3.2 Denial Of Service
Outline

• Objective
• Denial of Service Definition
• Example Types of DoS
• Distributed DoS
• Summary
• List of References
Objective

• To become familiar with the concept of Denial of Service and a few common types of this attack that have been employed.
Types of Attacks

• Disclosure of information
• Destruction of data (malicious intent)
• Alteration of data (forgery)
• Denial of Service (DOS)
Disclosure of Information

• Attacks against the confidentiality of data.
• Attempts to disclose otherwise private information such as finances, personal letters or information, proprietary documentation or marketing strategies.
  – Password or Email Sniffers
Destruction of Data

Deletion or “wiping” of information stored on a server (or even of the server itself).
Alteration of Data

Editing of data to hide or falsify data

– Inserting or substituting data in a IP stream (TCP session Hijacking)
– UDP data stream alteration
Denial of Service

Malicious attacks designed to prevent the victim server or that server’s clients from accessing resources that would otherwise be available.
Denial of Service Definition

• An attempt to make a computer resource unavailable to its intended users.
• It is the concerted, malevolent efforts of a person or persons to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely for revenge, blackmail, etc..
• Generally implemented by:
  – forcing the targeted computer(s) to reset, or consume its resources such that it can no longer provide its intended service; and/or,
  – obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

Example Types of DoS

• SYN Flood
  – This attack sends a flood of TCP/SYN packets, often with a spoofed sender address. Each of the packets are handled like a connection request, causing the server to spawn a half-open connection, by sending back a TCP/SYN-ACK packet, and waiting for a TCP/ACK packet in response from the sender address. However, because the sender address is forged, the response never comes.

SYN Flooding

- Attacker
- Legitimate Client
- Spoofered System
- Victim System
- SYN
- ACKs
Example Types of DoS

• Ping Flood
  – A type of ICMP Attack based on sending the victim an overwhelming number of ping packets, usually using the "ping -f" command. It is very simple to launch, the primary requirement being access to greater bandwidth than the victim.

Ping Flooding

Attacking System(s)

Internet

Victim System
“Smurf” Attacks

The “Smurf” attack is a modification of the classic ping flood attack.

Instead of sending ICMP echo packets (aka ping packets) from your system to the victims host/network, a packet is to sent to a broadcast address of intermediate network with a forged return address of the victim’s host.
“Smurf” Attacks

A Single System

Victim System

A Networks of Systems

A Single System
“Smurf” Attacks

As with ICMP Echo Flood attacks, the best protection is through IP filtering of ICMP_ECHOREPLY packets.
Land Attack

- Involves sending a spoofed TCP/SYN packet (connection initiation) with the target host's IP address with an open port as both source and destination. The attack causes the targeted machine to reply to itself continuously and eventually crash.

LAND Attack

Attacker

Victim System
Ping Of Death

The “Ping of Death” attack involves sending a unexpectedly large IP packet to the victim host. While this is most commonly sent as a ICMP_ECHO packet (AKA: ping), this attack may be done with almost any IP packet type. Attacks an implementation bug in the layer 3 code.
Ping Of Death

Most OSs have patches available to correctly deal with such IP packets

Up stream protection can be done at the router/IP filter level by not forwarding illegal packets (packet size > 65515), (although with fragmentation, this can be difficult to filter/stop).
Ping Of Death

One option is filter fragmented IP and TCP packets.

This solution can also inhibit connectivity from some remote sites that are forced to fragment due to high network congestion or bad network design on there local LANs.
IP Fragmentation Attacks

- Overlapping Fragment Attacks (AKA: “Teardrop”)
- IP Fragment Flooding
- Micro Fragments
- Overlapping Fragment Attack (hiding data)
Overlapping Fragment Attacks

Some implementations of the TCP/IP IP have bugs in the IP fragmentation re-assembly code and thus do not properly handle overlapping IP fragments.

“Teardrop” is a widely available attack tool that exploits this vulnerability.
Overlapping Fragment Attacks

There are patches available for Windows that make it resistant to this attack.

