Malicious Software

CSS322: Security and Cryptography

Sirindhorn International Institute of Technology
Thammasat University

Prepared by Steven Gordon on 29 December 2011
CSS322Y11S2L13, Steve/Courses/2011/S2/CSS322/Lectures/malicious.tex, r2070

Contents

Malicious Software

Viruses and Worms

Examples

Denial of Service Attacks

Classifying Malicious Software

Host Dependence

- Host Dependent: Code/programs are embedded in actual programs, e.g. viruses, backdoors
- Host Independent: Programs can be run separately by OS, e.g. worms, zombies

Replicating

- Non-replicating: programs usually activated by a trigger, e.g. logic bombs, backdoors
- Replicating: make copies of themselves, e.g. viruses, worms

Terminology of Malicious Software

- Virus: Attaches itself to a program and propagates copies of itself to other programs
- Worm: Program that propagates copies of itself to other computers
- Logic bomb: Triggers action when condition occurs
- Trojan horse: Program that contains unexpected additional functionality
- Backdoor (trapdoor): Program modification that allows unauthorized access to functionality
- Exploits: Code specific to a single vulnerability or set of vulnerabilities
- Downloaders: Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.

Terminology of Malicious Software

- Auto-rooter: Malicious hacker tools used to break into new machines remotely
- Kit (virus generator): Set of tools for generating new viruses automatically
- Spammer programs: Used to send large volumes of unwanted e-mail
- Flooders: Used to attack networked computer systems with a large volume of traffic to carry out a denial of service (DoS) attack
- Keyloggers: Captures keystrokes on a compromised system
- Rootkit: Set of hacker tools used after attacker has broken into a computer system and gained root-level access
- Zombie Program: activated on an infected machine that is activated to launch attacks on other machines

Contents

Malicious Software

Viruses and Worms

Examples

Denial of Service Attacks

Nature of Viruses

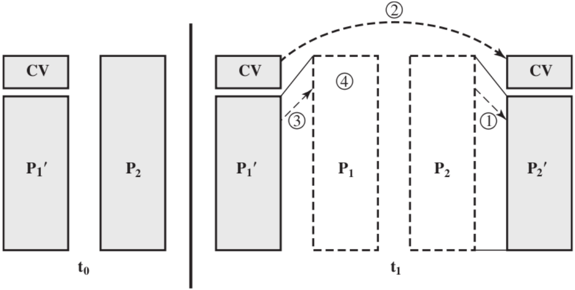
- A virus is piece of software that "infects" programs and copies itself to other programs
- The phases of a virus are:
  1. Dormant: virus is idle; will be activated by some event (like logic bomb)
  2. Propagation: virus copies itself into other programs or areas of operating system
  3. Triggering: virus is activated to perform some function; similar triggers to logic bombs, but also number of times virus copied
  4. Execution: function is performed, either harmless (display a message) or malicious (delete or modify files)
- Most viruses are specific to operating systems and/or hardware platforms

A Simple Virus

```
program V :=
{goto main;
   1234567;
   subroutine infect-executable :=
      {loop:
      file := get-random-executable-file;
      if (first-line-of-file = 1234567)
         then goto loop
      else
         prepend V to file; }
   subroutine do-damage :=
      {whatever damage is to be done}
   subroutine trigger-pulled :=
      {return true if some condition holds}
main: main-program :=
   {infect-executable;
   if trigger-pulled
      then do-damage;
   goto next;}
next:
}
```

Compression Virus

- The simple virus can be detected because file length is different from original program
- This detection can be avoided using compression
- Assume program P1 is infected with virus CV
  1. For each uninfected file P2, the virus compresses P2 to produce P2
  2. Virus CV is pre-pended to P2 (so resulting size is same as P2)
  3. P1 is uncompressed and (4) executed



A Compression Virus

```
program CV :=
{  goto main;
   01234567;
   subroutine infect-executable :=
      {loop:
         file := get-random-executable-file;
         if (first-line-of-file = 01234567)
            then goto loop;
      (1) compress file;
      (2) prepend CV to file;
   }
main: main-program :=
{  if ask-permission
      then infect-executable;
   (3) uncompress rest-of-file;
   (4) run uncompressed file;}
}
```

