Detect webshells dropped on Microsoft Exchange servers after 0day compromises

github.com/cert-lv/exchange_webshell_detection

cert-lv

This project has been discontinued

Please use Microsoft tools instead:

- Microsoft Safety Scanner
- Other detections and mitigations listed in: https://github.com/microsoft/CSS-Exchange/tree/main/Security

When assessing impact we strongly suggest to assume breach and to preemptively examine all MS Exchange servers that were publically exposed since January, even if there are no signs of active compromise.

Note that data exfiltration and configuration changes were possible (and were happening) just through SSRF part of the epxloit chain alone (i.e. without achieving code execution, dropping any files or spawning new processes on the Exchange host).

UPD: As of March 13, 2021 Windows Defender is detecting this script itself as a Webshell

This is a false positive, A/V products react to keywords listed in the script. `detect_webshells.ps1` is intentionally written as a standalone file using very simple PowerShell, so that you could inspect it yourself. The keywords that (rightly) trigger A/V are listed on line 94. If you do not feel confident that you understand what the script is doing, do not run it!

Hopefully the fact that antivirus software started detecting this script means it's capable of detecting real webshells as well, making `detect_webshells.ps1` unnecessary. Check that Exchange and inetpub directories are not whitelisted though and please realise that webshells where only used for the initial access. Once attackers achieved code execution they usually deployed additional persistence mechanisms - sometimes even removing initial webshell themselves to hide their tracks.

So **don't mistake lack of webshells for lack of compromise** - unfortunately your server still might have been hacked and either attackers removed webshell themselves or an antivirus did that (without completely booting attackers from your server).

This script looks for webshells dropped on Microsoft Exchange servers while they were vulnerable to following CVE's:

- CVE-2021-26855, pre-auth SSRF, CVSS:3.0 9.1 / 8.4
- CVE-2021-26857, insecure deserialization leading to privilege escalation to SYSTEM level, CVSS:3.0 7.8 / 7.2
- CVE-2021-26858, post-auth file write, CVSS:3.0 7.8 / 7.2
- CVE-2021-27065, post-auth file write, CVSS:3.0 7.8 / 7.2

Initial activity during January 2021 was attributed to HAFNIUM, however since then other threat actors got hold of these exploits and started using them. Prior to public disclosure & patches being published by Microsoft (since 27 February or so) publically exposed Exchange servers started being exploited indiscriminately. As such, installing latest Exchange updates soon after Microsoft published them **did not fully mitigate the risk of prior compromise**, therefore all Exchange servers should be inspected for signs of unauthorized access.

Running

`detect_webshells.ps1` will check for the presence of known webshells in typical locations:

- `inetpub/wwwroot/aspnet_client/` : system wide location, most common place of dropped webshells in current attacks; normally does not contain any files at all, so presence of anything there is suspicious

- `$($env:exchangeinstallpath)/Frontend/` : used by more sophisticated attackers in order to blend in with legitimate Exchange files (webshells could be added as new files or by modifying existing ones, including `web.config`); most common locations are `/owa/` and `/ecp/` , but webshells could be dropped anywhere within Frontend directory

Interpreting the results

`detect_webshells.ps1` only looks for webshells and does not attempt to detect past exploitation events directly (use https://github.com/microsoft/CSS-Exchange/tree/main/Security and other scripts mentioned below for this), nor is it looking for particularly stealthy threat actors (which could delete webshells after use or avoid dropping them altogether). As such, negative result can only mean absence of evidence of the compromise on this particular host. It does not guarantee that the host was not exploited by some other means.

More information

Writeups/disclosures (incl. IoC):

- https://www.microsoft.com/security/blog/2021/03/02/hafnium-targeting-exchange-servers/
- https://www.volexity.com/blog/2021/03/02/active-exploitation-of-microsoft-exchange-zero-day-vulnerabilities/
- https://blog.rapid7.com/2021/03/03/rapid7s-insightidr-enables-detection-and-response-to-microsoft-exchange-0-day/
- https://us-cert.cisa.gov/ncas/alerts/aa21-062a
- https://msrc-blog.microsoft.com/2021/03/05/microsoft-exchange-server-vulnerabilities-mitigations-march-2021/
- https://blog.truesec.com/2021/03/07/exchange-zero-day-proxylogon-and-hafnium/

Notable detection scripts: