Viruses & Malicious Code

Terminology

- Virus
- Worm
- Trojan Horse
- Logic Bomb
- Time Bomb
- Trapdoor
- Backdoor

What is a virus?

- A computer program file capable of attaching to (infecting) disks or files and replicating itself repeatedly.
- Some viruses….
  - attach to files so when the infected file executes, the virus also executes.
  - sit in a computer's memory and infect files as the computer opens, modifies or creates the files.
  - display symptoms (and some don't)
  - damage files and computer systems, (and some don't)
- A non-damaging virus is still a virus.
- Viruses NEED a <u>host</u> to live – they aren't standalone

Two Main Types of Viruses

- Viruses can either be:
  - Transient
    - A virus runs when its attached program executes and terminates when its attached program ends.
    - Note that during its execution, the transient virus may have spread to other programs.
  - Resident
    - A virus locates itself in memory so that it can remain active, or be activated, even after its attached program ends.

What is a worm?

- A program that replicates without "infecting" other programs with a copy of itself.
- Worms spread either as e-mail attachments or by being network-aware, and by using network specific calls to get form one place to another.
- Example:
  - Worms can spread by using network calls to find shared, writable drives across the network, to which they copy themselves

What is a worm?

- Worms spread copies of themselves as stand-alone programs.
- Unlike viruses, worms do not infect other computer program files.
- Worms can create copies on the same computer, or can send the copies to other computers via a network.

Worms & Viruses

- People get into stupid arguments over whether something is a “worm” or a “virus”
  - Is the Internet a host program?
    - See Mark W. Eichin and Jon A. Rochlis, With *Microscope and Tweezers: An Analysis of the Internet Virus of November 1988*
    - Is Outlook a host program for an email?
- Similarly, for worms/viruses/Trojans
  - If the user must open it (e.g., ILoveYou) is it self-replicating?

History of Trojan Horse

- Greeks and Trojans at war
- Greeks attacking Troy, bombarded city for 10 years, but couldn’t get through city walls.
- Pretended to leave, left big wooden horse as gift
- Trojans brought horse into city (had to tear down part of wall to do this), got silly drunk celebrating victory.
- Greeks jumped out, killed sentries, and let in Greek army.

Modern Trojan Horses

- A Trojan horse program is a malicious program that pretends to be a benign application
- User runs program that looks harmless
  - Program pretends to be “cool, dancing bears”, also erases your hard drive
- Most attacks today are Trojan Horses
  - ILoveYou, Melissa, etc.
- Rely on modern humans being as dumb as mythical Trojans
  - No matter how good your city/fire walls are, they don’t do any good if you can’t stop users from running random code

What is a Trojan Horse?

- Trojan horse programs purposefully do something a user does not expect.
- Trojans are not viruses since they do not replicate….
- But…. Trojan horse programs can be just as destructive.

Logic Bombs & Time Bombs

- A class of malicious code that lies dormant until a specified event happens or until a condition is true, and then the code is activated.
- Logic Bomb
  - Malicious code that “detonates” or goes off when a specified condition occurs.
- Time Bomb
  - A logic bomb whose trigger is a time or date.

Backdoors & Trapdoors

- A feature in a program by which someone can access the program other than by the obvious, direct call, perhaps with special privileges.
- Example:
  - Automated bank teller program allows anyone entering the number 990099 on the keypad to process the log of everyone’s transactions at that machine

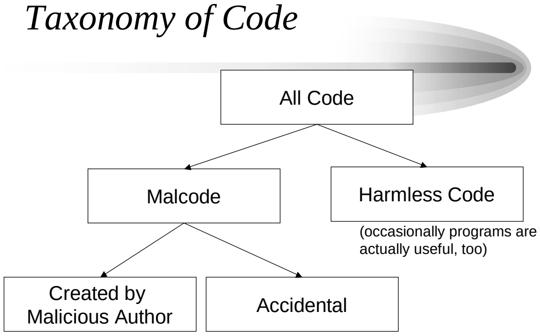
Terminology

- Virus
- Worm
- Trojan Horse
- Logic Bomb
- Time Bomb
- Trapdoor
- Backdoor

Attacking Malicious Code

- “Malicious Code” is a bad name
  - Code has no intent
  - Programmer’s intent doesn’t matter, what the code **<u>does</u>** matters
    - As networks get more programmable, accidentally harmful code will become common
- We’ll use “malcode” (mal = bad)
  - It’s not a great name either...

