جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني
Department of Cyber Security

Subject:

Malicious codes

Class:

THIRD

Lecturer:

Dr. Suha Alhussieny

Lecture: (2)

Logic Bomb, Trojan Horses , Mobile Code

Logic Bomb

One of the oldest types of program threat, predating viruses and worms, is the logic bomb .The logic bomb is code embedded in some legitimate program that is set to "explode" when certain conditions are met. Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application. Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage .

Logic bombs are often used with viruses, worms, and Trojan horses to time them to do maximum damage before being noticed. For example, a programmer may hide a piece of code that starts deleting files.

logic bombs are different because they **do not spread automatically**. Instead, they **stay hidden** until a specific condition is met.

- It is like a **time bomb**, but instead of counting seconds, it waits for a **logical condition**.
- When the condition is satisfied, the code **executes its malicious payload**.

Examples of triggers:

- A specific **date/time** (e.g., January 1st).
- **Deletion or creation** of a file.
- **Login** by a certain user.
- **Exceeding a value** in a database.

Key Characteristics

- **Dormant:** The logic bomb stays hidden until activated.
- **Triggered:** Activation occurs only when a predefined condition is met.

- **Payload:** Can delete files, steal data, crash systems, or corrupt databases.
- **Hidden inside legitimate software:** Often planted by insiders (e.g., employees, contractors).

How a Logic Bomb Works

1. **Insertion**
   - The attacker (or even a disgruntled insider) inserts malicious code inside a normal program or script.
2. **Dormancy**
   - The code sleeps and has no effect until the condition occurs.
3. **Trigger Event**
   - A logical condition is satisfied (e.g., date reaches May 1st, or user "Admin" logs in).
4. **Execution of Payload**
   - The malicious action occurs: file deletion, data leakage, shutdown, etc.
5. **Aftermath**
   - Sometimes, the code erases itself to hide traces.

Real-World Examples

- **2000s – UBS PaineWebber Case**

  A disgruntled employee planted a logic bomb on company servers, triggered by a date. It caused more than $3 million in damages.
- **NASA Example**

  Logic bombs were suspected in space research systems, where insiders planted time-based triggers.

- **Sony PlayStation Hack (alleged)**

  Some reports suggested insiders used logic bombs to disrupt systems during internal disputes.

Difference Between Logic Bomb and Other Malware

- **Virus:** Replicates and spreads – a logic bomb does not replicate.
- **Worm:** Self-contained and spreads via networks – a logic bomb just waits for a trigger.
- **Trojan:** Pretends to be useful software – a logic bomb can be hidden inside a Trojan.

Detection Methods

Detecting logic bombs is difficult because they remain dormant until triggered. However:

- **Code review:** Careful inspection of source code.
- **Behavioral analysis:** Looking for unusual conditions or hidden triggers.
- **System monitoring:** Watch for suspicious file modifications or scheduled tasks.
- **Integrity checking:** Compare files with known-good versions.

Defense and Prevention

- **Access control:** Limit who can modify source code and system scripts.
- **Change management:** Every code change should be reviewed and approved.
- **Regular audits:** Perform frequent system and code audits.
- **Monitoring triggers:** Watch for suspicious date checks, file conditions, or login-specific scripts.

- **Insider threat programs:** Since logic bombs are often insider-driven, monitor employee behavior.

Logic bombs remind us that cybersecurity is not only about external hackers.

Insider threats can be just as dangerous, if not more.

Always think about both **technical security** and **human behavior** in your defense strategies.

Trojan Horses

A **Trojan Horse** is a seemingly useful or legitimate program that contains **hidden malicious code**.

- When executed, it performs its normal expected function, **but also executes harmful actions** in the background.
- The name comes from the famous **Greek myth of the Trojan Horse**, where attackers hid inside a wooden horse to infiltrate Troy.

How a Trojan Horse Works

1. **Disguise** – The attacker disguises the program as something useful (a utility, installer, game, or update).
2. **Execution** – The victim runs the program, believing it is safe.
3. **Malicious Action** – In addition to the intended function, the Trojan secretly executes harmful code.
4. **Attacker Gains Advantage** – This may allow the attacker to steal data, install backdoors, or control the system.

Example Scenarios

- A program that appears to **list a user's files** but secretly changes file permissions, allowing the attacker to read all files.
- A **compiler modified** to insert malicious code into programs during compilation (e.g., modifying the login program to include a backdoor password).
- A "free game" download that actually **installs spyware** on the victim's device.

