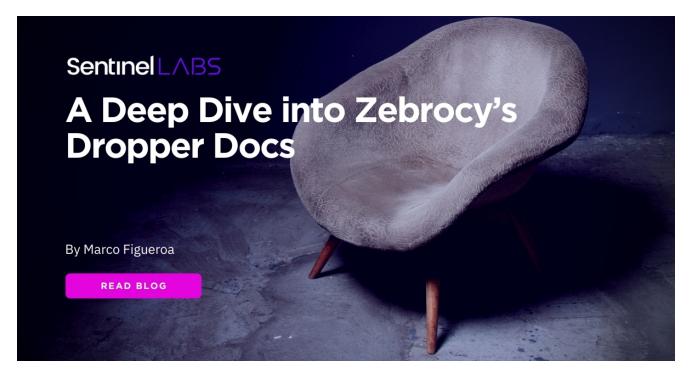
A Deep Dive into Zebrocy's Dropper Docs

(II) labs.sentinelone.com/a-deep-dive-into-zebrocys-dropper-docs/

Marco Figueroa



Contributor: Amitai Ben Shushan Ehrlich

Sofacy is an APT threat actor that's been around since 2008 and rose to prominence with the election hacks of 2016. Better known as FancyBear or APT28, this threat actor targets governments, military, and private organizations and has been known to engage in hack-and-leak operations. In the past couple of years, Sofacy has drastically retooled and largely evaded analysts. One of the more consistent <u>subgroups</u> is known as Zebrocy. Their targeting appears primarily focused on former Soviet Republics and, more recently, Asia.

In March 2021, we observed a cluster of activities targeting Kazakhstan with Delphocy – malware written in Delphi and previously associated with Zebrocy. The Word documents that were observed purport to be from a Kazakhy company named Kazchrome, a mining and metal company and one of the world's largest producers of chrome ore and ferroalloys.

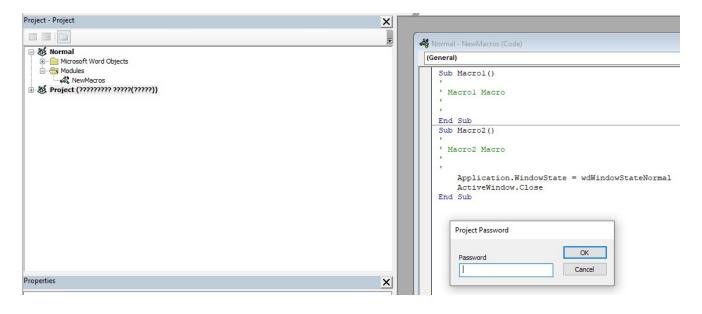
In total, we found six Delphocy Word documents that appear to be related to this cluster, all of which contain the same VBA script that drops a PE. Out of the six Word documents, two appear to be authentic uploads to VirusTotal by victims originating from Kazakhstan. The uploaded files contain what appeared to be the original filenames Авансовый отчет(новый).doc and Форма докладной (служебной) записки.doc.

In this post, we take a deep dive into these samples and share some techniques other analysts can employ to reverse engineer Delphocy dropper docs. We show how researchers can bypass password-protected macros and describe both how to decompile Delphi using IDR (Interactive Delphi Reconstructor) and how to import the saved IDC file into Ghidra using dhrake's plugin.

The results of our analysis led us to discover further Zebrocy clusters; a list of IOCs and YARA detection rules are provided to enable threat hunters to search for these and related artifacts in their environments.

Bypassing VBA Macro Password Protection

When analyzing Office documents with VBA macros, threat hunters have many different tools and techniques that do the job, but I've built a habit that I still use when I first started reversing malware to bypass password-protected macros manually.



- 1. Open up your favorite hex editor. I use HxD.
- 2. Load the Word Document.
- 3. Search for the following text:
 - 1. CMG=
 - 2. GC=
 - 3. DPB=
- 4. Add an \times to each of them:
 - 1. CMGx=
 - 2. GCx=
 - 3. DPBx=
- 5. Save the file with the changes.