Types of Viruses

- Parasitic Virus: virus attaches to executable file and copies itself to other executables that it can find
- Memory-resident virus: stored in main memory as part of current program executing; infects other programs that execute
- Boot sector virus: stored in boot sector of hard or floppy disk; spreads when system boots from disk (a popular method before computer networks were widespread)
- Polymorphic virus: changes (mutates) with each copy, so harder to detect based on signatures; e.g. Add extra, redundant code; re-order code
- Metamorphic virus: change appearance as well as behaviour; Very hard to detect

Worms

- Software that replicates itself and sends copies to other computers
  - And copies on new computers repeat the process (copy and send)
  - May perform some function as well (e.g. delete files)
- Is an email virus a virus or worm or both?
  - Email virus requires users to propagate
  - Worms propagate by themselves (without user intervention)
  - Virus infects other software
- Worms use network connections to propagate:
  - Email software, e.g. Simple Mail Transfer Protocol (SMTP)
  - Remote execution, Remote Procedure Call, sockets
  - Remote login, e.g. telnet, rlogin, rsh, ...
- Three main steps of worm:
  1. Search for other systems to infect
  2. Connect to a remote system
  3. Copy itself to remote system and cause the copy to execute

Distributions of Viruses and Worms

- Assume infect 4 new computers every hour
- How long to infect every personal computer in world?

Contents

Malicious Software

Viruses and Worms

Examples

Denial of Service Attacks

Examples

- Macro Virus
- Email Virus
- Melissa Virus
- Code Red Worm
- I Love You Worm

Contents

Malicious Software

Viruses and Worms

Examples

Denial of Service Attacks

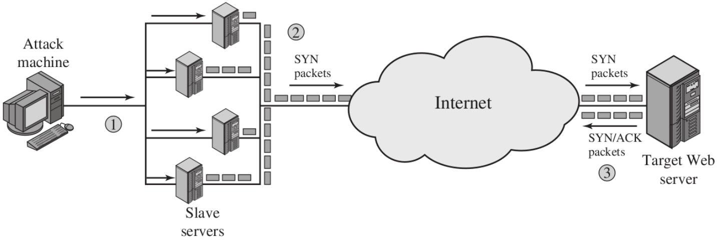
Distributed Denial of Service Attacks

- Security Service: Availability
  - A network or computer system should be available to users for the normal intended purpose
- Denial of Service (DoS) Attack:
  - Aim to prevent real users from using the system
  - Comes from a single computer towards a single computer or network
- Distributed DoS Attack:
  - DoS from multiple (often many) computers to single computer or network
  - Very hard to prevent and also sometimes hard to detect early
  - Typically involves an attacker taking control of many hosts on Internet, and these infected hosts perform the attacks on a single target

TCP SYN Flooding Attack

- Attacker takes control of many slave hosts
- Each slave sends TCP SYN segments to a single (target) host (e.g. web server)
  - Each TCP SYN has fake/incorrect source IP addresses
  - The target server responds to each TCP SYN with a SYN+ACK (if accepted) or a RST (if not accepted)
  - Target server also creates a data structure in memory for each accepted connection, as it is waiting for the final ACK to come back
  - As a result, target becomes overflowed with processing many SYNs, as well as storing data about each connection in memory
  - Target cannot process any legitimate connection requests
- Prevention: difficult; filter packets at routers; SYN cookies

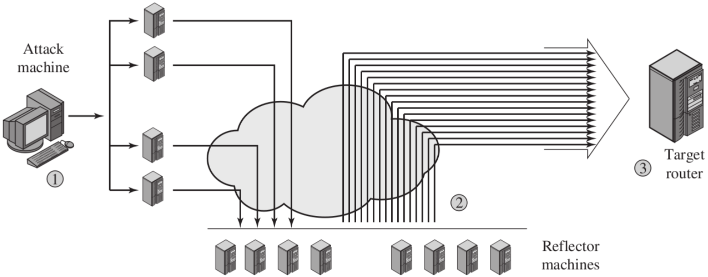
TCP SYN Flooding Attack



ICMP Attack

- Attacker takes control of many slave hosts
- Each slave sends ICMP ECHO messages (Pings) to set of reflector hosts
  - Reflector hosts are usually random hosts that are not infected or under control of attacker
  - ICMP ECHO from slaves has a spoofed source IP address—it is set to the target's IP address
  - Every reflector host sends a ICMP response to the source, that is to the target
  - Target's router is overloaded with ICMP packets, leaving no network resources for the target (or other nodes on its network)
- Prevention: Not respond to ICMP messages; routers drop ICMP messages

