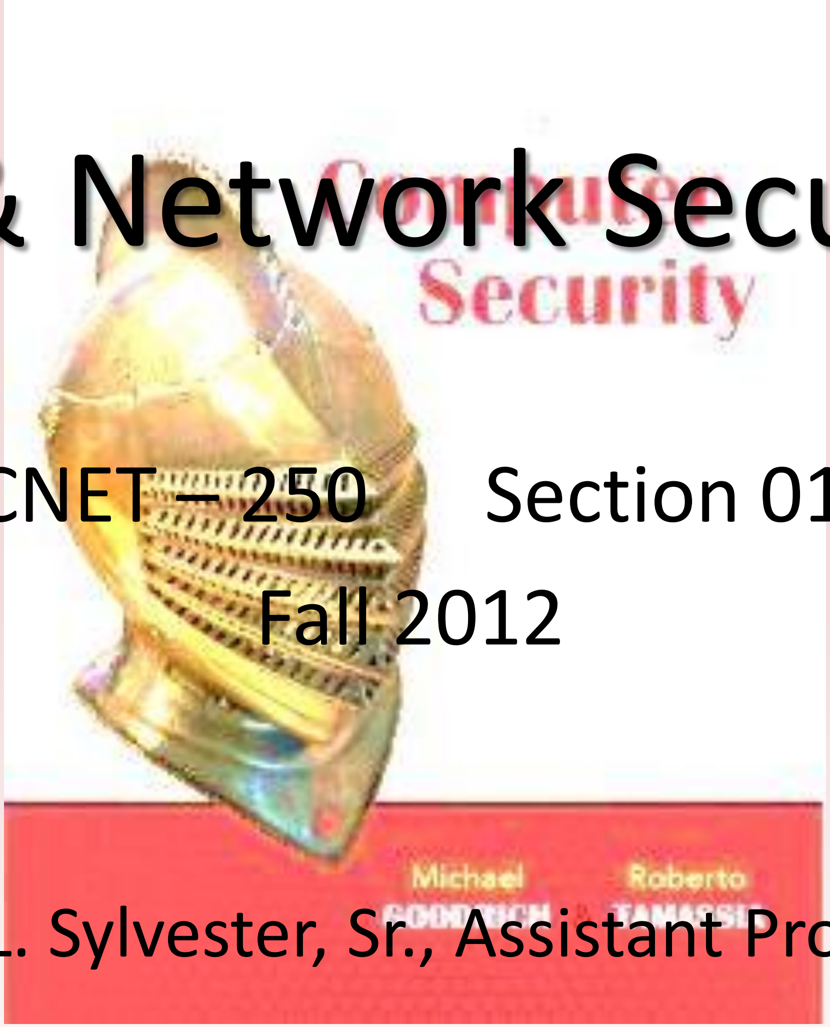
PC & Network Security



CNET – 250   Section 01

Fall 2012

David L. Sylvester, Sr., Assistant Professor

Chapter 4

Malware

Insider Attacks

Malware software, also known as malware, is software whose existence or execution has negative and unintended subsequences. In the case of malware, an insider attack refers to a security hole that is created in a software system by one of its programmers. An insider attack is a security breach that is caused or facilitated by someone who is a part of the very organization that controls or builds the asset that should be protected.

Such an attack is especially dangerous because it is initiated by someone that the organization should be able to trust.

Insider attack code can come embedded in a program that is part of a computer's operating system or in a program that is installed later by a user or system administrator. Either way, the embedded malware can initiate privilege escalation, can cause damage as a result of some event, or can itself be a means to install other malware.

Backdoors

A backdoor, which is also sometimes called a trapdoor, is a hidden feature or command in a program that allows a user to perform actions that would not normally be allowed. When used in a normal way, the program performs completely as expected or advertised. But if the hidden feature is activated, the program does something unexpected. (*Often in violation of security policies*) Since a backdoor is a feature or command embedded in a program, backdoors are always created by one of the developers or administrators of the software.

Backdoors Inserted for Debugging Purposes. Some backdoors are put into programs for debugging purposes. For example, if a programmer is working on an elaborate biometric authentication system for a computer login program, they may wish to also provide a special command or password that can bypass the biometric system in the event of a failure.

A backdoor left in a program even after it is fully debugged might not be intended to serve a malicious purpose. For instance, a biometric authentication system might contain a backdoor even after it is debugged, so as to provide a bypass mechanism in the case of an emergency or unanticipated problem.

