


PURDUE UNIVERSITY

Computer Security

CS 426

Lectures 9-10

Malicious Programs



Elisa Bertino
Purdue University
IN, USA
bertino@cs.purdue.edu

Center for Education and Research
in Information Systems and Security

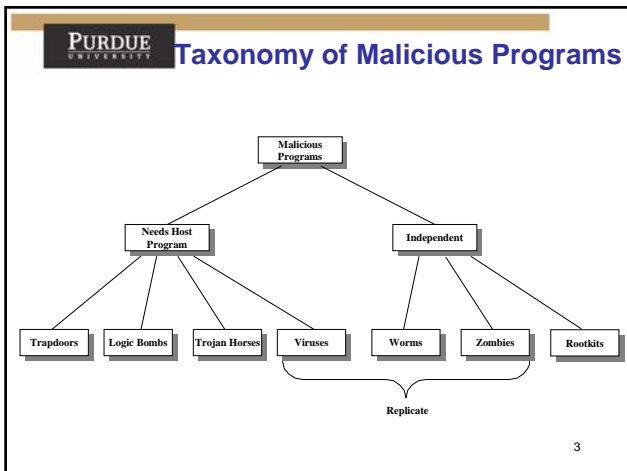
1

PURDUE UNIVERSITY

What is a Malicious Program


- **Malware:** software designed to infiltrate or damage a computer system without the owner's **informed consent**
- **Spyware:** software designed to intercept or take partial control over the user's interaction with the computer, without the user's **informed consent**
 - secretly monitors the user's behavior
 - collect various types of **personal information**

2



PURDUE UNIVERSITY

Trapdoor/Back Door



- Secret entry point into a system
 - Specific user identifier or password that circumvents normal security procedures.
- Presents a security risk
- Could be used for
 - Troubleshooting
 - Maintenance
 - Malicious intent

4

Logic Bomb



- Embedded in legitimate programs
- Activated when specified conditions met
 - E.g., presence/absence of some file; Particular date/time or particular user
- When triggered, typically damages system
 - Modify/delete files/disks

5

Trojan Horse



- Program with an overt (expected) and covert effect
 - Appears normal/expected
 - Covert effect violates security policy
- User tricked into executing Trojan horse
 - Expects (and sees) overt behavior
 - Covert effect performed with user's authorization

6

Virus



- Self-replicating code
 - Like replicating Trojan horse
 - Alters normal code with "infected" version
- No *overt* action
 - Generally tries to remain undetected
- Operates when infected code executed
 - If *spread condition* then
 - For *target files*
 - if not infected then *alter to include virus*
 - Perform malicious action
 - Execute normal program

7

Virus Types



- Boot Sector
 - Problem: How to ensure virus "carrier" executed?
 - Solution: Place in boot sector of disk
 - Run on any boot
 - Propagate by altering boot disk creation
 - *Similar concepts now being used for thumb drive*
- Executable
 - Malicious code placed at beginning of legitimate program
 - Runs when application run
 - Application then runs normally

8

Virus Types/Properties



- Terminate and Stay Resident
 - Stays active in memory after applications complete
 - Allows infection of previously unknown files
 - Trap calls that execute a program
- Stealth
 - Conceal Infection
 - Trap read and disinfect
 - Let execute call infected file
 - Encrypt virus
 - Prevents “signature” to detect virus
 - Polymorphism
 - Change virus code to prevent signature

9

Macro Virus



- Infected “executable” isn’t machine code
 - Relies on something “executed” inside application data
 - Common example: Macros
- Otherwise similar properties to other viruses
 - Architecture-independent
 - Application-dependent

10

Worm



- Runs independently
 - Does not require a host program
- Propagates a fully working version of itself to other machines
- Carries a payload performing hidden tasks
 - Backdoors, spam relays, DDoS agents; ...
- Phases
 - Probing → Exploitation → Replication → Payload



11

Cost of Worm Attacks

- Morris worm, 1988
 - Infected approximately 6,000 machines
 - 10% of computers connected to the Internet
 - cost ~ \$10 million in downtime and cleanup
- Code Red worm, July 16 2001
 - Direct descendant of Morris’ worm
 - Infected more than 500,000 servers
 - Caused ~ \$2.6 Billion in damages,
- Love Bug worm: May 3, 2000, \$8.75 billion

