Advanced Notification of APT34 Web Shell Attack Targeting UAE Entities

ae CERT
فريق الاستجابة لطوارئ الحاسب الآلي
Computer
Emergency
Response
Team

Security Advisory	ADV-19-09	**Criticality**	Critical	○○○●
Advisory Released On	9th of April 2019			

Impact

Once a web shell is uploaded to a web server, it may enable an attacker to remotely access it and interact with it by performing unauthorized activities such as uploading further malicious files, running further malicious processes on the system or exfiltrating content.

Solution

Refer to the solution section.

Affected Platforms

- Web Servers.

Summary

As the leading trusted secure cyber coordination center in the region, aeCERT has researched and discovered a web shell attack targeting multiple UAE governmental entities. This is a detailed follow up to our previous advisory ADV-19-08 which covered Web shell attacks targeting UAE entities.

Threat Details

The purpose of the web shell attack is to allow the attackers to gain persistence on the targeted servers, enabling them to maintain unauthorized accessed and further compromise the system and network.

The attackers are known to have gained persistence on web servers by utilizing malicious .ASPX files including web shells to execute commands on the affected machines. The files were hidden using names such as signin.aspx, logon.aspx, change_password.aspx, and various other names that are intended to seem legitimate.

After gaining persistence, the attackers have been known previously to use variations of Mimikatz. Mimikatz is a tool commonly used to obtain user passwords in order to escalate privileges on affected machines.

To assist with obtaining passwords, the attackers have previously been known to make changes to the Registry and could be using the same method in this attack by changing the registry value of:

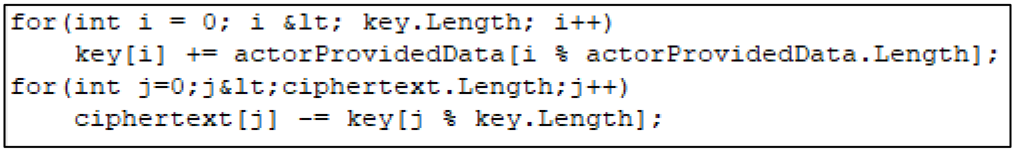
HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential to 1.

This enables the attacker to obtain cleartext passwords via Mimikatz once a user logs on.

One of the observed web shells is also tracked by some organizations under the name of Two Face. However, the attackers have been known to use other web shells so it is recommended that you ensure that your website contains only your approved files.

Two Face has two components, the loader and the payload component. The loader component contains heavily obfuscated code. This is to assist in evading detection, but it would be detectable as non-approved code on your website. The loader component is responsible for loading the payload and is usually accessed following authentication with it by the attacker. The payload enables the attacker to interact with the web server, to upload/download files, change their creation time and execute commands/processes.

A sample of code taken from the loader, which then decrypts the embedded payload is shown below:-



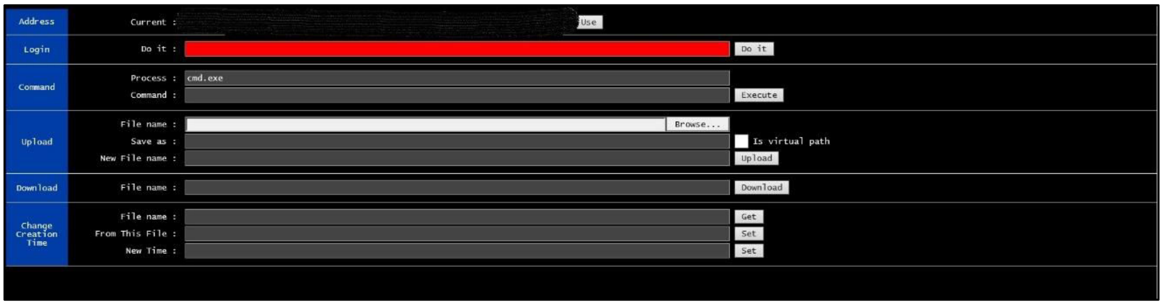
```
for(int i = 0; i &lt; key.Length; i++)
    key[i] += actorProvidedData[i % actorProvidedData.Length];
for(int j=0;j&lt;ciphertext.Length;j++)
    ciphertext[j] -= key[j % key.Length];
```

A known additional web shell used by the attacker is called IntrudingDivisor. Like Two Face, the attacker will have to authenticate with the code. This web shell has been observed to make requests to execute files named KB45253-ENU.exe and KB76862-ENU.exe.

