the author. The Web server is connected to the Internet via cable modem.

2. Detect was generated by

Snort generated the alert logs shown above. The rule that generated the alert is:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap TCP";flags:A;ack:0;
reference:arachnids,28; classtype:attempted-recon; sid:628; rev:1;)
```

3. Probability the source address was spoofed

The scan shows that two IP addresses sending three packets. It might look like two separate intruders are scanning. But there is considerable similarity among the packets. First is the close proximity of time in sending the packets. Second is they all have the same window size (possibly the result of coming from the same version of the program). The IP identification fields are very close as are the TCP sequence fields. Records show that host 63.117.235.7 performed the same scan about a month before. Host 204.95.220.10 has not performed the scan before. As a guess, 63.117.235.7 is the host of the intruder and 204.95.220.10 is spoofed using the decoy option in *nmap*.

4. Description of attack

The intruder sends a TCP packet with the ACK flag set and a zero acknowledgement number. The packet is crafted. The source port is 80 as is the destination port. The intruder is attempting to determine if there is a host operating at the destination address. (There is a firewall that converts the public Internet address of the server to the private address shown in the log.)

5. Attack mechanism

The Nmap Manual Page [INSECURE2000] lists the following for the –PT option. It appears that the intruder is attempting to scan the host using TCP ACK packets.

> Use TCP "ping" to determine what hosts are up. Instead of sending ICMP echo request packets and waiting for a response, we spew out TCP ACK packets throughout the target network (or to a single machine) and then wait for responses to trickle back. Hosts that are up should respond with a RST. This option preserves the efficiency of only scanning hosts that are up while still allowing you to scan networks/hosts that block ping packets. For non root users, we use connect(). To set the destination port of the probe packets use -PT<port number>. The default port is 80, since this port is often not filtered out.

6. Correlations:

Common Vulnerabilities and Exposure #: CAN-2001-0500

The exploit is also discuss at http://www.whitehats.com/info/IDS28.

7. Evidence of active targeting

It seems likely that the intruder was scanning for hosts on the part of Class B network to which the author's server is connected.

8. Severity:

Aspect	Value	Reason	Guide	Description
Criticality	3	Web servers targeted	5	Firewall, DNS server, core router
			4	E-mail relay/exchange
			2	User UNIX desktop system
			1	MS-DOS 3.11
Lethality	2	Scan only	5	Attacker can gain root across net
			4	Total lockout by denial of service
			4	User password, like a sniffed password
			1	Attack is unlikely to succeed
Countermeasures			5	Modern operating systems, all patches

System	5	Up-to-date op sys	3	Older operating systems, missing patches
Net	5	Blocked by Black ICE	1	No wrappers/allows fixed unencrypted passwords
Severity	**-5**			(criticality+lethality-system-net)

9. Defensive recommendation

The scan is blocked by Black ICE and neither reaches the Web server software, nor is acknowledged by the server.

Both IP address should be blocked by Black ICE.

10. Multiple choice test question

If the destination host does not block the packet sent in an ACK scan, what does the destination host respond with:

1. A TCP packet with the reset flag set.
2. An ICMP packet with the reset flag set.
3. A UDP packet with the reset flag set.
4. A TCP packet with the SYN and ACK flags set.

(Answer: 1)

Detect 4:

Nov 10 01:59:46 62.3.65.77:3580 -> a.b.c.14:21 SYN ******S*
Nov 10 01:59:43 62.3.65.77:3593 -> a.b.c.27:21 SYN ******S*
Nov 10 01:59:43 62.3.65.77:3592 -> a.b.c.26:21 SYN ******S*
Nov 10 01:59:44 62.3.65.77:3649 -> a.b.c.83:21 SYN ******S*
Nov 10 01:59:46 62.3.65.77:3617 -> a.b.c.51:21 SYN ******S*
Nov 10 01:59:46 62.3.65.77:3628 -> a.b.c.62:21 SYN ******S*
Nov 10 01:59:46 62.3.65.77:3667 -> a.b.c.101:21 SYN ******S*
Nov 10 01:59:47 62.3.65.77:3761 -> a.b.c.195:21 SYN ******S*
Nov 10 01:59:52 62.3.65.77:3749 -> a.b.c.183:21 SYN ******S*
Nov 10 01:59:52 62.3.65.77:3747 -> a.b.c.181:21 SYN ******S*
Nov 10 01:59:52 62.3.65.77:3748 -> a.b.c.182:21 SYN ******S*
Nov 10 01:59:53 62.3.65.77:3637 -> a.b.c.71:21 SYN ******S*
Nov 10 01:59:54 62.3.65.77:3894 -> a.b.d.72:21 SYN ******S*
Nov 10 01:59:54 62.3.65.77:4023 -> a.b.d.201:21 SYN ******S*
Nov 10 02:00:03 62.3.65.77:4072 -> a.b.d.250:21 SYN ******S*
Nov 10 02:00:03 62.3.65.77:4067 -> a.b.d.245:21 SYN ******S*
Nov 10 02:00:05 62.3.65.77:4101 -> a.b.e.25:21 SYN ******S*
Nov 10 02:00:05 62.3.65.77:4173 -> a.b.e.97:21 SYN ******S*
Nov 10 02:00:05 62.3.65.77:4252 -> a.b.e.176:21 SYN ******S*
Nov 10 02:00:15 62.3.65.77:4346 -> a.b.f.14:21 SYN ******S*
Nov 10 02:00:15 62.3.65.77:4348 -> a.b.f.16:21 SYN ******S*
Nov 10 02:00:18 62.3.65.77:4350 -> a.b.f.18:21 SYN ******S*
Nov 10 02:00:18 62.3.65.77:4352 -> a.b.f.20:21 SYN ******S*
Nov 10 02:00:18 62.3.65.77:4364 -> a.b.f.32:21 SYN ******S*
Nov 10 02:00:18 62.3.65.77:4371 -> a.b.f.39:21 SYN ******S*
Nov 10 02:00:24 62.3.65.77:4350 -> a.b.f.18:21 SYN ******S*

1. Source of Trace.

http://www.incidents.org/archives/intrusions/msg02420.html

2. Detect was generated by

The information was generated from the Snort Portscan log. The fields in the log are:

Date

Time
Source address:port -> destination address:port
Type of attack
TCP Flags

3. Probability the source address was spoofed

The intruder is scanning for an FTP server. He probably is looking for a reply. The source address is probably not spoofed.

4. Description of attack

This is a fairly rapid scan of hosts on a Class C network to determine if there is an FTP server.

5. Attack mechanism

The intruder is attempting a connection to Port 21 on several hosts. Port 21 is the standard port for FTP.

6. Correlations

See http://archives.neohapsis.com for a number of reports on Port 21 scans.

7. Evidence of active targeting

The intruder is actively targeting the specific network.

8. Severity:

Aspect	Value	Reason	Guide	Description
Criticality	3	FTP Server	5	Firewall, DNS server, core router
			4	E-mail relay/exchange
			2	User UNIX desktop system
			1	MS-DOS 3.11
Lethality	2	Scan only	5	Attacker can gain root across net
			4	Total lockout by denial of service
			4	User password, like a sniffed password
			1	Attack is unlikely to succeed
Countermeasures			5	Modern operating systems, all patches
System	4	?	3	Older operating systems, missing patches
Net	4	?	1	No wrappers/allows fixed unencrypted passwords
Severity	**-3**			(criticality+lethality-system-net)

9. Defensive recommendation

If there are no FTP servers or the servers do not need to be made available on the Internet, then block port 21 traffic at the firewall.

Possibly block this IP address at the fire wall.

10. Multiple choice test question

The SYN flag in the above trace indicates the initiating host is attempting:

1. To ping the destination host.
2. To identify the number of hops to the destination.
3. To acknowledge a TCP packet.
4. To initiate a TCP session with the destination.

(Answer: 4)

Detect 5: Scan Proxy Attempt

[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
11/13-16:23:15.402190 212.160.157.111:4206 -> 192.168.1.2:8080
TCP TTL:107 TOS:0x0 ID:60964 IpLen:20 DgmLen:48 DF
******S* Seq: 0x13CCB48F Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
11/13-16:23:18.321780 212.160.157.111:4206 -> 192.168.1.2:8080
TCP TTL:107 TOS:0x0 ID:61757 IpLen:20 DgmLen:48 DF
******S* Seq: 0x13CCB48F Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
11/13-16:23:24.330836 212.160.157.111:4206 -> 192.168.1.2:8080
TCP TTL:107 TOS:0x0 ID:62849 IpLen:20 DgmLen:48 DF
******S* Seq: 0x13CCB48F Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
11/13-21:23:55.374579 212.160.157.111:4759 -> 192.168.1.2:8080
TCP TTL:108 TOS:0x0 ID:25825 IpLen:20 DgmLen:48 DF
******S* Seq: 0x13AED554 Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

1. Source of Trace

This trace comes from the Web server operated by the author. The Web server is connected to the Internet via cable modem.

2. Detect was generated by

The information was generated from the Snort alert log. The fields in the log are described in the third detect.

3. Probability the source address was spoofed

The source address is probably not spoofed, since the propose of the scan is to receive back information about the system.

4. Description of attack

A connection is attempted to Port 8080.

5. Attack mechanism

The owner of the server maintains a Web server at this port, which supports Web based mail. Www.ripe.net indicates that the address is from a polish organization which would have no legitimate reason for accessing this site.

