Project Proposals for Cybersecurity Research and Reverse Engineering

Sazid Hasan

August 22, 2024

Project 1: Simulating Computer Virus Behavior for Cybersecurity Research

1. Project Title

Simulating Virus Behavior in a Secure Environment for Cybersecurity Research

2. Introduction and Background

- **Objective**: The goal of this project is to create a controlled simulation of virus-like behaviors to study how malware interacts with systems and how antivirus software detects and mitigates threats. This simulation will help deepen the understanding of modern security measures and vulnerabilities.
- **Background**: Computer viruses are among the most common forms of malware, posing serious threats to individual users and organizations alike. To develop better defenses, it is important to study how viruses operate, spread, and evade detection. However, creating and testing actual malicious software would pose ethical and legal risks. Therefore, this project will focus on simulating virus behavior in a secure environment for educational purposes.

3. Problem Statement

Cybersecurity measures need constant improvement to counter new malware techniques. This project aims to simulate virus behaviors and study how antivirus programs react to them. The purpose is to identify gaps in current detection methods and propose ways to strengthen cybersecurity defenses without causing harm.

4. Project Scope and Objectives

- **Scope**: Develop a virus-like program with behaviors such as file replication, evasion techniques, and system modifications. Conduct all testing in an isolated, sandboxed environment. Focus on monitoring antivirus responses and system integrity.
- **Objectives**:
  1. Create a sandboxed environment to contain the virus simulation.
  2. Develop virus-like behaviors (file replication, encryption, code obfuscation).
  3. Analyze how antivirus programs detect or fail to detect the simulated virus.
  4. Document insights and propose cybersecurity improvements.

5. Methodology

1. **Environment Setup**: Set up a secure virtual environment with a sandbox (VirtualBox, VMware) isolated from external networks.
2. **Simulation Development**: Write a program that simulates common virus behavior, such as file manipulation, stealth techniques, and basic network propagation. Ensure the virus is non-malicious and safe to use within the sandbox.
3. **Testing and Monitoring**: Test antivirus detection and system integrity during and after virus simulation using tools like Sysmon, Wireshark, and antivirus logs.
4. **Analysis**: Review data collected from system and antivirus logs, identify detection patterns, and evasion methods.
5. **Reporting**: Summarize findings and propose improvements to detection techniques.

6. Ethical Considerations

- **No harm**: All simulations will be conducted in a closed environment to prevent any real-world damage.
- **Legal compliance**: The simulated virus will not have any malicious payload, ensuring compliance with cybersecurity laws.

7. Timeline

- **Weeks 1-2**: Set up the environment and research virus behaviors.
- **Weeks 3-4**: Develop the simulation program.
- **Weeks 5-6**: Test the virus simulation and collect data.
- **Week 7**: Analyze results.
- **Week 8**: Document findings and submit the final report.

8. Expected Outcomes

A functional, controlled simulation of virus-like behavior, insights into antivirus detection mechanisms, and recommendations for improving malware detection and prevention.

Project 2: Learning Reverse Engineering by Cracking Small Software

1. Project Title

Learning Reverse Engineering through Software Cracking in a Legal and Controlled Environment

2. Introduction and Background

- **Objective**: The aim of this project is to learn reverse engineering techniques by analyzing small software applications to understand how they work and identify security vulnerabilities. The project focuses on educational purposes and legal reverse engineering techniques, avoiding any illegal cracking or piracy activities.
- **Background**: Reverse engineering is the process of analyzing software to understand its structure, function, and behavior. It is commonly used in cybersecurity to find vulnerabilities, detect malware, and ensure software integrity. This project will help develop practical reverse engineering skills using disassembly and decompilation tools, focusing on legal, ethical analysis of small programs.

3. Problem Statement

Software vulnerabilities are often hidden deep within application code. Reverse engineering allows cybersecurity professionals to uncover these weaknesses. However, reverse engineering can be legally challenging if not conducted in a responsible manner. This project will focus on learning the technical aspects of reverse engineering while complying with legal guidelines.

4. Project Scope and Objectives

- **Scope**: Analyze small, legally permitted software binaries using reverse engineering tools. Avoid illegal cracking or redistribution of proprietary software.
- **Objectives**:
  1. Gain a thorough understanding of reverse engineering tools and techniques.
  2. Reverse-engineer small software applications (e.g., freeware, open-source, or custom-built software) in a legal manner.
  3. Identify weaknesses, such as licensing checks or obfuscated code, to understand how software protections work.
  4. Document and demonstrate knowledge gained from analyzing the software.

5. Methodology

1. **Software Selection**: Choose legal targets such as open-source software, free trials (with permission), or self-made programs for reverse engineering.
2. **Tool Setup**: Install and configure reverse engineering tools like IDA Pro, Ghidra, and OllyDbg.
3. **Analysis**: Disassemble the target software and analyze its code, focusing on understanding the program flow, identifying key functions, and uncovering any protections. Experiment with bypassing basic protections purely for educational purposes.
4. **Documentation**: Document the reverse engineering process, including tools used, steps taken, and insights gained from software analysis.

6. Ethical and Legal Considerations

- **Ethical**: The project will respect software copyrights, and no illegal cracking, modification, or redistribution of proprietary software will take place.
- **Legal**: Only software that permits reverse engineering (e.g., open-source or self-developed) will be used.

7. Timeline

- **Weeks 1-2**: Choose target software and set up reverse engineering tools.
- **Weeks 3-4**: Begin reverse engineering analysis, focusing on understanding code structures.
- **Weeks 5-6**: Explore software protections and attempt to bypass them ethically for learning purposes.
- **Week 7**: Analyze findings and compile documentation.
- **Week 8**: Prepare the final report and demonstration.

8. Resources Required

Reverse engineering tools (IDA Pro, Ghidra, OllyDbg) and target software for analysis (open-source, freeware, or custom-developed).

9. Expected Outcomes

Improved understanding of software structure and reverse engineering techniques, practical experience with reverse engineering tools, and a detailed report of the reverse engineering process and insights gained.