The attackers also previously utilized process dumping to dump the process memory of lsass.exe which stands for Local Security Authority Subsystem Service. This Windows service is very crucial as it can handle password changes, verifies users logging on a PC or server, create access token as well as writing to the security log.

One variation of the web shell attack observed by aeCERT is the SLOWPOKE web shell. This is an ASPX .NET web shell that only allows for the execution of signed .NET code. The code is supplied through HTTP parameters and must be signed with a private key that matched the hard-coded public key.

Below is an example of a web shell opened in chrome:

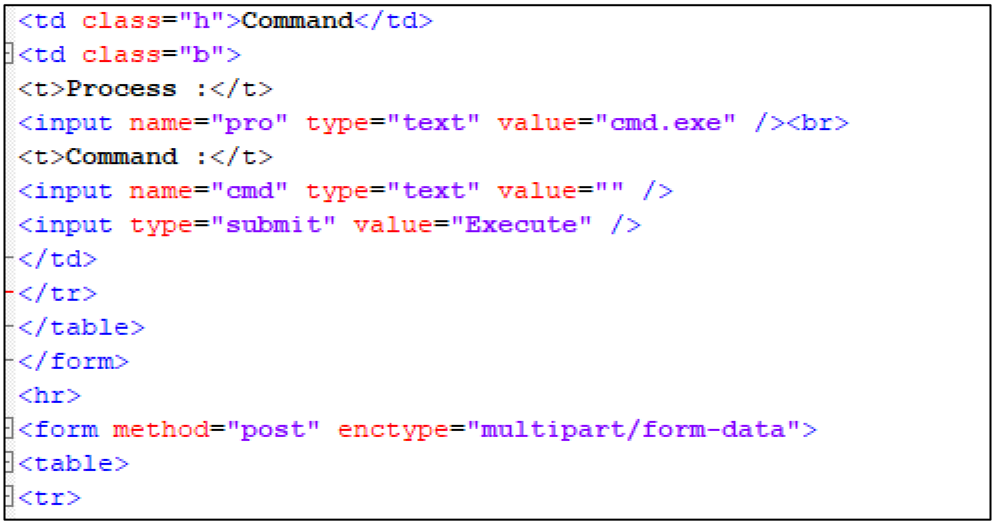


Current : [redacted] Use

Process : cmd.exe

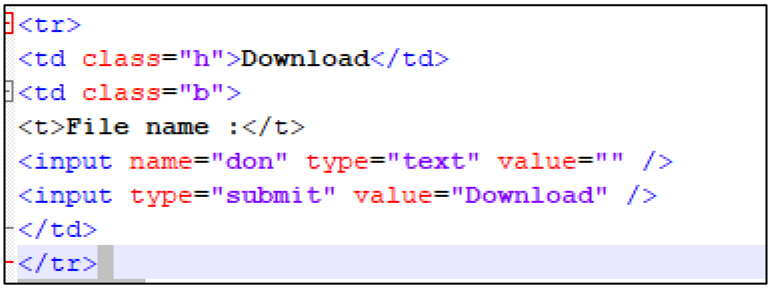
Below you can see some examples of the scripts/codes ran in this web shell (forming some of the code of the web shells):