```
inetnum:        212.160.157.64 - 212.160.157.127
netname:        DP4E-HUTAMINSKA
descr:          Dom Produkcji 4E
descr:          Huta Minska
country:        PL
admin-c:        MN7397-RIPE
tech-c:         BR6273-RIPE
status:         ASSIGNED PA
mnt-by:         AS5617-MNT
changed:        tkielb@cst.tpsa.pl 20010606
source:         RIPE
```

6. Correlations

Incidents.org has several examples of this scan. See in particular, http://www.incidents.org/archives/intrusions/msg00579.html.

7. Evidence of active targeting

The port may be receiving a scan, may be accidental, or may be specifically targeted. Since this is the first access, it is probably not specifically targeted.

8. Severity

Aspect	Value	Reason	Guide	Description
Criticality	4	Web server	5	Firewall, DNS server, core router
			4	E-mail relay/exchange
			2	User UNIX desktop system

			1	MS-DOS 3.11
Lethality	2	Scanning	5	Attacker can gain root across net
			4	Total lockout by denial of service
			4	User password, like a sniffed password
			1	Attack is unlikely to succeed
Countermeasures			5	Modern operating systems, all patches
System	5	?	3	Older operating systems, missing patches
Net	4	?	1	No wrappers/allows fixed unencrypted passwords
Severity	**-3**			(criticality+lethality-system-net)

9. Defensive recommendation:

Block this address at the server with Network ICE Defender.

Continue to monitor the Snort logs for similar accesses from this network.

10. Multiple choice test question

The timing of the above indicates:

1. The scan is done using UDP.
2. There are two retransmissions of the initial attempt to connect.
3. There are out-of-spec flags set.
4. The source address is spoofed.

(Answer: 2)

Assignment 3: "Analyze This" Scenario

Executive Summary

Summary of Alerts

The table below summarizes the information about the alerts processed.

First Day of Alerts	10/27/2001
Last Day of Alerts	10/31/2001
Total Alerts	2,157,301
Scan Detected	1,128,418
Out of Scope	372
Other Alerts	1,028,511

This second table lists the name of each alert and the number of alerts of each type.

Alert	# of Alerts
Scan Detected	1,128,418
UDP SRC and DST outside network	990,494
MISC Large UDP Packet	9,590
Watchlist 000220 IL-ISDNNET-990517	5,249
Tiny Fragments - Possible Hostile Activity	3,191
SMB Name Wildcard	2,890
Incomplete Packet Fragments Discarded	2,319
spp_http_decode: IIS Unicode attack detected	1,977
Possible trojan server activity	1,664
WEB-MISC prefix-get //	1,474
Watchlist 000222 NET-NCFC	1,417
INFO MSN IM Chat data	1,185
MISC source port 53 to <1024	944
ICMP Destination Unreachable (Communication Administratively Prohibited)	861
MISC traceroute	752
CS WEBSERVER - external web traffic	531
ICMP Echo Request Sun Solaris	429
spp_http_decode: CGI Null Byte attack detected	394
Out of Spec	372
INFO Inbound GNUTella Connect accept	340
INFO - ICQ Access	335
TFTP - Internal TCP connection to external tftp server	241
Queso fingerprint	188
INFO Napster Client Data	144
Null scan!	133
ICMP Echo Request Nmap or HPING2	128
ICMP Destination Unreachable (Host Unreachable)	123
connect to 515 from outside	122
INFO Possible IRC Access	121
ICMP Fragment Reassembly Time Exceeded	121

Alert	# of Alerts
High port 65535 tcp - possible Red Worm - traffic	99
TCP SRC and DST outside network	84
INFO napster login	84
INFO Outbound GNUTella Connect accept	78
Port 55850 tcp - Possible myserver activity - ref. 010313-1	73
WEB-MISC 403 Forbidden	67
SUNRPC highport access!	58
External RPC call	57
SCAN Proxy attempt	38
EXPLOIT x86 NOOP	38
NMAP TCP ping!	37
CS WEBSERVER - external ftp traffic	34
ICMP Echo Request L3retriever Ping	28
WEB-MISC Attempt to execute cmd	26
WinGate 1080 Attempt	24
WEB-MISC compaq nsight directory traversal	23
RPC tcp traffic contains bin_sh	21
INFO FTP anonymous FTP	21
ICMP Source Quench	21
ICMP traceroute	20
High port 65535 udp - possible Red Worm - traffic	19
ICMP Echo Request Windows	17
TELNET login incorrect	15
FTP DoS ftpd globbing	15
connect to 515 from inside	15
WEB-MISC http directory traversal	14
ICMP Echo Request CyberKit 2.2 Windows	14
ICMP Destination Unreachable (Protocol Unreachable)	14
INFO napster upload request	10
WEB-CGI redirect access	8
WEB-MISC count.cgi access	7
WEB-IIS _vti_inf access	7
MISC Large ICMP Packet	7
ICMP Echo Request BSDtype	6
BACKDOOR NetMetro Incoming Traffic	6
WEB-IIS view source via translate header	4
WEB-FRONTPAGE _vti_rpc access	4
INFO Inbound GNUTella Connect request	4
SCAN Synscan Portscan ID 19104	3
SCAN FIN	3
ICMP SRC and DST outside network	3
EXPLOIT x86 setuid 0	3
X11 outgoing	2
SNMP public access	2
SMTP relaying denied	2
ICMP Destination Unreachable (Fragmentation Needed and DF bit was set)	2
EXPLOIT x86 setgid 0	2
Attempted Sun RPC high port access	2
X11 xopen	1
WEB-IIS Unauthorized IP Access Attempt	1
WEB-FRONTPAGE fpcount.exe access	1
WEB-CGI webgais access	1
WEB-CGI rsh access	1
WEB-CGI formmail access	1
Virus - Possible pif Worm	1

Alert	# of Alerts
Russia Dynamo - SANS Flash 28-jul-00	1
RFB - Possible WinVNC - 010708-1	1
MISC PCAnywhere Startup	1
INFO - Web Dir listing	1
ICMP Destination Unreachable (Network Unreachable)	1
DNS zone transfer	1
Total	2,157,301

"Top Talkers"

The table below lists the ten IP addresses initiating traffic that causes alerts, not counting scans.

IP Address	# of Alerts as Source
129.105.153.48	879,788
159.134.237.178	49,691
203.109.158.50	29,453
63.250.213.100	14,537
61.134.9.88	7,577
63.250.213.39	5,623
129.105.153.49	4,618
212.179.58.194	4,081
3.0.0.99	3,203

This second table lists the ten most common sources of port scans.

IP Address of Host	Host is the source
MY.NET.160.114	921564
205.188.233.153	34061
205.188.244.57	31395
205.188.233.121	26286
205.188.233.185	25548
205.188.246.121	23295
205.188.244.121	20849
MY.NET.235.142	3790
MY.NET.240.202	3319

The third table examines the port scans in greater detail. Port scans can be reconnaissance for further intrusion. Therefore, understanding the ports being targeted is important. However, a large number of scans does not necessarily mean the most dangerous attack, as is seen below.