Linux systems should be upgraded to the current release.

IP filtering will also help protect unpatched systems.
IP Fragmentation Flooding

Any TCP/IP implementation has to deal with fragmented packets of one form or another.

A DOS attack can be executed by sending random IP fragments to a system.
IP Fragmentation Flooding

This will cause the system to buffer these fragments awaiting other IP or TCP fragments to reassemble the packets with.

A DOS attack situation can exist in cases where a joining fragment never arrives, thus causing a system to run low on memory and CPU resources.
Micro Fragments

One common method to circumvent IP filter based routers is to fragment one’s IP packets sufficiently so that the encapsulated TCP header can not be analyzed and filtered. This technique can also be used to evade detection from current IDS systems.
Distributed DoS

- Occurs when multiple compromised systems (sometimes called BOTS or ZOMBIES) flood the bandwidth or resources of a targeted system.
- The Bots may be compromised by attackers using a variety of methods.
  - “Break-in”
  - Malware
- The assembled Bots combine to form an attack that overwhelms the target.
Summary

• We have discussed the concept of Denial of Service and a few common types of this attack that have been employed.
List of References

- http://searchsoftwarequality.techtarget.com/sDefinition/0,290660,sid92_gci213591,00.html
- http://staff.washington.edu/dittrich/misc/ddos/
- http://www.irchelp.org/irchelp/nuke/
- http://attrition.org/security/denial/
- http://www.acm.org/ubiquity/views/m_kabay_1.html
3.3 Malware
Outline

• Objective
• Malware Definition
• How Did I Get It?
• Impacts
• Viruses
• Worms
• Trojan Horses
• Virus Hoax
• Spyware
• Botnet
• Summary
• List of References
Objective

• To provide familiarity with the different basic kinds of malicious software that may affect your system.
Malware Definition

• Software designed to infiltrate or damage a computer system without the owner's informed consent.
• Software is considered malware based on the perceived intent of the creator rather than any particular features.
• A catch-all term to refer to any software designed to cause damage to a single computer, server, or computer network.
• Popular press usually refers to any malware as a virus.

http://en.wikipedia.org/wiki/Malware
http://www.microsoft.com/technet/security/alerts/info/malware.mspx
How Did I Get It?

- In an email
- From a downloaded file or program
- From a purchased file or program
- From visiting a web site
- From a floppy disk/CD/DVD/USB Drive, etc.
- It sent itself through the network
Impacts

- Slows Down System
- Redirects Your Browser
- Displays strange things
- Destroys Information
- Infects Other Systems
- Used To Launch Attacks Against Others
- Causes Annoying Pop-ups
- Gathers Information About You and Sends it to Someone Else
- Dials a Long Distance Phone Number (1-900)
- Etc.
Viruses

• A computer program that can copy itself and infect a computer without permission or knowledge of the user.
• Spreads from one computer to another when its host is taken to the uninfected computer by a user sending it over a network or carrying it on a removable medium. Additionally, viruses can spread to other computers by infecting files on a network file system or a file system that is accessed by another computer.
• May be programmed to damage the computer by altering programs, deleting files, or reformatting the hard disk.
• Some are not designed to do any damage, but simply replicate themselves and perhaps make their presence known by presenting text, video, or audio messages. Even these benign viruses can create problems for the user. They exhaust memory used by legitimate programs causing erratic behavior possibly resulting in system crashes.

Worms

• A self-replicating computer program that uses the network to send copies of itself to other nodes and may do so without any user intervention. Unlike a virus, it does not need to attach itself to an existing program.

• Worms always harm the network (if only by consuming bandwidth), whereas viruses always infect or corrupt files on a targeted computer.

• Some may delete files on a host system, encrypt files in a cryptoviral extortion attack, or send documents via e-mail.

• A very common worm payload is to install a backdoor in the infected computer to allow the creation of a "zombie" under control of the worm creator. These are commonly used by spam senders for sending junk email or to hide their address during an attack.
Trojan Horses

• A program that unlike a virus, contains or installs a malicious program (sometimes called the payload or 'trojan') while under the guise of being something else.

