MOULE IV

Malware: Viruses, Worms and Trojans. Topological worms. Internet propagation models for worms.

Introduction:

A Malware is a set of instructions that run on your computer and make your system do something that an attacker wants it to do. Malware is any kind of unwanted software that is installed without your consent on your computer.Viruses, worms, Trojan horses, bombs, spyware, adware are subgroups of malware.

What malware does:

- Steal personal information
- Delete files
- Click fraud
- Steal software serial numbers
- Use your computer as relay

The purpose of malware

To partially control the user's computer, for reasons such as: – To subject the user to advertising – To launch DDoS on another service – To spread spam – To track the user's activity ("spyware") – To commit fraud, such as identity theft and affiliate fraud – For kicks (vandalism), and to spread FUD (fear, uncertainty, doubt).

Viruses and worms are the malwares which replicate themselves..

Virus	Worms
• Virus attach itself on to an executable file, • Uses this file as host and infects other files. • Rate of spread is generally slow. • Activated and propagated on human intervention.	• A worm is a standalone program, • Spreads from one computer to other. • Worms uses the network to spread so have an extra ordinary speed of propagation. • No need of human intervention.

A Trojan horse is a program with a malicious component masquerading as a useful piece of software. It do not replicate, but replicate by an action on the part of the victim. Trojans can enter through many ways, through emails, file sharing, website etc. it may read a file on disk and set up a network connection to hacker over which it communicates the files.

VIRUSES

Characteristics:

- When a virus program is run, the virus code gets executed first followed by the execution of original program.
- Virus code seeks other programs not yet infected and then passes on the infection to them.

- Virus may delete certain files.
- Virus file can be prepended or appended or both to the host file.
- Virus code can be split into segments and dispersed throughout the infected file using jump statements.
- The size of the infected file becomes greater than the original file so using these characteristics an antivirus can detect the infected file. To avoid detection, some viruses modify the **file service interrupt** file handler that returns attributes of files. Thus the service handler returns the original length only. Or by using **compression techniques** length of infected file remains same as that of original file. For this a compression routine need to be written in virus code.
- To infect a new file, the virus first compress the file and then prepends the virus code to compressed file. The infected file is uncompressed prior to execution.
- Viruses make the set of system calls. System calls are used for applications programs to request service of OS. They can make virus to read/write files, spawn new process, establish TCP connections.
- Some viruses call to copy their own code to other files, create/modify entries in windows registry etc.

Lifecycle of a virus

- **Dormant phase**- The virus program is idle during this stage. The virus program has managed to access the target user's computer or software, but during this stage, the virus does not take any action. The virus will eventually be activated by the "trigger" which states which event will execute the virus, such as a date, the presence of another program or file, the capacity of the disk exceeding some limit or the user taking a certain action (e.g., double-clicking on a certain icon, opening an e-mail, etc.). Not all viruses have this stage.
- **Propagation phase** - The virus starts propagating, that is multiplying and replicating itself. The virus places a copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.
- **Triggering phase** - A dormant virus moves into this phase when it is activated, and will now perform the function for which it was intended. The triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
- **Execution phase** - Payload: actions of the malware. This is the actual work of the virus, where the "payload" will be released. It can be destructive such as deleting files on disk, crashing the system, or corrupting files or relatively harmless such as popping up humorous or political messages on screen.

Types of viruses

1. Computer viruses infect a variety of different subsystems on their host computers and software.One manner of classifying viruses is to analyze whether they reside in binary executables (such as .EXE or .COM files), data files (such as Microsoft Word documents or PDF files), or in the boot sector of the host's hard drive.

a. **A memory-resident virus** / **resident virus** installs itself as part of the operating system when executed, after which it remains in RAM from the time the computer is booted up to when it is shut down. Resident viruses overwrite interrupt handling code or other functions, and when the operating system attempts to access the target file or disk sector, the virus code intercepts the request and redirects the control flow to the replication module, infecting the target.

b. **A non-memory-resident virus / non-resident virus** when executed, scans the disk for targets, infects them, and then exits (i.e. it does not remain in memory after it is done executing.