Deliberate Backdoors. Sometimes programmers deliberately insert backdoors so they can perform malicious actions later that would not otherwise be allowed by the normal use of the program. **For example**, imagine what could happen if a programmer who is designing a digital entry system for a bank vault adds a backdoor that allows access to the vault through the use of a specific sequence of keystrokes, known only to him. Such backdoors are clearly inserted for malicious purposes, and they have the potential for dramatic effects.

Another more subtle way of creating a backdoor involves deliberately introducing a vulnerability into a program such as a buffer overflow.

Because the programmer knows about the vulnerability, it maybe straightforward for him to exploit it and gain elevated privileges.

Easter Eggs. Software may include features that can be accessed similarly to backdoors. These features are called Easter eggs. An Easter egg is a harmless undocumented feature that is unlocked with a secret password or unusual set of inputs. **For example**, unlocking an Easter egg in a program could cause the display of a joke, a picture of a programmer, or a list of credits for the people who worked on the program.

Logic Bombs

A logic bomb is a program that performs a malicious action as a result of a certain logic condition. The classic **example of a logic bomb** is a programmer coding up the software for the payroll system who puts in code that makes the program crash should it ever process two consecutive payrolls without paying him.

The Y2K Problem. For a piece of software to be a logic bomb, there has to be malicious intent on the part of the programmer. Simply programming errors don't count. For example, programmers in the 20th century encoded dates as two digits, *xy*, to simplify 19*xy*. When the year 2000 came, this practice caused several problems with credit card transactions and other date-dependent calculations.

The Omega Engineering Logic Bomb. This bomb was created and triggered by a programmer named Tim Lloyd, who was convicted for triggering this bomb on July 31, 1996. On this date, the bomb was triggered on the server for Omega Engineering’s manufacturing operations, which ultimately cost the company millions of dollars in damages and led to the company laying off many of its employees.

After an investigation, investigator discovered that the files on the server had been destroyed and that Tim Lloyd was the administrator for that server. When they searched the for backup files on the server, they found only two – at Tim Lloyd’s house – and they were both erased.

The Omega Engineering Logic Bomb. After further investigation of the true copy of the server's memory, agents of the US Secret Service found a program containing the following sequence of six character strings:

- **7/30/96**

  This was the event that triggered the logic bomb – a date that caused the remaining code to be executed only if the current date was later than July 30, 1996.

- **F:**

  This focused subsequence commands to be run on the volume F, which contained the server's critical files.

- **F:\LOGIN\LOGIN 12345**

  This is the login for a fictitious user, 12345, that had supervisory and destroy permissions, but had no password.

- **CD\PUBLIC**

  This is a DOS command to change the current directory to a folder PUBLIC, which stores common programs and other public files.

- **FIX.EXE /Y F:\*.***

  FIX.EXE was an exact copy of the DOS program DELTREE, which can delete an entire folder (and recursively its subfolders).

  FIX.EXE displays the message “fixing…” instead of “deleting…” for each file that is deleted.

  The /Y option confirms that each file should indeed be deleted.

  F:\*.* identifies all the files on volume F as the ones to be deleted.

- **PURGE F:\ /ALL**

  Deleted files can often be easily recovered by a simple analysis of the disk. This command eliminates the information that would make such reconstruction easy, thereby making recovery of the deleted files.

This program was a time-bomb, which was designed to delete all the important files on Omega Engineering’s server after July 30, 1996

Defenses Against Insider Attacks

Protecting a system against backdoors and logic bombs is not easy, since each of these types of malware is created by a trusted programmer. But defense against these types of malware is not impossible. Possible defenses against insider attacks are:

- Avoid single points of failure. Let no one person be the only one to create backups or manage critical system.
- Use Code walk-throughs. Have each programmer present there source code to another programmer (line-by-line).
- Using archiving and reporting tools. Tools such as automatic documentation generators and software archiving tools.
- Limit authority and permissions. Use a least privilege principle which states that each program or user in a system should be given the least privilege required for them to do their job efficiency.
- Physically secure critical systems. Important systems should be kept in locked rooms, with redundant HVAC and power systems, and protected against flood and fire.

- Monitor employee behavior. Be especially on the lookout for system administrators and programmers that have become disgruntled.
- Control software installation. Limit new software installations to programs that have been vetted and come from reliable sources.

Computer Viruses

A computer virus, or simply virus, is computer code that can replicate itself by modifying other files or programs to insert code that is capable of further replication. The self-replication property is what distinguishes computer viruses from other kinds of malware, such as logic bomb.

