CTI Report Mapping

Students:

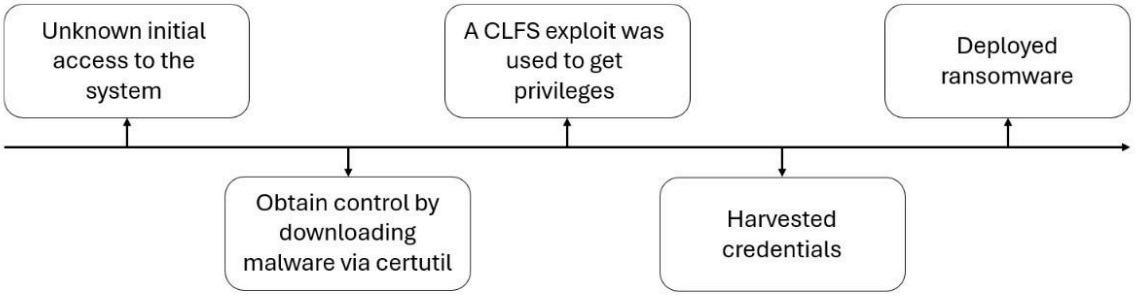
- Yossef Okropiridze, yossiu1@gmail.com
- Michael Naftalishen, michaelnafi99@gmail.com

Link to the Article:

Exploitation of CLFS zero-day leads to ransomware activity | Microsoft Security Blog

Schema Description:

An adversary group exploited a zero-day kernel vulnerability in the Windows Common Log File System (CLFS) to escalate from a standard user to system privileges. It hasn't been determined what initial access vectors led to the devices being compromised by adversary group but afterwards, they delivered a malicious MSBuild payload downloaded via *certutil*, used the CLFS exploit to overwrite a process token and inject into high-privilege processes, dumped LSASS memory with *procdump* to harvest credentials, and then deployed ransomware that deleted backups, cleared event logs and encrypted files (ransom note !_READ_ME_REXX2_!.txt).



***Tactics, Techniques and Behaviors*:**

- **Command and Control: T1105 — Ingress Tool Transfer.**
  Observation: The adversaries used certutil to download an MSBuild file (hosted on a compromised third-party site) that contained an encrypted payload.

- **Defense Evasion: T1140 — Deobfuscate/Decode Files or Information**
  Observation: The downloaded MSBuild file contained an encrypted payload that was decoded/processed at runtime (MSBuild + certutil used to deliver and activate the payload).

- **Defense Evasion: T1211 — Exploitation for Defense Evasion**
  Observation: The exploit targets a vulnerability in the CLFS kernel driver. It's notable that the exploit first uses the *NtQuerySystemInformation* API to leak kernel addresses to user mode.
- **Defense Evasion, Privilege Escalation: T1134 — Access Token Manipulation**
  Observation: The exploit overwrote the process token (article notes token bits set to 0xFFFFFFFF), effectively granting full privileges to the process - matching access-token manipulation/impersonation behaviors.

- **Defense Evasion, Privilege Escalation: T1055 — Process Injection (and subtechniques)**
  Observation: After exploitation, the attackers injected into winlogon.exe (process injection / in-memory techniques) to execute in a high-privilege context.

- **Credential Access: T1003.001 — OS Credential Dumping: LSASS Memory**
  Observation: The threat actors used procdump.exe to dump LSASS memory to harvest credentials (Sysinternals procdump -ma lsass.exe).

- **Impact: T1490 — Inhibit System Recovery**
  Observation: Commands recorded include deleting backup catalogs (*wbadmin* delete catalog -quiet) and disabling recovery options - typical ransomware behaviour to prevent restoration.

- **Defense Evasion: T1070.001 — Indicator Removal: Clear Windows Event Logs**
  Observation: The actors ran *wevtutil* cl Application to clear event logs and remove forensic evidence.

- **Impact: T1486 — Data Encrypted for Impact**
  Observation: The campaign included file encryption and placement of a ransom note (!_READ_ME_REXX2_!.txt), consistent with ransomware impact techniques.

Summary:

This attack is unusual because the adversaries initial access method is unknown, yet once the zero-day CLFS kernel vulnerability showed they immediately leveraged it to leak kernel addresses and overwrite access tokens. Instead of common loaders, they used certutil to deliver an encrypted, runtime-decoded payload. This blend of mystery entry point, true kernel-level zero-day exploitation, and system-process injection makes the operation far more advanced.