Detecting Cobalt Strike Beacons

By SOCFortress

Published: 2022-02-20 · Archived: 2026-04-05 19:04:00 UTC

Introduction

Cobalt Strike is a commercial tool for adversary **simulation**.

Created by Raphael Mudge in 2012, Cobalt Strike was one of the first public red team command and control frameworks.

It's also used by threat actors and the first activity detected for malicious purposes dates back to 2016.

The main components of Cobalt Strike are a **C2** server and a **beacon** installed on compromised machines.

The Cobalt Strike beacon comes with a number of capabilities, including a command-line interface. The beacon allows the execution of **scripts**, or commands native to the machine's **operating system**.

Cobalt strike for malicious purposes is known to be used by more than **50** threat actor groups. Its use is very common in **ransomware** attacks.

In terms of beacon types and methods used for connecting to the C2 servers the most common are **HTTP** (~67%), **HTTPS** (~29%) and **DNS** (~3%). Including JITTER in the beacon trying to avoid detection has been detected in roughly 15% of all the beacons analysed.

When it comes to spawning, around 90% of all beacons analysed were spawning **rundll32.exe** as the main process for lateral movement.

Sophisticated attacks using Cobalt strike beacons, like Nobelium usage of Cobalt strike linked to the **SolarWinds** campaign, try to evade detection by fine-tuning many of the configurable options.

PREVENTION, DETECTION AND RESPONSE TOOLS

- Static File Hash analysis: **Windows Defender**.
- Endpoint Telemetry: **Sysmon** (Sysinternals).
- EDR agent and SIEM: **Wazuh**.
- Threat Intel platform: **MISP**

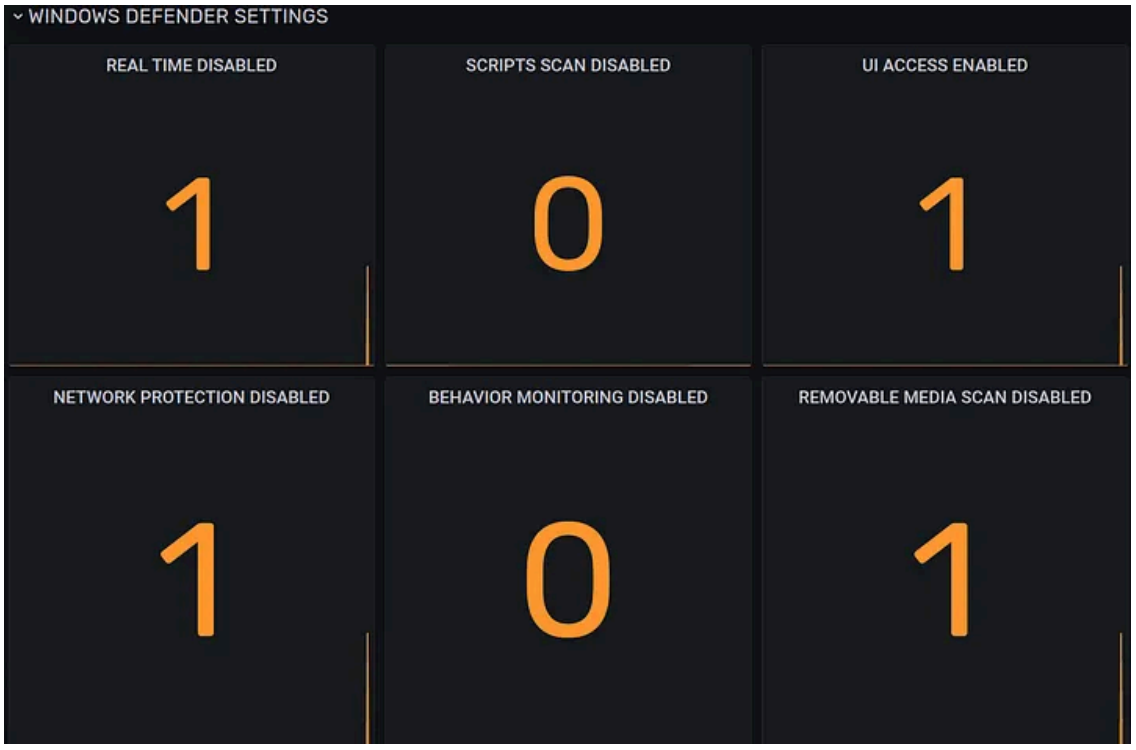
PREVENTION.