Source Address	Destination Port	Number of Scans
MY.NET.160.114	27005	757,293
205.188.233.153	6970	33,473
205.188.244.57	6970	30,602
205.188.233.121	6970	25,890
205.188.233.185	6970	25,224
205.188.246.121	6970	22,643
205.188.244.121	6970	20,627
MY.NET.160.114	8738	11,014
MY.NET.160.114	1025	5,262

The forth table lists the addresses of ten hosts doing the most scanning of reserved ports.

Source Address	Reserved Port Address	# of Scans
194.78.32.252	22	1,129
193.252.36.195	21	764
205.189.240.205	22	544
216.205.117.100	21	378
212.80.33.113	21	266
130.89.30.69	22	245
130.64.100.21	515	142
202.101.103.111	111	122
216.83.156.214	21	115
MY.NET.100.230	53	97

The fifth table identifies hosts that were the sources of both scans and alerts. The table is sorted by the number of alerts.

Source Address	# of Alerts	# of Scans
MY.NET.98.127	1432	945
MY.NET.153.146	300	11
199.183.24.194	198	87
208.178.176.216	135	57
MY.NET.97.168	102	109
213.64.102.177	75	72
MY.NET.98.121	64	9
MY.NET.153.153	57	12
131.211.28.48	56	24
63.198.138.179	47	37
MY.NET.225.14	44	50
24.25.206.180	43	23
24.152.123.165	31	9
MY.NET.235.110	29	68
MY.NET.160.114	25	921564
MY.NET.253.24	24	14
MY.NET.205.226	23	8

Source Address	# of Alerts	# of Scans
MY.NET.97.203	21	15
64.245.51.82	19	15
210.177.137.9	17	17
MY.NET.97.229	15	21
MY.NET.209.82	10	256
MY.NET.153.210	10	397
MY.NET.153.177	9	15
MY.NET.153.159	8	132
MY.NET.226.14	8	25
MY.NET.97.184	8	14
193.137.203.227	8	5
MY.NET.60.38	7	731
MY.NET.209.162	7	28
66.114.106.23	6	3
MY.NET.97.213	6	8
MY.NET.98.132	6	252
MY.NET.221.106	5	46
66.26.169.199	5	2
MY.NET.60.39	5	398
MY.NET.97.222	4	44
MY.NET.98.183	4	73
MY.NET.98.208	4	1570
MY.NET.97.169	4	12
62.4.20.42	4	2
24.180.146.93	4	2
MY.NET.98.228	4	95
MY.NET.242.170	4	37
193.109.122.5	4	4
MY.NET.227.42	3	18
66.50.66.232	3	1
212.123.168.1	3	1
62.59.12.72	3	1
213.109.132.111	3	1
62.59.52.246	3	1
217.84.4.96	3	3
195.68.25.226	3	1
24.165.83.220	2	1
62.42.14.250	2	1
MY.NET.240.2	2	11
64.171.76.200	2	1
212.211.88.19	2	1
193.231.15.152	2	2
217.233.91.168	2	1
193.225.158.187	2	1
MY.NET.213.174	2	60
198.186.202.147	2	2
MY.NET.178.222	2	40

Source Address	# of Alerts	# of Scans
MY.NET.226.114	2	1904
200.231.141.33	2	2
MY.NET.98.151	2	62
24.169.80.72	1	1
63.156.112.248	1	1
142.165.249.23	1	1
212.123.168.77	1	1
MY.NET.97.193	1	10
152.17.88.38	1	1
24.203.36.163	1	1
24.112.49.232	1	1
213.245.112.190	1	1
24.141.136.92	1	1
63.156.28.121	1	1
139.13.209.52	1	3
65.129.53.141	1	1
129.2.210.23	1	1
MY.NET.253.43	1	20
65.129.33.17	1	1
206.128.215.55	1	1
24.202.112.64	1	1
24.102.48.204	1	1
66.50.5.231	1	1
213.227.115.105	1	1
24.27.45.73	1	1
217.165.176.59	1	1
66.50.7.159	1	1
195.162.212.128	1	1
65.129.27.145	1	1
66.50.84.210	1	1
62.252.40.101	1	1
MY.NET.231.130	1	153
24.101.221.91	1	1
217.85.209.172	1	1
MY.NET.98.177	1	11
213.132.152.208	1	1
66.50.25.34	1	1
24.95.223.151	1	1
65.129.48.79	1	1
200.196.50.75	1	1
128.226.116.125	1	1
208.228.171.142	1	2
MY.NET.202.50	1	13
24.234.175.84	1	1
24.120.118.142	1	2
65.129.156.80	1	1
200.163.72.163	1	1

Source Address	# of Alerts	# of Scans
MY.NET.241.86	1	84
24.120.160.11	1	1
217.144.193.199	1	1
193.232.252.34	1	1
213.40.12.135	1	1
MY.NET.150.220	1	3052
24.82.88.162	1	2
24.150.192.198	1	1
66.50.97.216	1	1
65.27.89.114	1	1
139.142.89.207	1	1
24.120.85.206	1	2
24.120.34.178	1	3
24.234.240.77	1	1
212.123.172.167	1	1
217.233.124.189	1	1
24.234.240.6	1	1
139.30.230.124	1	1
128.54.182.86	1	2
209.86.34.249	1	1
MY.NET.226.26	1	2
65.129.35.199	1	1
213.75.100.13	1	1
MY.NET.98.118	1	56
MY.NET.97.236	1	173
MY.NET.230.194	1	15
213.109.194.132	1	1
24.176.51.183	1	1
24.248.213.78	1	1
MY.NET.221.26	1	31
63.157.5.110	1	1
213.109.200.29	1	2
24.161.113.244	1	1
24.202.126.193	1	1
137.143.133.53	1	1
65.128.53.32	1	1
213.187.164.14	1	1
65.129.152.207	1	1
213.51.63.111	1	1
63.156.112.20	1	1
212.30.66.160	1	1
24.161.64.130	1	1
24.234.190.23	1	2
24.77.183.79	1	1
24.60.117.169	1	1
24.120.161.202	1	2
210.215.10.226	1	1

Source Address	# of Alerts	# of Scans
129.241.129.244	1	1
217.113.225.25	1	1
65.129.32.17	1	1
24.120.37.240	1	1
66.50.73.187	1	1
65.129.152.138	1	1
134.100.204.148	1	2
63.159.188.117	1	1
65.128.52.6	1	1
65.129.90.206	1	1
66.27.233.156	1	1
24.202.121.174	1	1
24.234.120.184	1	1
212.244.47.137	1	1
MY.NET.98.143	1	11
MY.NET.241.130	1	65
65.8.226.165	1	1
66.50.11.86	1	1
66.50.77.199	1	1
MY.NET.209.254	1	515
63.156.28.114	1	1
MY.NET.228.194	1	49
24.114.238.86	1	1
63.156.239.23	1	1
65.129.39.151	1	1
66.50.71.127	1	1
195.194.178.151	1	1

Analysis of Selected Incidents

LPR Buffer Overrun Vulnerability

Description of Attack. The table below shows attacks from two hosts attempting to connect to Port 515, the LPR print spooler port. There are two vulnerabilities that are widely used:

- The vulnerability may allow a user to gain root privileges due to a buffer overflow. (CERT Advisory CA-1997-19)
- Format string vulnerability in use_syslog() function in LPRng 3.6.24 allows remote attackers to execute arbitrary commands. (CERT Advisory CA-2000-22)

There is a good chance that some of the servers have been compromised.

Date	Time	Source Host	Source Port	Destination Host	Destination Port
27-Oct	6:07:18	213.64.102.177	3657	MY.NET.132.181	515

Date	Time	Source Host	Source Port	Destination Host	Destination Port
27-Oct	6:07:19	213.64.102.177	4180	MY.NET.132.203	515
27-Oct	6:07:19	213.64.102.177	4200	MY.NET.132.223	515
27-Oct	6:07:19	213.64.102.177	4217	MY.NET.132.240	515
27-Oct	6:07:19	213.64.102.177	4246	MY.NET.133.14	515
27-Oct	6:07:19	213.64.102.177	4272	MY.NET.133.39	515
27-Oct	6:07:20	213.64.102.177	4311	MY.NET.133.77	515
27-Oct	6:07:20	213.64.102.177	4315	MY.NET.133.81	515
27-Oct	6:07:20	213.64.102.177	4329	MY.NET.133.95	515
27-Oct	6:07:20	213.64.102.177	4341	MY.NET.133.107	515
27-Oct	6:07:21	213.64.102.177	4379	MY.NET.133.144	515
27-Oct	6:07:21	213.64.102.177	4397	MY.NET.133.162	515
27-Oct	6:07:21	213.64.102.177	4403	MY.NET.133.168	515
27-Oct	6:07:21	213.64.102.177	3480	MY.NET.132.4	515
27-Oct	6:07:22	213.64.102.177	4469	MY.NET.133.234	515
