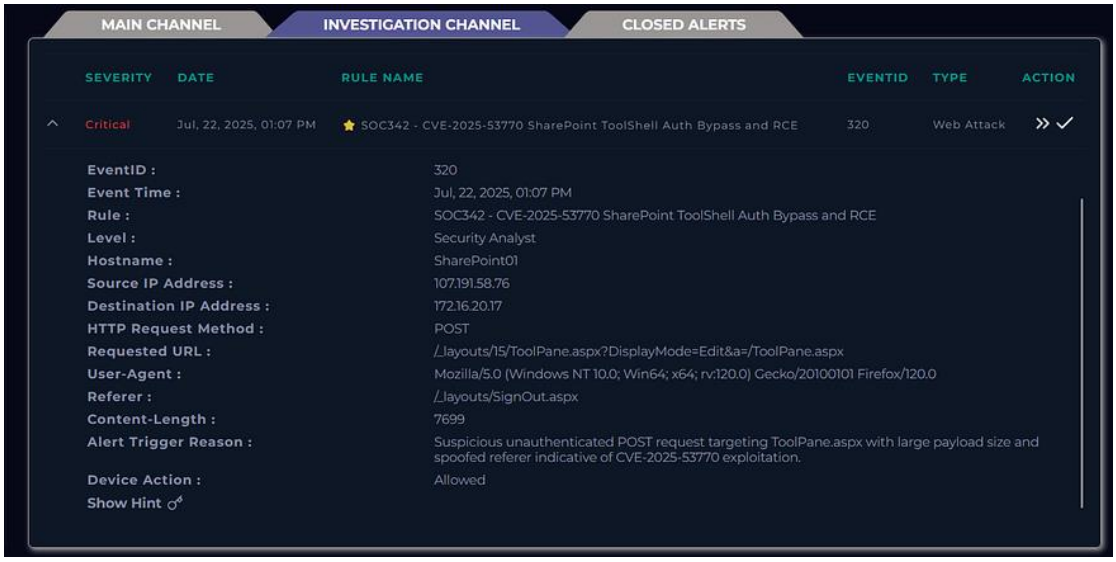
LetsDefend SOC342 — CVE-2025-53770 SharePoint ToolShell Auth Bypass and RCE

Overview of CVE-2025–53770

CVE-2025–53770 is a high-severity vulnerability affecting Microsoft SharePoint servers. It allows an attacker to bypass authentication and execute arbitrary code remotely by exploiting a flaw in how SharePoint handles requests to the `ToolPane.aspx` page. Specifically, this vulnerability can be triggered via crafted unauthenticated HTTP POST requests that manipulate internal SharePoint components, such as the ToolPane interface, and ultimately allow the execution of embedded scripts or dropped files. With a CVSS score of 9.8, this flaw represents a critical risk, particularly in internet-exposed environments where attackers can exploit it without credentials.

- **CVE-2025–49706**: Authentication bypass via crafted `Referer` headers.
- **CVE-2025–49704**: Insecure deserialization leading to remote PowerShell code execution.

