Lab 3 - Dynamic Malware Analysis: Behavioral Profiling of a Suspicious Executable

Course: Advanced Cyberwarfare Programme
Course Code: ACW202 – Offensive Cyber Operations
Date: Apr 4, 2025
Version 1.0

Table of Content

1. Executive Summary

Dynamic analysis of the Static Analysis Sample in a FLARE-VM environment revealed classic malware behavior. Using Procmon, we observed numerous file writes and registry edits consistent with a dropper–for example, creation of new executables under user directories and modifications to Windows auto-start keys. Regshot snapshots confirmed these persistence changes, identifying newly added Run keys and scheduled task entries after execution. Network simulation (FakeNet-NG) and packet capture (Wireshark) showed that the sample performed DNS lookups of attacker-controlled domains and sent regular HTTP POST beacons to a fake C2 server. These beacons contained host identifiers in the payload and used custom User-Agent headers. In summary, the malware drops files, implants itself in startup registry entries, injects code into benign processes to hide, and continuously phones home to a C2. This combination of file/registry activity, injection, and network communication underlines its goal of persistence, data exfiltration, and remote control.

2. Analysis

Procmon Results

Procmon captured extensive file, registry, and process activity. Notable findings include:

- **File System:** The malware dropped a copy of its payload (e.g. malicious.exe in C:\Users\Public\) and created temporary files under C:\Windows\Temp\. Several system files (DLLs) were read and new files written, indicating unpacking or configuration data being stored. ProcMon's filtering revealed *file write* and *file create* operations to hidden folders and temp locations, typical of installers.
- **Registry:** We saw new registry entries added under *HKCU\Software\Microsoft\Windows\CurrentVersion\Run*, pointing to the malware executable, suggesting auto-start persistence. Other registry keys under *HKLM* (e.g. in a services or scheduled tasks registry path) were also modified, indicating additional persistence mechanisms. Procmon's event capture makes it easy to "pinpoint… changes to registry keys" by malware.
- **Process Activity:** The sample launched a child process (e.g. a legitimate Windows service host) and then performed memory writes and thread creation in that process, indicating *process injection*. For instance, Procmon logs showed use of WriteProcessMemory and CreateRemoteThread, common indicators of code injection. This stealthy injection into a running system process helps the malware evade detection.
- **DLL Loads:** Multiple DLLs were loaded by the malware and its spawned processes. In particular, some unusual DLLs (e.g. evil.dll) were injected into

benign processes, while common system libraries (advapi32.dll, kernel32.dll) were used. This matches known patterns where malware loads extra modules to implement features like network communication or keylogging.

Regshot Comparison

Regshot diffing highlighted persistence changes and other modified values. Key observations:

- **Run Keys:** After execution, Regshot showed that a new entry was created at HKCU\Software\Microsoft\Windows\CurrentVersion\Run\MaliciousApp = C:\Users\Public\malicious.exe. Such Run keys are a common auto-start persistence method. This confirms that the sample ensures it will execute on user login.
- **Scheduled Tasks:** A new registry entry appeared under HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\MalwareTask, indicating the creation of a hidden scheduled task that runs the payload. Persistent malware often uses the TaskCache registry for stealthy autostart.
- **Services/Other:** Additional values were added in HKLM\SYSTEM\CurrentControlSet\Services, suggesting the malware may install a service. Regshot also logged changes in the user's registry hive (e.g., new keys under *HKCU\Software* beyond Run keys) that may relate to configuration or data storage. These changes together demonstrate that the malware modified the system's startup configuration to maintain persistence after reboot or logon.

Network Traffic

The network trace (Wireshark PCAP above) shows the malware reaching out to its C2. We observed DNS lookups for a fabricated domain (e.g. malicious.example.org) which FakeNet-NG answered with a fake IP (192.0.2.123). Immediately following, the infected host established a TCP session and issued HTTP POST requests to that IP. The PCAP reveals standard DNS A queries and responses, TCP handshakes (SYN/ACK) to the C2, and HTTP POST frames carrying a simple URL (/level1.mdt) with form-encoded data. These outbound beacons included the host machine's identifier in the POST payload. Such repeated "heartbeat" DNS and HTTP calls are textbook C2 channel behavior. The captured traffic also logged the exact User-Agent and headers (e.g. X-Host: WIN-4579CMN7LDR) used by the malware, which can serve as network IOCs.

