LNMIIT
The LNM Institute of
Information Technology

Web Attacks and Security IInd

Classroom Code:o7l60yk

Outline

- Malware—Viruses, Trojan Horses, and Worms
  - Precise History
  - Technical details
    - Effect on users, user's system and world
    - Transmission and Propagation
    - Activation
    - Stealth (hiding technique to avoid detection)
  - Counter measures
- Side Channel Attacks
- Demonstrations (Demo) of Attacks

MALWARE

- **Malicious code** or **rogue programs** or **malware** (short for MALicious softWARE) is the general name for programs or program parts planted by an agent with malicious intent to cause unanticipated or undesired effects.
- Malicious intent distinguishes this type of code from unintentional errors, even though both kinds can certainly have similar and serious negative effects.
- Most faults found in software inspections, reviews, and testing do not qualify as malicious code; their cause is usually unintentional.

Virus-code with malicious purpose; intended to spread

- A **virus** is a program that can replicate itself and pass on malicious code to other non-malicious programs by modifying them.
- The infection usually spreads at a geometric rate, eventually overtaking an entire computing system and spreading to other connected systems.
- A virus can be either **transient** or **resident**.

Virus Types

- A **transient virus** has a life span that depends on the life of its host. The virus runs when the program to which it is attached executes, and it terminates when the attached program ends. (During its execution, the transient virus may spread its infection to other programs.)
- A **resident virus** locates itself in memory, it can then remain active or be activated as a stand-alone program, even after its attached program ends.

Worm

- A **computer worm** is a standalone malware program that replicates itself in order to spread to other computers .
- Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it.

Worm vs Virus

- Worms almost always cause at least some harm to the network, even if only by consuming bandwidth, whereas viruses almost always corrupt or modify files on a targeted computer.
- The worm spreads copies of itself as a stand-alone program, whereas the virus spreads copies of itself as a program that attaches to or embeds in other programs.

Worm's Activity

- Any code designed to do more than spread the worm is typically referred to as the "payload".
- Typical malicious payloads might delete files on a host system (e.g., the ExploreZip worm), encrypt files in a ransomware attack, or steal data such as confidential documents or passwords.
- Probably the most common payload for worms is to install a backdoor.
- This allows the computer to be remotely controlled by the worm author. Networks of such machines are often referred to as botnets and used for a range of malicious purposes, including sending spam or performing DoS attacks.

Helpful Worm

- **Anti-worm** (sometimes **helpful worm**) is a computer worm designed to do something that its author feels is helpful, though not necessarily with the permission of the executing computer's owner.
- The concept of the "anti-worm", or "helpful worm", is a proactive method of dealing with virus and computer worm outbreaks.
- This type of worm delivers its payload by doing helpful actions.
- Just like malicious computer worms, anti-worms reach computers by scanning IP ranges and placing a copy of themselves on vulnerable hosts.
- **The anti-worm then may perform patching the computer's vulnerability and uses the affected computer to find other vulnerable hosts.**
- Anti-worms have the ability to spread just as fast as regular computer worms, utilizing the same "scan, infect, repeat" model that malicious computer worms use.
- Source: https://en.wikipedia.org/wiki/Anti-worm