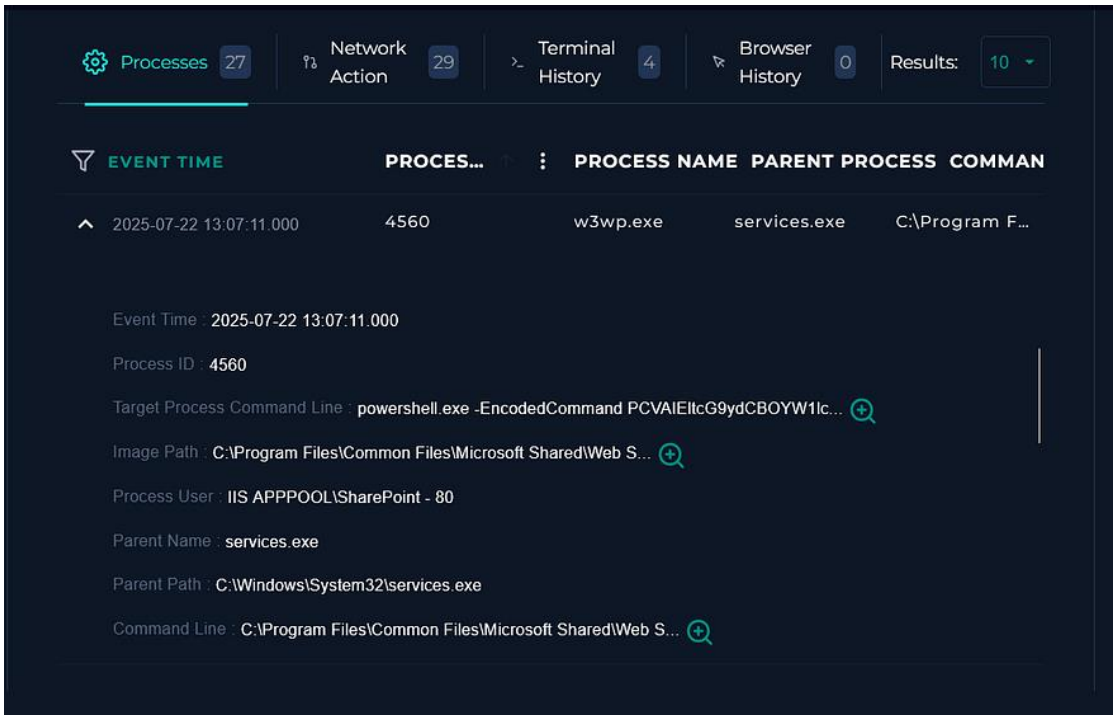
The Alert



This alert flags suspicious activity targeting Microsoft SharePoint's `ToolPane.aspx` endpoint, linked to **CVE-2025-53770** — a critical vulnerability allowing **unauthenticated attackers** to exploit SharePoint via a specially crafted POST request. Successful exploitation leads to **remote code execution** on the server without needing credentials.

Log Analysis

Press enter or click to view image in full size



The web traffic log revealed an unauthenticated POST request to SharePoint's ToolPane, a page not typically intended for direct access. The following anomalies were observed:

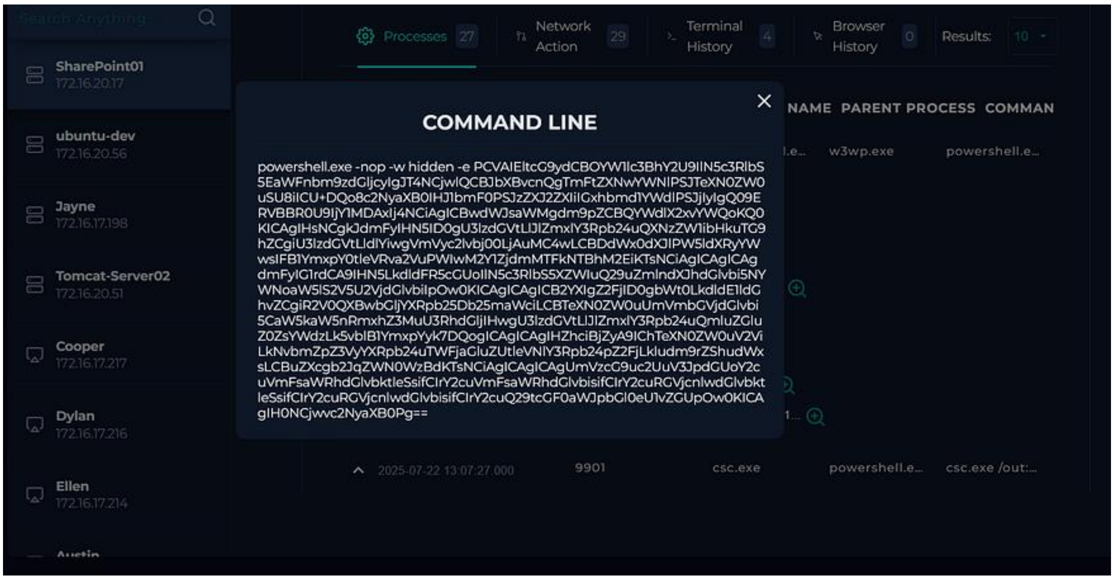
- **Spoofed Referer:** Set to `/layouts/SignOut.aspx` to appear legitimate
- **Large Payload:** 7699 bytes of encoded data (highly unusual)
- **No Authentication Headers:** Exploiting a known bypass flaw