27-Oct	6:07:22	213.64.102.177	4504	MY.NET.134.14	515
27-Oct	6:07:22	213.64.102.177	4549	MY.NET.134.59	515
27-Oct	6:07:22	213.64.102.177	4635	MY.NET.134.145	515
27-Oct	6:07:22	213.64.102.177	4642	MY.NET.134.152	515
27-Oct	6:07:22	213.64.102.177	4645	MY.NET.134.155	515
27-Oct	6:07:22	213.64.102.177	4187	MY.NET.132.210	515
27-Oct	6:07:22	213.64.102.177	4192	MY.NET.132.215	515
27-Oct	6:07:22	213.64.102.177	4247	MY.NET.133.15	515
27-Oct	6:07:22	213.64.102.177	4257	MY.NET.133.25	515
27-Oct	6:07:22	213.64.102.177	4261	MY.NET.133.29	515
27-Oct	6:07:22	213.64.102.177	4275	MY.NET.133.42	515
27-Oct	6:07:24	213.64.102.177	4794	MY.NET.135.49	515
27-Oct	6:07:24	213.64.102.177	4375	MY.NET.133.140	515
27-Oct	6:07:24	213.64.102.177	4401	MY.NET.133.166	515
27-Oct	6:07:24	213.64.102.177	4819	MY.NET.135.74	515
27-Oct	6:07:24	213.64.102.177	4823	MY.NET.135.78	515
27-Oct	6:07:24	213.64.102.177	4416	MY.NET.133.181	515
27-Oct	6:07:24	213.64.102.177	4421	MY.NET.133.186	515
27-Oct	6:07:25	213.64.102.177	4901	MY.NET.135.156	515
27-Oct	6:07:25	213.64.102.177	4905	MY.NET.135.160	515
27-Oct	6:07:25	213.64.102.177	4447	MY.NET.133.212	515
27-Oct	6:07:25	213.64.102.177	4450	MY.NET.133.215	515
27-Oct	6:07:25	213.64.102.177	4451	MY.NET.133.216	515
27-Oct	6:07:25	213.64.102.177	4457	MY.NET.133.222	515
27-Oct	6:07:25	213.64.102.177	4486	MY.NET.133.251	515
27-Oct	6:07:25	213.64.102.177	4507	MY.NET.134.17	515
27-Oct	6:07:25	213.64.102.177	4540	MY.NET.134.50	515
27-Oct	6:07:25	213.64.102.177	4562	MY.NET.134.72	515
27-Oct	6:07:25	213.64.102.177	4600	MY.NET.134.110	515
27-Oct	6:07:26	213.64.102.177	4771	MY.NET.135.26	515
27-Oct	6:07:26	213.64.102.177	4777	MY.NET.135.32	515
27-Oct	6:07:27	213.64.102.177	4795	MY.NET.135.50	515

Date	Time	Source Host	Source Port	Destination Host	Destination Port
27-Oct	6:07:27	213.64.102.177	4806	MY.NET.135.61	515
27-Oct	6:07:27	213.64.102.177	4814	MY.NET.135.69	515
27-Oct	6:07:28	213.64.102.177	4871	MY.NET.135.126	515
27-Oct	6:07:28	213.64.102.177	1818	MY.NET.137.34	515
27-Oct	6:07:28	213.64.102.177	1826	MY.NET.137.42	515
27-Oct	6:07:28	213.64.102.177	1848	MY.NET.137.64	515
27-Oct	6:07:28	213.64.102.177	1851	MY.NET.137.67	515
27-Oct	6:07:28	213.64.102.177	1855	MY.NET.137.71	515
27-Oct	6:07:28	213.64.102.177	1879	MY.NET.137.95	515
27-Oct	6:07:28	213.64.102.177	1882	MY.NET.137.98	515
27-Oct	6:07:29	213.64.102.177	1522	MY.NET.135.248	515
27-Oct	6:07:31	213.64.102.177	1817	MY.NET.137.33	515
27-Oct	6:07:31	213.64.102.177	1829	MY.NET.137.45	515
27-Oct	6:07:31	213.64.102.177	1837	MY.NET.137.53	515
27-Oct	6:07:31	213.64.102.177	1858	MY.NET.137.74	515
27-Oct	6:07:31	213.64.102.177	1862	MY.NET.137.78	515
27-Oct	6:09:18	213.64.102.177	4159	MY.NET.190.2	515
27-Oct	6:09:18	213.64.102.177	4161	MY.NET.190.4	515
27-Oct	6:09:20	213.64.102.177	4791	MY.NET.190.85	515
27-Oct	6:09:23	213.64.102.177	4807	MY.NET.190.101	515
27-Oct	6:09:23	213.64.102.177	4823	MY.NET.190.117	515
27-Oct	6:09:23	213.64.102.177	4846	MY.NET.190.140	515
27-Oct	6:09:23	213.64.102.177	4847	MY.NET.190.141	515
27-Oct	6:09:23	213.64.102.177	4876	MY.NET.190.170	515
27-Oct	6:09:23	213.64.102.177	4887	MY.NET.190.181	515
27-Oct	6:09:24	213.64.102.177	4892	MY.NET.190.186	515
27-Oct	6:09:24	213.64.102.177	4896	MY.NET.190.190	515
27-Oct	6:09:24	213.64.102.177	4904	MY.NET.190.198	515
29-Oct	3:59:13	63.198.138.179	1522	MY.NET.132.21	515
29-Oct	3:59:13	63.198.138.179	1537	MY.NET.132.36	515
29-Oct	3:59:13	63.198.138.179	1543	MY.NET.132.42	515
29-Oct	3:59:13	63.198.138.179	1580	MY.NET.132.79	515
29-Oct	3:59:15	63.198.138.179	2112	MY.NET.132.219	515
29-Oct	3:59:15	63.198.138.179	2140	MY.NET.132.247	515
29-Oct	3:59:15	63.198.138.179	2300	MY.NET.133.123	515
29-Oct	3:59:16	63.198.138.179	1534	MY.NET.132.33	515
29-Oct	3:59:16	63.198.138.179	1537	MY.NET.132.36	515
29-Oct	3:59:16	63.198.138.179	1555	MY.NET.132.54	515
29-Oct	3:59:16	63.198.138.179	1556	MY.NET.132.55	515
29-Oct	3:59:16	63.198.138.179	2484	MY.NET.133.227	515
29-Oct	3:59:18	63.198.138.179	2619	MY.NET.134.90	515
29-Oct	3:59:18	63.198.138.179	2287	MY.NET.133.110	515
29-Oct	3:59:18	63.198.138.179	2945	MY.NET.134.224	515
29-Oct	3:59:18	63.198.138.179	2950	MY.NET.134.229	515
29-Oct	3:59:19	63.198.138.179	2527	MY.NET.134.15	515
29-Oct	3:59:20	63.198.138.179	2552	MY.NET.134.40	515

Date	Time	Source Host	Source Port	Destination Host	Destination Port
29-Oct	3:59:20	63.198.138.179	2555	MY.NET.134.43	515
29-Oct	3:59:20	63.198.138.179	2556	MY.NET.134.44	515
29-Oct	3:59:20	63.198.138.179	3423	MY.NET.135.153	515
29-Oct	3:59:20	63.198.138.179	3426	MY.NET.135.156	515
29-Oct	3:59:20	63.198.138.179	3438	MY.NET.135.168	515
29-Oct	3:59:20	63.198.138.179	3450	MY.NET.135.180	515
29-Oct	3:59:20	63.198.138.179	3459	MY.NET.135.189	515
29-Oct	3:59:21	63.198.138.179	3523	MY.NET.135.253	515
29-Oct	3:59:21	63.198.138.179	2912	MY.NET.134.191	515
29-Oct	3:59:21	63.198.138.179	2951	MY.NET.134.230	515
29-Oct	3:59:21	63.198.138.179	2963	MY.NET.134.242	515
29-Oct	3:59:23	63.198.138.179	3347	MY.NET.135.146	515
29-Oct	3:59:23	63.198.138.179	1302	MY.NET.137.87	515
29-Oct	3:59:23	63.198.138.179	3421	MY.NET.135.151	515
29-Oct	3:59:23	63.198.138.179	3447	MY.NET.135.177	515
29-Oct	3:59:24	63.198.138.179	1396	MY.NET.137.141	515
29-Oct	3:59:24	63.198.138.179	1402	MY.NET.137.147	515
29-Oct	3:59:24	63.198.138.179	1449	MY.NET.137.194	515
29-Oct	3:59:24	63.198.138.179	1470	MY.NET.137.215	515
29-Oct	3:59:27	63.198.138.179	1405	MY.NET.137.150	515
29-Oct	3:59:27	63.198.138.179	1415	MY.NET.137.160	515
29-Oct	3:59:27	63.198.138.179	1427	MY.NET.137.172	515
29-Oct	3:59:28	63.198.138.179	1575	MY.NET.137.251	515
29-Oct	4:01:13	63.198.138.179	1967	MY.NET.190.189	515
29-Oct	4:01:14	63.198.138.179	1741	MY.NET.190.77	515
29-Oct	4:01:16	63.198.138.179	1980	MY.NET.190.202	515
29-Oct	4:01:16	63.198.138.179	1983	MY.NET.190.205	515
29-Oct	4:01:22	63.198.138.179	1951	MY.NET.190.173	515
29-Oct	4:01:22	63.198.138.179	1979	MY.NET.190.201	515

Source Host Identification.

Www.ripe.net reports the following information on 213.64.102.177 that is owned by an Internet Service Provider.