Many of the Cobalt Strike beacons in the wild and additional payloads downloaded as part of the attack chain are going to be flagged and removed by Windows Defender via local file analysis (file hash) or its real time analysis engine.

From a prevention point of view some key points are:

- Collecting, reviewing and alerting on Windows Defender Settings not matching minimum criteria (Real Time, Enabled, Exclusion extensions, Excluded folders, removable media, etc.). Deploy an inventory collection script (more info here) and alert on inadequate settings.

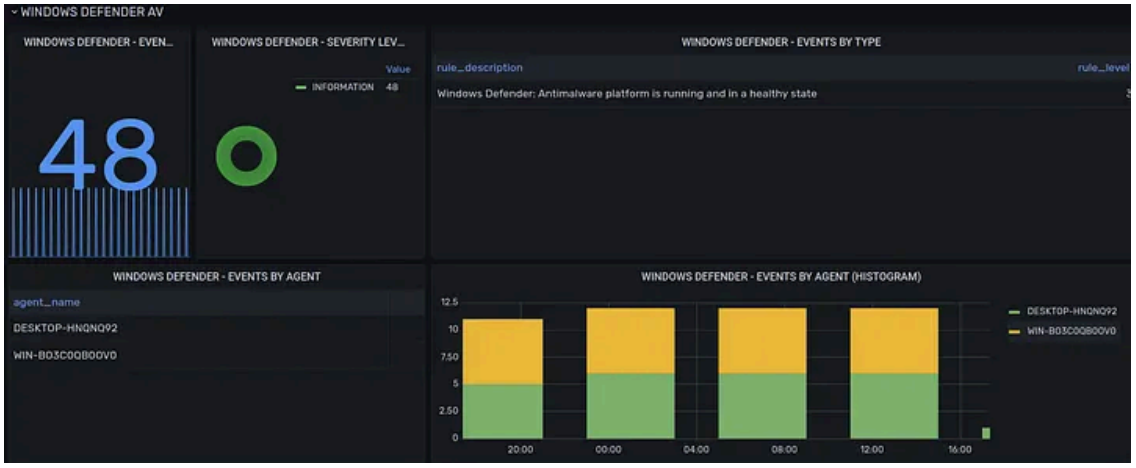
  Press enter or click to view image in full size



  Windows Defender Settings

- Ensuring that the antimalware platform is running and in a healthy state

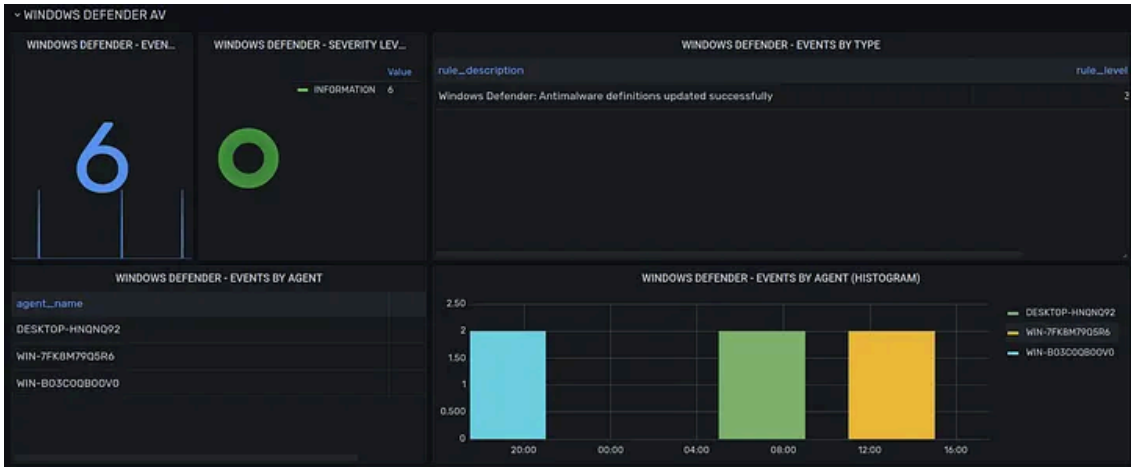
  Press enter or click to view image in full size