Viruses that replicate require some type of user assistance, such as clicking on an email or icon; and even sharing a USB drive.

Computer viruses mimic biological viruses, by being penetrated by the virus and then using the computers own systems to make copies of the virus and releasing the viruses to other computers.

Virus Classification

Computer Viruses follow four phases of execution:

1. **Dormant phase**. The virus just exists – the virus is laying low and avoiding detection.
2. **Propagation phase**. The virus is replicating itself, infecting new files on new systems.
3. **Triggering phase**. Some logical condition causes the virus to move from a dormant or propagation phase to perform its intended action.
4. **Action phase**. The virus performs the malicious action that it was designed to performed, called payload.
    - A silly picture on a computer's screen
    - Deleting all essential files on the hard drive.

Type of Viruses.

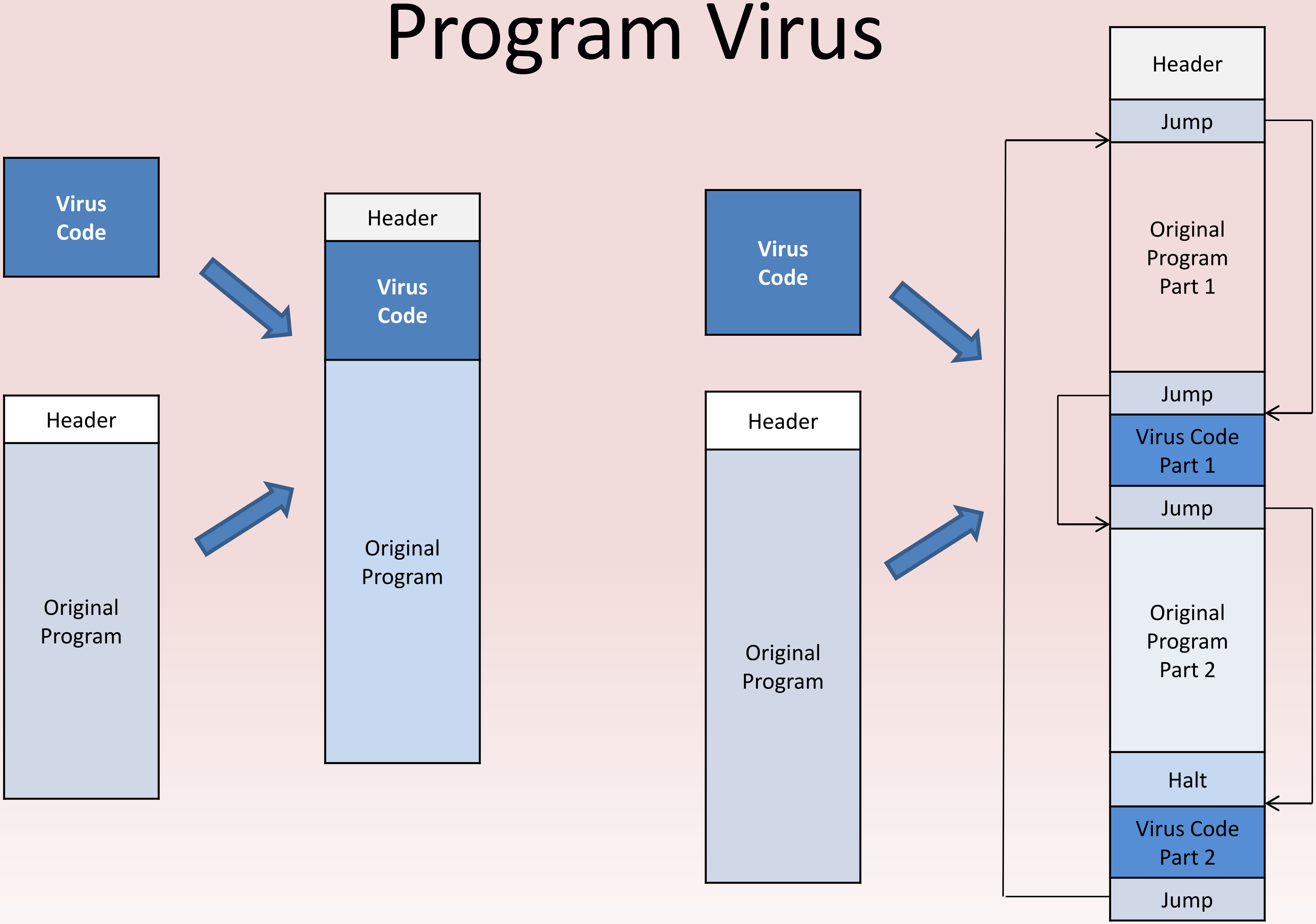
A **program virus**, also known as a file virus, infects a program by modifying the file containing its object code. Once the infection occurs, a program virus is sure to be run each time the infected program executes. If ran often, the virus is more likely to be able to be maintained and replicated. Macros are great targets for viruses, since they can behave similarly to an executable program.