2. **Macro viruses** Many common applications, such as Microsoft Outlook and Microsoft Word, allow macro programs to be embedded in documents or emails, so that the programs may be run automatically when the document is opened.
   a. **A macro virus / document virus** is a virus that is written in a macro language, and embedded into these documents so that when users open the file, the virus code is executed, and can infect the user's computer. }
   b. **Boot sector viruses** specifically target the boot sector and/or the Master Boot Record (MBR) of the host's hard drive or removable storage media (flash drives, floppy disks, etc.)

WORMS

Features:

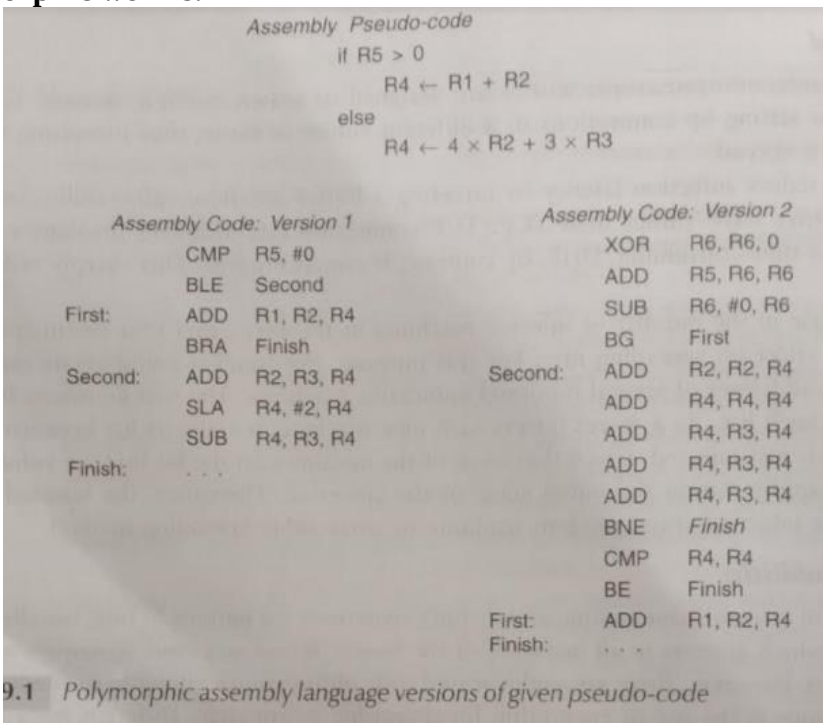
1. **Enhanced targeting :**
   Email Worms look into their victims mail box or email address book to find a set of targets. **A mobile worm** obtain phone numbers of the victim from the phone book in the cellphone hosting the worm. **Web worms** use search engines to harvest URLs of potentially vulnerable targets**. Internet scanning worms** scan the IP address space for vulnerable machines**. Random scanning** is choosing IP address at random eg: code red version1. **Localized scanning** attempts to connect with to victims with whom it share the network address eg: code red version2.
2. **Enhanced speed :**
   Some worms spawn multiple threads enhances the infection rate. Each thread set up connection to a different subset of hosts.
   Some worms reduce infection latency by targeting a buffer overflow vulnerability on an application that uses UDP than TCP as UDP is connectionless.
   Attacker creates multiple hit list carrying address of many vulnerable machines. The first worm could carry such a list, as it infects new machines, it splits its list between and the machine it has infected. There are after infected machine use their own strategies for spreading.
3. **Enhance capabilities :**
   Most worms have unique and distinct signatures – pattern of bits, assembly language, which appears in all instances of the worm. The signature turns the key for detection.

To avoid detection **encryption techniques** are used. Thus they fail to match any existing worm signatures. Such worms are called **polymorphic** – as different instances of worm use different keys of encryption. Polymorphic worm have to decrypted before executing. So a decryptor routine must also be a part of the worm code. Detecting a worm on the assumption that decryptor routine is invariant will not succeed always. As there are multiple mutation engines that create variations in the code of a given descriptor.