Information level events — Process running and in healthy state

- Ensuring that Windows Defender is successfully downloading and updating the latest file signatures

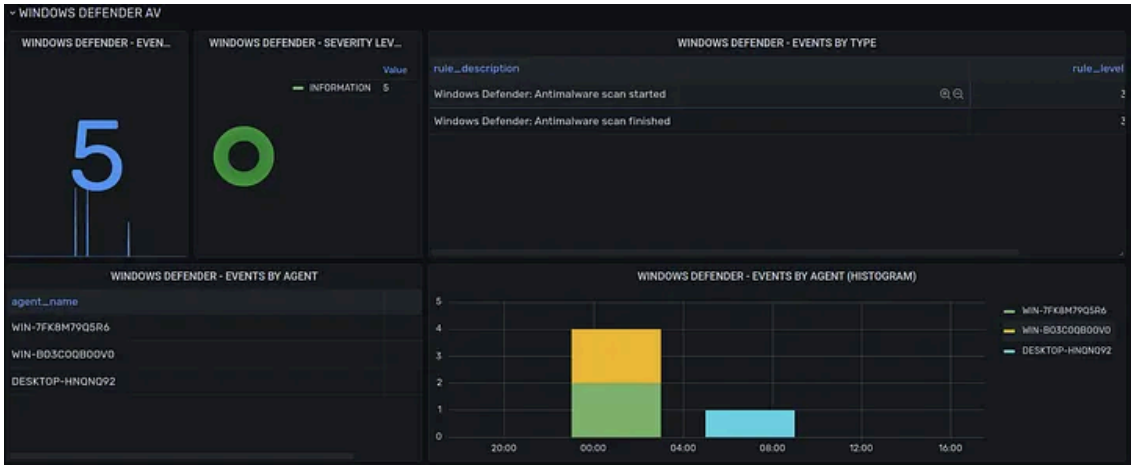
  Press enter or click to view image in full size



Information level events — Signatures Updated

- Ensuring that periodic scans are run in all agents and completed successfully.

  Press enter or click to view image in full size



Information level events — Periodic scans

DETECTION.