• Term is derived from the classical myth of the Trojan Horse. Trojan horses may appear to be useful or interesting programs (or at the very least harmless) to an unsuspecting user, but are actually harmful when executed.

• Two common types of Trojan horses:
  – Otherwise useful software corrupted by inserting malicious code that executes while the program is used. Examples include various implementations of weather alerting programs, computer clock setting software, and peer to peer file sharing utilities.
  – A standalone program that masquerades as something else, like a game or image file, in order to trick the user into some misdirected complicity that is needed to carry out the program's objectives.

• Impacts May Include:
  – Allowing Remote Access
  – Sending Emails
  – Destroy Data
  – Adding or copying data from the infected computer
  – Disabling security software
  – Participating in DoS attacks
  – Directing the infected computer to only connect to the internet via an expensive dial-up connection

http://en.wikipedia.org/wiki/Trojan_Horse_%28Computing%29
Virus Hoax

• A false email message warning the recipient of a virus that is going around. The message is usually in the form of a chain e-mail that tells the recipient to forward it to everyone they know.

• Usually harmless and do nothing more than annoy people who know it's a hoax or waste the time of people who forward it. However, a number of hoaxes have warned users that vital system files are viruses, and have encouraged the user to delete the file, possibly damaging the system.

• Most hoaxes are identified by the fact that they say the virus will do impossible things, like blow up the recipient's computer. They often claim to be from reputable organizations such as Microsoft and IBM, and include emotive language and encouragement to forward the message which would not come from an official source.

http://en.wikipedia.org/wiki/Virus_hoax
Spotting a Hoax

• IBM’s anti-virus site lists four typical components of hoax virus messages:
  – All capital letters with copious explanation marks in the announcement
  – Elaborate claims of extensive damage
  – Description of the virus as 'new and very malicious'
  – Request to send the message along to as many people as you can.

• Also:
  – Invoke some authority, such as CERT, FCC, NCSA.
Check before Circulating!

Before passing along any virus warnings, be sure to check one of the hoax/urban legend sites, such as:

  - CIAC's Internet Hoax Site. One of the best lists. CIAC is the US Department of Energy's Computer Incident Advisory Capability. They investigate this sort of thing

  - MCI's Internet Hoax Site

- [http://www.ncsa.com/services/consortia/anti-virus/alerthoax.html](http://www.ncsa.com/services/consortia/anti-virus/alerthoax.html)
  - NCSA's Hoax Site; Has a good overview section
Some Famous Hoaxes

- **Good Times Virus**
  - first of the “biggies” --
  - See
    http://www.public.usit.net/lesjones/goodtimes.html
- **Join The Crew**
  - Typical!
- **Irina**
  - According to CIAC, publicity stunt from Penguin gone wrong
- **Blot**
  - Used NASDAQ names and other authoritative ids

Here is some important information. Beware of a file called Goodtimes.

Happy Chanukah everyone, and be careful out there. There is a virus on America Online being sent by E-Mail. If you get anything called "Good Times", DON'T read it or download it. It is a virus that will erase your hard drive. Forward this to all your friends. It may help them a lot.

The FCC released a warning last Wednesday concerning a matter of major importance to any regular user of the InterNet. Apparently, a new computer virus has been engineered by a user of America Online that is unparalleled in its destructive capability. Other, more well-known viruses such as Stoned, Airwolf, and Michaelangelo pale in comparison to the prospects of this newest creation by a warped mentality.

What makes this virus so terrifying, said the FCC, is the fact that no program needs to be exchanged for a new computer to be infected.
Spyware

- Computer software (also called Adware) that collects personal information about users without their informed consent.
- Personal information is secretly recorded with a variety of techniques, including logging keystrokes, recording Internet web browsing history, and scanning documents on the computer's hard disk.
- Purposes range from criminal (password or credit card number theft) to merely annoying (recording Internet search history for targeted advertising or displaying pop-ups) while consuming computer resources.
- Indications are pop-ups, slow performance, lock-ups, etc.
- 2005 AOL/National Cyber-Security Alliance study
  - 61% of surveyed users had some form of spyware
  - 92% of surveyed users with spyware did not know they had it
  - 91% reported that they had not given permission for its installation

Botnet

• A collection of software robots, or bots, which run autonomously.