Creation of several code versions that are superficially different but functionally identical. Ways to create multiple versions are:

- Use dummy instructions(NOPs, ADDing 0 to a register)
- Use extraneous operands/variables
- Changing flow of control without disturbing the existing logic

The below given code shows two verisons that look different but perform same function. Worms that have multiple versions with/without relying on encryption are reffered to as **metamorphic worms.**



Assembly Pseudo-code

```
if R5 > 0
    R4 ← R1 + R2
else
    R4 ← 4 × R2 + 3 × R3
```

Assembly Code: Version 1

```
        CMP   R5, #0
        BLE   Second
First:  ADD   R1, R2, R4
        BRA   Finish
Second: ADD   R2, R3, R4
        SLA   R4, #2, R4
        SUB   R4, R3, R4
Finish: . . .
```

Assembly Code: Version 2

```
        XOR   R6, R6, 0
        ADD   R5, R6, R6
        SUB   R6, #0, R6
        BG    First
Second: ADD   R2, R2, R4
        ADD   R4, R4, R4
        ADD   R4, R3, R4
        ADD   R4, R3, R4
        ADD   R4, R3, R4
        BNE   Finish
        CMP   R4, R4
        BE    Finish
First:  ADD   R1, R2, R4
Finish: . . .
```

9.1 *Polymorphic assembly language versions of given pseudo-code*

Some worms **are time-aware**, they obtain the date and time of a network time protocol(NTP) server and can initiate specific actions at specified points of time. Such worms remain dormant for period of time and strikes in a concerted fashion by eg: launching dos attack

Some worms can **update** themselves by downloading code from given URLs. Else they may access a URL which in turn provides a set of URL from which updated worm code can be downloaded

4. **Enhanced destructive power :**

Cost is estimated based on lost productivity, clean-up costs, system downtime that affects business and revenues. fast spreading worms causes severe network congestion problems disrupting normal internet traffic and contributes to system downtime.

Some worms contributed attack packets to DDoS attack which caused website defacement.
The Witty worm was the first worm to carry destructive payload. It deleted a ranom section of victims harddisk leading to system crash.

Classes of worms:

Worms are classified based on their vector of propagation:

1. Internet scanning worms
2. Topological worms
    a. Email worms
    b. P2P worms
3. Web worms
4. Mobile worms

1. Internet scanning worms :

- are self activated.
- They scan the internet looking for vulnerable machines. This vulnerability can be a buffer overflow for a particular version of OS.
- Worm communicates with and delivers its malicious payload to the victim using TCP or UDP protocols.
- **Code Red:**
    - Started on June 18, 2001
    - Using buffer overflow vulnerability in MS IIS Web Server.
    - The worm itself was carrier in HTTP request messages targeted as IIS servers.
    - **Version 1** used a random number generator to generate new address of machine to infect. The same seed is used for every instance of the worm – so same machines infected over and over again.
    - **Variant of Code Red1 –** a random seed was generated in a worm.
    - About 360,000 machines were infected in 14 hours – **infection phase**
    - **Attack phase –** launched a DoS on www.whitehouse.gov had webpages defaced with phrase " hacked by chinese".
    - This was memory resient and destroyed by rebooting the system.
    - **CodeRed Version 2 –** installed a backdoor providing an attacker remote, administrator level access to the victim.
    - this persisted on the disk.
    - It spawned multiple threads, each sent probe packets to several targets.
- **Slammer:**
    - **The SQL Slammer** was launched on 25 Jan 2003.
    - Targeted a buffer overflow vulnerability on MS SQL Server 2000.
    - Worm sent packets to UDP port 1434 – the DB software's resolution service.
    - Uses simple random scanning to propagate.
    - Slammer payload is 384 bytes in length.
    - UDP is connection less protocol so no overhead of connection establishment.
    - Number of machines infected by slammer doubled every 8.5 seconds.

- It generated considerable traffic, thus disrupted the internet traffic. Thus crashed several switches and routers.