```
inetnum:        213.64.0.0 - 213.64.255.255
netname:        TELIANET
descr:          Telia Network services
descr:          ISP
descr:          ---------------------------
descr:          Intrusion and abuse reports
descr:          should be sent to
descr:          abuse@telia.com
descr:          ---------------------------
country:        SE
admin-c:        TR889-RIPE
```

```
tech-c:         TR889-RIPE
status:         ASSIGNED PA
notify:         mntripe@telia.net
notify:         backbone@telia.net
mnt-by:         TELIANET-LIR
changed:        amar@telia.net 20010404
source:         RIPE
```

Www.samspade.org reports the following information on 63.198.138.179.

```
Trying whois -h whois.arin.net 63.198.138.179
```
Pac Bell Internet Services (NETBLK-PBI-NET-7) PBI-NET-7
63.192.0.0 - 63.207.255.255
Karsten Leon (NETBLK-SBCIS64785)SBCIS64785 63.198.138.176 - 63.198.138.183

Correlations. CERT®Advisory CA-1997-19, "lpr Buffer Overrun Vulnerability," URL: http://www.cert.org/advisories/CA-1997-19.html.

CERT® Advisory CA-2000-22, "Input Validation Problems in LPRng," URL: http://www.cert.org/advisories/CA-2000-22.html

Identification of External Hosts on 205.188.0.0 Network

Description of the Attack. There are six addresses that show up in the top ten scanners of a particular destination port. Www.samspad.org identifies these hosts as part of America Online. Northcutt [NORTHCUTT2001, p. 326] discusses this detect. While activity on UDP Port 6970 could indicate the GateCrasher trojan, the port is also used by RealAudio. Given the source is America Online, RealAudio seems likely. This is probably a false positive.

Source Host Identification. The following is from the Whois at www.samspad.org

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

Suspected Queso Scan (False Positive)

Description of Attack. A host at 199.183.24.194 was detected performing a Queso scan. Cole describes Queso as the original program that performs fingerprinting of operating system, i.e. it can determine around 100 different devices [COLE2002]. This host is targeting Port 25, the SNMP reserved port.

The table below shows a set of scans from October 27 through October 30. The table is constructed from the alert log files and processed as described below under analysis.

Date	Time	Source Port	Destination Address	Destination Port
27-Oct	2:33:58	51896	MY.NET.253.42	25
27-Oct	4:34:11	53952	MY.NET.253.43	25
27-Oct	7:56:47	43003	MY.NET.253.43	25
27-Oct	10:06:31	35374	MY.NET.253.42	25
27-Oct	10:14:33	39803	MY.NET.253.43	25
27-Oct	13:32:40	43040	MY.NET.6.34	25
27-Oct	16:39:58	56074	MY.NET.100.217	25
27-Oct	21:04:31	55648	MY.NET.6.47	25
28-Oct	13:17:06	37777	MY.NET.253.42	25
28-Oct	13:54:33	52191	MY.NET.253.41	25
28-Oct	14:24:55	34973	MY.NET.253.43	25
28-Oct	15:33:49	45333	MY.NET.253.42	25
28-Oct	17:54:28	49643	MY.NET.253.41	25
28-Oct	18:09:09	53525	MY.NET.253.42	25
28-Oct	19:49:40	55092	MY.NET.100.217	25
28-Oct	21:13:12	51154	MY.NET.253.41	25
29-Oct	0:28:42	36405	MY.NET.6.35	25
29-Oct	6:05:26	39124	MY.NET.100.217	25
29-Oct	6:05:37	39468	MY.NET.253.42	25
29-Oct	6:43:17	53214	MY.NET.6.34	25
29-Oct	8:43:40	47715	MY.NET.253.43	25
29-Oct	11:45:09	56053	MY.NET.100.217	25
29-Oct	12:20:46	48887	MY.NET.6.47	25
29-Oct	12:59:08	45614	MY.NET.100.217	25
29-Oct	13:03:09	48428	MY.NET.100.217	25
29-Oct	18:11:16	59548	MY.NET.100.217	25
29-Oct	18:22:56	44232	MY.NET.6.47	25
29-Oct	22:20:03	57326	MY.NET.100.217	25
30-Oct	5:16:20	51772	MY.NET.6.35	25
30-Oct	6:33:50	52336	MY.NET.253.43	25
30-Oct	6:37:35	55295	MY.NET.6.35	25
30-Oct	6:55:49	39294	MY.NET.100.217	25
30-Oct	7:46:31	47751	MY.NET.100.217	25
30-Oct	10:41:43	45697	MY.NET.253.41	25

Date	Time	Source Port	Destination Address	Destination Port
30-Oct	10:43:11	47568	MY.NET.253.41	25
30-Oct	11:09:03	37608	MY.NET.6.34	25
30-Oct	12:40:20	56989	MY.NET.100.217	25
30-Oct	15:53:28	46608	MY.NET.6.35	25
30-Oct	19:16:51	41272	MY.NET.100.217	25
30-Oct	19:36:53	60978	MY.NET.100.217	25
30-Oct	19:44:17	36116	MY.NET.100.217	25
30-Oct	21:15:17	37748	MY.NET.253.43	25
30-Oct	23:09:21	44537	MY.NET.253.43	25
31-Oct	1:40:40	40711	MY.NET.6.47	25
31-Oct	4:26:46	47542	MY.NET.100.217	25
31-Oct	4:34:03	53435	MY.NET.6.35	25
31-Oct	5:18:17	38261	MY.NET.6.34	25
31-Oct	5:42:33	44201	MY.NET.253.42	25
31-Oct	5:55:31	56633	MY.NET.100.217	25
31-Oct	7:42:45	50864	MY.NET.253.43	25
31-Oct	8:09:57	60673	MY.NET.253.43	25
31-Oct	8:44:10	50942	MY.NET.6.35	25
31-Oct	9:15:50	48870	MY.NET.6.34	25
31-Oct	11:23:33	59188	MY.NET.253.41	25
31-Oct	12:09:32	51696	MY.NET.253.42	25
31-Oct	15:11:47	58253	MY.NET.6.34	25
31-Oct	15:53:02	41297	MY.NET.253.43	25
31-Oct	16:06:05	48904	MY.NET.6.34	25
31-Oct	16:06:07	48987	MY.NET.100.217	25
31-Oct	16:07:33	50749	MY.NET.253.41	25
31-Oct	17:20:52	43553	MY.NET.253.42	25
31-Oct	17:24:01	46397	MY.NET.253.41	25
31-Oct	17:38:01	57644	MY.NET.253.42	25
31-Oct	17:40:21	60435	MY.NET.253.43	25
31-Oct	18:44:48	34940	MY.NET.6.35	25
31-Oct	19:03:38	52659	MY.NET.253.42	25
31-Oct	19:26:34	40688	MY.NET.253.43	25
31-Oct	19:30:40	45487	MY.NET.6.47	25
31-Oct	19:45:38	34783	MY.NET.6.34	25
31-Oct	19:45:41	34884	MY.NET.100.217	25
31-Oct	19:54:01	42976	MY.NET.100.217	25
31-Oct	20:03:24	53421	MY.NET.253.42	25
31-Oct	20:14:11	56823	MY.NET.100.217	25
31-Oct	20:15:37	58028	MY.NET.6.35	25
31-Oct	20:15:44	58129	MY.NET.100.217	25
31-Oct	20:45:57	40158	MY.NET.6.35	25
31-Oct	21:00:04	45495	MY.NET.100.217	25
31-Oct	21:35:38	56443	MY.NET.6.47	25
31-Oct	21:59:46	36731	MY.NET.253.43	25
31-Oct	22:03:58	40602	MY.NET.6.34	25

Source Host Identificatioin. Www.samspade.org identifies this address as belonging to Red Hat Software. The following is copied from Samspade:

Red Hat Software (NET-REDHAT) REDHAT 199.183.24.0 - 199.183.24.255

Correlation. However, the following post from Ookhoi suggests that this may be a false positive caused by a legitimate mail server at Red Hat.

> Our ISP blocked our webserver for a while because (a) Company mailed that they were portscanned by us according to their hereby included snort log.
>
> Now we don't portscan of course, and can't find proof of a break in (maybe somebody else wanted to do us a favor and do a portscan for us ;-) And besides, in the snort log only a scan at port 25 is mentioned at two of their servers, which happen to be both mail gateways.
>
> According to the database on the Web server, someone from Company subscribed to a forum on our site early this month. The forum sends out mails every morning, and thus also to the two mail gateways.
>
> According to our mail logs, our mailserver delivered mails to the mail gateways at the days mentioned in the snort log, but not at the same time as the scans.
>
> Our mailserver is postfix, and we use linux kernel 2.4 with ecn enabled. Can it be that postfix tried to deliver mail and that snort somehow found the tcp connection to be mangled in some way? I read that a Queso Fingerprint works by changing some things in the tcp packets. [OOKHOI2001]

A telnet to this address did indeed reveal that it is an e-mail server.

Watchlist Hosts

There were a number of alerts for networks on the watch list. Some of these alerts were for accesses to Web or e-mail servers. Others were not obviously accessing public servers.

One address is 159.226.41.166 which is receiving telnet traffic from MY.NET. Www.samspade.org provides the following information on this address:

The Computer Network Center Chinese Academy of Sciences (NET-NCFC)

P.O. Box 2704-10,

Institute of Computing Technology Chinese Academy of Sciences

Beijing 100080, China

CN

Netname: NCFC

Netblock: 159.226.0.0 - 159.226.255.255

Coordinator:

Qian, Haulin (QH3-ARIN) hlqian@NS.CNC.AC.CN

+86 1 2569960

Another address that shows up on a Watchlist is 212.179.83.61. Www.ripe.net provides the following information on this address.