• Compromised computers (zombies) running programs, usually worms, Trojan horses, or backdoors, under a common command and control infrastructure.

• The botnet's originator (aka "bot herder") can control the group remotely usually for nefarious purposes.

• Botnets are exploited for various purposes, including DoS attacks, creation of mail relays for spam, click fraud (creating false web page accesses), and the theft of application serial numbers, login IDs, and financial information like credit card numbers.
Malicious Code

• Can cause much harm
• Can lie dormant until a trigger
  – Trigger such as: time, date, interval, event, or condition
• Can do different things each time invoked
• Same control (or more) than the user
• Malicious code has been studied for a long time.
**Definitions**

**Virus** - code that copies itself into other programs.

“**Bacteria**” replicates until it fills all disk space, or CPU cycles.

**Payload** - harmful things the malicious program does, after it has had time to spread.

**Worm** - a program that replicates itself across the network (usually riding on email messages or attached documents (e.g., macro viruses).

**Trojan Horse** - instructions in an otherwise good program that cause bad things to happen (sending your data or password to an attacker over the net).

**Logic Bomb** - malicious code that activates on an event (e.g., date).

**Trap Door** (or Back Door) - undocumented entry point written into code for debugging that can allow unwanted users.

**Easter Egg** - extraneous code that does something “cool.” A way for programmers to show that they control the product.
Kinds of Malicious code

• Virus: a program which can pass on malicious code to other non-malicious programs by modifying them.
  – Transient: runs when host program runs
  – Resident: remains active after host program ends.
Kinds of Malicious code (cont.)

• Trojan Horse: Has a non-malicious primary purpose and has a second non-obvious malicious effect.
  – Trojan Login scripts
  – Complete Control
    • BO2K, SubSeven
  – Many kinds!
Kinds of Malicious code (cont.)

• Logic Bomb: detonates, or goes off when a special condition occurs.
• Time Bomb: A logic bomb whose trigger is a time and date.
WESTBURY, N.Y. (AP) -- A computer company owner and his technician are accused of planting a virus in a dissatisfied customer's computer system, after the customer refused to pay for a program. Michael Lofaro, 29, owner of MJL Design of Manhattan, and his technician, John Puzzo, 22, were charged Monday with attempted computer tampering and coercion, said Lt. Lawrence Mulvey of the Nassau County police.

The article explains that the maximum penalties are 4-7 years and up to 5,000 in fines. The client, William Haberman, owner of Forecast Inc., a furniture company in Westbury, complained about poor performance in a program sold by MJL Design and refused to pay the full invoice when the vendor allegedly ignored his complaints.

According to the accusation, Lofaro and Puzzo planted a "computer virus" and threatened to detonate it. The accused were arrested when they came to defuse the logic bomb.
Kinds of Malicious code (cont.)

• Trapdoor or Backdoor – A feature in a program by which someone can access the program other than the obvious, direct call, perhaps with special privileges.
Trap Doors

• Secret, undocumented entry point, usually installed during development, in an OS or privileged program, that installs “hooks” to allow future expansion

• By performing appropriate action (e.g., weird flag on the command-line) user can "get in" by back/trap door

• Not always sinister
  – Some OS come with certain privileged accounts meant for the vendor's technicians to use at service times
Causes of Trapdoors

• Poor error checking
  – unacceptable input may not be caught
  – recent example, CERT advisory: JavaScript can be passed in html forms when overflows occur.
• Undefined opcodes - machine level (hardware). Newer machines are trapping these, however, test modes are now available
• Programmer:
  – forgets to remove them
  – leaves them in for production maintenance
  – leaves them in for testing
  – leaves them in as a covert means of access after program has been into production.
Kinds of Malicious code (cont.)

• WORM – spreads copies of itself through a network. Like a virus, but network capable spreading.