₩ HxD - [C:\Users\Marco\Desktop\Авансовый отчет(новый).doc]

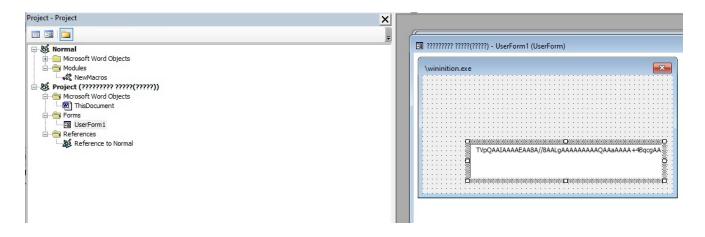
File Edit Search View Analysis Tools Window Help																	
🗋 👌 🕶 🛃				•	+ +	16	1	~ \	Vind	ows	(ANS	SI)		~	hex		~
🗱 Авансовый отчет(новый).doc																	
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	Decoded text
004B8110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
004B8120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
004B8130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
004B8140	49	44	ЗD	22	7B	30	30	30	30	30	30	30	30	2D	30	30	ID="{00000000-00
004B8150	30	30	2D	30	30	30	30	2D	30	30	30	30	2D	30	30	30	00-0000-0000-000
004B8160	30	30	30	30	30	30	30	30	30	7D	22	OD	OA	44	6F	63	000000000}"Doc
004B8170	75	6D	65	6E	74	ЗD	54	68	69	73	44	6F	63	75	6D	65	ument=ThisDocume
004B8180	6E	74	2F	26	48	30	30	30	30	30	30	30	30	OD	OA	42	nt/&H00000000B
004B8190	61	73	65	43	6C	61	73	73	ЗD	55	73	65	72	46	6F	72	aseClass=UserFor
004B81A0	6D	31	OD	OA	48	65	6C	70	46	69	6C	65	3D	22	22	OD	mlHelpFile="".
004B81B0	OA	4E	61	6D	65	3D	22	50	72	6F	6A	65	63	74	22	OD	.Name="Project".
004B81C0	OA	48	65	6C	70	43	6F	6E	74	65	78	74	49	44	ЗD	22	.HelpContextID="
004B81D0	30	22	OD	0A	56	65	72	73	69	6F	6E	43	6F	6D	70	61	0"VersionCompa
004B81E0	74	69	62	6C	65	33	32	3D	22	33	39	33	32	32	32	30	tible32="3932220
004B81F0	30	30	22	OD	43	4D	47	78	ЗD	22	31	35	31	37	42	39	00".CMGx="1517B9
004B8200	35	42	43	39	46	37	43	44	46	37	43	44	46	33	44	31	5BC9F7CDF7CDF3D1
004B8210	46	33	44	31	22	OD	OA	44	50	42	ЗD	22	41	44	41	46	F3D1"DPB="ADAF
004B8220	30	31	43	33	30	31	34	36	31	45	34	36	31	45	42	39	01C301461E461EB9
004B8230	45	32	34	37	31	45	36	31	36	46	30	31	44	30	36	30	E2471E616F01D060
004B8240	39	33	43	35	39	41	37	43	34	44	33	30	46	36	34	41	93C59A7C4D30F64A
004B8250	35	31	42	44	45	44	44	41	39	38	45	43	31	35	39	30	51BDEDDA98EC1590
004B8260	43	39	42	31	39	31	46	46	22	OD	47	43	78	3D	22	34	C9B191FF".GCx="4
004B8270	35	34	37	45	39	36	42	31	39	30	32	31	41	30	32	31	547E96B19021A021
004B8280	41	30	32	22	OD	OA	OD	OA	5B	48	6F	73	74	20	45	78	A02" [Host Ex

When opening the Word document and viewing the macro this time, you can see the script as well as the Forms. When analyzing the function, what immediately sticks out is the ert.DataType = "bin.base64", showing that the UserForm1 is encoded with <u>base64</u>.