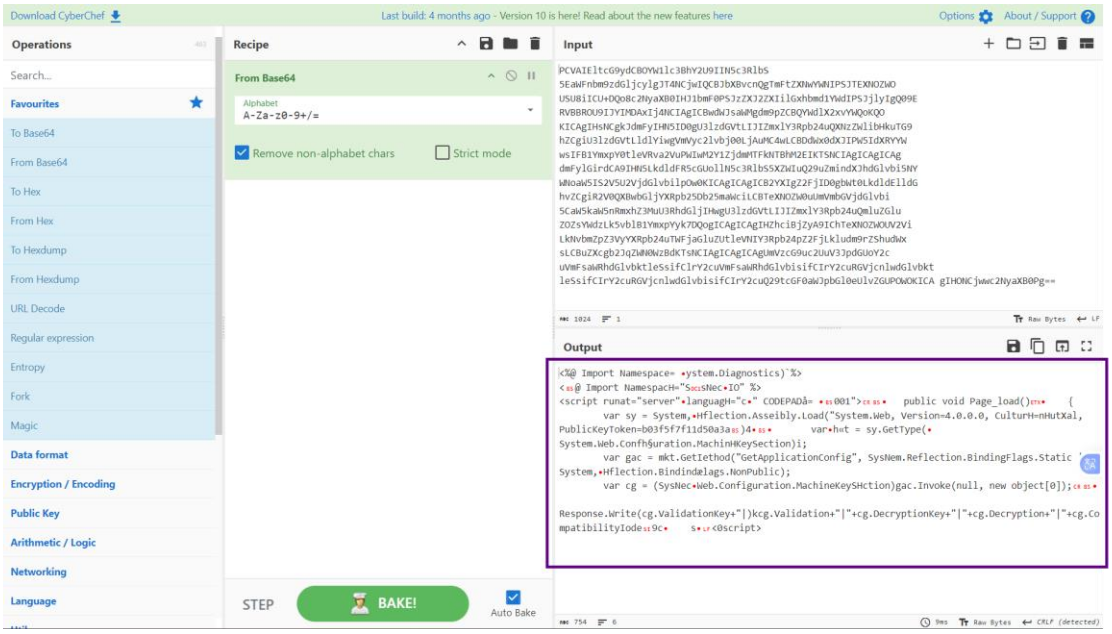
This directly aligns with CVE-2025–53770's exploitation pattern, where attackers bypass authentication and inject code through the ToolPane interface.

The delivery mechanism, headers, and payload structure were clearly crafted to avoid detection and execute commands stealthily.

EDR Analysis

EDR telemetry captured the post-exploitation activity in detail. A **PowerShell command** was executed that decoded into the following ASPX script