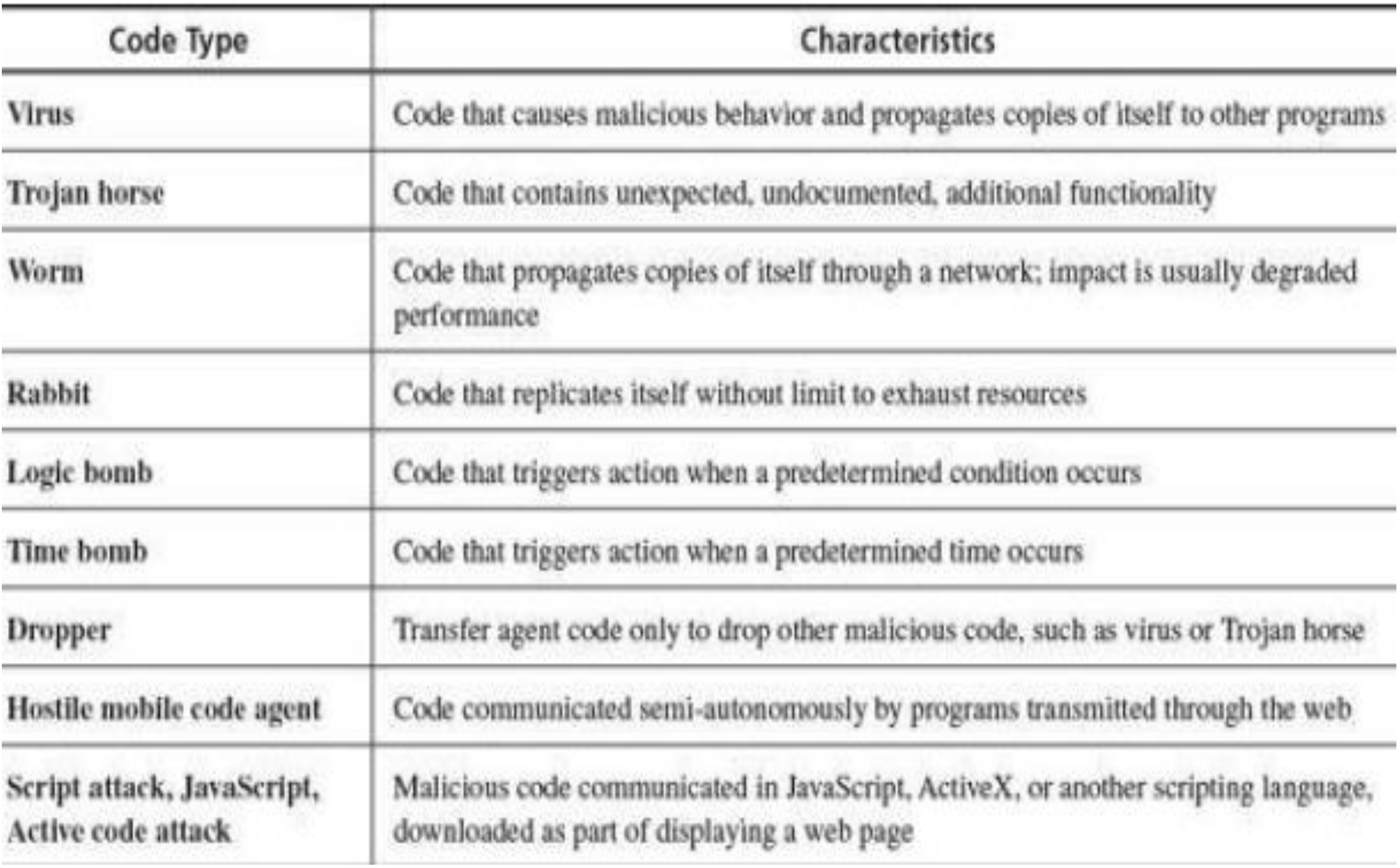
Trojan Horse: Program with benign apparent effect & hidden, malicious effect

- A **Trojan horse** is malicious code that, in addition to its primary effect, has a second, nonobvious, malicious effect.

- **Generally,** Trojan horse malware slips inside a program undetected and produces harmful effects later on.
- It does not necessarily try to propagate.