Network Activity

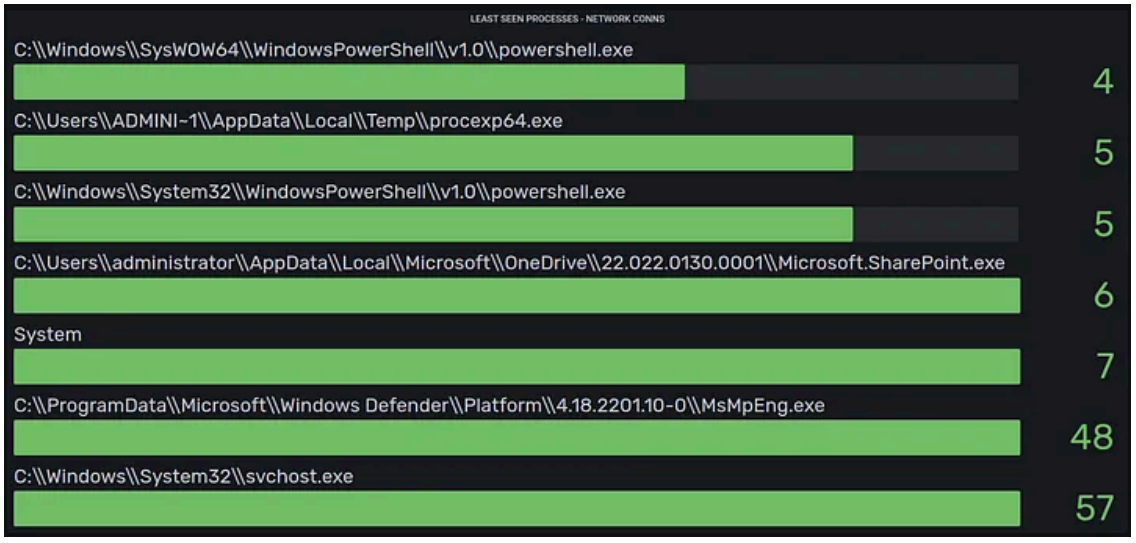
Communication to C2 Server:

- Suspicious processes (“winlogon.exe”, “rundll32.exe”, etc.) opening network connections to public IP addresses.

Detection Logic

Keep track and alert on unusual processes (least seen) opening network connections. Special attention to HTTP, HTTPS and DNS outbound connections. Sysmon (event ID 3) provides all the required telemetry:

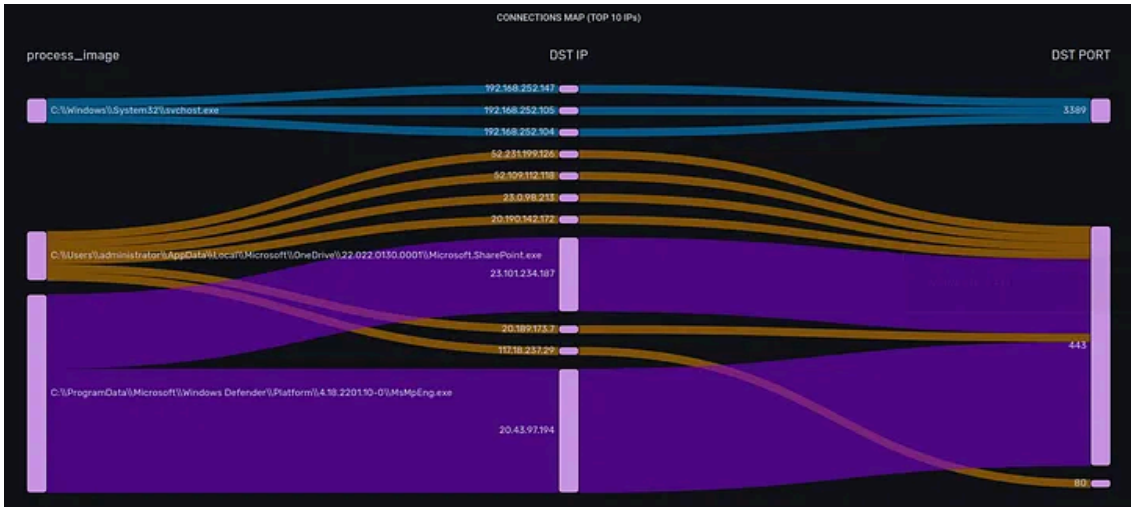
Press enter or click to view image in full size



Network Activity — Least Seen Processes

Visualisations such as network maps linking Process — DST IP — Connection port can help to quickly identify anomalies:

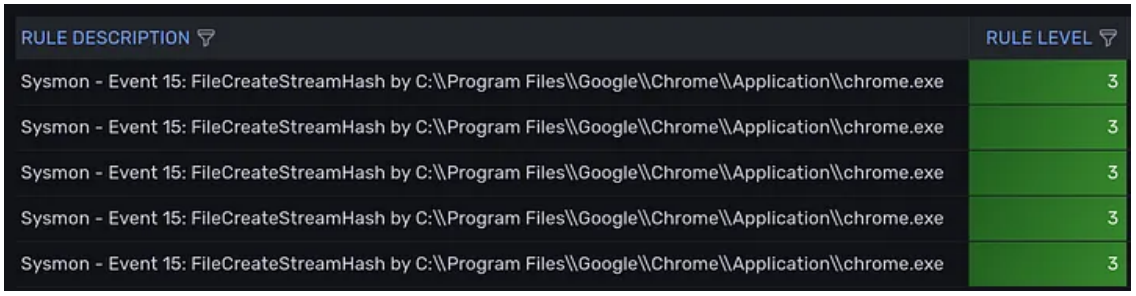
Press enter or click to view image in full size



Network Activity — Process Map

Keep track of downloaded payloads (.js files, .gif files, etc.). Sysmon (Event ID 15) provides this telemetry.

Press enter or click to view image in full size



RULE DESCRIPTION	RULE LEVEL
Sysmon - Event 15: FileCreateStreamHash by C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe	3
Sysmon - Event 15: FileCreateStreamHash by C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe	3
Sysmon - Event 15: FileCreateStreamHash by C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe	3
Sysmon - Event 15: FileCreateStreamHash by C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe	3
Sysmon - Event 15: FileCreateStreamHash by C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe	3

Network Activity — Sysmon Event 15

Send relevant observables (DNS requests, Destination IPs, files hashes of downloaded files) to security feeds/threat intel platform to identify IoCs related to these observables. More info here.

Process Activity.

rundll32.exe:

Get SOCFortress's stories in your inbox

Join Medium for free to get updates from this writer.

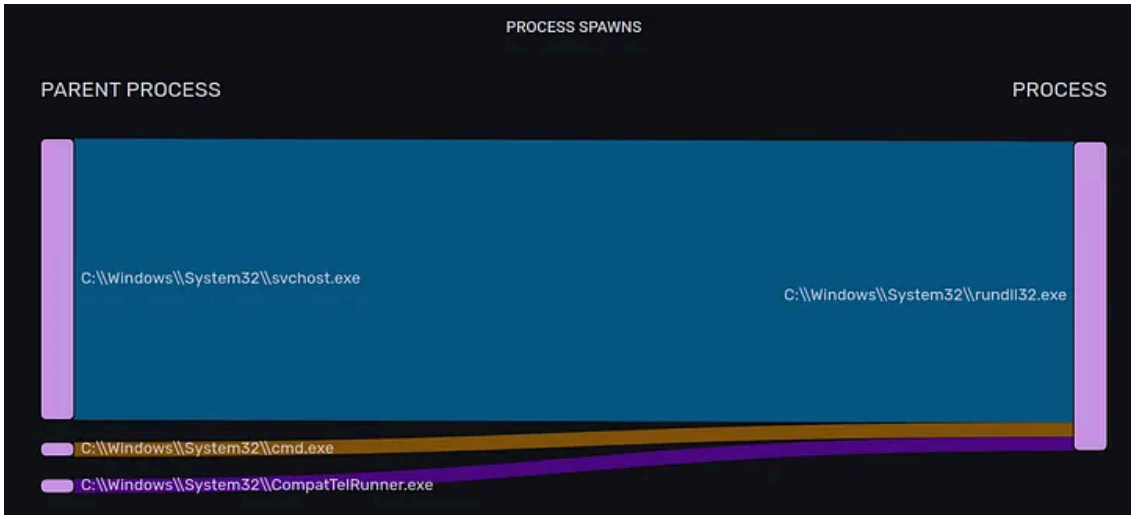
Remember me for faster sign in

Used in Lateral Movement by Cobalt Strike beacons: rundll32.exe being spawned by another process(es) and its process execution included no arguments.

