Slides adapted from "Foundations of Security: What Every Programmer Needs To Know" by Neil Daswani, Christoph Kern, and Anita Kesavan (ISBN 1590597842; http://www.foundationsofsecurity.com). Except as otherwise noted, the content of this presentation is licensed under the Creative Commons 3.0 License.
Agenda

- Worms spreading across Internet through vulnerabilities in software

- History of Worms
  - Morris Worm
  - Code Red
  - Nimda
  - Blaster & SQL Slammer

- Rootkits, Botnets, Spyware, and more Malware
What Is a Worm?

- **Virus**: program that copies itself into other programs
  - Could be transferred through infected disks
  - Rate dependent on human use

- **Worm**: a virus that uses the network to copy itself onto other computers

- Worms propagate faster than viruses
  - Large # of computers to infect
  - Connecting is fast (milliseconds)
An Abridged History of Worms

- Examples of how worms affect operation of entire Internet

- First Worm: Morris Worm (1988)
- Code Red (2001)
- Nimda (2001)
- Blaster (2003)
- SQL Slammer (2003)
Morris Worm: What It Did

- Damage: 6000 computers in just few hours
- Extensive network traffic by worm propagating
- What: just copied itself; didn’t touch data
- Exploited and used:
  - buffer overflow in fingerd (UNIX)
  - sendmail debug mode (execute arbitrary commands such as copying worm to another machine)
  - dictionary of 432 frequently used passwords to login and remotely execute commands via rexec, rsh
The Morris Worm: What We Learned

- Diversity is good: Homogenity of OSes on network -> attacker can exploit vulnerabilities common to most machines
- Large programs more vulnerable to attack
  - sendmail was large, more bug-prone
  - fingerd was small, but still buggy
- Limiting features limits holes: sendmail debug feature should have been turned off
- Users should choose good passwords: dictionary attack would have been harder
The Creation of CERT

- Computer Emergency Response Team (CERT) created due to damage and disruption caused by Morris worm
- Has become a leading center on worm activity and software vulnerability announcements
- Raises awareness about cyber-security
The Code Red Worm (1)

- Exploited
  - Microsoft IIS web server buffer overflow
  - "indexing server" feature: randomly scanned IP addresses to connect to other IIS servers
- Spread rapidly: > 2,000 hosts/min
- Evaded automated detection
  - Detectable more easily by humans than scanners
  - Resident only in memory, no disk writes
- Defaced home page of infected server
The Code Red Worm (2)

Web server defaced by Code Red
The Nimda Worm

- Propagation vector: method by which worm spreads to another machine
- Payload: data worm carries as it travels

Spread Rapidly, made Code Red worse
- Used multiple propagation vectors
- Spread from server to server (as in Code Red)
- But also from server to client (browser downloading infected file also became infected)
- Infected client sent e-mails with worm code as payload
Blaster Worm

- Exploited
  - buffer overflow in Microsoft OS: attacked Distributed Component Object Model service
  - Patch deployed but many users didn’t download it

- Caused infected machine to shut down

- Issued a DDoS attack against Windows Update website to prevent users from getting the patch
Blaster Worm

System shutdown
Dialog by
Blaster Worm

System Shutdown

This system is shutting down. Please save all work in progress and log off. Any unsaved changes will be lost. This shutdown was initiated by NT AUTHORITY\SYSTEM

Time before shutdown: 00:00:28

Message
Windows must now restart because the Remote Procedure Call (RPC) service terminated unexpectedly
SQL Slammer Worm

- Exploited another buffer overflow
  - Took a single 376-byte UDP packet
  - UDP connectionless -> spread quickly
  - Infected 75,000, 90% w/in 10 mins.

- Attacked Microsoft SQL Server DB App
- Disabled server, scanned random IPs to infect

- Impact
  - Excessive traffic due to the worm propagating caused outages in 13,000 BofA ATMs
  - Airlines were cancelled & delayed
More Malware

- *Rootkits*: imposter OS tools used by attacker to hide his tracks

- *Botnets*: network of software robots attacker uses to control many machines at once to launch attacks (e.g. DDoS through packet flooding, click fraud)

- *Spyware*: software that monitors activity of a system or its users without their consent
More Malware

- **Keyloggers**: spyware that monitors user keyboard or mouse input, used to steal usernames, passwords, credit card #s, etc…
- **Trojan Horses**: software performs additional or different functions than advertised
- **Adware**: shows ads to users w/o their consent
- **Clickbots**: bot that clicks on ads, leads to click fraud (against cost-per-click or CPC ad models)
Distributing Malware

- Most malware distribution through *drive-by downloads* (i.e. automatic installation of binary when visiting website)
  - Uses pull-based model (e.g. links)
  - Maximizes exposure by getting as many links as possible to malware distribution site

- Search engines such as Google mark pages as potentially malicious to prevent

---

Clickbot.A Botnet\(^2\) (1)

- Over 100,000 machines, HTTP-based botmaster
- Conducted *low-noise click fraud* against *syndicated search engines*
  - Syndication: get feeds of ad impressions
  - Sub-Syndication: partner with a syndicated engine
  - All get a share of revenue from click
- Only 7/24 anti-virus scanners detected it in 5/06
- IE browser helper object (BHO)
  - Capable of accessing entire DOM of web pages
  - Written in PHP with MySQL backend
Clickbot.A Botnet\(^1\) (2)

- Used *doorway-sites* (w/ links for bots to click) posing as sub-syndicated search engines

  ![Diagram showing the flow of money and control in the Clickbot.A botnet]

- Fine-grained control for botmaster
  - Low noise: set maxclicks bots could do to 20
  - Used redirectors & several layers below major search engine (harder to detect/track)

Summary

- Worms propagate rapidly, exploit common vulnerabilities and cause widespread damage

Prevention
- Eliminate Buffer Overflows (Programmers)
- Don’t open email attachments (Users, SAs)
- Disable unnecessary functionality (Users, SAs)
- Patch systems regularly (SAs)

Detection
- Update scanners with latest definitions
- Use auto-updating scanners when possible
- Employ programs such as Tripwire (SAs)