2. **Topological worms**

- Topological worms are so called because the machines vulnerable to such a worm can be represented as a graph with nodes representing a vulnerable machine. An edge between A and B exists if A stores/knows the address of B and is capable of directly infecting B by sending malicious payload. These worms have focused targets. Immediate targets are their neighbors which inturn spread infection to their neighbors and so on. There are two types of topological worms
  - Email worms
  - P2P worms
- **Email worms:**
  - Propogates through infected emails.
  - Victim receives an email that appears to be from a familiar/ trusted source. Or the email lures the recipient by an attachment with a catchy caption.
  - The worm is activated when the user clicks on the attachment.
  - Eg: "I love you" worm unleashed in 2000. It appears with the title I love you in victims inbox with a text file loveletter.text.vbs contains Visual Basic Script. On clicking on this attachment, the embedded VB script executes sending a copy of itself to every one in victims contact list.
  - Eg: macro worms – software macro are embedded in documents and this macro executes while document is opened. – Melissa
  - Nimda worm – had multiple vectors of propogation.
  - SoBig ia a popular email worm with many versions - one version update itself by downloading code from websites. URL of these sites were contained in a file that itself was downloadable from geocities.com.
  - Malicious code received installed a keystroke logger and stole passwords from victims.
- **P2P worms**
  - A P2P network is a massively distributed system of computers where each peer or node plays the role of both client and server. They are used for sharing files. Each peer maintains within itself a shared folder of files that is willing to share with others. Users download from their peers located across the world not from a central server.
  - most P2P networks use an overlay network – a logical network of peers. To peers are said to be neighbors if there exist an active TCP connection between them.
  - A peer, A that wishes to obtain a file, abc creates a "query request" message for abc which it sends to all its neighbors. Each neighbor that does not have the file, in turn queries to its neighbors and so on. if a peer has abc it returns a "query hit" message in reverse. The requesting node may receive multiple query hit responses. It can choose one among them, say if A received hit messages from B and C, it can choose one among them, and directly contacts it. A sends its IP address to node B with a request that abc be downloaded to that IP address. The file is then downloaded using FTP/HTTP.

Potential ways of spreading of P2P worms:

- A malicious peer responds positive to any query. When the requester chooses to download the file from the malicious peer, the latter sends the infected file by altering the name, so as to match the requested file. Once infected the requester mimics the behavior of the malicious peer thus propagates the infection. Various popular files stored in the shared folder f peer gets infected and any of them on downloading the infection spreads. → this type of worms are called as **passive,** as they propagates only when requested to download a file.
- Peers in a P2P network run the same protocol. An exploitable buffer overflow vulnerability is a familiar starting point. Peer maintains a list of neighbors implies that a worm has ready targets and does not need to perform random scanning.--> This type of worm is called **active** as it propagates on its own without receiving request from its peers.

P2P worms may result in no apparent traffic anomaly, so an intrusion detection system monitoring network traffic is unlikely to raise an alert.

3. Web worms

Web worms are executed in browsers which run on diverse hardware / OS platforms. Webworms are written in high level language making it easy to perform complex operations but difficult to execute low-level operations.
Eg: XSS worm – as it exploit cross-site-scripting vulnerability. First step in creating an XSS worm is to inject attack code into vulnerable web server. When a user access the infected website through a browser, malicious code is downloaded into the browser.

Propagation of XSS worm: / case study:
XSS worm, Samy infected the social networking site, Myspace. Samy added a bunch of carefully crafted Javascript to his profile. When a visitor to Samy's website, say V1, downloaded samys profile on to his profile, the javascript in samys profile executed. Thus samy is added as a friend of V1 and included the message "but most of all, samy is my hero".
Within 20 hours of the first visit to samys profile , samy have been added as a friend to more than a million users profiles. The javascript uploaded itself on to V1's profile on Myspace server, thus infecting it. This is done by an HTTP Post request sent from the browser to the server. That cause the screen to freeze sending the request and receiving the HTTP response from the server. To ensure that the viewer had a normal screen experience, samy's javascript created an XMLHttpRequest object which was used to send the malicious javascript to myspace server. The message from an XMLHttpRequest object is asynchronous and runs in the background, it was not notice by the user. The XMLHttpRequest object sent a post message to update V1's profile with malicious javascript.