Statistics: Computer Economics Inc., Carlsbad, California

12

Morris Worm (First major attack)

- Released November 1988
 - Program spread through Digital, Sun workstations
 - Exploited Unix security vulnerabilities
 - VAX computers and SUN-3 workstations running versions 4.2 and 4.3 Berkeley UNIX code
- Consequences
 - No immediate damage from program itself
 - Replication and threat of damage
 - Load on network, systems used in attack
 - Many systems shut down to prevent further attack

13

Morris Worm Description

- Two parts
 - Program to spread worm
 - look for other machines that could be infected
 - try to find ways of infiltrating these machines
 - Vector program (99 lines of C)
 - compiled and run on the infected machines
 - transferred main program to continue attack
- Security vulnerabilities
 - fingerd – Unix finger daemon
 - sendmail - mail distribution program
 - Trusted logins (.rhosts)
 - Weak passwords

14

Morris Worm Spread Mechanisms

- Sendmail
 - Exploit debug option in sendmail to allow shell access
- Fingerd
 - Exploit a buffer overflow in the gets function
 - Apparently, this was the most successful attack
- Rsh
 - Exploit trusted hosts
 - Password cracking

15

sendmail

- Worm used debug feature
 - Opens TCP connection to machine's SMTP port
 - Invokes debug mode
 - Sends a RCPT TO that pipes data through shell
 - Shell script retrieves worm main program
 - places 40-line C program in temporary file called x\$\$,l1.c where \$\$ is current process ID
 - Compiles and executes this program
 - Opens socket to machine that sent script
 - Retrieves worm main program, compiles it and runs

16

fingerd

- Written in C and runs continuously
- Array bounds attack
 - Fingerd expects an input string
 - Worm writes long string to internal 512-byte buffer
- Attack string
 - Includes machine instructions
 - Overwrites return address
 - Invokes a remote shell
 - Executes privileged commands

17

Remote shell

- Unix trust information
 - /etc/host.equiv – system wide trusted hosts file
 - /.rhosts and ~/.rhosts – users' trusted hosts file
- Worm exploited trust information
 - Examining files that listed trusted machines
 - Assume reciprocal trust
 - If X trusts Y, then maybe Y trusts X
- Password cracking
 - Worm was running as daemon (not root) so needed to break into accounts to use .rhosts feature
 - Read /etc/passwd, used ~400 common password strings & local dictionary to do a dictionary attack

18

The worm itself

- Program is shown as 'sh' when ps
 - Clobbers argv array so a 'ps' will not show its name
 - Opens its files, then unlinks (deletes) them so can't be found
 - Since files are open, worm can still access their contents
- Tries to infect as many other hosts as possible
 - When worm successfully connects, forks a child to continue the infection while the parent keeps trying new hosts
 - find targets using several mechanisms: 'netstat -r -n', /etc/hosts, ...
- Worm did not:
 - Delete system's files, modify existing files, install trojan horses, record or transmit decrypted passwords, capture superuser privileges

19

Detecting Morris Internet Worm

- Files
 - Strange files appeared in infected systems
 - Strange log messages for certain programs
 - System load
 - Infection generates a number of processes
 - Password cracking uses lots of resources
 - Systems were reinfected => number of processes grew and systems became overloaded
 - Apparently not intended by worm's creator
- Thousands of systems were shut down

20

Trivia questions

- What happened to Morris?
- Where is now Morris?
- Who was the first to analyze the Morris worm?

21

Trivia questions

- What happened to Morris
 - **Robert T. Morris was convicted of violating the computer Fraud and Abuse Act (Title 18), and sentenced to three years of probation, 400 hours of community service, a fine of \$10,050, and the costs of his supervision.**
- Where is now Morris?
- Who was the first to analyze the Morris worm?

22

Trivia questions

- What happened to Morris
 - Robert T. Morris was convicted of violating the computer Fraud and Abuse Act (Title 18), and sentenced to three years of probation, 400 hours of community service, a fine of \$10,050, and the costs of his supervision.
- Where is now Morris?
 - **Professor at MIT**
- Who was the first to analyze the Morris worm?