Detection Logic

Keep track and alert on **unusual** processes spawning rundll32.exe

Press enter or click to view image in full size



Processes Spawning Rundll32.exe

Used in Post Exploitation in Cobalt strike related attacks: rundll32.exe spawning processes like **Adfind.exe**, Net.exe, or any other Windows processes used for systems, services or network discovery.

Parent Process Spoofing:

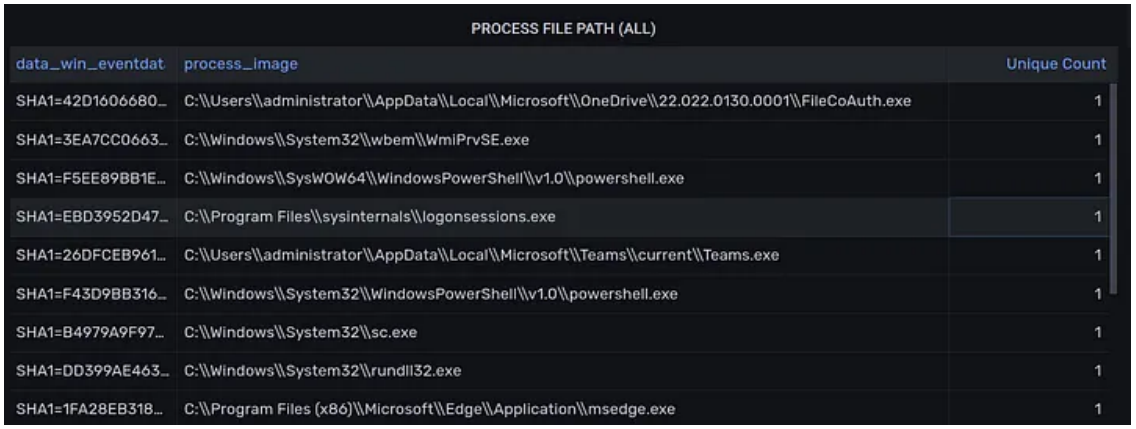
Parent process spoofing is a common technique used by Cobalt Strike beacons. With this technique the beacon tries to evade common detection methods such as processes related to the Office suite launching unusual child processes.

Cobalt strike beacons often spoof processes like “word.exe”, or “excel.exe” to “explorer.exe” so that when the child process is launched the telemetry reported by the EDR agent makes the detection of unusual process chains difficult.

Detection Logic

Keep track on unique file **hashes** (process image) and their mapping to file process image name and location. Spoofed processes will have same process name but different file hash and possibly executed from an unusual location.

Press enter or click to view image in full size



PROCESS FILE PATH (ALL)

data_win_eventdat	process_image	Unique Count
SHA1=42D1606680…	C:\\Users\\administrator\\AppData\\Local\\Microsoft\\OneDrive\\22.022.0130.0001\\FileCoAuth.exe	1
SHA1=3EA7CC0663…	C:\\Windows\\System32\\wbem\\WmiPrvSE.exe	1
SHA1=F5EE89BB1E…	C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe	1
SHA1=EBD3952D47…	C:\\Program Files\\sysinternals\\logonsessions.exe	1
SHA1=26DFCEB961…	C:\\Users\\administrator\\AppData\\Local\\Microsoft\\Teams\\current\\Teams.exe	1
SHA1=F43D9BB316…	C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	1
SHA1=B4979A9F97…	C:\\Windows\\System32\\sc.exe	1
SHA1=DD399AE463…	C:\\Windows\\System32\\rundll32.exe	1
SHA1=1FA28EB318…	C:\\Program Files (x86)\\Microsoft\\Edge\\Application\\msedge.exe	1

UNIQUE RELATIONSHIP PROCESS FILE HASH — PROCESS EXECUTED

Send relevant observables (file hash of the executed file image) to security feeds/threat intel platform to identify IoCs related to these observables. More info here.

