New Locky variant – Zepto Ransomware Appears On The Scene

securityaffairs.co/wordpress/49094/malware/zepto-ransomware.html

July 7, 2016



July 7, 2016 By Pierluigi Paganini

New threat dubbed Zepto Ransomware is spreading out with a new email spam campaign. It is a variant of the recent Locky Ransomware.

The news was recently reported in a blog post by the Cisco Talos team:

"We are watching Zepto very carefully. It's closely tied to Locky, sharing many of the same attributes," said Craig Williams, senior technical leader and global outreach manager at Cisco Talos.

"There is still a lot to learn about Zepto. As far as we can tell, it's either a new variant of Locky or an entirely new ransomware with many copycat Locky features," he said.

In the last week, experts observed more than 140,000 emails using a particular naming convention to deliver a malicious attachment.

That email is generated by a template body text, where it fetches the header greeting randomly from an array followed by the [NAME] of the receiver.

As previous variants of the same malware family, the text of the email attempts to trick the victim to open the attachment.

The attachment is a .zip archive containing the hard-coded js downloader.

The naming conventions used to rename the js downloader have the following format "swift [XXX|XXX].js" where X are some combination of letters (a-f) and numbers (0-9).

Once the js file is executed through wscript it downloads the main payload binary from the C2 Servers.

Many of them have a list of hardcoded domains for download the binary, other variations use just a few domains.

That is done through HTTP GET requests to define C2 domains and the server functionalities are implemented in PHP.

We observed through dynamic analysis that it uses the same technique of Locky ransomware to decode the main payload, spawning the process through wScript with the argument '321', otherwise, the decryption routine will produce junk code and the execution flow will jump into that junk code and crash the process.

The encrypted files have the ".zepto" extensions and it targets the same extensions files of Locky taking care of the system files, it uses a lot of code of Locky ransomware to implement its malicious behaviors.

One of the smartest features of the ransomware is the fact they do not encrypt all the files needed for the correct functioning of the OS, otherwise, how can the victims pay?

Once the encryption routine of all the files is over, it shows the instructions on how to get the files back:

one picture (_HELP_instructions.jpg) and one html page (_HELP_instructions.html) are prompted to the victim for the explanation on how to unlock the files.

Following an image of a machine infected by the Zepto Ransomware:



Cisco Talos researchers tracked all the attachment they found and on 137,731 spam emails and discovered that there were 3,305 unique samples. They collected them <u>here</u>.

OUR ANALYSIS

Our main contribution will be to find the actual code that differs from the previous version of the <u>Locky Ransomware</u>. We hope to help in detecting variation on some core features (as encryption routine, files enumeration, drive enumeration...), and to allow experts to distinguish the <u>Locky ransomware</u> family from the Zepto ransomware family.

We will do this through <u>bindiff</u> software that let us to compare two binary files and calculate the differences, we will use a Locky Ransomware sample with the following hash SHA256:e5a6828f732bea6b66c4f6d850b235f6c1f139b10f8d9f2c3760298cfd88c163. So Cisco Talos researchers give us a good advice on where to start for this new variant, unfortunately, they didn't publish some samples to use in our analysis so we found some way to get them.