23

Trivia questions

- What happened to Morris
 - Robert T. Morris was convicted of violating the computer Fraud and Abuse Act (Title 18), and sentenced to three years of probation, 400 hours of community service, a fine of \$10,050, and the costs of his supervision.
- Where is now Morris?
 - Professor at MIT
- Who was the first to analyze the Morris worm?
 - **Prof. Spafford at Purdue**
 - **The Internet Worm Program: An Analysis**

24

Nimda Worm

- Spreads via 5 methods to Windows PCs and servers
 - e-mails itself as an attachment (every 10 days)
 - runs once viewed in preview plane (due to bugs in IE)
 - scans for and infects vulnerable MS IIS servers
 - exploits various IIS directory traversal vulnerabilities
 - copies itself to shared disk drives on networked PCs
 - appends JavaScript code to Web pages
 - surfers pick up worm when they view the page.
 - scans for the back doors left behind by the "Code Red II" and "sadmind/IIS" worms

29

Nimda Worm

- Nimda worm also
 - enables the sharing of the c: drive as C\$
 - creates a "Guest" account on Windows NT and 2000 systems
 - adds this account to the "Administrator" group.
- 'Nimda fix' Trojan disguised as security bulletin
 - claims to be from SecurityFocus and TrendMicro
 - comes in file named FIX_NIMDA.exe
 - TrendMicro calls their free Nimda removal tool FIX_NIMDA.com

30

Other Worms

- Warhol worms
 - infect all vulnerable hosts in 15 minutes – 1 hour
 - optimized scanning
 - initial hit list of potentially vulnerable hosts
 - local subnet scanning
 - permutation scanning for complete, self-coordinated coverage
- Flash worms
 - infect all vulnerable hosts in 30 seconds
 - determine complete hit list of servers with relevant service open and include it with the worm
- Stealthy worms

31

January 2009; Coming to a Computer Near You: Downadup

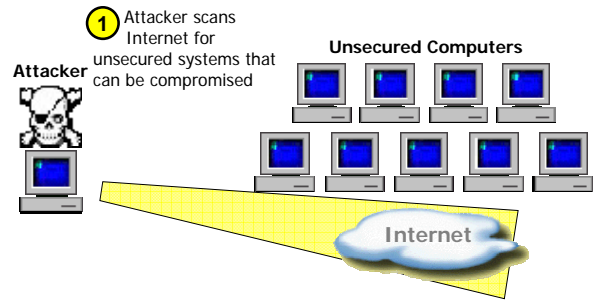
- Downadup aka Conficker aka Kido
 - has infected over 9 million machines so far.
- Known to steal personal and financial information.
- The worm is spreading through low security networks, memory sticks, and PCs without current security updates (exploits a bug in Windows)
- Home or individual PC's are known to be at less risk from this virus as it is programmed to target large networks.
- The structure, design and functions not entirely understood, the purpose still unclear, it is being researched and reverse engineered as of now.
- Waiting for the attack ...

32

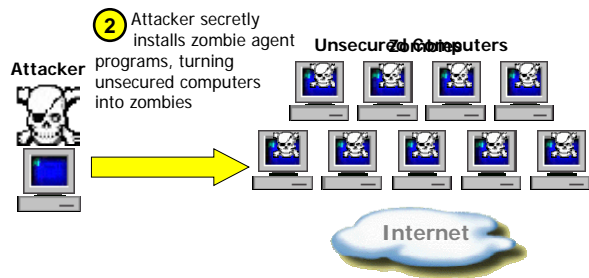
Zombie & Botnet

- Secretly takes over another networked computer by exploiting software flaws
- Builds the compromised computers into a zombie network or botnet
 - a collection of compromised machines running programs, usually referred to as worms, Trojan horses, or backdoors, under a common command and control infrastructure.
- Uses it to indirectly launch attacks
 - E.g., DDoS, phishing, spamming, cracking

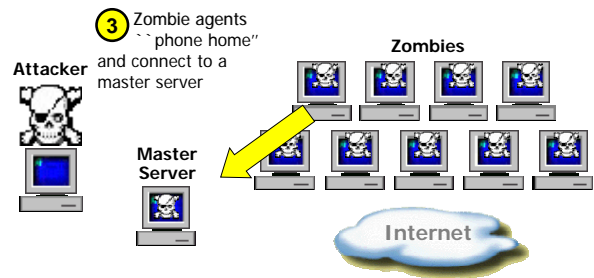
Detailed Steps (1)

