CEN429 Secure Programming Week-1

Introduction to Secure Programming and Computer Viruses

Author: Asst. Prof. Dr. UÄŸur CORUH

Contents

List of Figures

List of Tables

1 CEN429 Secure Programming

1.1 Week-1

1.1.0.1 Course Plan and Communication, Secure Programming, and Computer Viruses

Download

- PDF[1]
- DOC[2]
- SLIDE[3]

[1] pandoc_cen429-week-1.en_doc.pdf
[2] pandoc_cen429-week-1.en_word.docx
[3] cen429-week-1.en_slide.pdf

- PPTX[4]

1.1.1 Outline

- Secure Programming and Computer Viruses
- Application Protection Plan
  - Code Splitting
  - Code Verification
  - Timing
  - Protocol Monitoring
- Computer Viruses
  - Virus Characteristics
  - Virus Types
  - Virus Countermeasures
- Attack Trees and Security Models
- Attack Methods
- Secure Communication Goals

1.2 Application Protection Plan

1.2.1 1. Code Splitting

1.2.1.1 Theoretical Explanation: Code splitting refers to moving operations performed in an untrusted environment to a trusted one, thus minimizing security vulnerabilities.

1.2.1.2 Practical Application:

- **Task:** Implement a system where encryption is handled on the server instead of the client in a client-server model. This method ensures that critical operations are executed in a secure environment.

1.2.2 2. Code Verification

1.2.2.1 Theoretical Explanation: We verify that a system performs as expected by asking an untrusted device or site, "Are you running the correct code?"

1.2.2.2 Practical Application:

- **Task:** Develop a system that checks whether an application can solve specific mathematical problems correctly and quickly during runtime. The system will cease operations if it cannot prove its correctness.

1.2.3 3. Timing

1.2.3.1 Theoretical Explanation: In an untrusted system, a computational challenge is set up and expected to be solved within a certain timeframe. This technique prevents attackers from having enough time for analysis.

1.2.3.2 Practical Application:

- **Task:** Create a "Time-Based Question-Answer" application. If no response is received within the specified time, terminate the session.

1.2.4 4. Protocol Monitoring

1.2.4.1 Theoretical Explanation: By monitoring protocol flows during data transfer, we can detect potential security vulnerabilities or malicious activities.

[4] cen429-week-1.en_slide.pptx

1.2.4.2 Practical Application:

- **Task:** Set up a logging system that monitors HTTP requests on a web server. Block users when suspicious requests are detected.

1.3 Computer Viruses

1.3.1 1. Virus Characteristics

- **Dormant State:** A virus may remain inactive for a while, avoiding detection.
- **Propagation:** It spreads to new files or systems.
- **Triggering:** An event determines when the virus will activate.
- **Action:** The virus performs a harmful task, often called a "payload."

1.3.1.1 Practical Application:

- **Task:** Create a simulation where the virus stays dormant and activates on a specific date to delete a file.

1.3.2 2. Virus Types

- **Program/File Virus:** Infects program files.
- **Macro Virus:** Infects Word/Excel documents and executes when the document is opened.
- **Boot Sector Virus:** Infects the boot sector of a hard drive, activating when the computer starts.

1.3.2.1 Practical Application:

- **Task:** Create a simulation showing how different virus types work. Each type should be triggered by different events.

1.3.3 3. Virus Countermeasures

- **Signature-Based Detection:** Detects viruses based on known code snippets.
- **Encryption:** Viruses can encrypt their code to evade signature-based detection.

1.3.3.1 Practical Application:

- **Task:** Create a simulation of an encrypted virus. The virus's code should be encrypted with a different key each time it is executed.

1.4 Security Models and Attack Trees

1.4.1 1. What is an Attack Tree?

An attack tree helps us understand the strategies an attacker might use to reach a goal. This model visualizes vulnerabilities, aiding in the development of effective defenses.

1.4.1.1 Practical Application:

- **Task:** Build a simple attack tree. For example, model the steps from an SQL injection in a web application to gaining access to the database.

1.4.2 2. Cost Modeling

Each step in an attack has a cost. These costs can be calculated to make it harder for an attacker to achieve their goal. In an attack tree, costs are assigned to each node, and the least costly path is calculated.

1.4.2.1 Practical Application:

- **Task:** Create a simulation that calculates the cost of each step in an attack tree. Simulate reaching the goal with the lowest cost.

1.5 Attack Methods

1.5.1 1. Dynamic Analysis

This helps understand which parts of a program are triggered during execution and how the system behaves based on inputs.

1.5.1.1 Practical Application:

- **Task:** Develop a tracker that monitors which functions of software are called during runtime and what inputs trigger these functions.

1.5.2 2. Static Analysis

This involves analyzing the source code or compiled version of a program. This analysis helps identify potential security vulnerabilities.

1.5.2.1 Practical Application:

- **Task:** Use a disassembler to analyze the compiled code of a simple program and identify weaknesses.

1.5.3 3. Editing Phase

After understanding the internal workings of software, an attacker may edit the program to disable license checks or remove restrictions.

1.5.3.1 Practical Application:

- **Task:** Edit a program's binary to bypass a license check. Track which restrictions are lifted.

1.6 Secure Communication Goals

- **Mutual Authentication:** Both parties in a communication verify each other.
- **Key Revocation:** Revocation of invalid keys.
- **High Performance:** Speed and low latency are essential for secure communication.

1.6.0.1 Practical Application:

- **Task:** Create a simple authentication protocol where both parties mutually verify each other.

1.7 Week Summary and Next Week

1.7.1 This Week:

- **Application Protection Plan**
- **Computer Viruses and Types**
- **Attack Trees and Security Models**
- **Attack Methods and Secure Communication Goals**

1.7.2 Next Week:

- **Data Security**
- **Cryptographic Techniques**
- **Applied Encryption**

$$End - Of - Week - 1$$