Search results for	swift			II Launch Fannd 🕈 5			Jser Comment	Extracted Hiles Archived Report	← Network Irathc
▲ Timestamp	Input		Threat level	Analysis Summary	Countries	Enviro	onment		
July 4 2016, 14:23 (CEST)	b8385356c0acb7c3c9f67510be217921d04b483f599e875c55783dd9115770ab_14676 010716_copy,jar Java Jar Hie data (zip) b8385356c0acb7c3c9f67510be217921d04b483f599e875c55783dd9115770ab	34230781_swift_ @ Sample (132K(B)	malicious	Threat Score: 93/100 AV Multiscan: 14% Matched 27 Signatures 🚡 🛱 Classified as <i>Agent PA</i>		Wind	ows 7 32 bit		
July 4 2016, 3:38 (CEST)	swift copy.zip Java Jar file data (21p) d188b1554d908397d151609887b996e3c31b2b67f041c0e7b8b79150faf02998	Sample (593KiB)	malicious	Threat Score: 65/100 AV Multiscan: 11% Matched 22 Signatures 🕌 🚅 Classified as <i>Agent PA</i>	•	Wind	ows 7 32 bit		
July 2 2016, 9:03 (CEST)	swift ca6.js ASCII text, with very long lines, with CRLF line terminators 068e08f01e117f66f607a27492a500cc?c3ffa91cac76dcebbe97667394a9cde	Sample (6.8KiB)	malicious	Threat Score: 100/100 Matched 31 Signatures 📱 🚅	1999 III 1	Wind	ows 7 32 bit		
July 1 2016, 19:35 (CEST)	swift transfer.doc data ef60357252d7ba61d89atd826740259f5b471f79147f4et12ec27756dc5eftd7		malicious	Threat Score: 100/100 AV Multiscan: 15% Matched 24 Signatures b ≓ f f Classified as <i>Exploit Rtf Heuristic</i>		Wind	ows 7 32 bit		
July 1 2016, 18:14 (CEST)	Swift Copy.pub Composite Document File V2 Document, Little Endian, Os: Windows, Version 6.2, Co b07c3e58c9e3090204269bf55f0c449ca3b0b8c86a0806d0eaff8f710a3feae4	Sample (158KiB)	malicious	Threat Score: 45/100 AV Multiscan: 15% Matched 15 Signatures Classified as <i>Downloader.act</i>	-	Wind	ows 7 32 bit		
July 1 2016, 18:09 (CEST)	Inv_payment_swift.exe PE32 executable (GUI) Intel 80386 Mono/. Net assembly, for MS Windows e00fc0a08805dcdff84062d3adce902f258d8814dd4f50a7c33bc490fc3408d4	Sample (786KiB)	suspicious	Threat Score: 43/100 Matched 19 Signatures	-	Wind	ows 7 32 bit		
July 1 2016, 14:50 (CEST)	pdf_copy-manuel-ortiz_802890.zip ASCII text, with CRLF, LF line terminators 6fe554961bb4337bb1589f39d9e0482085fc081a8f7b7b029d2d8a53118d8a73	() Sample (7.6KiB)	malicious	Threat Score: 100/100 Matched 36 Signatures 📱 🛤		Wind	ows 7 32 bit		
July 1 2016, 14:13 (CEST)	swift 2c1,is ASCII text, with very long lines, with CRLF, LF line terminators Sceb478b2af8d75f7ca0S5b7d67e4d62e7bbc36b4299ta7757d0ae7ct25373d	Sample (7.3KiB)	malicious	Threat Score: 100/100 AV Multiscan: 42% Matched 25 Signatures 👔 ≓ Classified as <i>Trojan GenericKD</i>		Wind	ows 7 32 bit		
July 1 2016, 3:00 (CEST)	swift O6ae.js ASCII text, with CRLF, LF line terminators 8191195dbf4976ae2cte2a07111ad8bf733c8e27862b34423f3b1aded3dc586	Sample (7.9KiB)	malicious	Threat Score: 100/100 AV Multiscan: 11% Matched 31 Signatures 🗈 😅 Classified as <i>Nemucod.FJ</i>	35 	Wind	ows 7 32 bit		

We grabbed the most recent one in order to study the most recent variant.

The file name is "swift ca6.js"

SHA256:068e08f01e117f66f607a27492a500cc7c3ffa91cac76dcebbe97667394a9cde. As we can see, the file has the same name pattern discovered by Cisco Talos researchers.

Now we will need to extract the main payload from the execution of the JavaScript file. We will monitor our file system activities with procMon tools and we will take care on the dropped files of the malicious js.

