Technical Report: Memory Forensic Analysis of the Stuxnet Case

Luca Volonterio

January 20, 2026

1 Introduction

This report details the forensic findings from the analysis of the `stuxnet.vmem` memory dump. The investigation focuses on identifying the advanced evasion techniques and rootkit capabilities characteristic of the Stuxnet worm.

2 System Identification

The analysis was performed using the following profile parameters:

- **Operating System:** Windows XP Service Pack 2 (x86)
- **Kernel Architecture:** 32-bit (Non-PAE)
- **Primary Evidence:** Memory image `stuxnet.vmem`

3 Process Hierarchy Analysis

A review of the process tree via the `pstree` plugin revealed significant structural anomalies involving the Local Security Authority Subsystem Service (`lsass.exe`).

3.1 Identification of Malicious Clones

In a standard Windows environment, only one instance of `lsass.exe` should exist, spawned by `winlogon.exe`. The analysis identified three instances:

- **PID 680:** Legitimate process (Parent: `winlogon.exe` - PID 624).
- **PID 868 & 1928:** Malicious clones (Parent: `services.exe` - PID 668).

The presence of multiple instances and the incorrect parentage (Services.exe) are definitive indicators of process manipulation.

4 Evidence of Process Hollowing

The `malfind` plugin confirmed that the malicious `lsass.exe` instances were victims of Process Hollowing.

4.1 Memory Artifacts

At memory addresses `0x80000` and `0x1000000`, the analysis detected the **MZ header** ($4D\ 5A$) within private VAD regions marked with `PAGE_EXECUTE_READWRITE` (RWX) permissions.

- **Technical Impact:** The malware initiated a legitimate process in a suspended state, unmapped the original code, and replaced it with a malicious payload. This allows the malware to execute under the guise of a trusted system process.

5 Kernel-Mode Rootkit Drivers

Analysis of the loaded modules identified two core Stuxnet drivers acting as a kernel-level rootkit:

1. `mrxcls.sys` (Base: `0xf895a000`): Responsible for payload persistence and automatic execution.

2. `mrxnet.sys` (Base: `0xb21d8000`): A file-system filter driver used to hide malicious files on removable media (USB) and intercept industrial control communications.

6 Network and Lateral Movement

The `sockets` analysis identified the infrastructure used for worm propagation:

- **TCP Port 445 (System):** Active SMB listener used for exploiting vulnerabilities such as **MS08-067**.

- **TCP Port 135 (PID 940):** RPC Endpoint Mapper. Stuxnet utilized **RPC-based remote execution** to trigger code on remote targets, facilitating lateral movement without user interaction.

7 Conclusion

The forensic evidence confirms a sophisticated Stuxnet infection. By utilizing **Process Hollowing** into `lsass.exe` and deploying **signed kernel**

drivers, the malware achieved high-level stealth and full SYSTEM privileges. The active RPC and SMB listeners confirm the machine's role in the worm's network-wide propagation.