Example 1



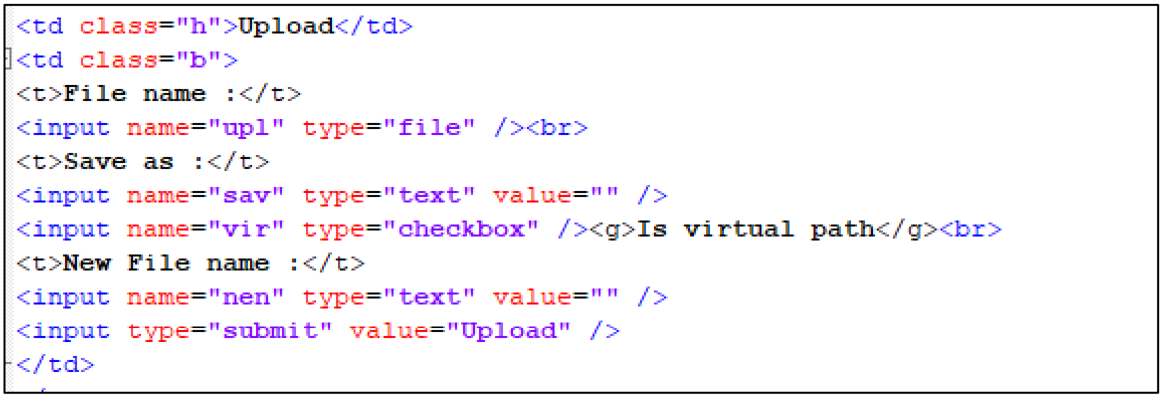
```
<td class="h">Command</td>
<td class="b">
<t>Process :</t>
<input name="pro" type="text" value="cmd.exe" /><br>
<t>Command :</t>
<input name="cmd" type="text" value="" />
<input type="submit" value="Execute" />
</td>
</tr>
</table>
</form>
<hr>
<form method="post" enctype="multipart/form-data">
<table>
<tr>
```

Example 2



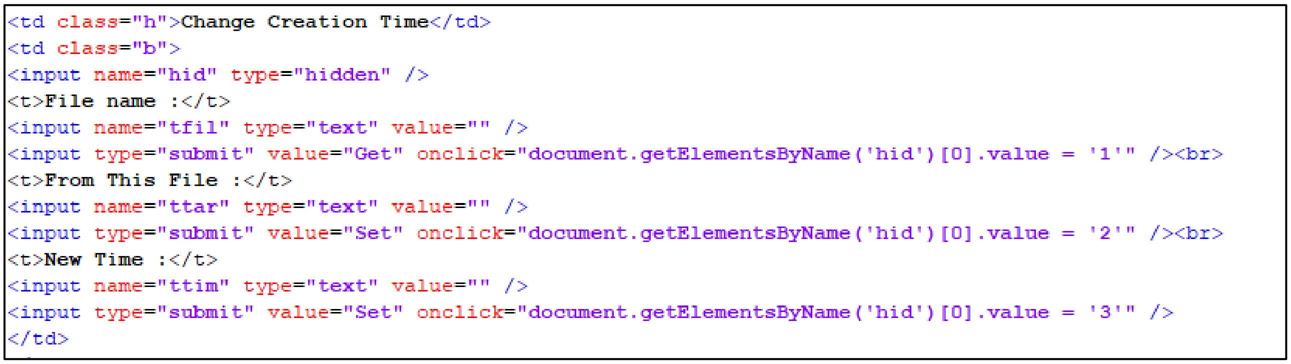
```
<tr>
<td class="h">Download</td>
<td class="b">
<t>File name :</t>
<input name="don" type="text" value="" />
<input type="submit" value="Download" />
</td>
</tr>
```

Example 3



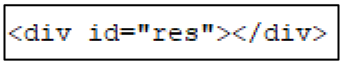
```
<td class="h">Upload</td>
<td class="b">
<t>File name :</t>
<input name="upl" type="file" /><br>
<t>Save as :</t>
<input name="sav" type="text" value="" />
<input name="vir" type="checkbox" /><g>Is virtual path</g><br>
<t>New File name :</t>
<input name="nen" type="text" value="" />
<input type="submit" value="Upload" />
</td>
```