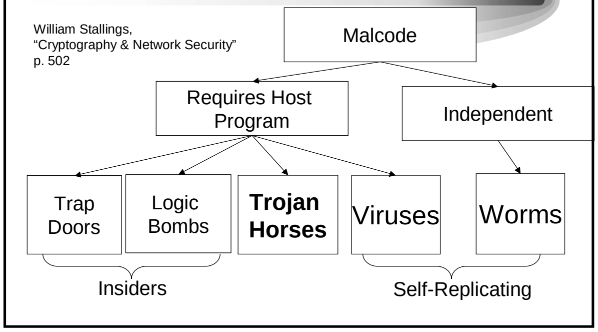
Taxonomy of Code



(occasionally programs are actually useful, too)

Taxonomy of Malcode

William Stallings,
“Cryptography & Network Security”
p. 502



A Touch of History

- 1981
  - First virus “in the wild” spread on Apple II floppy disks (which contained the operating system) and reputed to have spread from Texas A&M. It died out by itself and was largely forgotten.
- 1983
  - Fred Cohen & Len Adleman created the first documented experimental computer virus as an experiment for a computer security seminar.
  - Term “Virus” was conceived

A Touch of History

- 1986
  - **Brain** - first non-experimental PC boot sector virus created by 2 Pakistani brothers. It spread widely on MS-DOS PC systems.

  - ***PC-Write Trojan*** - first PC-based Trojan was released in the form of the popular shareware program *PC-Write*.

A Touch of History

- 1987 - first file viruses started to appear.
  - **Lehigh virus** - first to infect COMMAND.COM
  - **Suriv-02** - The first EXE infector(Suriv = Virus backward). Later evolved into **Jerusalem** virus.
  - **IBM Christmas Worm** - A fast-spreading (500,000 replications per hour) worm hit IBM mainframes by e-mail

A Touch of History

- 1988
  - **MacMag** - First Macintosh virus. Attached Hypercard stack.
  - **Scores** virus - First major Macintosh outbreak.
  - **The Internet Worm** (Robert Morris' creation) causes the first Internet crisis and shut down many computers.

A Touch of History

- 1988
  - **MacMag** - First Macintosh virus; it attached itself to Hypercard stack.
  - **Scores** virus - First major Macintosh outbreak.
  - **The Internet Worm** (Robert Morris' creation) causes the first Internet crisis and shut down many computers.

A Touch of History

- **1989**
  - **AIDS Trojan Horse -** sent out under the guise of an AIDS information program. When run it encrypted the user's hard drive and demanded payment for the decryption key.
- **1991**
  - **Tequila -** first polymorphic virus which changed itself in an attempt to avoid detection.

A Touch of History

- **1992**
  - **Michelangelo** - was the first media darling. A wordwide alert went out with claims of massive damage predicted. Actually, little happened.
  - **Dark Avenger Mutation Engine (DAME)** became the first toolkit that could be used to turn any virus into a polymorphic virus.
  - **Virus Creation Laboratory (VCL)** became the first actual virus creation kit. It had pull-down menus and selectable payloads.

A Touch of History

- **1995**
  - **Concept** - First macro virus to attack Word.
- **1996**
  - **Laroux** - First Excel macro virus.
  - **Staog** is the first Linux virus
- **1998**
  - **Strange Brew** - First Java virus.
  - **Back Orifice** is the first Trojan designed to be a remote administration tool that allows others to take over a remote computer via the Internet.
  - Access macro viruses begin to appear

A Touch of History

- **1999**
  - – **Melissa** - first combination Word macro virus and worm to use the Outlook and Outlook Express address book to send itself to others via E-mail.
  - – **Tristate** - first multi-program macro virus; it infects Word, Excel, and PowerPoint files.
  - – **Bubbleboy** - first worm that would activate when a user simply opened an E-mail message in Microsoft Outlook (or previewed the message in Outlook Express). No attachment necessary. Bubbleboy did no damage; it was a proof of concept; *(Seinfeld)*
  - – **Kak** - spread widely using this technique.

A Touch of History

- **1999**
  - – **Happy99** – one of the first worms to patch winsock.ddl to watch for outgoing e-mail addresses to send itself to. Displayed Fireworks graphics occasionally.
  - – **CIH** - Time bomb that would overwrite a portion of victim's hard drive and attempt to overwrite the flash ROM of the PC (Devastating for laptops….must return to factory). US and Europe not affected to badly but Asian computers were hit hard on April 26 trigger date.

A Touch of History

- **2000**
  - – **Love Letter** worm - fastest-spreading worm (to that time); shutting down E-mail systems around the world.
  - – **Liberty Trojan Horse** - first Trojan developed for the Palm PDA.
  - – **Pirus** - a proof of concept, written in the PHP scripting language, which attempts to add itself to HTML or PHP files.