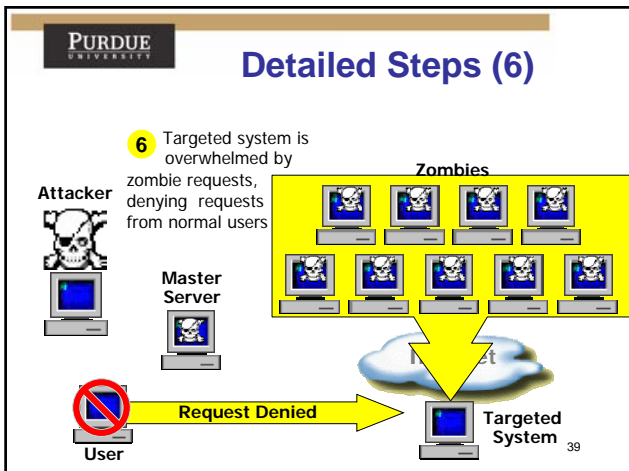
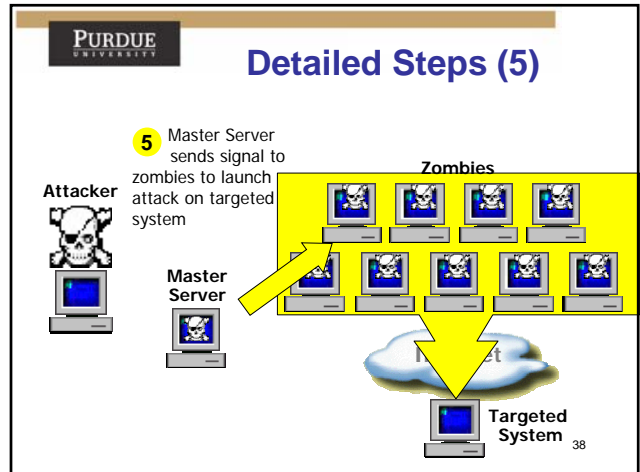
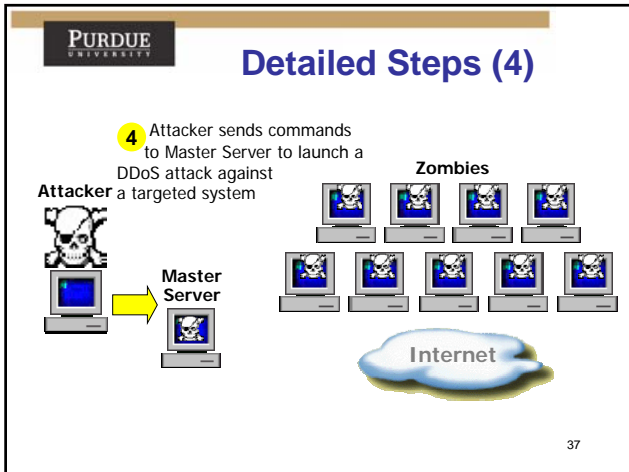


Detailed Steps (2)



Detailed Steps (3)

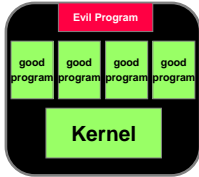




- PURDUE UNIVERSITY**
- ### Rootkit
- Software used after system compromise to:
 - Hide the attacker's presence
 - Provide backdoors for easy reentry
 - Simple rootkits:
 - Modify user programs (ls, ps)
 - Detectable by tools like Tripwire
 - Sophisticated rootkits:
 - Modify the kernel itself
 - Hard to detect from userland
- 40

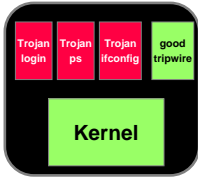
Rootkit Classification

Application-level Rootkit



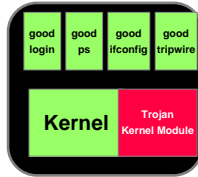
Hxdef, NTIllusion

Traditional RootKit



Lrk5, t0rn

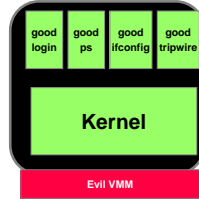
Kernel-level RootKit



Shadow Walker, adore

Rootkit Classification

Under-Kernel RootKit



SubVirt, ``Blue Pill''