How does server know where the XMLHttpRequest **has come from :**
HTTP is a stateless protocol, as session id is created by the server upon user login. It is included in all messages exchanged between client and server. The malicious javascript in samy's profile contained the code that retrieved the session id and included it in the XMLHttpRequest. The infection propagates the same way

4. Mobile worms

New genearation smartphones work as a cell phone and a low end PC. They provide a rich set of APIs. In Symbion OS users can download new applications onto their smartphones. New application and updations of existing applications are packaged in Symbian Installation Source(SIS) files. The mobile worms was packaged in well formed SIS files. The installer was tricked into believing that this is updation of a software. So the new version with malicious code replaces the old. So when application involked the malware also run.

Bluetooth is another vector of mobile worm propagation. Worm Cabir attempts to discover other Bluetooth enabled phones set in discoverable mode. When a phone found, it sends the worm payload in SIS file. The receiver needs to accept and install the file. Continuos scanning for new victims by an infected phone deplets the battery power.

Commwarrior is the worm which spread through Bluetooth and mms.

DIFFERENCE BETWEEN VIRUS AND WORM

Virus	Worm
1. A computer virus attaches itself to a program or file enabling it to spread from one computer to another, leaving infections as it travels.	1. A worm is similar to a virus by design and is considered to be a sub-class of a virus. Worms spread from computer to computer
2. A computer virus is a small program written to alter the way a computer operates, without the permission or knowledge of the user.	2. The worm consumes too much system memory (or network bandwidth), causing Web servers, network servers and individual computers to stop responding.
3. A virus must meet two criteria: • It must execute itself. It often places its own code in the path of execution of another program. • It must replicate itself. For example, it may replace other executable files with a copy of the virus infected file.	3. A worm must meet two criteria: • Worms exploit holes in operating system security so it is important to download and install all patches. • The weak security and similar network configuration is required to travel.
4. Viruses, which requires the spreading of an infected host file.	4. Worms are programs that replicate themselves from system to system without the use of a host file.
5. Spread with uniform speed as programmed.	5. Worms spread more rapidly than viruses.
6. It can be attached to .EXE,.COM,.DOC,.XLS etc.	6. It can be attached to any **attachments of an email, any file on network.**
7. EXAMPLE: Michelangelo, I LOVE YOU, Melissa, Cascade(file infector virus) etc.	7. **EXAMPLE:** Blaster Worm, the worm has been designed to tunnel into your system and allow malicious users to control your computer remotely, W32.Mydoom.AX@mm

INTERNET PROPAGATION MODELS FOR WORMS

Modeling worm propagation model is important as it helps to obtain insight into the factor which governs the speed, to study the efficacy of different schemes designed to retard the spread of a worm.

1. The Simple Epidemic Model:

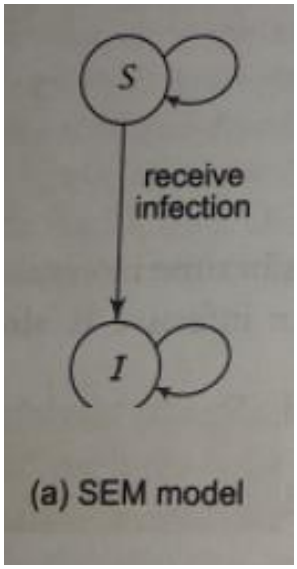
The model assumes that there are only two types of entities in the population.

a. Susceptible entity
b. Infected entity

An infected entity can infect a susceptible entity. Once infected an entity remains infected an doesnot recover.

N = size of total population
I(t) = number of infected individuals at the time t

The number of susceptible at time t = N – I(t)
β = initial infection rate (each infected person attempts to pass on the infection to β susceptibles in 1 time unit)