Two ways of Infecting a Program File.

1. A simple injection at the beginning of a program
2. A more complex injection that splits the virus code into two parts and inject them at different points in the program

This is accomplished by beginning execution with the virus code using jump instructions and then passing control to the original program code.

Program Virus



(a)

(b)

A **macro virus**, known as a document virus, is launched when a document is opened, at which time the virus then searches for other documents to infect. Macro viruses can insert themselves into the standard document template, which makes every newly created document infected. Propagation occurs when infected documents are emailed to other users.

A **boot sector virus** is a special type of program virus that infects the code in the boot sector of a drive, which is run each time the computer is turned on or restarted. This type of virus can be difficult to remove, since the boot program is the first program that a computer runs. If the boot sector is infected with a virus, then that virus can make sure it has copies of itself carefully placed in other operating system files. As a result, antivirus software routinely monitors the integrity of the boot sector.

Real-World Computer Viruses.

Jerusalem Virus. This is a virus that originated in the 1980s and infected DOS operating system files. Discovered first in Jerusalem, Israel, it loaded itself into main memory and infected other executable files that are ran on the system.

Melissa Virus. This was the first recorded virus that spread itself via mass emailing. It is a macro virus that infects Microsoft Word 97 or 2000 documents and Excel 97 or 98 documents. Once infected documents are opened, the Melissa virus would email infected documents to the first 40 or 50 address of the victim's address book. When first launched many email servers had to shut down due to the overload of emails.

Elk Cloner Virus. This is a boot sector virus that infected Apple II operating system in the early 1980s. It infected systems by writing itself to the hard drive any time an infected disk was inserted. It was a fairly harmless virus that simply printed out a poem each 50th time the computer was rebooted.

Sality Virus. This a a recent executable file virus. Once executed, it disables antivirus programs and infects other executable files. It obscures it presence in an executable file by modifying its entry point. It also checks if it is running on a computer with an internet connection. And if so, it may connect to malware web sites and download other malware.

Defenses Against Viruses

Computer viruses act in the same manner as biological viruses, in that once the virus gets into your body, it may spread rapidly and infect many other cells. Once the body is attacked by a virus, the body fights to immune itself from the virus.

Virus Signatures. Virus signatures, or its digital fingerprint, are the character string that identifies the virus. Expert study infected files looking for code fragments that are unique to a particular computer virus. Once they locate a set of characteristic instructions, they can construct a character string that uniquely identifies the virus.

Once the signature of a virus is found, it can be loaded into a virus detection program. Virus detection software packages have to be frequently updated, so that they always are using the most up-to-date database of virus signatures. Detecting the presence of a virus signature in a file is an instance of a pattern-matching problem, which consists of finding a search pattern in the text.