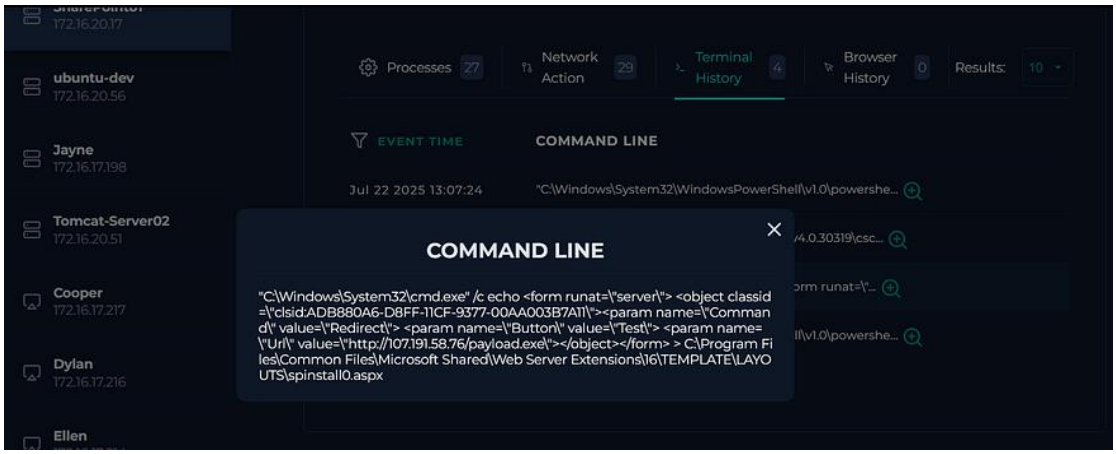
This script abuses **.NET reflection** to access private configuration and extract:

- ValidationKey
- DecryptionKey
- Encryption mode + compatibility settings

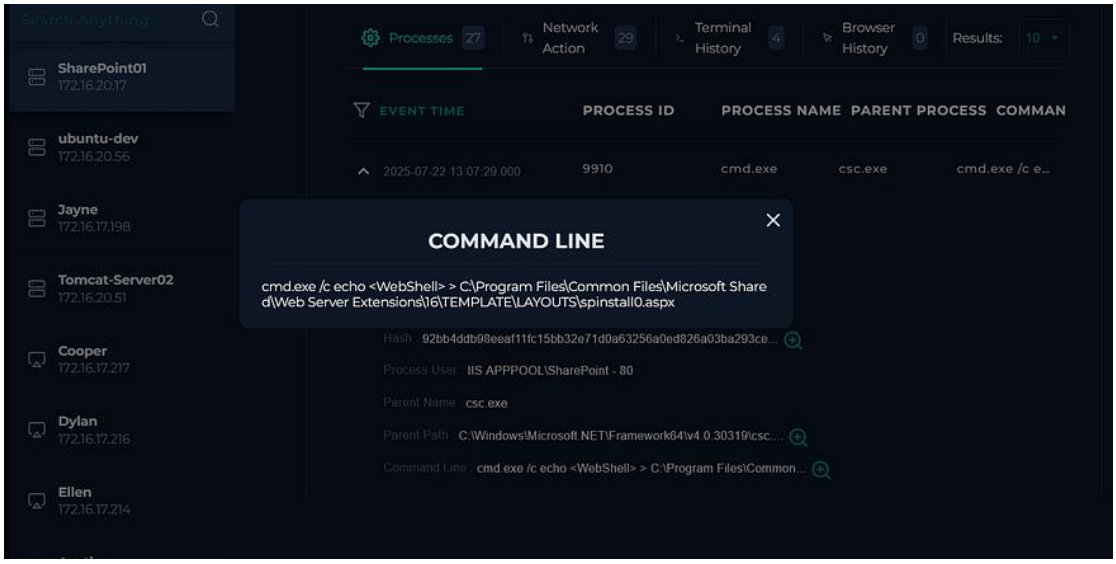
These keys can allow attackers to **forge authentication cookies** or decrypt protected data — a major step toward **persistence and lateral movement**.