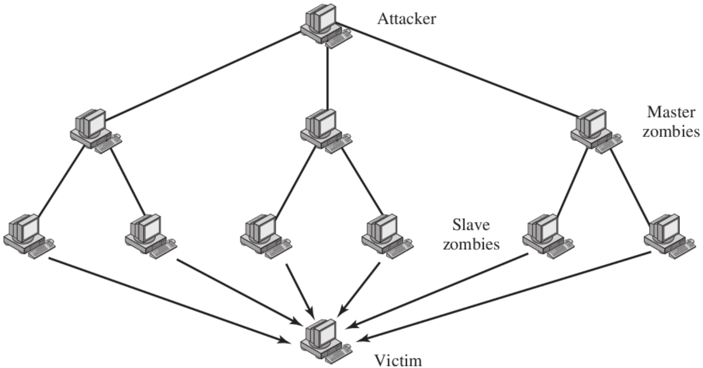
ICMP Attack



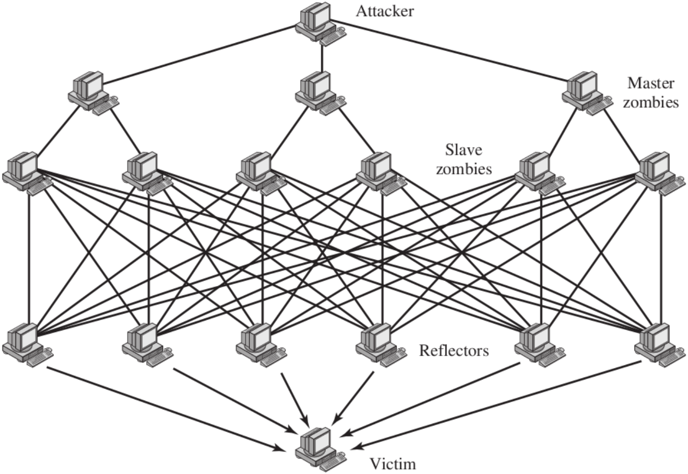
Classifying DDoS Attacks

- Resource consumed:
  - Internal host resources such as CPU and memory, e.g. TCP SYN flood
  - Data transmission capability of network, e.g. ICMP Ping flood
- Source of attacks
  - Direct DDoS Attack
    - Attacker controls slaves (or hierarchy of slaves), and the slaves attack the target directly
  - Reflector DDoS Attack
    - Attacker controls slaves (or hierarchy of slaves), and the slaves send data to reflectors which then forward to the target
    - Reflectors are not under control of attacker
    - Easier to involve more hosts than direct DDoS and hence send more traffic and create more damage
    - Harder to trace back to original attacker if reflectors are used

Direct DDoS Attack



Reflector DDoS Attack



Constructing Attack Network

- Attacker must get many slave hosts under its control
- Infect the hosts with zombie software

1. Create software that will perform the attacks. This should:
    - Be able to run on different hardware architectures and OSes
    - Hide, that is not be noticeable to the normal user of the zombie host
    - Be able to be contacted by attacker to trigger an attack
2. Identify vulnerability (bug) in large number of systems, in order to install the zombie software
3. Locate vulnerable machines, using scanning:
    - Attacker finds vulnerable machines and infects with zombie software
    - Then the zombie software searches for vulnerable machines and infects with zombie software
    - And so on, until a large distributed network of slaves is constructed

Preventing DDoS Attacks

- Prevention
  - Allocate backup resources and modify protocols that are less vulnerable to attacks
  - Aim is to still be able to provide some service when under DDoS attack
- Detection
  - Aim to quickly detect an attack and respond (minimise the impact of the attack)
  - Detection involves looking for suspicious patters of traffic
- Response
  - Aim to identify attackers so can apply technical or legal measures to prevent
  - Cannot prevent current attack; but may prevent future attacks