A Touch of History

- **2001**
  - – **Gnuman** (Mandragore) – a worm cloaked itself from the Gnutella file-sharing system (the first to specifically attack a peer-to-peer communications system) and pretended to be an MP3 file to download.
  - – **Winux** - a proof of concept virus designed to infect both Windows and Linux (and cross between them) was released.

A Touch of History

- **2001**
  - – **PeachyPDF**-A worm became the first to spread using Adobe's PDF software. Only the full version, not the free PDF reader, was capable of spreading the worm so it did not go far.
  - – While not new in concept, a couple of worms created a fair amount of havoc during the year: **Sircam** (July), **CodeRed** (July & August), and **BadTrans** (November & December).

A Touch of History

- 2002
  - – **LFM-926** - the first virus to infect Shockwave Flash (.SWF) files. It was named for the message it displays while it's infecting: "Loading.Flash.Movie...". It drops a Debug script that produces a .COM file which infects other .SWF files.
  - – **Donut** - first worm directed at .NET services.
  - – **Sharp-A** - first native .NET worm written in C#, was announced. Sharp-A was also unique in that it was one of the few malware programs reportedly written by a woman.

A Touch of History

- 2002
  - **SQLSpider -** a Javascript worm that attacked installations running Microsoft SQL Server (and programs that use SQL Server technology).
  - **Benjamin** - uses the KaZaa peer-to-peer network to spread.
  - **Perrun** - a proof-of-concept virus which attached itself to JPEG image files.
    - Note: despite the hype, JPEG files are still safe as you must have a stripper program running on your system in order to strip the virus file off the image file.
  - **Scalper** – a worm that sets up a flood net to attack FreeBSD/Apache Web servers.

Malcode Varieties

- Viruses
  - Virus Factories/ Code Generators
  - Program Infector Viruses
    - Polymorphic Viruses, Stealth, Retroviruses, Memory-Resident Viruses, Nonresident Infectors, Prependers, Appenders, Overwriters, Midfile Infectors, and Multipartite Viruses
  - Boot Viruses
- Worms
- Non-Replicating Threats
- Hoaxes

The Race

- There is always a competition between virus writers, who want to find some way to evade detection, and the AV folks, who change their software to detect it.

vXers vs. AVers

Virus Factories/Code Generators

- Code generators make it easy for untrained & unskilled people to write viruses
- Most popular
  - VCL – Virus Creation Laboratory code generator (1993) -
    - Can generate source assembler files of the viruses, the OBJ modules and infected master files. VCL contains the standard pop-up menu interface. By using VCL menus it is possible to choose the virus type, enable or disable self encryption, anti-debugging code, internal text strings.
  - PSMPC – Phalcon/Skism Mass-Produced Code generator
  - VX Heavens: **Virus** Creation Tools

Virus Factories/Code Generators

Two major flaws

- Serious virus writers write their own code – it's not COOL to modify someone else's.
- There is so much common code that AVers find all possible generations using just a few signatures
  - 15,000 samples produced by PSMPC
  - AV companies detected all versions within a week

Program Infector Viruses

- Viruses that infect executable files such as those that end in .exe, .com, or .sys
- Categorized by the:
  - Means used to Evade Detection
    - polymorphic, stealth, retro, companion, and path-companion
  - Infection Method & Location
    - memory-resident, nonresident, prepending, appending, and midfile.

Polymorphic Viruses

- Viruses that use encryption to evade signature scanners
- Two methods:
  - – encrypt the main code of the virus with nonconstant key with random sets of decryption commands, or
  - – by executable virus code changing.
    - mostly used by macro viruses, which randomly change the names of their variables, insert empty lines or change their code in some other way while making copies of themselves. Therefore the operating algorithm of a virus remains unchanged, but the virus code changes virtually completely from one infection to another.

Polymorphic Viruses

- Dark Avenger created the first polymorphic virus writing kit called DAME (Dark Avenger’s Mutation Engine).
  - – An .obj file that can be linked to any virus to create a polymorphism
- Also TPE (Trident Polymorphic Engine) and DSME (Dark Slayer’s Mutation Engine)

Stealth Viruses

- Developed to evade check-summing in AV programs.
- Check summers monitor each program on the hard drive.
  - If no change => Probably no virus active
  - If change(s) => a virus probably on system
- Virus removes itself, allows check sum to be computed, and then reinfects orig file

Stealth Viruses

- Virus follows AV program thru entire disk check and in one pass, it infects every program on hard drive
- VERY effective spreaders & hard to remove
- Fortunately, stealth viruses are hard to write and therefore not widely disseminated.