Project - Project X	<pre></pre>
Properties - ThisDocument	qw.SaveToFile ghj, a End Function
ThisDocument Document	= <u>=</u> <

Wininition UserForm

When selecting on UserForm1, the textbox reveals a **base64** encoded string; we know this because of the function we discussed above. The next step is to copy the entire string into a file so it can be decoded.



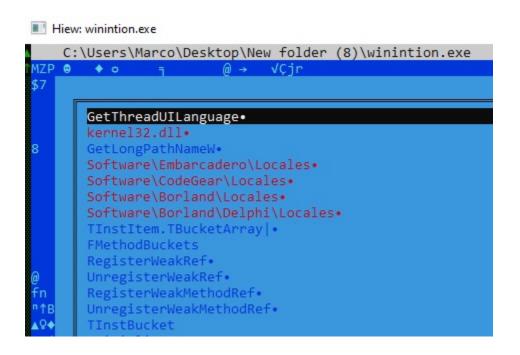
Now we decode the binary from **base64** and save it to disk as **wininition.exe**.



Following that, clean the headers using <u>HxD</u>, and then use <u>PE-Bear</u> to fix the sections headers to move to the next phase of the analysis.

winintion.exe		📓 winintion.exe
Offset(h) 00 01 (1 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text	Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 4D 5A 3	A 50 00 02 00 00 00 04 00 OF 00 EF BF BD EF MZP	00000000 4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00 MZPÿÿ
00000010 BF BD (00000010 B8 00 00 00 00 00 00 00 40 00 1A 00 00 00 FB 80 ,ĝĝ€û€
	00 00 EF BF BD EF BF BD 6A 72 00 00 00 00i¿¾jź	00000020 6A 72 00 00 00 00 00 00 00 00 00 00 00 00 00
	00 00 00 00 00 00 00 00 00 00 00 00 00	00000030 00 00 00 00 00 00 00 00 00 00 0
	0 00 00 00 00 00 02 00 00 EF BF BD 10 00 0Ei¿4 F BF BD 09 EF BF BD 21 EF BF BD 01 4C EF BF .i¿4.i¿4!i¿4.Li¿	00000050 54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73 This program mus
	L EF BF BD EF BF BD 54 68 69 73 20 70 72 6F %!i¿%i¿%This pro	00000060 74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57 t be run under W
	2 61 6D 20 6D 75 73 74 20 62 65 20 72 75 6E gram must be run	00000070 69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 in32\$7
00000080 20 75 (5 6E 64 65 72 20 57 69 6E 33 32 0A 24 37 00 under Win32.\$7.	00000080 00 00 00 00 00 00 00 00 00 00 0
00000090 00 00 0	0 00 00 00 00 00 00 00 00 00 00 00 00 0	00000090 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	000000A0 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	000000B0 00 00 00 00 00 00 00 00 00 00 0
	0 00 00 00 00 00 00 00 00 00 00 00 00 0	
		000000E0 00 00 00 00 00 00 00 00 00 00 0
		000000F0 00 00 00 00 00 00 00 00 00 00 0
00000100 00 00 0	0 00 00 00 00 00 00 00 00 00 00 00 00 0	00000100 00 00 00 00 00 00 00 00 00 00 0
00000110 00 00 0	0 00 00 00 00 00 00 00 00 00 00 00 00 0	00000110 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	00000120 00 00 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	00000130 00 00 00 00 00 00 00 00 00 00 00 00 0
	0 00 00 00 00 00 00 00 00 00 00 00 00 0	
	0 00 00 00 00 00 00 00 00 00 00 00 00 0	
		00000170 00 00 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	00000180 00 00 00 00 00 00 00 00 00 00 00 00 0
00000190 00 00 0	0 00 00 00 00 00 00 00 00 00 00 00 00 0	00000190 00 00 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	000001A0 00 00 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	000001B0 00 00 00 00 00 00 00 00 00 00 00 00 0
	0 00 00 00 00 00 00 00 00 00 00 00 00 0	000001C0 00 00 00 00 00 00 00 00 00 00 00 00 0
	0 00 00 00 00 00 00 00 00 00 00 00 00 0	000001E0 00 00 00 00 00 00 00 00 00 00 00 00 0