Models of Trojan Horses

Trojan horses typically fall into three models:

1. **Original function + malicious function**
    - The program continues to perform its expected task, **but also runs hidden malicious actions**.
    - Example: A calculator app that also records keystrokes.
2. **Modified function + disguise**
    - The program performs its function but in a modified, malicious way.
    - Example: A login program that works normally but **stores passwords for the attacker**.
    - Another example: A task manager that hides certain malicious processes.
3. **Replacement function**

- The program **completely replaces** the original functionality with something harmful.
- Example: A fake antivirus program that "scans" but actually installs malware.

Key Characteristics

- **Deceptive:** Appears to be harmless or useful.
- **User-driven:** Requires the user to install or run it.
- **Payload:** Can steal data, open backdoors, disable security, or install other malware.
- **Difficult to detect:** Especially if it continues to provide expected functionality.

Real-World Examples

- **Zeus Trojan:** Used for stealing online banking credentials.
- **Emotet:** Originally a Trojan that spread through malicious email attachments.
- **Fake antivirus software:** Programs that pretend to clean your system but actually infect it.

Detection and Defense

- **Antivirus/EDR solutions:** Can detect known Trojan signatures.

- **Behavior analysis:** Monitoring unusual activity (e.g., a text editor trying to access the network).
- **User awareness:** Educating users not to download untrusted software.
- **Access controls:** Limiting installation rights on systems.
- **Code review:** For critical systems, review compilers and core software.

Don't trust software just because it looks useful. Always verify the source, apply security controls, and train users—because many attacks succeed not through technology, but through **deception**.

Mobile Code:

A program or application that can **move or be transferred** from one computer to another through emails, documents, websites, or removable media.

It is often embedded inside **HTML emails, email attachments, documents, or web pages**. Mobile code may be useful (e.g., Java applets, scripts, macros), but it is frequently abused for malicious purposes

Mobile code is not always malicious, but it becomes dangerous when attackers use it as a **delivery mechanism** for viruses, worms, or Trojans.

Mobile code itself is not always bad, but when used maliciously, it becomes MMC.

How Mobile Code Spreads

The most common ways mobile code is used for malicious activity are:

1. **Cross-Site Scripting (XSS)** – Code injected into websites to steal data or execute unauthorized actions.
2. **Interactive & Dynamic Websites** – Attackers hide scripts in web pages that run automatically when opened.
3. **Email Attachments** – Malicious macros or scripts hidden inside documents or files.
4. **Untrusted Downloads** – Applications or tools from unreliable sources that contain hidden malware.
5. **Removable Media** – USB drives carrying mobile code that executes when plugged into a computer.

Characteristics of Mobile Code

- **Portable:** Moves easily across networks and storage media.
- **Embedded:** Hidden inside legitimate files, emails, or websites.
- **Executable Content:** Runs automatically in some environments without user awareness.
- **Varied Impact:** Can be harmless, disruptive, or destructive.

Examples

- A JavaScript code on a website that silently redirects users to a malicious site.

- A Microsoft Word document containing a macro virus.
- A PDF file with embedded code that executes when opened.
- A USB stick infected with an autorun worm.

Risks of Mobile Code

- Unauthorized data access.
- System modification or corruption.
- Spreading malware to other machines in the network.
- Data theft and credential harvesting.
- Disabling security mechanisms.

Defense Against Malicious Mobile Code

To protect against MMC, organizations and users should:

- **Disable or restrict mobile code execution** in browsers and email clients.
- **Use antivirus and endpoint protection** to detect malicious attachments or scripts.
- **Apply software patches** to fix vulnerabilities in applications.
- **Educate users** about risks of opening unknown attachments or downloads.
- **Use network filtering** to block suspicious scripts or content.
- **Restrict removable media use** (USB drives)

Mobile code shows how **flexibility in modern software** can also create **security vulnerabilities** .As cybersecurity professionals, we must carefully balance **usability and security**, ensuring mobile code does not become a hidden weapon for attackers

H.W/

1. A logic bomb is best described as:

2. Which of the following is a trigger for a logic bomb?

3. The main difference between a virus and a logic bomb is:

4. What happens when a logic bomb is triggered?

5. Who most often plants logic bombs?

6. A logic bomb is similar to:

7. In the UBS PaineWebber case, the logic bomb caused:

8. Which method can help detect a logic bomb?

9. Which is NOT a characteristic of logic bombs?

10. A logic bomb may:

11. A Trojan horse is named after:

12. The main feature of a Trojan horse is:

13. A Trojan requires:

14. Which Trojan model keeps original function AND adds malicious function?

15. A login program that collects passwords is an example of:

16. A fake antivirus that only installs malware is:

17. Which is NOT a Trojan characteristic?

18. Which is a famous Trojan used for banking theft?

19. Emotet originally spread through:

20. Which defense can detect Trojans?

21. Mobile code is:

22. Malicious mobile code (MMC) includes:

23. Which is NOT a way mobile code spreads?

24. An example of mobile code is:

25. Which defense helps against malicious mobile code?