Retroviruses

- A virus that “fights back” by detecting popular AV programs and rendering them ineffective.
- vXers reverse engineer AV programs to look for ways to deactivate the AV program
  - – Example: Sometimes AV programs need to remove themselves from memory so they can be upgraded. Once this is detected, vXers make the same call for removal.

Resident & Non-resident Viruses

- Memory resident viruses
  - – Once they run, they don’t terminate
  - – They usually detach themselves from a host program but eventhough the host terminates, they stay active in memory looking for files to infect
- Non-resident Infectors
  - – They run when the host program is run, but once they infect one (or more) files, they terminate.

Prependers, Appenders, Overwriters, & Midfile Infectors

- All viruses (except overwriters) try to maintain functionality of orig program. Use JUMP codes to branch from virus to back to original code
  - Prependers – place viral code at beginning of victim file
  - Appenders – place viral code at end of victim file
  - Overwriters – overwrite all or part of victim file – thus, easy to detect
  - Midfile Infectors – move some code from middle of victim file & write virus in middle; hard to do

See Figure 5.2 – 5.4, pp182-3, Pfleeger

Companion Viruses

- Companion Viruses
  - Don’t touch victim file at all; position virus so that it is found and run first
    - Example: .com files are run before .exe files. If you want to infect hello.exe file, call virus “hello.com” and run hello.exe file after.
- Path Companions
  - Find a directory closer to the beginning of the search path and place virus (with same name as a real program) in that directory.

Multipartite Viruses

- Viruses that infect more than one category of targets to give the virus a better chance of spreading.
- Example
  - Infect both the boot record and program files. User gets infected if he/she boots from an infected disk or runs an infected program.

Non-replicating Threats

- No need to have malicious code replicate itself, the Internet **is** a replicator!!!
- RATs (Remote Access Trojans) posted to Usenet often disguised with two file extensions (you only see one):

  HardCore.avi.scr, FunnyBlooper.mpg.exe
- Often time victim runs file and nothing seems to happen – usually concluding that file was corrupted in the download.

HOAXES

- Rather than write a virus, just tell people you did and tell them to warn their friends.
- **Good Times**: (1994)
  - Not first hoax but very successful and became template for other hoaxes that followed
  - Warning: “Delete all e-mail with submect line of “Good Times” because it is a new, undetectable, and deadly virus that puts the CPU into an nth degree binary loop which causes the processor to overheat and burn itself out.”
  - vXers eventually wrote a real virus called “Good Times” expecting it to be ignored

Hoaxes

- Signs of Hoaxes
  - “Virus is undetectable”
  - “Tell everyone you know”
  - Experts recognize the threat as not possible
- How to be certain
  - Visit an AV vendor’s website
  - Check sites like http://www.truthorfiction.com/

Differences between Morris Worm (1988) and Melissa/ILoveYou (1999)

Vulnerabilities Exploited

- Morris Worm:
  - Buffer overflow: fingerd uses gets
  - sendmail debug mode
  - Weak Unix passwords
- Melissa:
  - Word enables macros by default, no limitations on macro behavior
- ILoveYou:
  - Dumb people will run code attached to email

Morris Worm Objectives

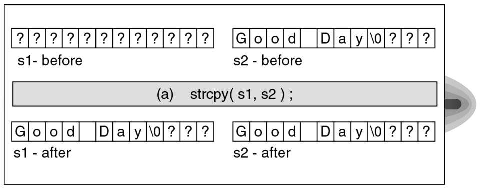
- Determined where it could spread
  - Find user accounts on target machine by guessing passwords
  - Exploit buffer overflow bug in finger program
  - Use trapdoor in sendmail mail handler – debug mode: executes received command string as root instead of the destination address
- Spread its infection
  - The Worm's bootstrap loader was sent and compiled on the target machine.
  - Bootstrap would then request rest of worm

Morris Worm Objectives

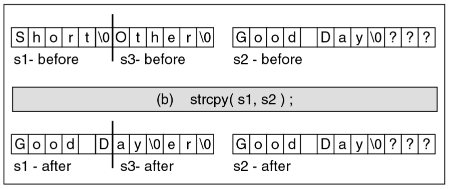
- Remained undiscovered and undiscoverable
  - If any problems occurred during spread, it removed all traces of itself.
  - Once full code was received and loaded into memory, it encrypted it, and deleted original copies from disk
  - Worm periodically changed its process identifier

What is a buffer overflow?