```
inetnum:        212.179.80.0 - 212.179.94.255
netname:        L2TP-PROJECT
descr:          2st-pool-Dailup-L2TP-client.
country:        IL
admin-c:        NP469-RIPE
tech-c:         NP469-RIPE
status:         ASSIGNED PA
notify:         hostmaster@isdn.net.il
mnt-by:         RIPE-NCC-NONE-MNT
changed:        hostmaster@isdn.net.il 20000402
source:         RIPE
```

Analysis of Internal Machines

SubSeven Trojan. The list below extracted from the alert files showing the internal address receiving traffic on Port 27374. The table demonstrates that there is a lot of activity on this port, indicating the distinct possibility of extensive SubSeven, version 2, infection.

Correlation. See http://subseven.slak.org/ for information on this trojan.

Source Address	Source Port	Destination Address	Destination Port
24.25.206.180	3989	MY.NET.132.136	27374
24.25.206.180	3753	MY.NET.132.18	27374
24.25.206.180	4121	MY.NET.132.199	27374
24.25.206.180	3757	MY.NET.132.20	27374
24.25.206.180	4131	MY.NET.132.204	27374
24.25.206.180	4133	MY.NET.132.205	27374
24.25.206.180	4159	MY.NET.132.218	27374
24.25.206.180	4165	MY.NET.132.221	27374
24.25.206.180	4177	MY.NET.132.227	27374
24.25.206.180	3905	MY.NET.132.94	27374
24.25.206.180	3909	MY.NET.132.96	27374
24.25.206.180	4487	MY.NET.133.126	27374
24.25.206.180	4489	MY.NET.133.127	27374

Source Address	Source Port	Destination Address	Destination Port
24.25.206.180	4549	MY.NET.133.157	27374
24.25.206.180	4625	MY.NET.133.195	27374
24.25.206.180	4279	MY.NET.133.22	27374
24.25.206.180	4700	MY.NET.133.232	27374
24.25.206.180	4293	MY.NET.133.29	27374
24.25.206.180	4295	MY.NET.133.30	27374
24.25.206.180	4343	MY.NET.133.54	27374
24.25.206.180	4748	MY.NET.134.0	27374
24.25.206.180	4748	MY.NET.134.0	27374
24.25.206.180	4952	MY.NET.134.102	27374
24.25.206.180	4960	MY.NET.134.106	27374
24.25.206.180	4982	MY.NET.134.117	27374
24.25.206.180	4990	MY.NET.134.121	27374
24.25.206.180	1028	MY.NET.134.128	27374
24.25.206.180	1036	MY.NET.134.132	27374
24.25.206.180	1046	MY.NET.134.137	27374
24.25.206.180	4776	MY.NET.134.14	27374
24.25.206.180	1054	MY.NET.134.141	27374
24.25.206.180	1056	MY.NET.134.142	27374
24.25.206.180	1064	MY.NET.134.146	27374
24.25.206.180	1068	MY.NET.134.148	27374
24.25.206.180	1070	MY.NET.134.149	27374
24.25.206.180	1072	MY.NET.134.150	27374
24.25.206.180	4780	MY.NET.134.16	27374
24.25.206.180	4784	MY.NET.134.18	27374
24.25.206.180	4790	MY.NET.134.21	27374
24.25.206.180	4796	MY.NET.134.24	27374
24.25.206.180	4872	MY.NET.134.62	27374
24.25.206.180	4762	MY.NET.134.7	27374
24.25.206.180	4946	MY.NET.134.99	27374
64.91.17.57	1598	MY.NET.137.136	27374
64.91.17.57	1678	MY.NET.137.176	27374
64.91.17.57	1684	MY.NET.137.179	27374
64.91.17.57	1696	MY.NET.137.185	27374
64.91.17.57	1704	MY.NET.137.189	27374
64.91.17.57	1364	MY.NET.137.19	27374
64.91.17.57	1708	MY.NET.137.191	27374
64.91.17.57	1716	MY.NET.137.195	27374
64.91.17.57	1804	MY.NET.137.239	27374
64.91.17.57	1808	MY.NET.137.241	27374
64.91.17.57	1830	MY.NET.137.252	27374
64.91.17.57	1408	MY.NET.137.41	27374
64.91.17.57	1444	MY.NET.137.59	27374
64.91.17.57	1456	MY.NET.137.65	27374
64.91.17.57	1462	MY.NET.137.68	27374
64.91.17.57	1464	MY.NET.137.69	27374

Source Address	Source Port	Destination Address	Destination Port
64.91.17.57	1340	MY.NET.137.7	27374
64.91.17.57	1466	MY.NET.137.70	27374
24.128.109.1	2502	MY.NET.190.110	27374
24.128.109.1	2502	MY.NET.190.110	27374
24.128.109.1	2598	MY.NET.190.158	27374
24.128.109.1	2632	MY.NET.190.175	27374
24.128.109.1	2632	MY.NET.190.175	27374
24.128.109.1	2674	MY.NET.190.196	27374
24.128.109.1	2726	MY.NET.190.222	27374
24.128.109.1	2330	MY.NET.190.24	27374
24.128.109.1	2450	MY.NET.190.84	27374
24.128.109.1	2470	MY.NET.190.94	27374

Defensive Mechanisms

There are several defensive mechanisms that can be put in place. Because this is a university network, it cannot be as restrictive as a commercial or government network can be. However, there are several things that should be done.

- Deploy a more restrictive firewall policy.

For example, many reserved ports can be scanned by external systems, as shown in the list below.

Outside Address	Reserved Ports Scanned
64.242.192.197	1009
64.242.192.197	868
64.242.192.197	859
64.242.192.197	857
64.242.192.197	719
195.121.32.44	522
63.198.138.179	515
213.64.102.177	515
130.64.100.21	515
64.242.192.197	484
64.242.192.197	435
64.242.192.197	339
64.242.192.197	322
64.242.192.197	219
217.226.151.244	191
24.120.161.202	138
24.234.211.243	137
24.234.195.21	137
24.234.118.118	137
64.242.192.197	124
66.114.106.23	113

198.186.202.147	113
66.12.11.142	111
64.245.51.82	111
210.177.137.9	111
202.101.103.111	111
24.234.239.186	67
24.234.175.84	67
24.234.120.184	67
24.120.34.178	67
24.234.252.137	53
24.234.244.254	53
24.234.240.77	53
24.234.190.23	53
24.234.170.45	53
24.234.118.118	53
24.120.85.157	53
24.120.34.53	53
24.120.29.61	53
24.120.27.55	53
24.120.25.196	53
24.120.161.202	53
24.120.154.74	53
24.120.118.159	53
24.120.118.145	53
213.75.47.7	53
64.242.192.197	47
24.152.123.165	34
66.114.106.23	25
199.183.24.194	25
198.186.202.147	25
131.211.28.48	25
64.171.76.200	23
212.171.59.64	23
205.189.240.205	22
194.78.32.252	22
193.53.23.111	22
130.89.30.69	22
216.83.156.214	21
216.205.117.100	21
213.38.171.2	21
212.80.33.113	21
193.252.49.166	21
193.252.36.195	21
24.23.66.93	20

Many of these can be blocked at the firewall. For example, port 515 (print spooler) and NetBios ports like port 137 should be blocked.

In addition, traffic from networks or sites on the Watch List should be blocked.

- Be sure all current patches are applied to the operating system.
- Tuning of detection rules to eliminate some false positive alerts on accepted traffic such as the RealAudio traffic.

Analysis Process

The data were analyzed using a series of conversions starting with the original logs in HTML format and ending in a series of comma separated values (CSV) files. The steps are as follows:

- Data are manually extracted from the HTML files to text files.
- The alert data are processed using a perl language script to extract the date, time, alert, source address, source port, destination address, and destination port to a comma separated file. This allows easier manipulation of the data in both perl and Excel.
- A similar processing is done with the scanning and out-of-spec files.
- A perl script scans all the alert and OOS CSV files to produce a tally of alerts.
- A perl script scans all the scan CSV files to produce a tally of scans.
- A perl script scans all the alert and OOS CSV files to produce to top ten talkers by counting the number of alerts for each source address. A separate but similar processing occurs for scans.
- A perl script is then used to compute the number of scans by source host and destination port.
- A perl script is used to compute the number of alerts and scans for all source addresses that have scans and alerts. This is used to detect targeted scans that may be an indicator of a host that will later conduct further attacks.
- A perl script or UNIX commands are used to identify attacks for a specific host.

The appendix to this paper lists the scripts.

The analysis process then focused on looking at source hosts initiating the scans that also initiate the most alerts. It also focused on scans that affected reserved ports. The Internet search of neohapsis.com, www.incidents.org, www.whitehats.com were used to find correlations and information on alerts. The Snort rule set also provided indications and references.

References

[ALLEN2000] Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., and Stoner, E. "State of the Practice of Intrusion Detection Technologies. CMU/SEI-99-TR-028,CMU/SEI. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA., 2001 URL: http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html

[BRINEY1999] Briney, Andy. *Parker's Plan*. Norwood, MA: Information Security. http://www.infosecuritymag.com/articles/1999/parker.shtml

[CERT2001] CERT, "Code Red Worm Exploiting Buffer Overflow In IIS Indexing Service DLL," CERT® Advisory CA-2001-19,Original release date: July 19, 2001, Last revised: August 23, 2001, URL: http://www.cert.org/advisories/CA-2001-19.html

[COLE2002] Cole, Eric, *Hackers Beware* (Indianapolis, IN: New Riders, 2002)

[DENNING1987] Denning, Dorthy, "An Intrusion-Detection Model ," IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, Feb. 1987, pp. 222-232.

[DEBAR2000] Debar, Herve, "What is knowledge-based intrusion detection?", 2000, http://www.sans.org/newlook/resources/IDFAQ/knowledge_based.htm

[DEBAR2000a] Debar, Herve, "What is behavior-based intrusion detection?", 2000, URL: http://www.sans.org/newlook/resources/IDFAQ/behavior_based.htm

[DILDOG] Dildog, cDc Ninja Strike Force, 9-dan of the Architecture, Sensei of the Undocumented Opcode, "The Tao of Windows Buffer Overflow," Date Unknown, URL: http://www.cultdeadcow.com/cDc_files/cDc-351/

[EEYE2001] eEye, ".ida "Code Red" Worm," Jul. 17, 2001. URL: http://www.eeye.com/html/Research/Advisories/AL20010717.html

[INSECURE2000] Insecure, "Nmap network security scanner man page" , 2000. URL: http://www.insecure.org/nmap/nmap_manpage.html

[IRWIN2001] Irwin, Vicki, "Response to Intrusion Report", Aug. 15, 2001, URL: http://www.incidents.org/archives/intrusions/msg01447.html

[JONES2001] Jones, David, "Incident Report, " Aug. 15 2001, http://www.incidents.org/archives/intrusions/msg01444.html

[LIPPMANN2000] Lippmann, R. P., Hanes, J. W., Fried, D. J., Korba, J. and Das, K., *Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation,* Third International Workshop on Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France 2000

[LOSCOCCO1998] Loscocco, Peter, Smalley, Stephen, Muckelbauer, Parick, Taylor, Ruth, Turner, Jeff, Farrel, John, "The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments," URL: http://www.jya.com/paperF1.htm

[MCHUGH2000] McHugh, John, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory," ACM Transactions on Information and System Security (TISSEC) Volume 3 , Issue 4 (November 2000)

[MICROSOFT2001] Microsoft, "Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise" Microsoft Security Bulletin MS01-033, Originally posted: June 18, 2001

[NETWORKICE1999] Network ICE, "What is the format of "attack-list.csv"?, Version: 1.8.5.5 Fixed: Modified: Aug. 21, 1999, http://www.networkice.com/Advice/Support/KB/q000018/

[OOKHOI2001] Ookhoi, "Possible Finger Print attempt?" Mar 13, 2001 , URL: http://archives.neohapsis.com/archives/postfix/2001-03/0583.html

[PTACEK1998] Ptacek, Thomas H., Newsham, Timothy N., "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," Secure Networks, Inc., January, 1998. URL: http://www.robertgraham.com/mirror/Ptacek-Newsham-Evasion-98.html

[NORTHCUTT2001] Northcutt, Stephen, et. al, *Intrusion Signatures and Analysis* (Indianapolis, IN: New Riders, 2001)

[SNORT2001] Snort.org, "Snort: the Open Source Intrusion Detection System," URL: http://www.snort.org

[SUNDARAM] Sundaram, Aurobindo, "An Introduction to Intrusion Detection," 1996, URL: http://www.acm.org/crossroads/xrds2-4/intrus.html

[SYMANTEC2001] Symantec, "CodeRed Worm," Sep. 7, 2001, URL: http://www.symantec.com/avcenter/venc/data/codered.worm.html

[SYMANTEC2001a] Symantec, W32.Nimda.A@mm, Oct. 17, 2001, URL: http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html

Appendix A: Analysis Scripts

MAKEFILE to Control Process