Powershell execution (and command line arguments).

Suspicious command line arguments used:

nop, hidden, **encodedcommand**, nologo, noprofile

The “encodedcommand” (base64) can be extracted and its literal content further analysed, looking for commands like “Net.Webclient”, “Invoke-WebRequest”, etc., commonly used for lateral movement on the source machine.

On the destination machine, a common detection is spotting powershell executions where the parent process = “**wsmprovhost.exe**” and with a command line = “-Version 5.1 -s -nologo -noprofile”

Detection Logic

Detection rule to spot powershell executions with the *encodedcommand* command line and decode the content of the base64 string to “clear text”. Include the decoded command as an additional alert in Wazuh manager.

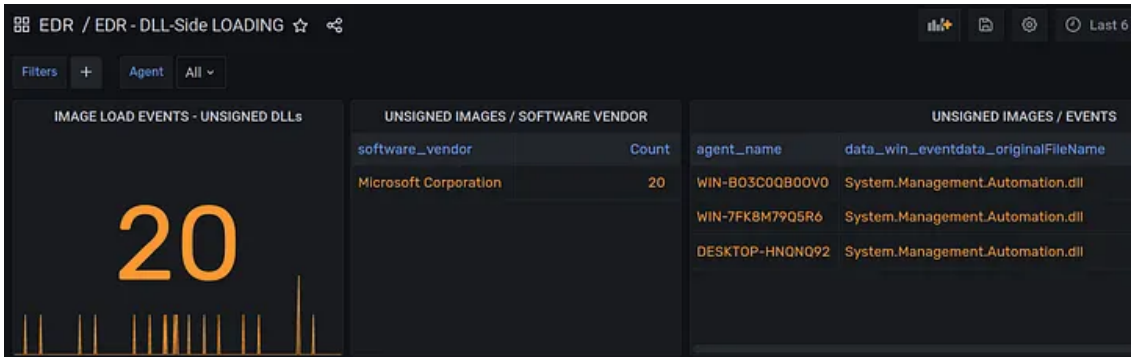
Memory artifacts / Process Injection.

Downloaded payloads try to be executed under the memory space of “Rundll32.exe”.

Detection Logic

Keep track and alert on **unsigned** DLLs or with no valid certificate loaded into memory. Sysmon (Event ID 7) provides this telemetry:

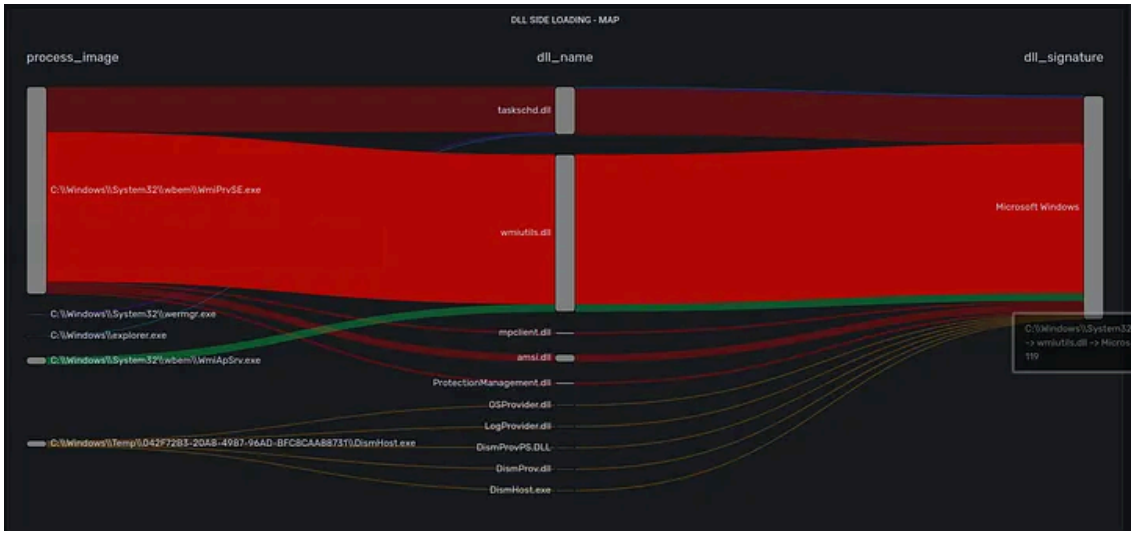
Press enter or click to view image in full size



Unsigned DLLs loaded in memory.