- Write unexpected memory area by overflowing buffer
- The most famous hacking technique
  - About ½ of recent vulnerabilities are buffer overflows
- For almost all cases, buffer overflow means stack buffer overflow, not heap overflow
- Overflows attack three areas of security:
  - Availability – run denial of service attack
  - Integrity – run code to modify data
  - Confidentiality – code reas sensitive information

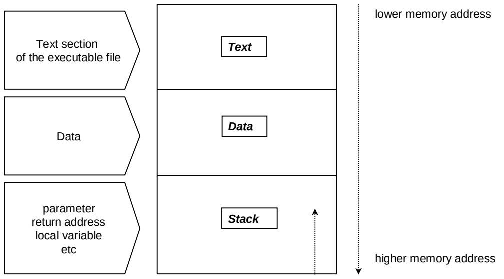


COPYING STRINGS

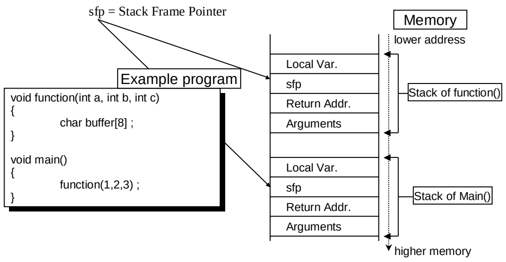


COPYING LONG STRINGS

Memory Structure

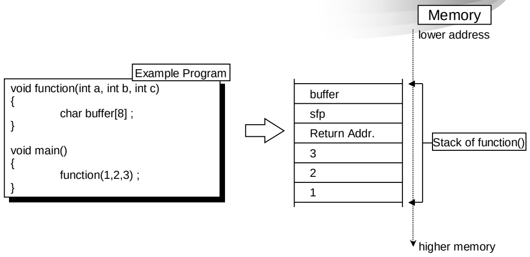


Stack Overflow Example



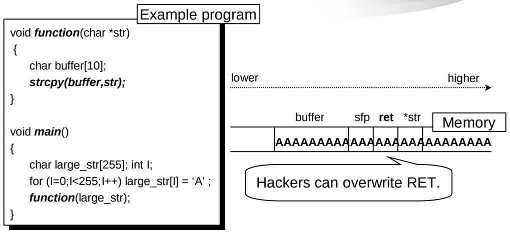
```
void function(int a, int b, int c)
{
        char buffer[8] ;
}
void main()
{
        function(1,2,3) ;
}
```

Stack Overflow Example



```
void function(int a, int b, int c)
{
        char buffer[8] ;
}
void main()
{
        function(1,2,3) ;
}
```

Stack Overflow Example



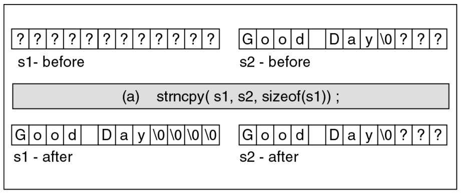
```
void function(char *str)
 {
     char buffer[10];
     strcpy(buffer,str);
}

void main()
{
     char large_str[255]; int I;
     for (I=0;I<255;I++) large_str[I] = ‘A’ ;
     function(large_str);
}
```

Stack Overflow

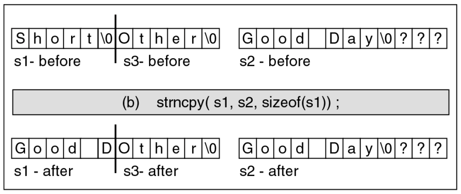
- Vulnerable program
  - Doesn’t check buffer boundary
  - Executes with root permission
- Exploit the program
  - Put instructions into memory
    - Ex) execute /bin/sh
  - Change return address to the instructions

Preventing Buffer Overflows

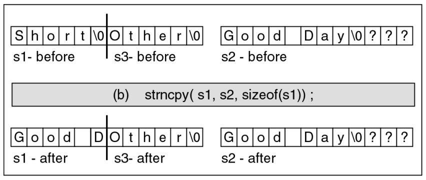
- Use run-time checks on all memory references
  - Safe languages (Java, Eiffel, etc.)
  - Safe libraries for C (don’t use gets, strcpy, etc.)
- Use Compiler Checks
  - Compile with /s option => stack check
- Static analysis
  - Check binary or source code



COPYING STRINGS



COPYING LONG STRINGS



COPYING LONG STRINGS

```
strncpy(s1,s2,sizeof(s1)-1);
*(s1 + (sizeof(s1)-1)) = ‘\0’;
```

Morris Worm

- Damage
  - – Infected ~6000 computers (10% of Internet)
- Responses
  - – Disconnect from network
  - – Disorganized
    - Anonymous message (probably from Robert Morris) explaining how to disable virus was not noticed or distributed
  - – DARPA established CERT