🧿 Process Monitor - Sysinterr	nals: www.sysinternals.com		
File Edit Event Filter To	ools Options Help		
🛎 🖬 🔍 隆 🖾	▽ Δ 🚱 🗉 🛤 🐬 😹 🔜 🖙 🗷		
Time Process Name	PID Operation Path	Result	Detail
01:37: a WScript.exe	1072 mcRegCloseKey HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	SUCCESS	
01:37: a WScript.exe	1072 🔐 RegOpen Key HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	NAME NOT FOU	JND Desired Access: Q
01:37: a WScript.exe	1072 戱 RegOpenKey HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	SUCCESS	Desired Access: Q
01:37: 🗃 WScript.exe	1072 🍂 RegQueryValue HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun	NAME NOT FOU	JND Length: 144
01:37: a WScript.exe	1072 📸 RegCloseKey HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	SUCCESS	
01:37: 🗃 WScript.exe	1072 📸 RegOpenKey HKCU\Software\Microsoft\Windows\CurrentVersion\App Paths\9sX6D0L0xLSf.exe	NAME NOT FOU	JND Desired Access: R
01:37: 🗃 WScript.exe	1072 KgegOpenKey HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths\9sX6D0L0xLSf.exe	NAME NOT FOU	JND Desired Access: R
01:37: 🗃 WScript.exe	1072 📸 RegQueryKey HKCR\exefile\shell\open	SUCCESS	Query: Name
01:37: 📺 WScript.exe	1072 🕰 RegOpenKey HKCU\Software\Classes\exefile\shell\open	NAME NOT FOU	JND Desired Access: M
01:37: 👛 WScript.exe	1072 🧟 RegQueryValue HKCR\exefile\shell\open\SetWorkingDirectoryFromTarget	NAME NOT FOU	JND Length: 144
01:37: 👝 WScript.exe	1072 戱 RegQueryKey HKCR\exefile\shell\open	SUCCESS	Query: Name
01:37: 🚰 WScript.exe	1072 🕰 RegOpenKey HKCU\Software\Classes\exefile\shell\open	NAME NOT FOU	JND Desired Access: M
01:37: 👝 WScript.exe	1072 🧟 RegQueryValue HKCR\exefile\shell\open\NoWorkingDirectory	NAME NOT FOU	JND Length: 144
01:37: 👝 WScript.exe	1072 <a>KegOpenKey <a>HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume	SUCCESS	Desired Access: R
01:37: 👝 WScript.exe	1072 RegOpenKey HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume\{5d9a6f44-941e-11e5-a771-806e6f6e6963}\	SUCCESS	Desired Access: R
01:37: 👝 WScript.exe	1072 KgcgCloseKey HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume	SUCCESS	
01:37: 👝 WScript.exe	1072 Mc BegQueryValue HKCU/Software/Microsoft/Windows/Current/Version/Explorer/MountPoints2/CPC/Volume/{5d9a6f44-941e-11e5-a771-806e6f6e6963}/Generation	SUCCESS	Type: REG_DWO
01:37: 👝 WScript.exe	1072 RegCloseKey HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume\{5d9a6f44-941e-11e5-a771-806e6f6e6963}	SUCCESS	
01:37: 👝 WScript.exe	1072 C:\Users\Administrator\Desktop	SUCCESS	Desired Access: R
01:37: 👝 WScript.exe	1072 🛃 Query Basic InforC:\Users\Administrator\Desktop	SUCCESS	Creation Time: 26/1
01:37: 👝 WScript.exe	1072 ScloseFile C:\Users\Administrator\Desktop	SUCCESS	
01:37: 👝 WScript.exe	10/2 🖹 CreateFile C:\Users\Administrator\AppData\Local\Iemp\9sX6D0L0xLSf.exe	SUCCESS	Desired Access: R
01:37: 👝 WScript.exe	1072 🛃 WriteFile C:\Users\Administrator\AppData\Local\Temp\9sX6D0L0xLSf.exe (1)	SUCCESS	Offset: 0, Length: 1
01:37: 👝 WScript.exe	1072 SetEndOfFileInfC:\Users\Administrator\AppData\Local\Temp\9sX6D0L0xLSf.exe	SUCCESS	EndOfFile: 165.888
01:37: ov vv Scnpt.exe	1072 🖻 Create HielwappC. Nusers vacministrator vappulata Nocal Niemp V35Abuuluk Lor exe	SUCCESS	SyncType: SyncTy
01:37: 👝 WScript.exe	1072 🛃 CreateFileMappC:\Users\Administrator\AppData\Local\Temp\9sX6D0L0xLSf.exe	FILE LOCKED W	/I SyncType: SyncTy
01:37: 👝 WScript.exe	1072 🛃 QueryStandardIC:\Users\Administrator\AppData\Local\Temp\9sX6D0L0xLSf.exe	SUCCESS	AllocationSize: 167
01:37: 🍙 WScript.exe	1072 🛃 Create File MappC:\Users\Administrator\AppData\Local\Temp\9sX6D0L0xLSf.exe	SUCCESS	SyncType: SyncTy
01:37: 👝 WScript.exe	1072 KagopenKey HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\9sX6D0L0xLSf.exe	NAME NOT FOU	JND Desired Access: Q
01:37: 👝 WScript.exe	1072 🛃 Query Security File C:\Users \Administrator \App Data \Local \Temp \9sX6D0L0xLSf.exe	SUCCESS	Information: Label
D1:37: 👝 WScript.exe	1072 🗟 QueryNameInfoC:\Users\Administrator\AooData\Local\Temo\9sX6D0L0xLSf.exe	SUCCESS	Name: \Users\Ad
01:37 💣 WScript.exe	1072 🎝 Process Create C:\Users\Administrator\AppData\Local\Temp\9sX6D0L0xLSf.exe	SUCCESS	PID: 3224, Comma
01:37 🗸 9sX6D0L0xLSf.exe	3224 27 Process Start (2)	SUCCESS	Parent PID: 1072,
01:37 V9sX6D0L0xLSf.exe	3224 🖓 Thread Create	SUCCESS	Thread ID: 2100
01:37 e w scnpt.exe	1072 🚉 Regupenkey HkLim System (CurrentControliSet (Control Session Manager AppCertUlis	REPARSE	Desired Access: Q
01:37: a WScript.exe	1072 dt RegOpenKey HKLM\System\CurrentControlSet\Control\Session Manager\AppCertDlls	NAME NOT FOU	JND Desired Access: Q