		000001F0 00 00 00 00 00 00 00 00 00 00 00 00 0
	00 00 00 00 00 00 00 00 00 00 00 00 00	00000200 50 45 00 00 4C 01 09 00 3F DA 1C 60 00 00 00 00 PEL?Ú.`
00000210 00 00 (0 00 00 00 00 00 50 45 00 00 4C 01 09 00 3FPEL?	00000210 00 00 00 00 E0 00 0E 01 0B 01 05 00 00 10 30 00à
	F BD 1C 60 00 00 00 00 00 00 00 00 EF BF BD 1245.`1245	00000220 00 40 02 00 00 00 00 00 30 22 00 00 00 10 00 00 .@0"
	E 01 0B 01 05 00 00 10 30 00 00 40 02 00 000@	00000230 00 20 30 00 00 00 40 00 00 10 00 00 02 00 00 . 0@
	0 00 30 22 00 00 00 10 00 00 00 20 30 00 000"0.	00000240 04 00 00 00 00 00 00 00 05 00 00 00 00 00
	0 00 00 10 00 00 00 02 00 00 04 00 00 00 00 0 0 00 05 00 00 00 00 00 00 00 00 30 38 00 00	
		00000270 00 00 00 10 00 00 00 00 D0 32 00 C7 00 00 00
	0 00 00 10 00 00 10 00 00 00 00 00 00 10	00000280 00 80 32 00 4A 38 00 00 00 E0 32 00 00 58 01 00 .€2.J8à2X
00000290 00 00 0	00 00 EF BF BD 32 00 EF BF BD 00 00 00 00i242.i24	00000290 00 00 00 00 00 00 00 00 00 00 00 00 0
	F BD 32 00 4A 38 00 00 00 EF BF BD 32 00 00 ï¿52.J8ï¿52	000002A0 00 40 34 00 74 ED 03 00 00 00 00 00 00 00 00 00 .@4.ti
	1 00 00 00 00 00 00 00 00 00 00 00 00 00	000002B0 00 00 00 00 00 00 00 00 00 00 00 00 0
	0 00 00 40 34 00 74 EF BF BD 03 00 00 00 00@4.tä¿≒	000002C0 00 70 32 00 18 00 00 00 00 00 00 00 00 00 00 00 .p2 000002D0 00 00 00 00 00 00 00 00 00 00 00 00 0
	0 00 00 00 00 00 00 00 00 00 00 00 00 0	000002E0 00 C0 32 00 1A 0A 00 00 00 00 00 00 00 00 00 00 .A2
	0 00 00 00 00 00 00 00 00 00 00 00 00 0	000002F0 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00text
	0 00 00 00 00 EF BF BD 32 00 1A 0A 00 00 00iz+2	00000300 00 10 30 00 00 10 00 00 10 30 00 00 06 00 0000
00000310 00 00 (00 00 00 00 00 00 00 00 00 00 00 00 00	00000310 00 00 00 00 00 00 00 00 00 00 00 00 0
	5 78 74 00 00 00 00 10 30 00 00 10 00 00 00 text0	00000320 2E 64 61 74 61 00 00 00 00 40 02 00 00 20 30 00 .data@ 0.
	00 00 06 00 00 00 00 00 00 00 00 00 00 0	00000330 00 54 01 00 00 16 30 00 00 00 00 00 00 00 00 00 .T0 00000340 00 00 00 00 40 00 00 C0 2E 74 6C 73 00 00 00 00@
	0 00 20 00 00 60 2E 64 61 74 61 00 00 00 00	00000340 00 00 00 00 40 00 00 C0 2E 74 6C 73 00 00 00 00@Å.tls 00000350 00 10 00 00 00 60 32 00 00 02 00 00 00 6A 31 00`2il.
	2 00 00 20 30 00 00 54 01 00 00 16 30 00 00 @ 0T0 0 00 00 00 00 00 00 00 00 00 40 00 00 EF BF	00000360 00 00 00 00 00 00 00 00 00 00 00 00 0
	E 74 6C 73 00 00 00 00 00 00 10 00 00 00 60 32 %.tls2	00000370 2E 72 64 61 74 61 00 00 00 10 00 00 70 32 00 .rdatap2.
	0 02 00 00 00 6A 31 00 00 00 00 00 00 00 00 00 00jl	00000380 00 02 00 00 00 6C 31 00 00 00 00 00 00 00 00 0011
	0 00 00 00 40 00 00 EF BF BD 2E 72 64 61 74@ï¿≒.rdat	00000390 00 00 00 00 40 00 00 50 2E 69 64 61 74 61 00 00@P.idata
	0 00 00 10 00 00 00 70 32 00 00 02 00 00 00 ap2	000003A0 00 40 00 00 00 80 32 00 00 3A 00 00 00 6E 31 00 .@€2:nl.
	1 00 00 00 00 00 00 00 00 00 00 00 00 00	000003B0 00 00 00 00 00 00 00 00 00 00 00 00 0
	0 50 2E 69 64 61 74 61 00 00 00 40 00 00 00P.idata@	000003C0 2E 64 69 64 61 74 61 00 00 10 00 00 00 C0 32 00 .didataÀ2. 000003D0 00 0C 00 00 0A 31 00 00 00 00 00 00 00 00 001.
	F BD 32 00 00 3A 00 00 06 E 31 00 00 00 0 i¿*2nl	