Morris Worm

- Lessons Learned
  - – There’s an inherent danger in running the same code in many places; diversity is key; Running a popular app makes you a target for malcode
  - – Big programs contain many bugs and therefore should NOT run with high privileges (sendmail)
  - – Poor password selection can be disastrous

Melissa Virus (1999)

- Written in only 107 lines of VB code in the form of a Word macro
- First known appearance on newsgroup alt.sex
- When triggered, creates a Word doc and sends it as an attachment to the first 50 names in a user’s Outlook Express mailbox
- Used a keyword in the registry to see if the computer was already infected

Melissa Virus (1999)

- How & Why it Worked
  - – People received mail from someone they knew
  - – Lots of people use Word and Outlook Express
    - Other e-mail apps (pine, Eudora, etc) immune to Melissa
  - – No separation of applications
    - Why does a Word macro have enough privledge to construct and send e-mail messages?

Melissa Virus – Lessons Learned

- When possible, avoid most popular type of program
- Heed warnings in pop-up boxes re enabling macros
- No longer safe to trust something just because you trust the sender.

I LOVE YOU Virus (1999)

- Message with subject line: “I LOVE YOU” contained a VB attachment.
- When triggered, infects computer and sends it as attachment to the 500 people in a user’s Outlook or Outlook Express mailbox
- Reinstalls itself on a reboot
- Copied itself to:
  - files with the same name but with .vbs extension added to them
  - References in the Registry

LoveLetter.VBS

- This 328-line program caused ~$10B in damage
- How much work and smarts was required?

Main Loop

```
rem  barok -loveletter(vbe) <i hate go to school>
rem by: spyder  /  ispyder@mail.com  /
  @GRAMMERSoft Group  / Manila,Philippines
On Error Resume Next
...
 wscr.RegWrite "...Scripting Host\
     Settings\Timeout", 0
sub main()
  ...
  Set c = fso.GetFile(WScript.ScriptFullName)
  c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
  ...
  spreadtoemail()
  ...
end sub
```

Smart people would convey more interesting message.

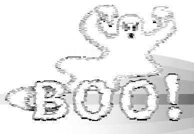
Smart virus writers don’t include their contact information.

spreadtoemail (edited to fit)

```
sub spreadtoemail()
   for ctrlists=1 to mapi.AddressLists.Count
     set a=mapi.AddressLists(ctrlists)
     x=1
     for ctrentries=1 to a.AddressEntries.Count
        malead=a.AddressEntries(x)
        set male=out.CreateItem(0)
        male.Recipients.Add(malead)
        male.Subject = “ILOVEYOU”
        male.Body = “kindly check the attached
                     LOVELETTER coming from me.”
        male.Attachments.Add(dirsystem&
                 “\LOVE-LETTER-FOR-YOU.TXT.vbs”)
        male.Send
       x=x+1
      next
     next
end sub
```

Smart virus writers can spell “mail”.

Be Very Afraid...



- When “not so smart” people with no resources write malicious programs, and…..it costs ~$10B.
- What would happen if “smart” people with resources wrote a malicious program?

I LOVE YOU Virus (1999)

- How & Why it Worked
  - Attachment named LOVE-LETTER-FOR-YOU.TXT.vbs (double extension) – it appears that it is a text msg that would be safe to open
  - People received mail from someone they knew and hard to resist opening it….it's a love letter!!
  - It installed lots of copies of itself so very likely to stay infected

I LOVE YOU Virus – Lessons Learned

- Demonstrated how quickly a "properly" written virus/worm can be spread
- "It was like a million SCUD missiles with no payload"
- No longer safe to trust something just because you trust the sender.

Damage

- Morris Worm
  - Infected ~6000 computers (10% of Internet)
- Melissa
  - Infected 1.2 Million machines in a few hours
- I LOVE YOU
  - ~$10 Billion in damage

Outcomes

- Internet Worm (Robert Morris, Jr.)
  - Convicted in '90 under 1986 Computer Fraud & Abuse Act
  - 3 years suspended sentence (no jail time), $10,000 fine & 400 hrs community service
- Melissa (David Smith) (~$80m damages)
  - Plead guilty, Dec 1999 (second successful prosecution of virus author), sentenced 20 months
  - Hired by Rutgers as Computer Technician while awaiting sentencing
- ILoveYou ($10B damages)
  - Released without penalty, no laws in Philippines