• Rabbit – virus or worm which replicates without bound.

• Easter Eggs – like a Trojan, undesired program code. Example, Demo.
Malicious code combinations

• The types of Malicious codes may be combined. For example, A **logic bomb**, which is a **time bomb**, which features a **trapdoor**, and spreads itself like a **worm** when triggered.

• Popular Press generally only uses the term “**virus**”
Malicious Logic

• **Malicious logic**: *Instructions designed to violate the security policy of a site.*

• **Sometimes called “malware”:**
  - Malicious code which moves itself (active)
    • **Virus**: code that copies itself into another program
    • **Worm**: migrating programs.
  - Malicious code which stays in one place (static)
    • **Trojan Horse**: covert code in a 'known' function.
    • **Trap Door**: bypass authentication
    • **Logic Bomb**: perform (destructive) action upon external condition.
  - Other
    • **Rabbit, bacterium**: a “virus” or other denial of service-style attack that continuously tries to use resources until they have all been absorbed.
    • **Black Widows**: malicious applets.
Summary

• We have covered some of the different kinds of malicious software that may affect your system.
List of References

• http://en.wikipedia.org/wiki/Malware
• http://www.microsoft.com/technet/security/alerts/info/malware.mspx
• http://en.wikipedia.org/wiki/Computer_virus
• http://en.wikipedia.org/wiki/Computer_worm
• http://en.wikipedia.org/wiki/Trojan_Horse_%28Computing%29
• http://en.wikipedia.org/wiki/Virus_hoax
• http://en.wikipedia.org/wiki/Spyware
• http://en.wikipedia.org/wiki/Botnet
3.3A Malware Addendum
Internet Worm

• Cornell University student Robert T. Morris Jr. (1988)
• Convicted in 1990 ($10,000 fine, 3 year suspended jail, 400 hrs community service - ten 40 hours workweeks)
• http://www.websteruniv.edu/philosophy/~umbaugh/courses/frosh/dairy/morris.htm
Case study: 1988 Internet Worm

• Disabled 2600--6000 machines, or 5% of the machines on the Internet then.
  – Targets: Sun and VAX Unix (BSD variant).
  – *Really a worm carrying a virus using trapdoors and behaving like a rabbit (!)*

• So very common and well-publicized that most of the exploited vulnerabilities have been closed (though not everywhere!).
Internet Worm Objective

- Determine where it could spread
- Spread its infection
- Remain undiscovered (hidden)
A Flaw

• The worm was to negotiate that a newer infection or the existing infection would terminate.
• However, a flaw in the code, new copies did not terminate.
• The effect: unlimited, uncontrolled reproduction and spreading.
How??

• Exploited known flaws in Berkeley V.4 Unix.
• Where to spread? Used a bug in finger, sendmail to find users on target machine.
  – Tried guessing passwords. 1) 432 common PWs, 2) tried the systems dictionary file.
  – Fingerd daemon. Used a buffer overflow to gain remote shell.
  – Sendmail. Used a trapdoor left as a debugging tool.
The Spread

- Sent a bootstrap loader written in C
- the bootstrap loader would fetch the remaining code.
- The loader supplied a one-time password back to the sender.
- If transmission error occurred, it zeroed out the bootstrap loader, deleted all code, and exited.
Stealth

- Once established on a host with its full code it brought into memory its full code
- It encrypted the code in memory.
- It deleted original copies from disk. No traces were left on the disk!
- A memory dump would not expose the worm either!
- It would periodically change its name and process ID.
Good outcomes

• 400 hours community service
• Computer Emergency Response Team (CERT) at Carnegie Mellon University (Pittsburgh, PA)
• WWW.CERT.ORG
Initial Attack

• How did it start?