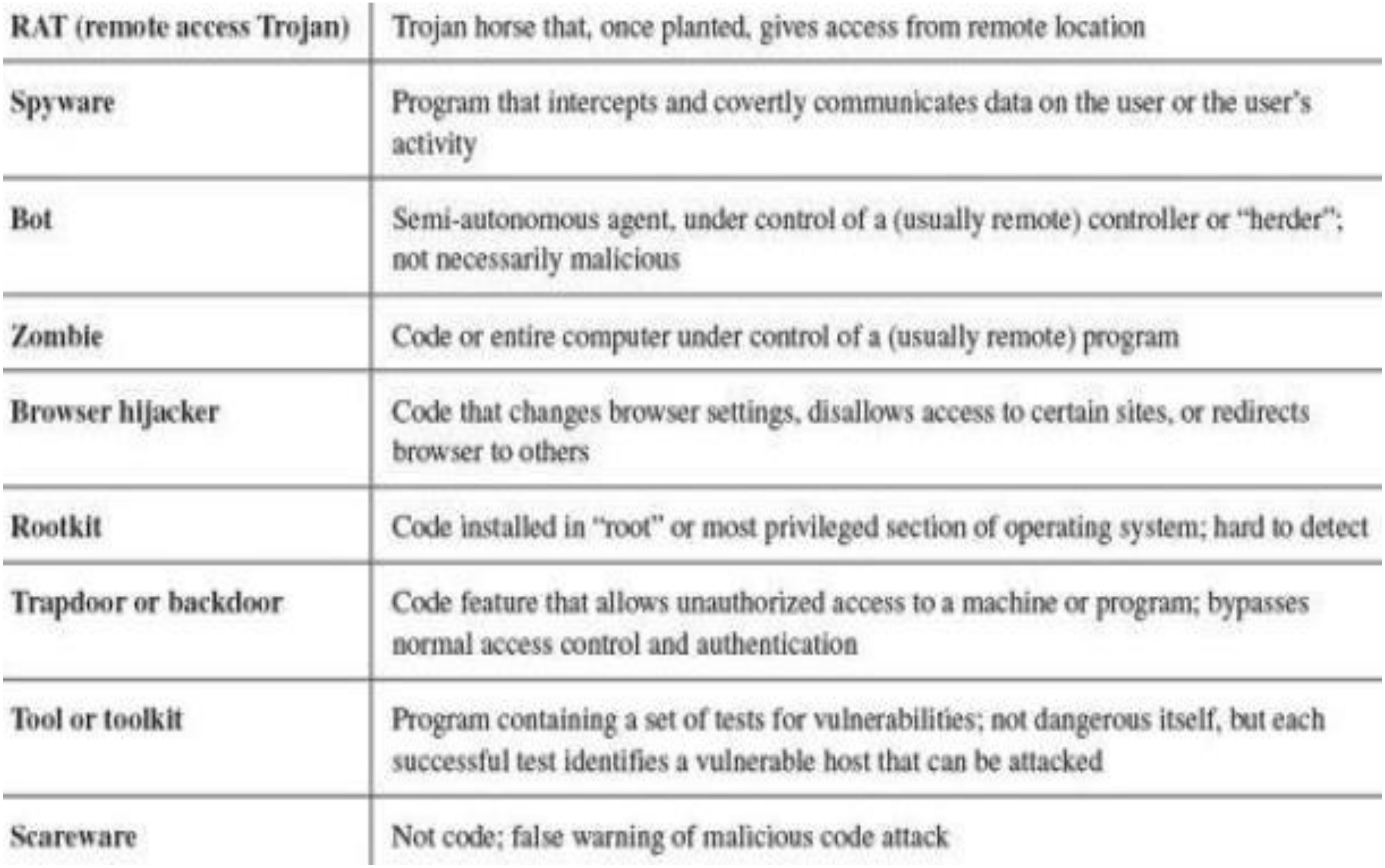
Various Malicious Codes: Don’t Rote them ??

LNMIIT



Code Type	Characteristics
Virus	Code that causes malicious behavior and propagates copies of itself to other programs
Trojan horse	Code that contains unexpected, undocumented, additional functionality
Worm	Code that propagates copies of itself through a network; impact is usually degraded performance
Rabbit	Code that replicates itself without limit to exhaust resources
Logic bomb	Code that triggers action when a predetermined condition occurs
Time bomb	Code that triggers action when a predetermined time occurs
Dropper	Transfer agent code only to drop other malicious code, such as virus or Trojan horse
Hostile mobile code agent	Code communicated semi-autonomously by programs transmitted through the web
Script attack, JavaScript, Active code attack	Malicious code communicated in JavaScript, ActiveX, or another scripting language, downloaded as part of displaying a web page

