Project Proposal

Reconstructing and Analyzing the Morris Worm in a Simulated Internet

Overview and Motivation

In this project, we will reconstruct and analyze the Morris Worm by recreating a vulnerable network environment and executing the original worm code or a faithful reimplementation in a sandbox setting. By observing how the malware propagates across interconnected machines, we aim to understand the vulnerabilities it exploits and the mechanisms behind early internet attacks.

We will then extend this setup to a more modern operating system to compare how architectural and security improvements affect malware behavior, providing a concrete, experimental perspective on the evolution of network security.

Project Goals

1. Reproduce a historically accurate environment vulnerable to Morris Worm style attacks.
2. Execute the original worm code or a minimally modified equivalent in a safe, sandbox network.
3. Observe and record how malware propagates across a simulated internet.
4. Build tooling to visualize infection paths and network traversal.
5. Compare outcomes between legacy systems and more modern operating systems.

Minimum Viable Product (MVP)

The MVP is a functioning simulated network in which a Morris Worm style program successfully infects multiple machines. The system will dynamically log infection events, demonstrate lateral movement, and provide verifiable proof that each host was compromised as intended.

Key MVP criteria:

- A working emulation of multiple connected vulnerable Unix machines.
- Successful compilation and execution of the worm code.

- Instrumentation that records infection timestamps and host identifiers.

Technical Approach

Environment and Infrastructure

We will evaluate multiple approaches to virtualization and isolation, including virtual machines, emulators, and Docker based container networks. A key question is whether Docker containers can accurately model historical vulnerabilities or whether full system emulation is required. We will investigate the limitations of containers with respect to kernel behavior, privilege escalation, and legacy service support.

To simulate the internet, we plan to construct a virtual network using Docker networking and Docker Compose or emulator level networking tools. This network will include multiple hosts, configurable routing behavior, and controlled exposure of vulnerable services.

Vulnerabilities and Exploits

The Morris Worm exploited several weaknesses, including buffer overflows, weak authentication, and trust relationships between machines. We will study these vulnerabilities in detail and replicate them where feasible, including vulnerable versions of services such as fingerd.

Where original exploits cannot be executed verbatim, we will simulate their effects using equivalent vulnerabilities or custom implementations. This includes shellcode injection and stack based buffer overflow scenarios designed specifically for educational analysis.

Instrumentation and Analysis

Each host in the network will provide proof of infection through logs, file system artifacts, or process level indicators. All attack attempts and successful infections will be recorded in a centralized repository. We also plan to construct a visual map showing how the worm traverses the network over time, highlighting entry points, propagation paths, and reinfection behavior.

Challenges and Risks

Key challenges include accurately reproducing historical vulnerabilities, compiling legacy code with modern toolchains, and learning sufficient emulator and networking concepts within the project timeline. There is also a tradeoff between realism and feasibility, especially when deciding between containers and full system emulation.

We will mitigate these risks by starting with a minimal setup, validating each component independently, and progressively increasing complexity.

Timeline and Responsibilities

Week 2

- Complete prerequisite labs by the start of Week 3.

Week 3

- Changwoo: Evaluate Docker containers and follow the Rapid7 walkthrough.
- Nathan: Research prior reconstructions and available tooling.
- Lee: Research Morris Worm exploits and vulnerability mechanisms.

Weeks 4 to 5

- Build sandboxed environments and virtual networks.
- Validate networking, services, and exploit prerequisites.
- Test whether the worm code runs on a single host.

Weeks 6 to 7

- Execute the worm across the network to complete the MVP.
- Collect data and generate infection visualizations.

References

- https://www.rapid7.com/blog/post/2019/01/02/the-ghost-of-exploits-past-a-deep-dive-into-the-morris-worm/
- https://www.tuhs.org/Archive/Distributions/UCB/ (The Unix Archive)
- https://sourceforge.net/projects/bsd42/files/4BSD%20under%20Windows/ (Windows SIMH Unix)
- https://www.cs.unc.edu/~jeffay/courses/nidsS05/attacks/seely-RTMworm-89.html#p4.5 (discusses the individual vulnerabilities)