These Slides are prepared from
Matt Bishop slides and book "Introduction to Computer Security"
Benefiting from the Slides posted by Ahmad Al-Mulhem

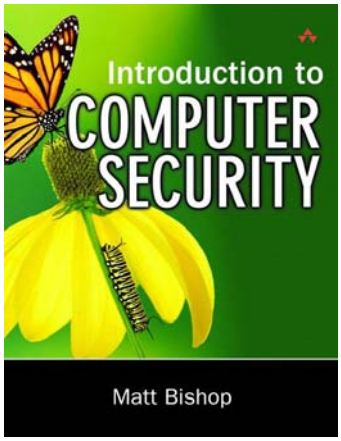
Malicious Logic

Chapter 19

Adnan Gutub

gutub@kfupm.edu.sa

Computer Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia



Overview

Defining malicious logic

Types
- Trojan horses
- Computer viruses and worms
- Other types:
  - Rabbits/Bacreria
  - Logic Bombs

Defenses
- Characteristics
  - Trust
- Countermeasures
  - Anti-Virus Software

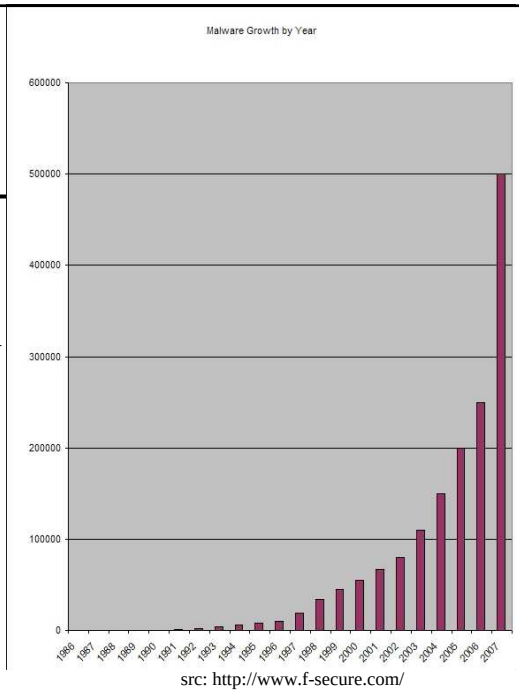
Malicious Logic

Set of instructions that cause site security policy to be violated

- Also called *Malicious software* (malware)
- Incorrectly called "computer viruses"
- Based on intention (e.g. rm -fr *)
- Should not be confused with defective software (bugs)

Malicious Logic

- Most of released code today are malicious
- F-Secure: 25,000 malware samples every day!
- Symantec: web-based malware instead of direct attacks
- Delivered through the Internet, by email and websites



src: http://www.f-secure.com/

Trojan Horse

Program with an *overt* purpose (known to user) and a *covert* purpose (unknown to user)
- Often called a Trojan
- Named by Dan Edwards in Anderson Report

Example: previous script is Trojan horse
- Overt purpose: list files in directory
- Covert purpose: create setuid shell

Usually superficially attractive (eg game, s/w upgrade etc)

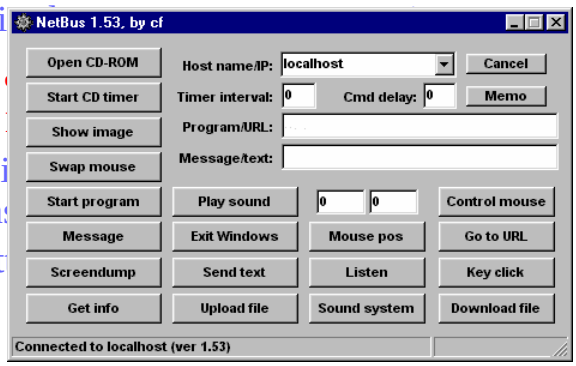
The covert purpose is done by a hidden payload:
- Remote Access
- Data Destruction
- Downloader
- Server Trojan (Proxy, FTP , IRC, Email, HTTP/HTTPS, etc.)
- Security software disabler
- Denial-of-service attack (DoS)