Various Malicious Codes



RAT (remote access Trojan)	Trojan horse that, once planted, gives access from remote location
Spyware	Program that intercepts and covertly communicates data on the user or the user's activity
Bot	Semi-autonomous agent, under control of a (usually remote) controller or "herder"; not necessarily malicious
Zombie	Code or entire computer under control of a (usually remote) program
Browser hijacker	Code that changes browser settings, disallows access to certain sites, or redirects browser to others
Rootkit	Code installed in "root" or most privileged section of operating system; hard to detect
Trapdoor or backdoor	Code feature that allows unauthorized access to a machine or program; bypasses normal access control and authentication
Tool or toolkit	Program containing a set of tests for vulnerabilities; not dangerous itself, but each successful test identifies a vulnerable host that can be attacked
Scareware	Not code; false warning of malicious code attack

Precise History

- Malicious code dates certainly to the 1970s, and likely earlier.
- Its growth has been explosive, but it is certainly not a recent phenomenon.

See **TABLE 3-3:** Notable Malicious Code Infections, Page no 203, in the text book by Charles P. Pfleeger.

Harm from Malicious Code

We can divide the payload from malicious code into three categories:

1. *Nondestructive*. Examples of behavior are sending a funny message or flashing an image on the screen, often simply to show the author's capability.
2. *Destructive*. This type of code corrupts files, deletes files, damages software, or executes commands to cause hardware stress or breakage with no apparent motive other than to harm the recipient.

Harm from Malicious Code

3. *Commercial or criminal intent*. An infection of this type tries to take over the recipient's computer, installing code to allow a remote agent to cause the computer to perform actions on the agent's signal or to forward sensitive data to the agent.

Examples of actions include collecting personal data, for example, **login credentials** to a banking web site, **collecting proprietary data**, such as **corporate plans**, or serving as a **compromised agent** for **sending spam email** or mounting a **denial-of-service attack**

Activation, Transmission and Propagation

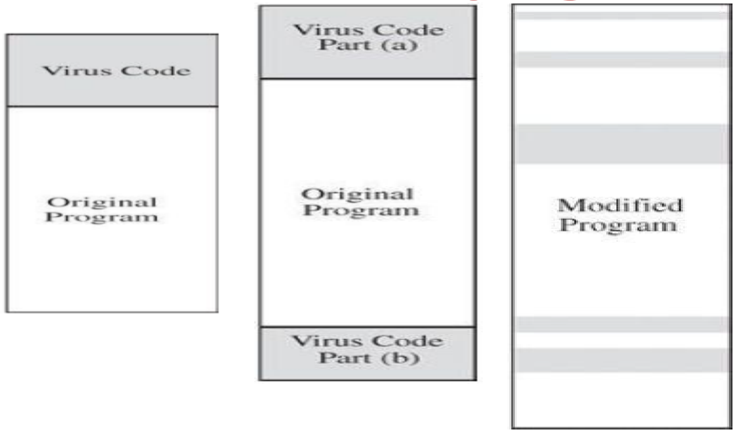
Activation

- Setup and Installer Program Transmission
  - Executed by human
  - Triggered by date and/ or elapsed time
- Attached File
- Document Viruses
- Autorun

Activation, Transmission and Propagation

Propagation

- Appended Virus
- Surrounding Viruses
- Integrated Viruses and Replacements



Stealth

- The final objective for a malicious code writer is stealth: avoiding detection during installation, while executing, or even at rest in storage.
- **Installation Stealth**
- **Execution Stealth**
- **Stealth in Storage**

Countermeasures

- As per McAfee Labs quarterly report-Jun-2018 [1] There are 700,000,000 (Seventy Crores) malwares.
- The available countermeasures are not perfect, some are reactive—after the attack succeeds—rather than preventive, and all parties from developers to users must do their part.

[1] https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-jun-2018.pdf

Countermeasures for Users

- User Vigilance:
  - use commercial software
  - test new software on isolated computer
  - open only safe email attachments and USB drive files
  - avoid visiting potential malicious website
  - Make recoverable system images
- Virus Detectors

Counter Measures for Developers

- **Software Engineering Techniques:** General software engineering principles are intended to lead to correct code, which is also a security objective.