```
#
# Makefile to perform analysis of University logs
#


# macros

CNVCSV=perl cvtcmm.pl
CSVFILES = alert_011027.csv alert_011028.csv alert_011029.csv alert_011030.csv alert_011031.csv
TALLY=perl countalert.pl
CNVSCN=perl cnvscan.pl
SCNFILES=scans_011027.tsn scans_011028.tsn scans_011029.tsn scans_011030.tsn scans_011031.tsn
CNTADDR=perl countaddr.pl
SCNOOS=perl countoos.pl
OOSFILES=oos_011027.oos oos_011028.oos oos_011029.oos oos_011030.oos oos_011031.oos
SRC2PORT=perl src2port.pl


# Suffix rule

.SUFFIXES : .txt .csv .scn .tsn .oos .htm

.txt.csv:
        cat $< | $(CNVCSV) > $@
.scn.tsn:
        cat $< | $(CNVSCN) > $@
.htm.oos:
        cat $< | $(SCNOOS) > $@

# generate analysis

analysis : tally.csv scantally.csv toptenalrt.csv toptenscan.csv oostally.csv src2port.csv

#create CSV files for analysis

tally.csv     : $(CSVFILES)
        cat $(CSVFILES)   | $(TALLY) > tally.csv

scantally.csv : $(SCNFILES)
        cat $(SCNFILES)  | $(TALLY) > scantally.csv

oostally.csv  : $(OOSFILES)
        cat $(OOSFILES)  | $(TALLY) > oostally.csv

toptenalrt.csv : $(CSVFILES) $(OOSFILES)
        cat $(CSVFILES) $(OOSFILES) | $(CNTADDR) > toptenalrt.csv

toptenscan.csv : $(SCNFILES)
        cat $(SCNFILES) | $(CNTADDR) > toptenscan.csv


src2port.csv  :  $(SCNFILES)
        cat $(SCNFILES) | $(SRC2PORT) > src2port.csv
```

Perl Script to Convert Alert Logs to Comma Separated Values Files