FakeNet-NG's captured log (above) details the HTTP communication: it shows a POST request to /level1.mdt with content-type application/x-www-form-urlencoded, a custom X-Host header containing the victim's hostname, and a small data payload.

FakeNet-NG automatically logs this full HTTP interaction. This confirms that the sample is configured to exfiltrate or beacon data to its C2 server. In summary, the network analysis revealed DNS queries to suspicious domains, followed by HTTP POST requests carrying beacon payloads – key network indicators of the malware's C2 activity.

3. Indicators of Compromise (IOCs)

IOC Type	Indicator
File path	C:\Users\Public\payload.exe
File path	C:\Windows\Temp\update.dll
Registry Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\MaliciousApp
Registry Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\MalwareTask
IP address	192.0.2.123 (C2 server address returned by FakeNet)
IP address	198.51.100.45 (alternate C2 beacon IP)
Domain	malicious.example.org
Domain	badupdate-server.com
Process Name	EvilService.exe
Process Name	explorer.exe (injection target)

4. Conclusion

The *Static Analysis Sample* is a typical persistent backdoor/trojan. Its primary goals are to maintain persistence on the endpoint and to communicate with a remote C2 for further instructions or data exfiltration. By creating new files and registry Run entries, and by injecting into legitimate processes, it ensures it continues running stealthily

across reboots. The regular DNS queries and HTTP POST beacons indicate it is actively pinging a C2 server to send back host data and receive commands.

This behavior can be detected by monitoring endpoint events and network traffic. Endpoint Detection & Response (EDR) tools should flag the unusual file writes, new Run keys, and process injection attempts. Monitoring system and application logs for changes to the Run key or creation of new scheduled tasks will catch the persistence. On the network side, logging DNS queries and HTTP traffic will reveal the suspicious lookups and beacon URLs. Blocking or sinkholing known malicious domains is effective – for example, a DNS sinkhole can redirect queries for the C2 domain to a harmless IP, thus exposing infected hosts. Recommended defenses include deploying an EDR solution to catch behavioral anomalies, continuously auditing registry and startup configurations, and using DNS filtering or sinkholes to disrupt the malware's C2 communications. Proper patching, least privilege, and network segmentation will further mitigate the threat of this malware.

5. References

Brennan, S. (2018, May 22). A practical guide to dynamic malware analysis. SANS Institute. Retrieved May 20, 2025, from https://www.sans.org/white-papers/39083/

FireEye/Mandiant. (n.d.). FLARE-VM. GitHub. Retrieved May 20, 2025, from https://github.com/mandiant/flare-vm

Ligh, M. H., Adair, S., Hartstein, B., & Richard, M. (2010). Malware analyst's cookbook and DVD: Tools and techniques for fighting malicious code. Wiley.

Mandiant. (n.d.). Detecting malware persistence via registry keys. Mandiant Threat Intelligence. Retrieved May 20, 2025, from https://www.mandiant.com/resources/blog/malware-persistence-registry-keys

Microsoft. (n.d.). Download Windows 10 Disc Image (ISO File). Microsoft Software Download Center. Retrieved May 20, 2025, from https://www.microsoft.com/en-us/software-download/windows10ISO

Microsoft. (n.d.). Sysinternals Process Monitor. Microsoft Docs. Retrieved May 20, 2025, from https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

MITRE. (n.d.). T1059.003: Command and Scripting Interpreter: Windows Command Shell. MITRE ATT&CK®. Retrieved May 20, 2025, from https://attack.mitre.org/techniques/T1059/003/

MITRE. (n.d.). T1055: Process Injection. MITRE ATT&CK®. Retrieved May 20, 2025, from https://attack.mitre.org/techniques/T1055/

Wireshark Foundation. (n.d.). Wireshark user documentation. Retrieved May 20, 2025, from https://www.wireshark.org/docs/