

MALWARE EXAMINATION ON GNU/LINUX: WIRENET CASE STUDY

Adolfo Montironi, Prof. Marcus Thompson
Department of Computer and Information Technology
Purdue University

Linux.Wirenet

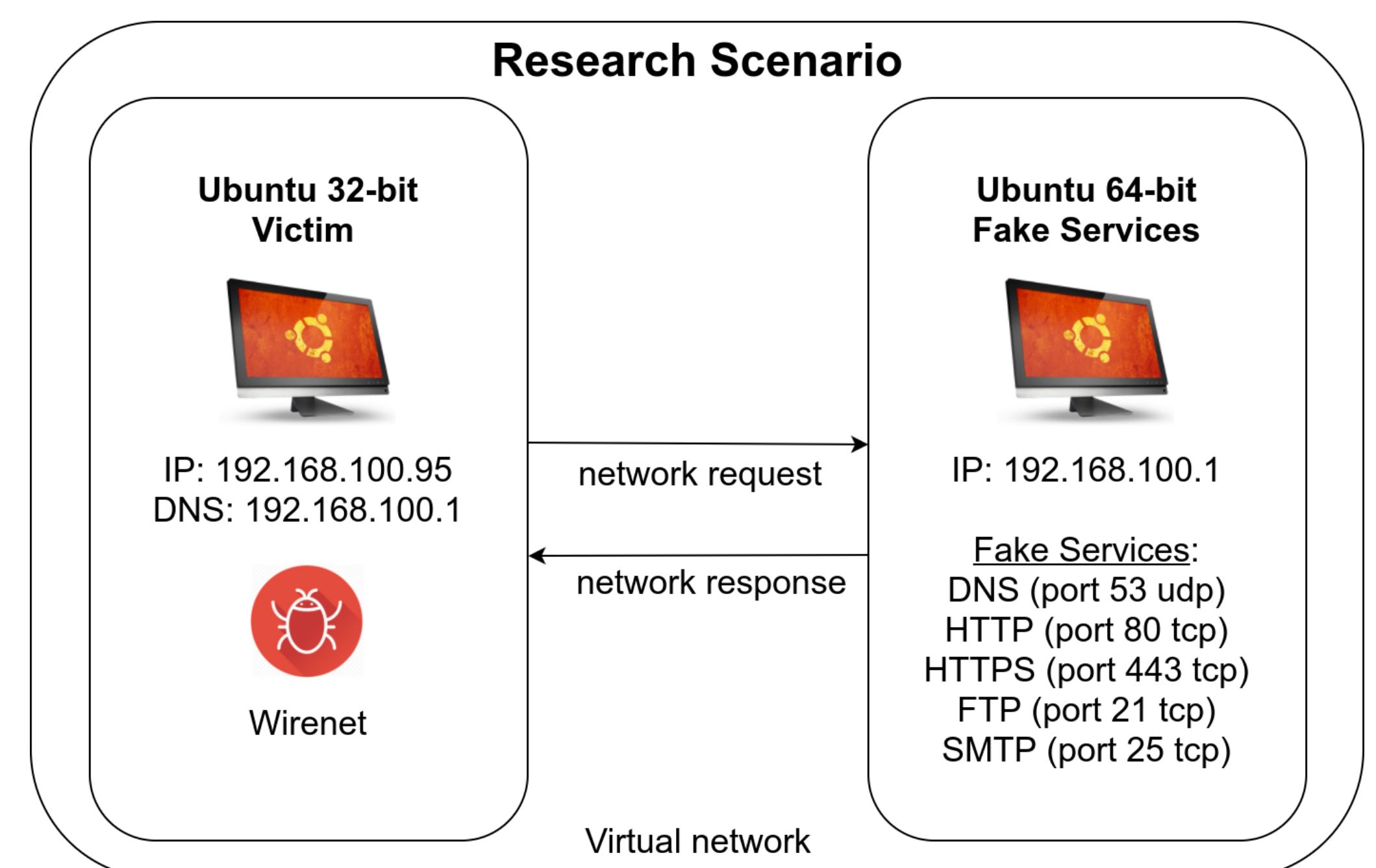
It is a password-stealing trojan designed to target GNU/Linux systems. Once the system is infected, it can access all the user's sensitive data and upload it to a command and control (C&C) server.

Study Objectives

- Utilize static, dynamic, and memory forensic techniques to analyze a system infected with Wirenet.
- Develop a general practical procedure to perform such analysis in a GNU/Linux environment.

Research Process

1. The victim's system was infected.
2. A memory dump was created.
3. Basic static analysis was performed to examine the malware file.
3. Basic dynamic analysis was performed.
4. Memory forensic techniques were used to find memory artifacts.
5. Fake services logs were examined to search for network indicators.



Procedure and Tools

Static Analysis

System details and context:

- lsb_release, uname, mount, df

Malicious file profiling:

- ls, stat, istat, debugfs, file

Hash values and ELF signature:

- md5sum, sha1sum, od, hexdump

Embedded artifacts:

- strings, ldd, nm, readelf, objdump, exiftool, extract

Dynamic Analysis

Process monitoring:

- ps, pstree, top, /proc/<pid>

Filesystem monitoring:

- lsof, fuser

Network monitoring:

- netstat, tcpdump

System and dynamic library calls:

- strace, ltrace

Fake services log:

- /var/log/inetsim/service.log

Memory Forensics

Acquire memory:

- LiME (Linux Memory Extractor) Module

Analyze memory:

- The Volatility Framework

Conclusion

Wirenet was examined as a case study through the application of static analysis, dynamic analysis, and memory forensic techniques. As a result, a practical procedure was developed to deal with malware infections in a GNU/Linux environment.