Ransomware Simulation – Detailed Project Report

Executive Summary

This project is a controlled ransomware simulation designed to demonstrate the full lifecycle of a ransomware attack without introducing any real-world risk. The goal is to educate, train, and prepare security professionals, red teamers, and incident responders on how ransomware operates, how it impacts systems, and how it can be mitigated or recovered from. The simulation focuses on realistic behaviors of ransomware while ensuring 100% safety: no real cryptography or data destruction, operating only on isolated lab systems, and a fully reversible process using a manifest-based restore mechanism.

Project Objectives

- Simulate ransomware behavior safely to study its operational flow.
- Enhance red team capabilities by practicing the attack lifecycle.
- Improve defensive understanding to identify prevention and response strategies.
- Demonstrate professional reporting through a client-ready document.

Lab Environment

The simulation is conducted in a fully isolated virtual machine network to ensure zero risk to production systems.

- Attacker Machine: Kali Linux 2024 with Python 3
- Victim Machine: Windows 10 Pro (VM) / Ubuntu Linux
- Virtualization: VirtualBox with host-only network
- Tools: Python 3, Draw.io (diagrams), GitHub for version control
- Safety: No internet connection, dummy files only

Technical Implementation

- File Discovery: Scans target directory while skipping manifest and ransom note.
- Simulated Encryption: Renames files with a .locked extension (no cryptography).
- Ransom Note Generation: Drops a README_RESTORE_FILES.txt file with fake payment instructions.
- Manifest Tracking: Records original and renamed filenames with timestamps.
- Restoration Process: Reverts files using the manifest and removes ransom note.

Attack Lifecycle Simulation

Simulates MITRE ATT&CK; Technique T1486 – Data Encrypted for Impact:

- Initial Access: In real attacks via phishing, RDP brute force, or downloads. In simulation, executed manually in target directory.
- Execution: File renaming begins and ransom note is created.
- Impact: Files appear 'locked' and unusable until restored.
- Recovery: Restoration process reverts files to original state.

Usage Instructions

```
- Create Dummy Files: python3 ransomware_simulation.py --target test_files --create
- Simulate Attack: python3 ransomware_simulation.py --target test_files
- Restore Files: python3 ransomware_simulation.py --target test_files --restore
```

Sample Ransom Note

```
Your files have been encrypted! To recover them, send 1 Bitcoin to the address
below: [FAKE BITCOIN ADDRESS] Contact: fakehacker@example.com (This is a simulated,
harmless demonstration. No real encryption was performed.)
```

Defensive Considerations

- Use EDR solutions to detect suspicious file renaming patterns.
- Block unauthorized script execution using AppLocker or Linux MAC.
- Maintain regular offline backups.
- Segment networks to reduce lateral movement.
- Conduct incident response drills regularly.

Potential Extensions

- Integration with a simulated Command and Control (C2) server.
- Simulation of data exfiltration before encryption.
- Integration with SIEM tools to test alerting capabilities.

Conclusion

This ransomware simulation serves as a complete red team training module. It demonstrates offensive skills through attack simulation, defensive knowledge with mitigation strategies, and professional communication via detailed reporting. The project can be used for stakeholder education, security team training, and forensic practice.

Legal Disclaimer

This simulation is intended solely for educational purposes in a controlled, isolated environment. Running this code on production systems or without explicit permission is illegal and unethical.