Malcode Defenses

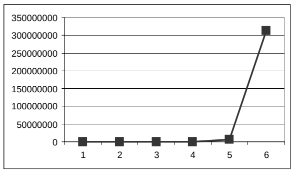
1. Prevent malcode from running
   - Virus scanners – recognize known malcode
   - Firewalls – strip malcode from incoming packets
   - Education – make users smarter
2. Limit damage it can do
   - Sandbox ("Playpen") – run malcode in protected virtual machine
   - Regular system maintenance
3. Discourage attackers
   - Legal – pass laws to penalize attackers
   - Education

Virus Scanners

- Compare code to a database of known malicious code
  - Smart authors create self-mutating viruses
- Reasonably useful in days of "sneaker" net (viruses spread on floppies)
- Reasonably useless when viruses spread as fast as email

Virus Spreading

- Read email every hour
- Everyone’s address book contains 50 people
- Infects 300M people in 6 hours!



(For more complex model, see Wang/Knight/Elder paper on manifest.)

Virus Scanners - Timeline

- Early AV programs
  - used simple scans to search for a specific text string hidden within a specific file
  - Only detected - didn’t disinfect
  - One often had to (1) know which virus they thought they had (2) then search infected file and (3) find a program that would remove it

Virus Scanners - Timeline

- Next Generation AV programs
  - As # of viruses increased, creating and distributing individual fixes was no longer feasible
  - Comprehensive scanners evolved with
    - Scanning Engine – User interface and prg that scanned files
    - Signature Files – DB of fingerprints of known viruses
- Problems (of late 80s - early 90s)
  - Lots of AV vendors emerged - no standards
  - Lots of viruses known by several names
  - Public confused

Virus Scanners - Timeline

- Two events that revolutionized AV market
  - Joe Wells (1993) assembled library of viruses from experts to coordinate reporting procedures (see www.wildlist.org); naming conventions emerged
  - National Computer Security Association (NCSA) started commercial AV testing and certification
    - Consortium of AV vendors who, for a fee, submitted their products for testing and certification.
    - Initially less than 80% of viruses in a list were detected
    - Lab provided measurable improvements in effectiveness of AV technology

Scanner Internals

- Four basic methods of operation:
  - **Detection** – looking for infections by known viruses
  - **Prevention** – monitoring changes, or attempted changes to files
  - **Heuristics** – scanning for previously unknown viruses using rule-based scan (take lots of time)
  - **Immune Systems** – monitoring the health of all systems that are connected to a centrally based reporting system. Try to heal itself when sick or call for help when it can’t. Will it work?

AV Engines

- GUI and library of commonly used functions
- Knows nothing about viruses without signature database
- Includes dozens of complex searching engines, CPU emulators

*AV Databases (*dat *file)*

- Contains fingerprints of thousands of viruses
- Database encrypted – lots of false-positives if second AV product installed on same machine
- Some apps check for updates automatically
- Also contains rule sets used in heuristic scans

Scanner Methodologies

- Detection
  - As a virus copies itself from one executable to another, it leaves bits of its code in the infected file
  - The sequence of code is referred to as the *fingerprint* or signature of that virus.
  - Some viruses have very short or no signatures

Scanner Methodologies

- Prevention
  - Find the viruses before they infect and prevent them from doing harm.
  - Uses CTC or checksums as first step
  - Some viruses interrupt a program at a specific point that is quite different from normal operation. AV scanners check if interrupt exists.

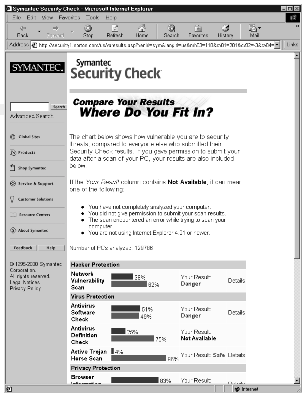
Virus Scanner Issues

- AV scanners often misconfigured
- AV scanners & signature files often out of date
  - Today users advised to update signature files WEEKLY!!
- AV scanners are largely based on what has happened before; they try to anticipate new viruses, but it's not their strong point.
- Sys admins are swamped so they install just what they have time for and call it good enough.
- Upper management often views AV scanners as high-cost/low return items and are given low priority in the budget

How vulnerable are you?

http://security1.norton.com/us/intro.asp

Scans your computer to see if you have purchased enough Symantec software. Tells you in bright red letters you should buy some.



http://security1.norton.com/us/varesults.asp?venid=sym&langid=us&mh03=110&cv01=201&cv02=-3&cv04=

Symantec Security Check

Compare Your Results Where Do You Fit In?

The chart below shows how vulnerable you are to security threats, compared to everyone else who submitted their Security Check results. If you gave permission to submit your data after a scan of your PC, your results are also included below.