Starting Host → Ending Host
Overview of the Morris Worm

- Worm selects host for infection.
- Place hook (grappling hook) on host.
- Causes hook to compile and run.
- Executing hook copies remainder of code over.
Step 1: About to Infect

- Status: infecting machine has found a victim

Server Worm

Socket established on victim
Step 2: Placing Grappling hook

- Status: Send code across

Server Worm

TCP or SMTP connection
Step 3: Pull in code from server

- Status: Send challenge string back, transfer Sun, VAX binaries and vector Server Worm

Send challenge
Copy files back
Execl shell and
maintain socket

Server Worm
Check challenge string; transmit files
Step 4: Get further commands from server worm

- Status: Server sends code to compile

Execute compile commands as sent by Server Worm; only the appropriate binary will link.

Send shell commands to attempt compile of binaries (one after the other)
Step 5: Hide and gather information

• Status: New worm generates victim lists
Step 6: Attempt infection

- Read through hosts.equiv, /.rhosts, .forward (not user .rhosts!)
- Attempt to break each user password ...
  - with simple guesses
  - with internal dictionary
  - with UNIX online dictionary
- Loop “forever” trying to infect new hosts
Step 7: success!

• For any successful password compromise
  – look for remote machines where the user has an account (.forward .rhosts)
  – For each, attempt a remote shell (rexec)

• Upon success, continue as before
First step: placing the hook

• Techniques:
  – Sendmail attack
  – Trusted remote host attack
  – Finger attack
  – Dictionary-based password attack

• Sendmail Attack
  – sendmail implements a mail delivery protocol. Not the only implementation, but most common. Developers put trapdoor in sendmail to assist with debugging:
    • Trapdoor executed command on remote host
    • Intended for in-house debugging
  – Many vendors didn't realize/understand the trapdoor, and left it enabled upon shipment
Sendmail, continued

• Simply put, the worm did the following:
  – Put commands in the body of a mail message
  – While running, deleted mail header
  – Body was run through an interpreter, and it:
    • Stored small 99-line program in a file (grappling hook)
    • Compiled program
    • Started it executing
Timeline (1988)

Timeline, courtesy of Gene Spafford:

- 11/2, 6pm believed started on MIT machine
- 11/2, 6:24p West Coast site infected
- 11/2, 7:04p UC Berkeley attacked
- 11/2, 7:54p UMD College Park attacked
- 11/2, 8:40p Berkeley figures out sendmail and trusted host potential finger problem
  - shuts down those services
- 11/2, 8:49p U Utah infected
- 11/2, 9p Stanford infected
- 11/2, 9p RAND corp attacked
- 11/2, 10:06p 100 active procs on Utah; unusable
- 11/2, 10:20p Utah clean
- 11/2, 10:41p Utah again infected and unusable
- 11/2, 10:49p Utah shut down and rebooted
- 11/2, 11:21p Utah yet again infected and unusable
- 11/2, 11:28p General "attack" message posted
Whodunnit?

• Identified: Robert T. Morris, then-Cornell grad student and son of famous computer expert Robert Morris of NCSC.

• Intent? Morris (and friends) have made several comments ... most recent is that the worm was intended to benignly spread slowly throughout the Internet. The report is that Morris told a friend to post directions on how to shut down the worm on a BB but the friend was unable to do so since machines were offline (due to the attack).
And what happened to Morris?

- One-count felony indictment and conviction under 1986 Computer Fraud and Abuse Act.
- 3 years probation, $10K fine, 400 hours of community service.
How did they find the author?

- Software forensics and other things ...!
- Spaford of Purdue, Seely of Utah, Eichin and Rochlis of MIT decompiled the worm.
  - Coding techniques seen within the worm indicated authorship (ex: no apparent knowledge of linked lists where appropriate, and author hadn't taken a data structures class)
  - Discovered that the grappling hook could have transferred twenty files, but used only three ... leading to speculation that this version was still preliminary and might have had viral code, etc, added at a later date.
Development Speculations

• Logs at Cornell of the mailer indicated development work done on worm
  – first test seen Oct 19
  – continued thru Oct 28
  – intensive testing on Oct 29 ... mechanism for propagating worm apparently failed
  – grappling hook program appeared Nov2. First tested 5pm.
  – possible second infection on Cornell machine at 5:04p, which might be the real source.