Example 4



```
<td class="h">Change Creation Time</td>
<td class="b">
<input name="hid" type="hidden" />
<t>File name :</t>
<input name="tfil" type="text" value="" />
<input type="submit" value="Get" onclick="document.getElementsByName('hid')[0].value = '1'" /><br>
<t>From This File :</t>
<input name="ttar" type="text" value="" />
<input type="submit" value="Set" onclick="document.getElementsByName('hid')[0].value = '2'" /><br>
<t>New Time :</t>
<input name="ttim" type="text" value="" />
<input type="submit" value="Set" onclick="document.getElementsByName('hid')[0].value = '3'" />
</td>
```

In addition, aeCERT has determined that in some instances a separate component associated with the web shell attack has been observed in the form of the presence of .ASPX files uploaded by attackers than will contain the following code:-



```
<div id="res"></div>
```

IOC

Note: Please be aware some of these .aspx files could match the default configuration of your server and therefore could have an identical name to some of your.aspx files so we advise checking the contents of the .aspx before marking it as malicious.

/owa/auth/outlookfilles.aspx
/signproces.aspx
/owa/auth/signon.aspx
/owa/auth/logontimeout.aspx
/owa/auth/expirepw.aspx
/owa/auth/change_password.aspx
/owa/auth/outlookservice.aspx
/owa/auth/getidtoken.aspx
/_layouts/WrkStatLog.aspx
/EnterpriseVault/js/jquery.aspx
/owa/auth/espw.aspx
/_layouts/workpage.aspx
/owa/auth/outlooklogonservice.aspx
/owa/auth/OutlookCName.aspx
/owa/auth/RedirSuiteService.aspx
/owa/auth/handlerservice.aspx
/petrol.aspx
/owa/auth/outlooklogon.aspx
/web/tofollowup.aspx
/owa/auth/OutlookEN.aspx
/english/resources.aspx
/tax.aspx
/owa/auth/owaauth.aspx
/owa/auth/outlooktoken.aspx
/owa/auth/owalogin.aspx
/owa/auth/outlookdn.aspx
/signin.aspx
/Global.aspx
/Exchange.aspx

HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential to 1.

“compilerParams.GenerateInMemory”
“<div id= “res”>”
“n.getElementsByName(”cmd”)[0].value”
“wEPDwUKMTMxNzM0MTI3N2RkdhFFEUsPWyojQS1nhFFdcorP2pdVVi+fQq6YzX9Pmd”
[http-request-ext:request_header.'Cookie' LIKE 'data=pro#=##|#cmd#=##%']

FireEye Webshell detection:
FE_APT_Webshell_ASPX_SLOWPOKE_1,APT.Webshell.ASPX.SLOWPOKE

Solution

aeCERT has created a tool in which it detects if a web-server contains the malicious IOCs. To ensure that you have not been attacked, run the tool on your webservers and validate the detected IOCs. If the IOCs is detected and validated, it is recommended that you forward the samples to aeCERT in a password protected zip file and then proceed with eliminating the detected IOC ASAP.

If you suspect infection on the web server, ensure to scan the whole drive with our tool on the web server.

In case you find a web shell, we advise that you check the server's IIS logs for any access on the page hosting the web shell.

We recommend that you e-mail us any web shells or suspicious files you find for analysis at: Incident@aecert.ae

We recommend that you scan the whole drive with this tool on the web server.
The tool can be found in our FileShare:
https://fs.aecert.ae/index.php/s/JrDjpnRQEPy6QkH

As the tool is still in its development stages, please do a re-scan if it was not completed. Also, you can periodically access the link and check if there are updates for the tool.

Contact Us

aeCERT
P.O. Box 116688
Dubai, United Arab Emirates

Tel	(+971) 4 777 4003
Fax	(+971) 4 777 4100
Email	info[at]aeCERT.ae
Instagram	@TheUAETRA
Twitter	@TheUAETRA

For secure communications with aeCERT with regards to sensitive or vulnerability information please send your correspondences to aeCERT[at]aeCERT.ae