Trojan Horse Example: NetBus

Designed for Windows NT system

Victim uploads and installs this
- Usually disgui

Acts as a server,
commands for
- This includes i
mouse motion
- Also allows at



Replicating Trojan Horse

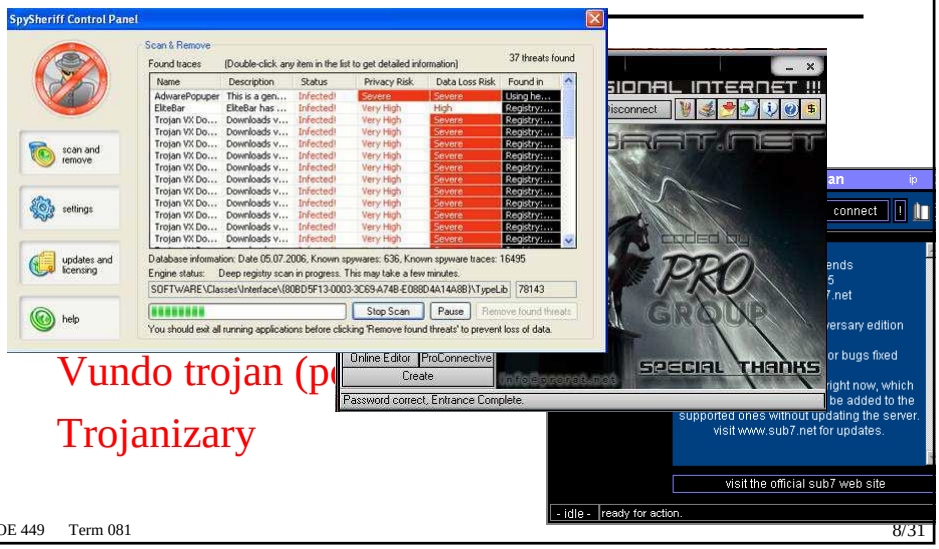
Trojan horse that makes copies of itself

- Also called *propagating Trojan horse*
- Early version of *animal* game used this to delete copies of itself

Hard to detect

- 1976: Karger and Schell suggested modifying compiler to include Trojan horse that copied itself into specific programs including later version of the compiler
- 1980s: Thompson implements this

Trojan Horses: Examples



Vundo trojan (p

Trojanizary

Computer Virus

Program that inserts itself into one or more files and performs some action

- *Insertion* (*infection*) *phase* is inserting itself into file
- *Execution phase* is performing some (possibly null) action

Insertion phase *must* be present

- Need not always be executed
  - Spreads without permission or knowledge of the user
- Lehigh virus inserted itself into boot file only if boot file not infected
- Requires a host (program) to spread

Trojans vs Viruses

- Trojans have an overt (good) purpose and a covert (bad) purpose
- Viruses have only one purpose

Trojan Horse Or Not?

Yes

- Overt action = infected program's actions
- Covert action = virus' actions (infect, execute)

No

- Overt purpose = virus' actions (infect, execute)
- Covert purpose = none

Semantic, philosophical differences

- Defenses against Trojan horse also inhibit computer viruses

History

Programmers for Apple II wrote some
- Not called viruses; very experimental

Fred Cohen
- Graduate student who described them
- Teacher (Adleman) named it “computer virus”
- Tested idea on UNIX systems and UNIVAC 1108 system

Cohen’s Experiments

UNIX systems: goal was to get superuser privileges
- Max time 60m, min time 5m, average 30m
- Virus small, so no degrading of response time
- Virus tagged, so it could be removed quickly

UNIVAC 1108 system: goal was to spread
- Implemented simple security property of Bell-LaPadula
- As writing not inhibited (no *-property enforcement), viruses spread easily

First Reports

Brain (Pakistani) virus (1986)

- Written for IBM PCs
- Alters boot sectors of floppies, spreads to other floppies

MacMag Peace virus (1987)

- Written for Macintosh
- Prints “universal message of peace” on March 2, 1988 and deletes itself

More Reports

Duff’s experiments (1987)

- Small virus placed on UNIX system, spread to 46 systems in 8 days
- Wrote a Bourne shell script virus

Highland’s Lotus 1-2-3 virus (1989)

- Stored as a set of commands in a spreadsheet and loaded when spreadsheet opened
- Changed a value in a specific row, column and spread to other files

Types of Viruses

- Boot sector infectors
  - A virus that inserts itself into the boot sector of a disk
  - Executed when system boots
- Executable infectors
  - A virus that infects executable programs (eg .exe)
- Multipartite viruses
  - A virus that can infect either boot sectors or executable
  - Contains a boot sector infector and executable infector
- Memory-resident (TSR) viruses
  - A virus that stays active in memory
- Stealth viruses
  - A virus that conceals infection of files
  - Intercepts system calls
  - Example: Request for file length: return length of uninfected file
- Encrypted viruses
  - A virus that is enciphered except for a small deciphering routine
  - Uses random key; harder to detect!
- Polymorphic viruses
  - A virus that changes its form each time it inserts itself into another
  - program
  - Use different instructions with same effect (eg add/subtract/xor 0)
  - Harder than encrypted viruses
- Macro viruses
  - A virus composed of a sequence of instructions that are interpreted
  - rather than executed directly
  - Code is platform independent (eg MS Word/Excel)
  - Melissa virus (MS Word)