Visualisations such as DLL side loading maps linking Process —DLL — DLL vendor can help to quickly identify anomalies:

Press enter or click to view image in full size



Process and DLL side loading map.

Send relevant observables (DLL file hash) to security feeds/threat intel platform to identify IoCs related to these observables. More info here.

Lateral Movement.

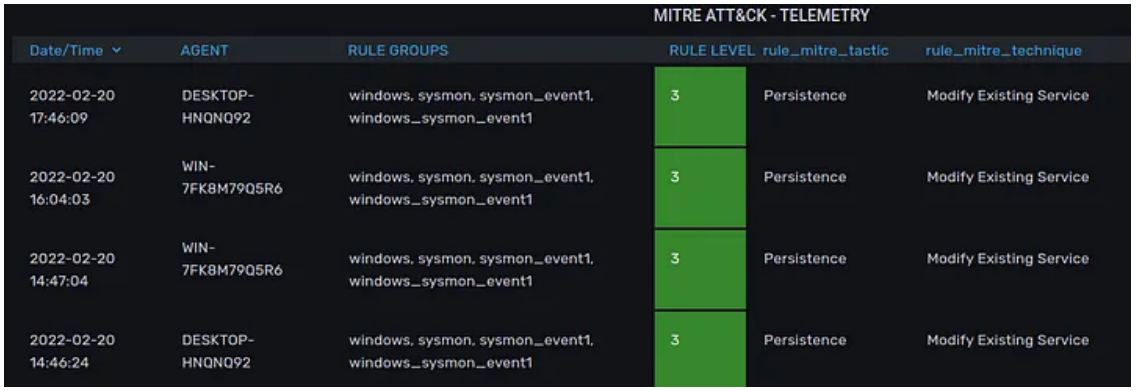
PSEXEC is one of the most common processes used by Cobalt strike beacons for lateral movement.

PSEXEC is used to drop a payload in a shared folder (normally ADMIN$) and then to start a new service on the target machine that executes that payload. The payload will spawn another process and finally the remote service is removed.

Detection Logic

Detection rules based on **frequency** alerting on service creation/modification/deletion activity:

Press enter or click to view image in full size



MITRE ATT&CK - TELEMETRY

Date/Time	AGENT	RULE GROUPS	RULE LEVEL	rule_mitre_tactic	rule_mitre_technique
2022-02-20 17:46:09	DESKTOP-HNQNQ92	windows, sysmon, sysmon_event1, windows_sysmon_event1	3	Persistence	Modify Existing Service
2022-02-20 16:04:03	WIN-7FK8M79Q5R6	windows, sysmon, sysmon_event1, windows_sysmon_event1	3	Persistence	Modify Existing Service
2022-02-20 14:47:04	WIN-7FK8M79Q5R6	windows, sysmon, sysmon_event1, windows_sysmon_event1	3	Persistence	Modify Existing Service
2022-02-20 14:46:24	DESKTOP-HNQNQ92	windows, sysmon, sysmon_event1, windows_sysmon_event1	3	Persistence	Modify Existing Service

System Services Activity and Telemetry.

Keep track and alert on unusual executables launched by **services.exe**.

Send relevant observables (process file hash) to security feeds/threat intel platform to identify IoCs related to these observables. More info here.

Source: https://socfortress.medium.com/detecting-cobalt-strike-beacons-3f8c9fdcb654