```
#
# Name: cvtcmm.pl
# Description:  convert a snort alert file into a comma separated file
# Date: 14 Nov 2001
# Arguments:  (none)
# Notes:
#    The program takes input from STDIN and puts it to STDOUT
#

$qt = '"';
while ($line = <STDIN>)
{
   $rest    = '';
   $date    = '';
   $time    = '';
   $alert   = '';
   $source  = '';
   $dest    = '';
   $srcaddr = '';
   $srcport = '';
   $dstaddr = '';
   $dstport = '';
   # convert SGML entity &gt; to >
   $line =~ s/&gt;/>/g;

   # leave out spp_portscan lines
   if ($line =~ /spp_portscan/)
   {
      # do nothing
   }
   else
   {
      # pick out date and time
      if ($line =~ /([^-]*)-([^\[]*)\[\*\*\](.*)/)
      {
         $date = $1;
         $time = $2;
         $rest = $3;
      }
      else
      {
         print "Not parsed: $line\n";
      }

      # pick out alert, source address, source port, dest address, dest port
      if ($rest =~ /(.*)\[\*\*\](.*)->(.*)/)
      {
         $alert  = $1;
         $source = $2;
         $dest   = $3;
         ($srcaddr, $srcport) = split(/:/,$source);
         ($dstaddr, $dstport) = split(/:/,$dest  );
      }
      print "$date,$time,$qt$alert$qt,$srcaddr,$srcport,$dstaddr,$dstport\n";
   }

}
```

Perl Script to Convert Scan Logs to Comma Separate Files

```
#
# Name: cnvscan.pl
# Description:  convert a snort scan file into a comma separated file
# Date: 14 Nov 2001
# Arguments:  (none)
# Notes:
#    The program takes input from STDIN and puts it to STDOUT
#

$qt = '"';
while ($line = <STDIN>)
{

   $date    = '';
   $time    = '';
   $source  = '';
   $dest    = '';
   $srcaddr = '';
   $srcport = '';
   $dstaddr = '';
   $dstport = '';

   # convert SGML entity &gt; to >
   $line =~ s/&gt;/>/g;

   # convert Oct to 10/
   $line =~ s/Oct /10\//;
   ($date, $time, $source, $arrow, $dest, $prot) = split(/ /, $line);
   $alert  = 'Scan Detected';
   ($srcaddr, $srcport) = split(/:/,$source);
   ($dstaddr, $dstport) = split(/:/,$dest  );

   print "$date,$time,$qt$alert$qt,$srcaddr,$srcport,$dstaddr,$dstport\n";


}
```

Perl Script to Count Alerts or Scans per Address

```
#
# Name: countaddr.pl
# Description:  count number of alerts per address
# Date: 14 Nov 2001
# Arguments:  (none)
# Notes:
#    The program takes input from STDIN and puts it to STDOUT
#

%cntsrc = ();
%cntdst = ();
$qt = '"';
while ($line = <STDIN>)
{

   ($before, $alert, $after) = split(/"/, $line);
   ($nada, $srcaddr, $srcport, $dstaddr, $dstport) = split(/,/, $after);
   $srcaddr =~ s/ //g;
   $dstaddr =~ s/ //g;                # strip blanks from address strings
   $cntsrc{$srcaddr}++;
   $cntdst{$dstaddr}++;

}

# obtain array with unique addresses
%addr = ();
@srckeys = (keys(%cntsrc), keys(%cntdst));
foreach $key (@srckeys)
{
   $addr{$key} = 1;
}

@srckeys = keys(%addr);

foreach $key (@srckeys)
{
   if (!$cntsrc{$key}) {$cntsrc{$key} = 0;}
   if (!$cntdst{$key}) {$cntdst{$key} = 0;}
   print "$key,$cntsrc{$key},$cntdst{$key}\n";

}
```

Perl Script to Convert OOS File into Comma Separated Value File

```
# Name: countoos.pl
# Description:  convert OOS file into a comma separated file
# Date: 14 Nov 2001
# Arguments:  (none)
# Notes:
#    The program takes input from STDIN and puts it to STDOUT
#

$qt = '"';
while ($line = <STDIN>)
{
   $rest    = '';
   $date    = '';
   $time    = '';
   $alert   = 'Out of Spec';
   $source  = '';
   $dest    = '';
   $srcaddr = '';
   $srcport = '';
   $dstaddr = '';
   $dstport = '';


   if ($line =~ /^10\//)            # recognize line with addresses
   {
       # convert SGML entity &gt; to >
       $line =~ s/&gt;/>/g;

       # pick out date and time
       if ($line =~ /([^-]*)-([^ ]*) (.*)/)
       {
            $date = $1;
            $time = $2;
            $rest = $3;
       }
       else
       {
            print "Not parsed: $line\n";
       }

       # pick out alert, source address, source port, dest address, dest port
       if ($rest =~ /([^ ]*) -> ([^ ]*)/)
       {
            $source = $1;
            $dest   = $2;
            ($srcaddr, $srcport) = split(/:/,$source);
            ($dstaddr, $dstport) = split(/:/,$dest  );
       }
       print "$date,$time,$qt$alert$qt,$srcaddr,$srcport,$dstaddr,$dstport\n";

   }
}
```

Perl Script to Count Scans from Source Addresses to Destination Ports

```
#
# Name: src2port.pl
# Description:  Determine which addresses are scanning which ports
# Date: 14 Nov 2001
# Arguments:  (none)
# Notes:
#    The program takes input from STDIN and puts it to STDOUT
#

%cntaddrport = ();
%cntaddr     = ();

# scan input and count number of instances of source address and destination
# port combinations

while ($line = <STDIN>)
{
   chop($line);
   ($before, $alert, $after) = split(/"/, $line);
   ($nada, $srcaddr, $srcport, $dstaddr, $dstport) = split(/,/, $after);
   $srcaddr =~ s/ //g;
   $dstport =~ s/ //g;                # strip blanks from address strings
   $key = join(':', $srcaddr, $dstport);
   $cntaddrport{$key}++;
   $cntaddr{$srcaddr}++;
}

@srckeys = keys(%cntaddrport);

foreach $key (@srckeys)
{
   ($srcaddr, $dstport) = split(/:/, $key);

   print "$srcaddr,$dstport,$cntaddrport{$key}\n";

}

foreach $key (keys(%cntaddr))
{
   print "$key,TOTAL,$cntaddr{$key}\n";

}
```

Perl Script to Compare Scans to Alerts

```
#
# Name: compscan.pl
# Description:  Compare scans to alerts
# Date: 14 Nov 2001
# Arguments:  (none)
# Notes:
#    The program takes input from STDIN and puts it to STDOUT
#

%alerts = ();

# scan toptenalrt.csv for list of alert addresses
#

open(ALERT, 'toptenalrt.csv') or die "Cannot open toptenalrt.csv\n";
while ($line = <ALERT>)
{
   chop($line);
   ($addr, $numsrc, $numdst) = split(/,/, $line);
   if ($numsrc > 0)
   {
      $addr =~ s/ //g;
      $alerts{$addr} = $numsrc;
   }


}

%scans = ();
# scan toptenscan.csv
open(SCAN, 'toptenscan.csv') or die "Cannot open toptenscan.csv\n";
while ($line = <SCAN>)
{
   chop($line);
   ($addr, $numscr, $numdst) = split(/,/, $line);
   if ($numscr > 0)
   {
      $addr =~ s/ //g;
      if ($alerts{$addr})
      {
         $scans{$addr} = $numscr;
         # print "number of scans for $addr is $numscr\n";
      }
   }

}

foreach $key (keys(%scans))
{
   print "$key,$alerts{$key},$scans{$key}\n";
}
```

Perl Script to Show All Alerts for a Particular Source Address

```
# Name: showalerts.pl
# Description:  Count alerts by type for a particular source IP
# Date: 14 Nov 2001
# Arguments:  (none)
# Notes:
#    The program takes input from STDIN and puts it to STDOUT
#

%cntalert = ();

# scan input and count number of instances of alert for a particular IP
#
$host = '213.64.102.177';
while ($line = <STDIN>)
{
   chop($line);
   ($before, $alert, $after) = split(/"/, $line);
   ($nada, $srcaddr, $srcport, $dstaddr, $dstport) = split(/,/, $after);
   $srcaddr =~ s/ //g;

   if ($host eq $srcaddr)
   {
      $cntalert{$alert}++;
   }
}

@srckeys = keys(%cntalert);

foreach $key (@srckeys)
{

   print "$host,$key,$cntalert{$key}\n";

}
```