When triaging a binary, the go-to tool is <u>Hiew</u> to investigate and look for clues for a deeper understanding. With <u>wininition</u>, I notice the <u>Embarcadero</u> string, which means that this binary was written in Delphi. When reversing Delphi binaries I've always used <u>IDR</u> (Interactive Delphi Reconstructor). IDR is a decompiler of executable files and dynamic libraries (DLL) written in Delphi.



Reversing Delphi Binaries with Ghidra and dhrake

When searching for the latest developments with IDR, I came across a fantastic plugin for Ghidra, a collection of scripts for reverse engineering Delphi binaires in Ghidra using IDR's output to IDC. It was published over a year ago, but it is a gem if threat hunters are using Ghidra.

<u>dhrake</u> allows you to import the IDC file from IDR into Ghidra. This will import the Symbol names, function signatures and create structs for Delphi classes. This plugin extracts and applies the Delphi symbols from the IDC file, which is generated by IDR, and attempts to find cases where Ghidra has incorrectly determined the entry point of a function. If you've never imported a plugin to Ghidra please read this <u>post</u>. I've saved the IDC to a selected folder. I then install the plugin in Ghidra and run the script it prompts for the IDC file and then load it!

Units (F2) Types (F4)	Forms (F5)	CodeViewer (F6)	ClassViewer (F7) Str	ings (F8) Names (F9)	SourceCode (F10) Map (F11)	
00401000 #001 00402A78 #002 00402A78 #003 00402A80 #004 00402A80 #004 00402B48 #005 00402B48 #006 0041F084 #008 0041F084 #008	_Unit1 ^ _Unit2 _Unit3 _Unit4 _Unit5 System;Generics.Defau: _Unit7 System.Types yeit4	EP <-> Src EntryPoint 00402230>		8) (8)	ⓒ 🌶 🕫▼	
094218FC #009 094218FC #019 09422164 #011 09422164 #012 094231A0 #013 094231A0 #014 09423570 #016 09423570 #016 09423570 #016 09423580 #019 0942358C #021 0942358C #021 0942358C #022 0942358C #022 0942398C #024 09423910 #025	Unit9 Unit10 Unit11 Unit12 Unit13 Unit14 Unit15 Unit15 Unit16 Unit17 Unit18 Unit19 Unit20 Unit21 Unit22 Unit22 Unit23 Unit24 Unit25 Unit26	Quick acce Desktop Libraries This PC		No item	Date modified s match your search.	Type
00428898 #027 00428898 #028 00428804C #029 00428804C #030 0042804C #031 0042605 #031 00420605 #032 00427468 #033 00437468 #034 0043746C #034 0043746C #036 0043794C #036 0043FFF0 #037	_Unit27 _Unit28 _Unit29 System.SysUtils _Unit31 System.SysUtils;SysUt: _Unit33 _Unit34 _Unit35 _Unit36 _Unit37	Network	< File name: Files of type: Split output	winintion.ldc IDC	~ ~	> Open Cancel