Boot Sector Infectors

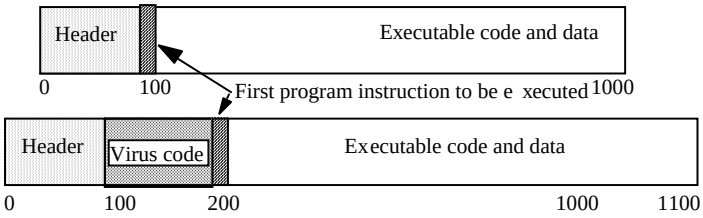
A virus that inserts itself into the boot sector of a disk

- Section of disk containing code
- Executed when system first "sees" the disk
  - Including at boot time …

Example: Brain virus

- Moves disk interrupt vector from 13H to 6DH
- Sets new interrupt vector to invoke Brain virus
- When new floppy seen, check for 1234H at location 4
  - If not there, copies itself onto disk after saving original boot block

Executable Infectors



A virus that infects executable programs

- Can infect either .EXE or .COM on PCs
- May prepend itself (as shown) or put itself anywhere, fixing up binary so it is executed at some point

Executable Infectors (*con't*)

Jerusalem (Israeli) virus

- Checks if system infected
  - If not, set up to respond to requests to execute files
- Checks date
  - If not 1987 or Friday 13th, set up to respond to clock interrupts and then run program
  - Otherwise, set destructive flag; will delete, not infect, files
- Then: check all calls asking files to be executed
  - Do nothing for COMND.COM
  - Otherwise, infect or delete
- Error: doesn't set signature when .EXE executes
  - So .EXE files continually reinfected

Multipartite Viruses

A virus that can infect either boot sectors or executables

Typically, two parts

- One part boot sector infector
- Other part executable infector

TSR Viruses

A virus that stays active in memory after the application (or bootstrapping, or disk mounting) is completed

- TSR is "Terminate and Stay Resident"

Examples: Brain, Jerusalem viruses

- Stay in memory after program or disk mount is completed

Stealth Viruses

A virus that conceals infection of files

Example: IDF virus modifies DOS service interrupt handler as follows:

- Request for file length: return length of *uninfected* file
- Request to open file: temporarily disinfect file, and reinfect on closing
- Request to load file for execution: load infected file

Encrypted Viruses

A virus that is enciphered except for a small deciphering routine

- Detecting virus by signature now much harder as most of virus is enciphered

Computer Worms

A program that copies itself from one computer to another

Spreads over a network

Morris Internet worm in 1988

- Written by Robert Morris (Cornell University student) and launched
- from MIT
- Targeted Berkeley, Sun UNIX systems
- Disabled several thousand systems in about 6 hours
- Used virus-like attack to inject instructions into running program and

Worms vs Viruses

Worms spread through network --- Viruses spread through files

Worms harm network bandwidth --- Viruses corrupt or modify files

Rabbits/Bacteria

A program that absorbs all of some class of resources

Example: for UNIX system, shell commands:

```
while true
do
   mkdir x
   chdir x
done
```

Exhausts either disk space or file allocation table (inode) space

Logic Bombs

A program that performs an action that violates the site security policy when some external event occurs

- A program that performs malicious actions when specified conditions are met:
  - presence/absence of some file
  - particular date/time
  - particular user
- When triggered typically damage system
  - modify/delete files/disks, halt machine, etc

Example: program that deletes company's payroll records when one particular record is deleted

Other Malware types

- The list goes on . . .
  - Malware for profit
    - spyware
    - adware
    - botnet
    - key-loggers
  - rootkits
  - zombies
  - backdoor

Defenses

Distinguish between data, instructions

Limit objects accessible to processes

Inhibit sharing

Detect altering of files

Detect actions beyond specifications

Analyze statistical characteristics

Trust

Trust the user to take explicit actions to limit their process' protection domain sufficiently

- That is, enforce least privilege correctly

Trust mechanisms to describe programs' expected actions sufficiently for descriptions to be applied, and to handle commands without such descriptions properly

Trust specific programs and kernel

- Problem: these are usually the first programs malicious logic attack

Countermeasures

- Best countermeasure is prevention
  - Don't allow a virus to get in
- Prevention in general is not possible
- Hence need to do one or more of:
  - detection - of viruses in infected system
  - identification - of specific infecting virus
  - removal - restoring system to clean state
- Advances in viruses and anti-virus technology go hand in hand!
- Earlier viruses were easier to detect

Anti-Virus Software

- First-generation (simple scanners)
  - Uses virus signature to identify virus
  - Detect changes in length of programs
- Second-generation (heuristic scanners)
  - Uses heuristic rules to spot viral infection
  - Uses checksum/hash of program to detect changes
- Third-generation (activity traps)
  - Memory-resident programs identify virus by actions
- Fourth-generation (full-featured protection)
  - Packages with a variety of anti-virus techniques
  - eg scanning & activity traps, access-controls

Key Points

A perplexing problem

- How do you tell what the user asked for is *not* what the user intended?

Strong typing leads to separating data, instructions

File scanners most popular anti-virus agents

- Must be updated as new viruses come out