Modularity, Encapsulation, information hiding, mutual suspicion, simplicity, testing (correctness, usability, performance), **Validation etc.**

Counter Measures for Developers: Specifically for Security

"Top 10 Secure Coding Practices" from the Computer Emergency Response Team (CERT) of the Software Engineering Institute at Carnegie Mellon University.

1. Validate input.

2. Heed (Pay Attention to) compiler warnings.

3. Architect and design for security policies.

4. Keep it simple.

5. Default to deny.

6. Adhere to the principle of least privilege.

7. Sanitize data sent to other systems.

8. Practice defense in depth.

9. Use effective quality-assurance techniques.

10. Adopt a secure coding standard.

Counter Measures for Developers: Specifically for Security

- **Penetration Testing for Security:**

penetration testing or **tiger team analysis** is a strategy often used in computer security and it is also called **ethical hacking**, because it involves the use of a team of experts (Tiger Team) trying to crack the system being tested.

Counter Measures for Developers: Specifically for Security

- **Proofs of Program Correctness using a technique “Program Verification”.**
- Program verification involves making initial assertions about the program’s inputs and then checking to see if the desired output is generated.

Counter Measures for Developers: Specifically for Security

- Defensive Programming : Program designers and implementers need not only write correct code but must also anticipate what could go wrong. For example: A program expecting a date as an input must also be able to handle incorrectly formed inputs such as 31-Nov-1929 and 42-Mpb-2030.

How Antivirus/ Virus Scanner Works?

Signature Detection: antivirus applications come with a directory of already checked-viruses (called virus definitions and keep updating them) and match the codes and patterns in files and web pages to unique bits and patterns that make up the code of a virus.

If they match, the file is quarantined, means that it is moved to a new and safe location so that it does not infect any other files on the system.

Antivirus programs also checks for any malicious behavior on a system such as suspicious registry entries or executing an unknown program automatically upon system startup

Detection Methods

- ***Heuristics-based detection*** aims at detecting malware by statically examining files for suspicious characteristics without an exact signature match.
- For instance, an antivirus tool might look for the presence of rare instructions or junk code in the examined file.
- The tool might also emulate running the file to see what it would do if executed, attempting to do this without noticeably slowing down the system.
- The biggest downside of heuristics is it can inadvertently flag legitimate files as malicious.

Detection Methods

- **Behavioral detection** observes how the program executes, rather than merely emulating its execution. This approach attempts to identify malware by looking for suspicious behaviors, such as
- unpacking of malcode,
- modifying the hosts file,
- observing keystrokes.
- Changing settings/content of other programs
- Dozens of files are modified or deleted
- Remotely connecting to computers

Detection Methods

- **Sandbox Detection :** This is a type of detection method in which antivirus software run programs in a virtual environment and record the actions it performs to identify whether the programs are malicious or not.
- If the program is found safe, it is then executed in the real environment.
- This technique is rarely used in consumer antivirus solutions as it is both heavy and slow but antivirus solutions designed for corporate and network use offer this.

Detection Methods

- **Security Analytics** is the recent development in malware detection that security companies now provide with their antivirus products to detect and eliminate forms of malware that has just been released.
- First, a series of features of files are extracted from files and then data mining and machine learning algorithms are used to determine the behavior of a file to detect whether the file is malicious or not.

Side Channel Attacks

- Cache Attack
- Power Monitoring/ Analysis Attack
  - (SPA and DPA)
- Timing Attack

Side Channel Attacks on Web

- Cache Attack (using JavaScript)
- Timing Attack (for example User Enumeration)

User enumeration: when a malicious actor can use brute-force to either guess or confirm valid users in a system. It is a web application vulnerability that is applicable to any user authentication based system.

Two of the most common areas where user enumeration occurs are in a site's login page and its ‘Forgot Password' functionality.

LNMIIT
The LNM Institute of Information Technology

Thank you, for your time and attention!

The LNMIIT: Where young dreams take shape …..