💋 CodeBrowser: S1:/winint	ion.exe										
File Edit Analysis Naviga	ation Search Sele	ect Tools W	/indow Help	0							
			. F K W	B - 🏹 🍅	6 CH 🗸 🖁	🐒 🖽 🛅 😋 .	h 🜔 🛄 🔶 🗐 📑] -		
Program Trees	💋 Script Manager	[CodeBrowser	r: S1:/winintio	n.exe]						- [×
🖃 🕅 winintion.exe	Help										
Headers	🜔 Script Manager - 2	2 scripts (of 24)	2)					0 9 🛛	ء 🌾 🥥 🖨	2 5	🗄 🕂 🗙
.text	Scripts	^ In Te	ool Status	Name		Description					
				DhrakeInit.java			from an IDR generated II		mbols and fix cert		
.rdata	Analysis		2	DhrakeParseClass.jav	/a	Creates structs	s based on Delphi type info	ormation.		F	8
.idata .didata	Assembly	💋 IDC File P	ath						×		
.edata	Cleanup 🗸	(-	C: \Users\Marco	Desktop Wew folder (8	3)			S	۲ 🖪		
.rsrc	CodeAnalysi		6732897	162395648.zip							
in cioc	CustomerSul	-	winintion								
	Data Data Types	My Computer	winintion								
	Delphi	3									
	Examples										
	FunctionID	Desktop									
	- FunctionStar										
Program Tree ×	ELP	Home									
Symbol Tree	Images Import										
🕀 🔂 Imports	Instructions										
🕀 🛅 Exports	ios	Recent									
Eurotions	iteration iteration										
Labels Classes	Mac OS X										
Classes Classes Classes	Memory										
	MultiUser										
	PCode Processor										
	Program										
	Project										
	References										
	Repair Control Resources										
	E Search		File name: w	vinintion.idc							
	<		_	All Files (*.*)					~		
	Filter:		Type:	All Files (1.1)					~	*	😰 🗄 🔹
Filter:	DhrakeInit.java				Load an IDC f	e Cancel					^
可 Data Type Manager	L										~

In the wininition binary, the first function WinMain has SetWindowsHookExW function, which is a hook procedure to monitor a system for certain types of events. The hook procedures low-level keyboard input events is WH_KEYBOARD_LL, which is the number 13 in the parameter. This hook is a mechanism that intercepts keystroke events. All the events are then saved to a log file to be sent to a C2.