CMD Execution and Dropped Files

Shortly after, `cmd.exe` was executed with the following command:

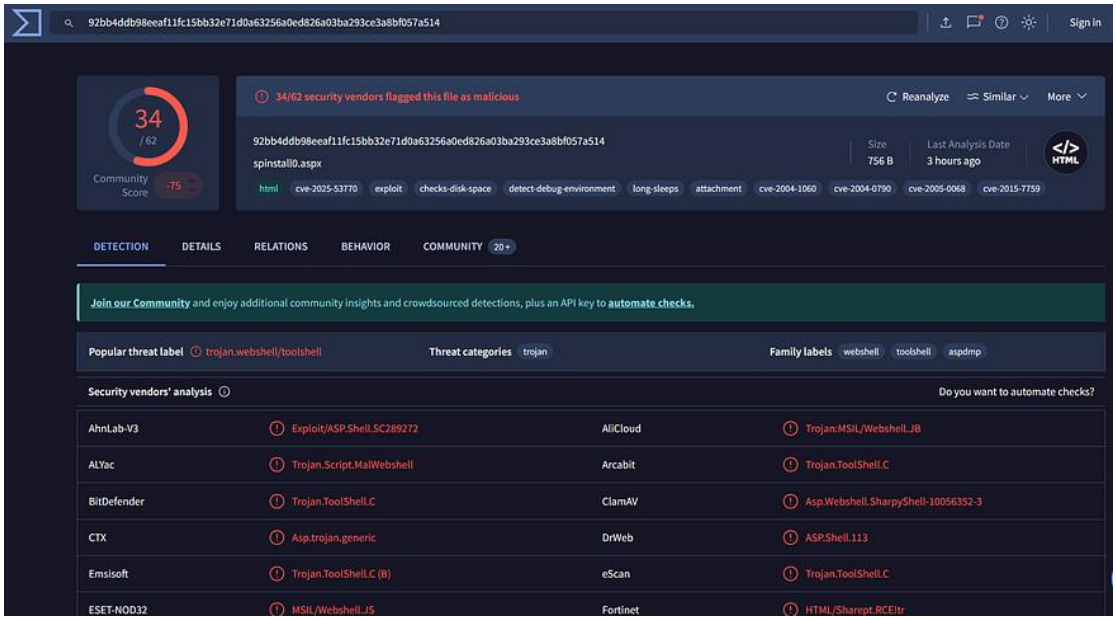


- Creates a malicious ASPX file (spinstall0.aspx) directly in the **SharePoint layouts directory**
- Embeds an <object> ActiveX tag that points to: http://107.191.58.[]76/payload[.]exe
- Acts as a **remote downloader** when the page is visited by a browser or triggered internally



To confirm the malicious nature of the dropped ASPX file, I uploaded the hash of `spinstall0.aspx` to VirusTotal. The result showed that **34 out of 64 security vendors** flagged it as malicious, further validating the presence of a webshell capable of downloading remote payloads and executing arbitrary commands on the server.

