Lab 3: Behavioral Anomaly Detection & Evasion Analysis

“it said mp4 but it definitely was not a video”

El Hani Mohammed-Rida

Note : I did this lab out of genuine passion, and I spent a lot of time coding everything by hand (I used LLMs mainly to debug and improve parts of the code). I know I missed the deadline and I understand any penalty for late submission. Still, what matters most to me is the amount of experience and takeaways I gained from this analysis, and I'm really excited to try building my own sandbox someday. :)

1 Overview

Based on the logs from the CAPEv2 sandbox I extracted behavior of the malware sample from the API trace. The run shows a small process tree, file drops, persistence via Startup folder, cleanup via a BAT file, and a suspicious DLL-load trick using a fake `.mp4` extension.

2 Environment Notes

User profile in the trace: `Stati`. The sample initially runs from: `C:\Users\Stati\AppData\Local\Temp\e8207e8c31a8613112223d12.exe`. I treated this as the main process and followed children.

3 Process Tree (observed)

- **PID 2940:** `e8207e8c31a8613112223d12.exe` (main)
- **PID 1884:** `21607.exe` (copied into Startup, then executed)
- **PID 2916:** `cmd.exe` (used for delay + cleanup)
- **PID 3052:** `PING.EXE` ("sleep" via `ping localhost`)
- **PID 2688:** `cmd.exe` (spawned by cmd)

4 Key Behavioral Findings

4.1 Install / Working Directory

The sample creates a working directory: `C:\Users\Stati\AppData\Local\zzStati`. Inside it, it drops two scripts: `Stati.vbs` and `Stati.bat`. (So yes, the malware is basically doing homework too: *write scripts, run scripts, repeat.*)

4.2 Evasion Trick: "DLL in an MP4 costume"

The weirdest (and most important) behavior:

- It copies `C:\Windows\System32\ntdll.dll` to `C:\Users\Stati\AppData\Local\zzStati\slideshow.mp4`.
- Then `21607.exe` loads `slideshow.mp4` using `LdrLoadDll`.

This looks like extension-masquerading: a DLL (or DLL-like payload) is stored with a non-DLL extension and still loaded into memory. The copy source being `ntdll.dll` suggests the goal may be hook-evasion / unhooking experiments (or just a noisy trick to confuse analysts).

4.3 Persistence: Startup Folder

The main process copies itself into the Startup folder as: `...\Startup\21607.exe` and executes it (via `ShellExecuteExW`). This is a basic persistence method: "start me again next login".

4.4 Cleanup / Anti-Forensics: `del.bat`

A BAT file is dropped and executed: `C:\Users\Stati\AppData\Roaming\del.bat`. Later, a command is executed: `cmd /c del "... \del.bat"` to remove the BAT itself. So the script basically rage-quits and deletes its own notes. Mood.

4.5 Anti-Analysis Spice (lightweight)

- A delay trick is used: `ping localhost -n 3`.
- The trace resolves `IsDebuggerPresent` from `kernel32.dll`.

No strong VM-detection was obvious from this trace, but the sample does show the usual "are you watching me?" vibes.

4.6 Network Activity

No real outbound network behavior was observed. I saw `WSAStartup` and local `ping localhost`, but no external connections/domains/IPs in the trace.

5 Provenance / Behavior Graph

I generated a provenance graph showing: `ntdll.dll` → `slideshow.mp4` → loaded by `21607.exe`, plus the BAT drop/cleanup edges.

- Static image: `outputs/provenance_graph.png`
- Interactive HTML (recommended): `provenance_graph.html`

6 Conclusion

Overall, the sample behaves like a small loader with: (1) a working directory under `%LOCALAPPDATA%`, (2) persistence via Startup folder, (3) cleanup via a self-deleting BAT, and (4) the standout move: copying `ntdll.dll` into a fake `.mp4` and loading it with `LdrLoadDll`. If I had to summarize it in one sentence: *it tries to look like a video file, but it definitely wants to be a DLL when it grows up.*

7 Indicators of Compromise (IOCs)

Type	**Value**	**Notes**
file	`C:\Users\Stati\AppData\Local\zzStati\slideshow.mp4`	Masqueraded DLL (loaded via `LdrLoadDll` despite `.mp4`)
file	`C:\Windows\System32\ntdll.dll`	Copy source for `slideshow.mp4` (system DLL)
file	`C:\Users\Stati\AppData\Roaming\del.bat`	Cleanup BAT (dropped and later deleted)
process	`21607.exe`	Executed from Startup (persistence) and performs the suspicious DLL load
process	`e8207e8c31a8613112223d12.exe`	Initial sample / main process (drops scripts, performs copy/rename)
process	`cmd.exe`	Used for cleanup and command execution (`/c del ...`)
process	`ping.exe`	Used as a delay trick (`ping localhost -n ...`)
process	`dfrgui.exe`	Referenced during execution (Windows binary)
script	`stati.vbs`	Dropped/used scripting component (VBS launcher)
script	`stati.bat`	Dropped/used scripting component (BAT executed via VBS)
COM / script object	`WScript.Shell`	Used by VBS for process execution
COM / script method	`oShell.Run`	VBS method used to run commands (likely hidden window)
loaded module	`kernel32.dll`	Windows API module observed in trace
loaded module	`advapi32.dll`	Registry/security-related API module observed
loaded module	`shell32.dll`	Shell / execution API module observed
loaded module	`user32.dll`	UI API module observed
loaded module	`setupapi.dll`	Setup/installation API module observed
loaded module	`comctl32.dll`	Common controls module observed
loaded module	`netapi32.dll`	Networking/admin API module observed
loaded module	`mscoree.dll`	.NET runtime loader module observed
file	`sortdefault.nls`	NLS/locale file referenced by Windows during execution

Table 1: Indicators and artifacts extracted from CAPE dynamic analysis.