🌮 📄 🖉 💼 🗕 🕉

```
Decompile: WinMain - (winintion.exe)
16
     undefined2 local 24;
17
     int local 18;
   undefined4 local 10;
18
19
   undefined4 local_c;
    undefined4 local_8;
20
22
   FUN_006e8980(&DAT_007037b4);
23
   local 24 = 0x18;
24
   uVar2 = FUN_00402e80();
25
     local_18 = local_18 + 1;
   FUN_0040358c();
26
27
   CVar1 = FUN 0043127c(local 8,CONCAT31((int3)((uint)extraout EDX >> 8),1),uVar2);
28
   bVar5 = cVar1 == '\0';
29
    local_18 = local_18 + -1;
30
     FUN_006ec8f0();
    if (bVar5 != false) {
      local 24 = 0x24;
      FUN_00402e80();
      local_18 = local_18 + 1;
34
35
      FUN_0040358c();
36
      (**(code **)(*gvar_007021B8 + 0x80))(gvar_007021B8,local_c);
37
      local_18 = local_18 + -1;
38
      FUN_006ec8f0();
39
     }
40
    local_24 = 0x30;
41
    uVar2 = FUN_00402e80();
42
   local_18 = local_18 + 1;
43
    FUN_00402d7c();
     cVar1 = FUN_0043127c(local_10,CONCAT31((int3)((uint)extraout_EDX_00 >> 8),1),uVar2);
44
45
   uVar4 = (uint) (cVar1 == '\0');
   local_18 = local_18 + -1;
46
47
    FUN_006ec8f0();
48
    if ((char)uVar4 != '\0') {
49
      gvar_00717404 = 0;
50
   }
51
    *gvar 007021B4 = 1800000;
52
     local_38 = FUN_00402bc0(VMT_704AE8_THREAD,CONCAT31((int3)(uVar4 >> 8),1),0);
53
     gvar_00717408 = USER32.SetWindowsHookExW(0xd,FUN_004037a0,(HINSTANCE)0x0,0);
54
     do {
55
      BVar3 = USER32.GetMessageW((LPMSG) &local_54
                                                        Hex Decimal
56
    } while (BVar3 != 0);
                                                   byte Dh
                                                                 13
57
     *in FS OFFSET = local 34;
58
     return;
                                                            '\r'
                                                   char
59 }
60
```

The C2 is obfuscated using hex that can be converted to ascii:

```
68747470733A2F2F7777772E786268702E636F6D2F646F6D696E61726772656174617369616E6F64797373
hxxps://www.xbhp[.]com/dominargreatasianodyssey/wp-content/plugins/akismet/style.php
68747470733A2F2F7777772E63346373612E6F72672F696E636C756465732F736F75726365732F66656C69
hxxps://www.c4csa[.]org/includes/sources/felims.php
```

Note: These appear to be compromised domains.

Conclusion

Analysis of these documents led us to find other Zebrocy clusters. As Zebrocy continues to evolve its scope, organizations must have the proper visibilities and detection capabilities to find this threat actor. We hope the techniques discussed in this post will be useful to other

researchers in analyzing Delphocy dropper docs in particular, and documents with passwordprotected macros in general.

Indicators of Compromise

Word Documents

SHA256

3b548a851fb889d3cc84243eb8ce9cbf8a857c7d725a24408934c0d8342d5811 1dd03c4ea4d630a59f73e053d705185e27e2e2545dd9caedb26a824ac5d11466 1e8261104cbe4e09c19af7910f83e9545fd435483f24f60ec70c3186b98603cc c213b60a63da80f960e7a7344f478eb1b72cee89fd0145361a088478c51b2c0e 2bf088955007b4f47fe9187affe65fffea234ff16596313a74958a7c85129172 d9e7325f266eda94bfa8b8938de7b7957734041a055b49b94af0627bd119c51c

SHA1

fc0b7ad2ae9347d6d2ababe2947ffb9f7cc73030 71b4b9f105de94090fc36d9226faaa1db6d9f3d1 6a8f63c4491adcf2cf7f76cd1481c5647615a6c9 a3ecf1fdc1206e9d3061530fa91775cf3d97f788 ae01ca2cf0dc07abb3a7bef9930e38c9212975d5 66b39f4fd1dd51c2f548330e5818f732dad0aa28

VBA

SHA256

a442135c04dd2c9cbf26b2a85264d31a5ac4ec5d2069a7b63bc14b64a6dd82b7

SHA1

6ec4eb883752b70db134ac0f4e0d5b4a77196184

Wininition

SHA256

ee7cfc55a49b2e9825a393a94b0baad18ef5bfced67531382e572ef8a9ecda4b SHA1 afbdb13d8f620d0a5599cbc7a7d9ce8001ee32f1