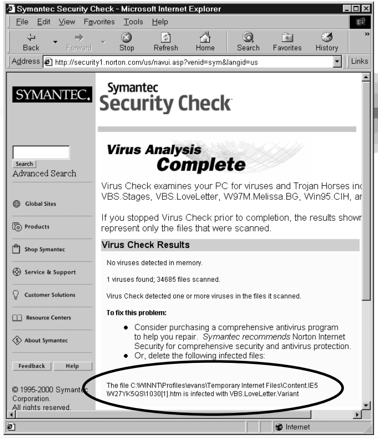
If the *Your Result* column contains **Not Available**, it can mean one of the following:

- You have not completely analyzed your computer.
- You did not give permission to submit your scan results.
- The scan encountered an error while trying to scan your computer.
- You are not using Internet Explorer 4.01 or newer.

Number of PCs analyzed: 129786

Hacker Protection			
Network Vulnerability Scan	38% 62%	Your Result: **Danger**	Details
Virus Protection			
Antivirus Software Check	51% 49%	Your Result: **Danger**	Details
Antivirus Definition Check	25% 75%	Your Result: **Not Available**	
Active Trojan Horse Scan	4% 96%	Your Result: **Safe**	Details
Privacy Protection			
Browser Information	83%	Your Result:	Details

© 1995-2000 Symantec Corporation.
All rights reserved.
Legal Notices
Privacy Policy



http://security1.norton.com/us/navui.asp?venid=sym&langid=us

Symantec Security Check

Virus Analysis Complete

Virus Check examines your PC for viruses and Trojan Horses inc VBS.Stages, VBS.LoveLetter, W97M.Melissa.BG, Win95.CIH, ar

If you stopped Virus Check prior to completion, the results showr represent only the files that were scanned.

Virus Check Results

No viruses detected in memory.

1 viruses found; 34685 files scanned.

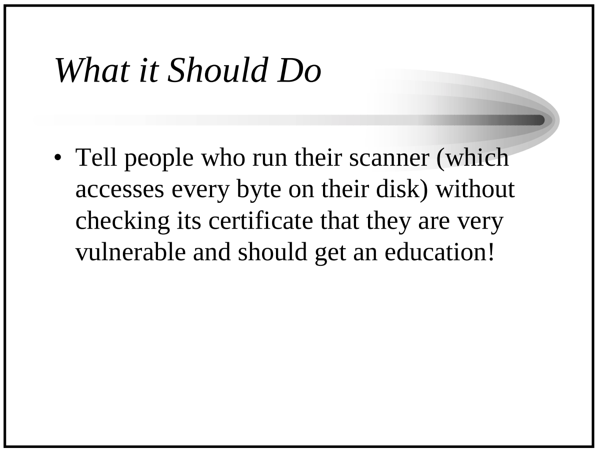
Virus Check detected one or more viruses in the files it scanned.

To fix this problem:

- Consider purchasing a comprehensive antivirus program to help you repair. *Symantec recommends* Norton Internet Security for comprehensive security and antivirus protection.
- Or, delete the following infected files:

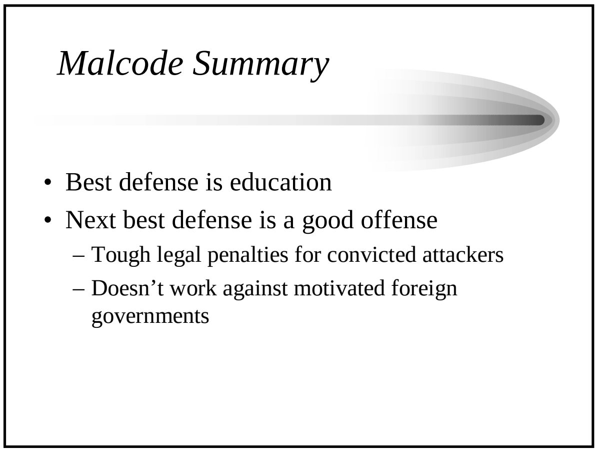
The file C:\WINNT\Profiles\evans\Temporary Internet Files\Content.IE5 W27YK5QS\1030[1].htm is infected with VBS.LoveLetter.Variant

© 1995-2000 Symantec Corporation.
All rights reserved.



What it Should Do

- Tell people who run their scanner (which accesses every byte on their disk) without checking its certificate that they are very vulnerable and should get an education!



Malcode Summary

- Best defense is education
- Next best defense is a good offense
  - Tough legal penalties for convicted attackers
  - Doesn’t work against motivated foreign governments