sha256 hash:
92bb4ddb98eeaf11fc15bb32e71d0a63256a0ed826a03ba293ce3a8bf057a514

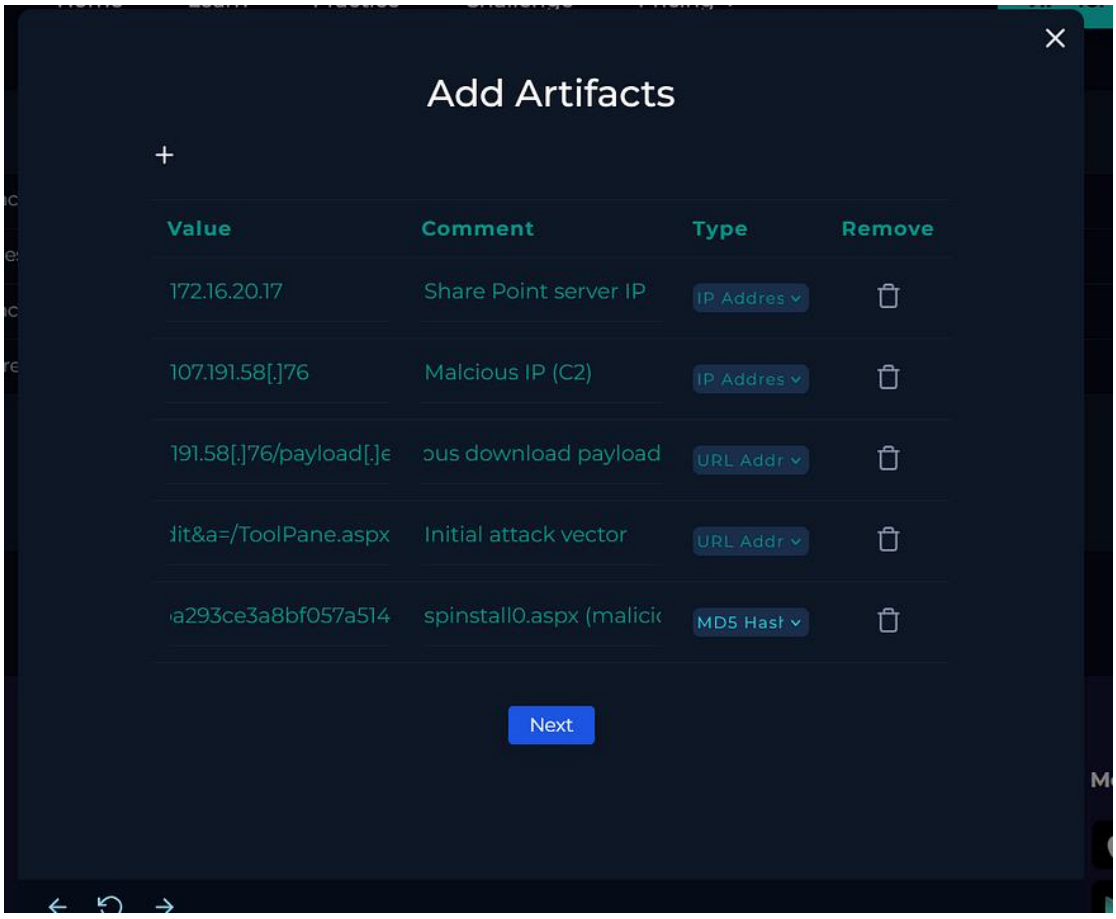


Final Stage

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -Command
"[System.Web.Configuration.MachineKeySection]::GetApplicationConfig()
```

In the final observed stage, the attacker manually invoked the method via Power-shell. This command accessed the server's cryptographic machine key configuration, extracting values such as the `ValidationKey`, `DecryptionKey`, and encryption modes. These keys are critical components used in securing authentication tokens and view state data in ASP.NET applications. By obtaining them, the attacker could potentially forge authentication cookies or decrypt sensitive data, allowing unauthorized access to SharePoint or other web applications relying on the same cryptographic configuration. This confirms the initial webshell's intent — to extract machine-level secrets for later exploitation or privilege escalation.

IOC's



Analyst Note:

The alert was triggered by suspicious PowerShell activity on a SharePoint server, which was later identified as an exploitation attempt of CVE-2025–53770. The attacker first dropped and compiled a custom C# executable (`payload.exe`) using `csc.exe`, followed by writing a malicious ASPX webshell (`spinstall0.aspx`) to a web-accessible SharePoint directory. Finally, a PowerShell command was executed to extract sensitive cryptographic machine keys from the server's configuration using .NET reflection. This behavior indicates a multi-stage attack aiming to establish persistent access and potentially forge authentication tokens. The webshell was confirmed malicious via Virus-total (34/64 detections). The incident has been classified as a true positive and further containment and remediation actions are recommended.

Conclusion

This was a super interesting alert to dig into, from weaponized Power-shell to on the fly compilation and a sneaky webshell, it had all the ingredients of a classic exploitation chain. CVE-2025–53770 is no joke, extracting machine keys can lead to token forgery and full compromise of SharePoint environments. While it was a fun analysis, it's a serious vulnerability that should be patched immediately in any exposed system.

By: Asad Dafalla