URLs

hxxps://www.xbhp[.]com/dominargreatasianodyssey/wp-content/plugins/akismet/style.php hxxps://www.c4csa[.]org/includes/sources/felims.php

Yara Rules

```
rule apt_RU_delphocy_encStrings {
 meta:
    desc = "Hex strings in Delphocy drops"
    author = "JAG-S @ SentinelLabs"
   version = "1.0"
   TLP = "White"
    last modified = "04.09.2021"
    hash0 = "ee7cfc55a49b2e9825a393a94b0baad18ef5bfced67531382e572ef8a9ecda4b"
   hash1 = "07b2d21f4ef077ccf16935e44864b96fa039f2e88c73b518930b6048f6baad74"
 strings:
    $enc_keylogger2 = "5B4241434B53504143455D" ascii wide
    $enc_keylogger3 = "5B5441425D" ascii wide
    $enc_keylogger4 = "5B53484946545D" ascii wide
    $enc_keylogger5 = "5B434F4E54524F4C5D" ascii wide
    $enc_keylogger6 = "5B4553434150455D" ascii wide
    $enc_keylogger7 = "5B454E445D" ascii wide
    $enc_keylogger8 = "5B484F4D455D" ascii wide
    $enc_keylogger9 = "5B4C4546545D" ascii wide
    $enc_keylogger10 = "5B55505D" ascii wide
    $enc_keylogger11 = "5B52494748545D" ascii wide
    $enc_keylogger12 = "5B444F574E5D" ascii wide
    $enc_keylogger13 = "5B434150534C4F434B5D" ascii wide
    scnc1 =
"68747470733A2F2F7777772E786268702E636F6D2F646F6D696E61726772656174617369616E6F6479737
 ascii wide
    scnc2 =
"68747470733A2F2F7777772E63346373612E6F72672F696E636C756465732F736F75726365732F66656C6
ascii wide
 condition:
   uint16(0) == 0x5a4d and (any of ($cnc*) or all of ($enc_keylogger*))
}
```

```
rule apt_RU_Delphocy_Maldocs {
 meta:
    desc = "Delphocy dropper docs"
    author = "JAG-S @ SentinelLabs"
   version = "1.0"
   TLP = "White"
    last modified = "04.09.2021"
    hash1 = "3b548a851fb889d3cc84243eb8ce9cbf8a857c7d725a24408934c0d8342d5811"
   hash2 = "c213b60a63da80f960e7a7344f478eb1b72cee89fd0145361a088478c51b2c0e"
   hash3 = "d9e7325f266eda94bfa8b8938de7b7957734041a055b49b94af0627bd119c51c"
    hash4 = "1e8261104cbe4e09c19af7910f83e9545fd435483f24f60ec70c3186b98603cc"
 strings:
    $required1 = "_VBA_PROJECT" ascii wide
    $required2 = "Normal.dotm" ascii wide
    $required3 = "bin.base64" ascii wide
    $required4 = "ADODB.Stream$" ascii wide
    $author1 = "Dinara Tanmurzina" ascii wide
    $author2 = "Hewlett-Packard Company" ascii wide
                               = "wininition.exe"" ascii wide
    $specific = "Caption
    $builder1 = "Begin {C62A69F0-16DC-11CE-9E98-00AA00574A4F} UserForm1" ascii wide
    $builder2 = "{02330CFE-305D-431C-93AC-29735EB37575}{33D6B9D9-9757-485A-89F4-
4F27E5959B10}" ascii wide
    $builder3 = "VersionCompatible32="393222000"" ascii wide
    $builder4 = "CMG="1517B95BC9F7CDF7CDF3D1F3D1"" ascii wide
    $builder5 =
"DPB="ADAF01C301461E461EB9E2471E616F01D06093C59A7C4D30F64A51BDEDDA98EC1590C9B191FF""
ascii wide
    $builder6 = "GC="4547E96B19021A021A02"" ascii wide
 condition:
   uint32(0) == 0xE011CFD0 and all of ($required*) and (all of ($author*) or
$specific or 5 of ($builder*))
}
```