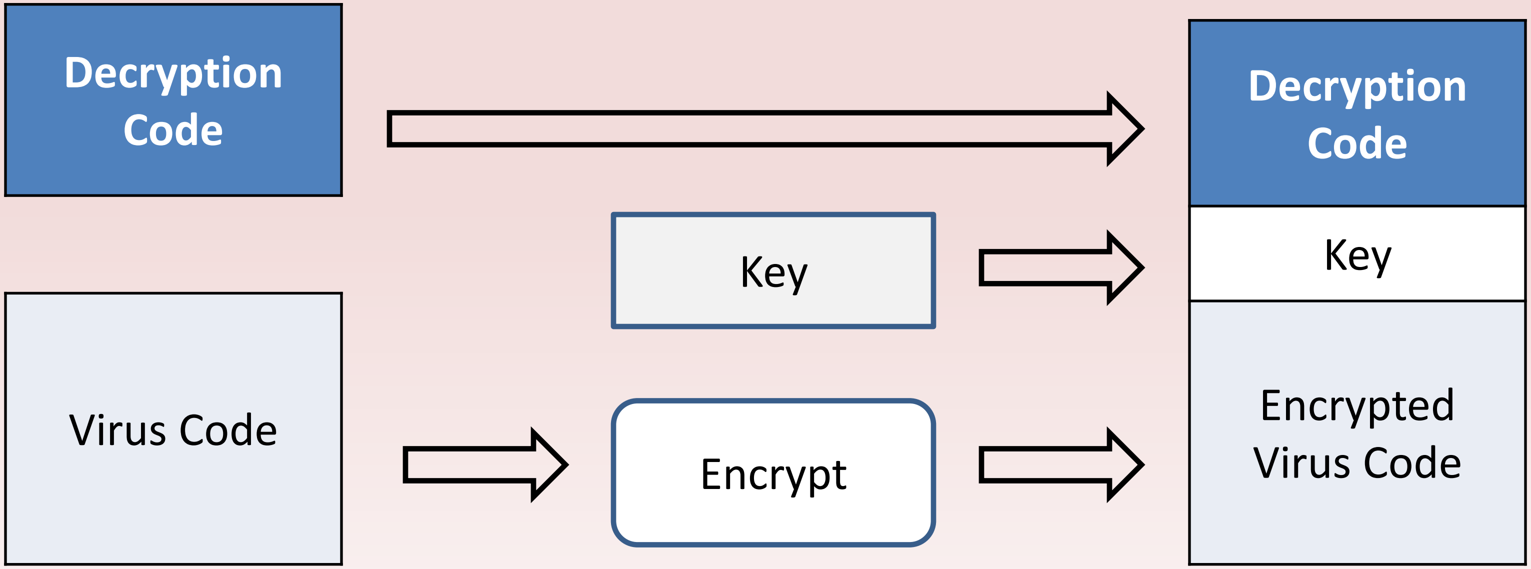
Virus Detection and Quarantine. Checking for viruses can be done in various ways.

1. Periodically (by scanning the entire filesystem.)
2. In Real-time (by examining each newly created or modified file and each email attachment received.

Real-time virus checking relies on intercepting system calls associated with file operations so that a file is scanned before it is written to disk. Any file that contains a part that matches a virus signature and is set aside into protected storage is considered to be quarantined.

Encrypted Viruses

Before antivirus software systems target virus signatures, computer virus writers often tried to hide their code. A virus may subdivide itself into multiple pieces and inject them into different locations in a program file. A technique used by virus writers to make the presence of their virus in a file more stealthy is to encrypt the main body of their program.



By encrypting the main part of its code, a virus hides many of its distinct features, including its replication code and, more importantly, its payload, such as searching for and deleting important files. This modification results in the virus code taking on a different structure: the decryption code, the key, and the encrypted virus code.

Polymorphic and Metamorphic Viruses

Another technique used by viruses to fight back against signature-based detection is mutating as they replicate, thereby creating many different varieties of the same virus. Such viruses are known as polymorphic or metamorphic viruses. Although these terms are sometimes used interchangeably, a polymorphic virus achieves its ability of taking on many forms by using encryption, with each copy of the virus being encrypted using a different key. A metamorphic virus, uses noncryptographic obfuscation (confusing) technique, such as instruction reordering and the inclusion of useless instructions. Polymorphic and metamorphic viruses are difficult to detect, since they often have few fixed characteristic patterns of bits in their code that can be used to identify them.

Detecting Polymorphic Viruses. One way to detect a polymorphic virus is to focus on the fact that it uses a different encryption key each time the virus encrypts and replicates itself.

Detecting Metamorphic Viruses. Finding a single string that serves as the signature for a metamorphic virus may be impossible. Instead, more complex signature schemes are used.

1. A conjunction signature consists of a set of strings that must appear, in any order, in the infected file.
2. A sequence signature consists of an ordered list of strings that must appear in the given order in the infected file.
3. A probabilistic signature consists of a threshold value and a set of string-score pair. A file is considered infected if the sum of the scores of the strings present in the file exceeds the threshold.

Malware Attacks

When malware was first discovered as a real-world risk to computer security, malicious software was distributed primarily via infected floppy disks. USB drives, CD-ROMs, and DVD-ROMs had not been invented yet, and the internet was restricted to researches in universities and industrial labs. Friends and coworkers would share files and collaborate using floppy disk and would inadvertently transmit computer viruses to each other. The explosive growth of the Internet gave rise to a whole new crop of malware, which didn’t need to be transmitted via media sharing in order to spread.