$$dI = \beta I(t)\left(1 - \frac{I(t)}{N}\right)dt$$

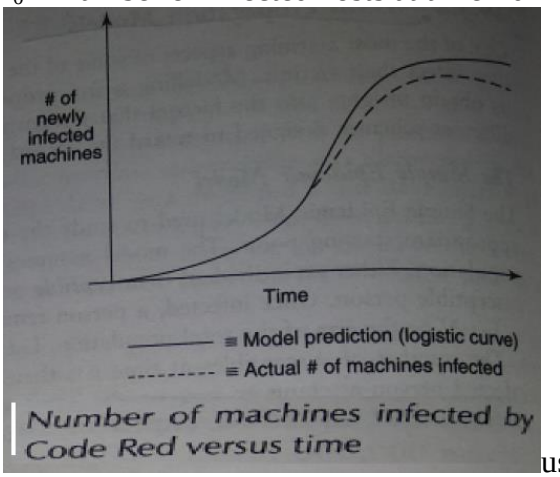
$$\beta\, dt = \left(\frac{dI(t)}{I(t)\left(1 - \frac{I(t)}{N}\right)}\right)$$

In an infected population each infected person infects β dt susceptibles in time interval dt.
In an infinite population of size N, the probability that the target of an infective is already infecte = I(t)/N → which do not add to the population of newly infected.
Previously uninfected entities added to the count of freshly infected in time interval dt = 1-[I(t)/N]

If we integrate both sides we get:

$$I(t) = \frac{I_0 N}{I_0 + (N - I_0)e^{-\beta t}}$$

I_0 = number of infected hosts at time =0

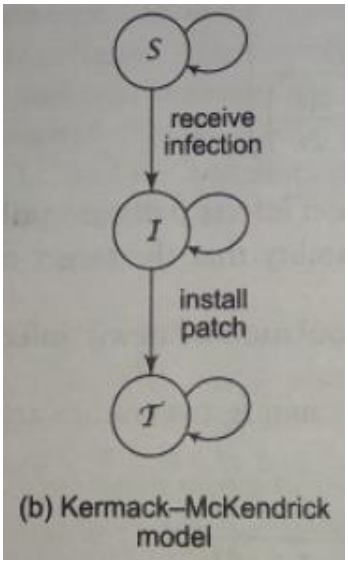


using SEM

2. **Kermack-McKendrick Model**

KM Model more accurately models the spread of infectious malwares by considering three categories of people:

a. Susceptibles (state S)
b. Infectious (state I)
c. Neither in S nor I(terminal state T)



(b) Kermack–McKendrick model

- Initially all individuals in population are in S
- Can move from S to I but not vice versa
- An infected person may or may not be cured
- If cured they are never again vulnerable to disease. Moving from I → T corresponds to an infected machine being pathched.

- N → size of population
- I(t) → number of infected peoples at a time t
- S(t) → number of Susceptible at t
- Number of Machine at terminal T
- T = N - S(t) - I(t)

The K-M set of equation :

$$\frac{dS}{dt} = -\beta I(t)\left(\frac{S(t)}{N}\right)$$

The rate of decreasing in susceptible

and

$$\frac{d(N - S(t) - I(t))}{dt} = \gamma I(t)$$

Define rate of increase in terminal state machine.
γ(Gamma) → Rate at which infected machine transit to Terminal state - Opposite of Beta

Some drawback in assumptions of K-M model

1. Susceptible machine can be patched. So movement from a S to T is possible
2. Beta is network dependent. In K-M model data is considered as a constant . But in real time as network traffic increases infection rate decreases. So, Beta is not a constant
3. Gamma is constant. But in Real time, after infection most systems will be patched due to awareness and rate of infection decreases. So Gama is not a constant

TROJAN HORSES:

A Trojan horse program is one that can do something malicious in addition or instead of what the person thinks it is doing. The term has recently also come to mean any malicious program that is added to your system without your knowledge or authorization. A Trojan horse is a program with an overt (documented or known) effect and a covert (undocumented or unexpected) effect.

Trojan horses can make copies of themselves. One of the earliest Trojan horses was a version of the game animal. When this game was played, it created an extra copy of itself. These copies spread, taking up much room. The program was modified to delete one copy of the earlier version and create two copies of the modified program. Because it spread even more rapidly than the earlier version, the modified version of animal soon completely supplanted the earlier version. After a preset date, each copy of the later version deleted itself after it was played.

A propagating Trojan horse (also called a replicating Trojan horse) is a Trojan horse that creates a copy of itself.