In the above image, we can see in (1) that the script creates the binary file and create a process launching it (2).

We found interesting that the js downloader calls the binary with an argument needed for

decoding the main payload like the Locky ransomware and most weird is that it uses the same argument for the decryption routine:

vent Proce	ss Stack					
Image						
	dout					
•	Siber Systems					
Name:	9sX6D0L0xLSf.exe					
Version:	1, 0, 2, 1					
Path:						
C:\Users\	Administrator \AppData \Loc	al\Temp\9s	X6D0L0xLSf.	exe		
Command Lines						
Command L	ine:					
Command L	ine: Administrator\AppData\Loo	cal\Temp\9	sX6D0L0xLSf	.exe" 321		
Command L "C:\Users	ine: (Administrator (AppData (Loc	cal\Temp\9	sX6D0L0xLSf	exe" 321.		
Command L "C:\Users" PID:	ine: Administrator\AppData\Loo 3224	cal\Temp\9 A	sX6D0L0xLSf	exe" 321		
Command L "C:\Users PID: Parent PID:	ine: Administrator\AppData\Loo 3224 1072	cal\Temp\9 A V	sX6D0L0xLSf rchitecture: irtualized:	exe" 321 32-bit False		
Command L "C: \Users" PID: Parent PID: Session ID:	ine: (Administrator\AppData\Loo 3224 1072 1	cal\Temp\9 A V Ir	sX6D0L0xLSf rchitecture: irtualized: ntegrity:	exe" 321 32-bit False High		
Command L "C:\Users" PID: Parent PID: Session ID: User:	ine: Administrator\AppData\Loo 3224 1072 1 username-PC\Administrat	cal\Temp\9 A V Ir tor	sX6D0L0xLSf rchitecture: irtualized: ntegrity:	exe" 321 32-bit False High		
Command L "C:\Users" PID: Parent PID: Session ID: User: Auth ID:	ine: Administrator\AppData\Loo 3224 1072 1 username-PC\Administrat 00000000:00017606	cal\Temp\9 A V Ir tor	sX6D0L0xLSf rchitecture: irtualized: ntegrity:	exe" 321 32-bit False High		
Command L "C: \Users" PID: Parent PID: Session ID: User: Auth ID: Started:	ine: Administrator\AppData\Loo 3224 1072 1 username-PC\Administrat 00000000:00017606 04/07/2016 01:37:20	cal\Temp \9 A V Ir tor E	sX6D0L0xLSf rchitecture: irtualized: ntegrity: nded:	exe" 321 32-bit False High 04/07/201	16 01:37:20	
Command L "C: \Users" PID: Parent PID: Session ID: User: Auth ID: Started: Modules:	ine: (Administrator\AppData\Loo 3224 1072 1 username-PC\Administrat 00000000:00017606 04/07/2016 01:37:20	cal\Temp\9 A V Ir tor E	sX6D0L0xLSf rchitecture: irtualized: ntegrity: nded:	exe" 321 32-bit False High 04/07/201	16 01:37:20	

Once identified the dir location of the ransomware payload we could identify him: SHA256:5bbc9afa3128956b3f6116037cc97d0ea1c79d8bb5d3e15473d1e9c5c8eecfdf

The only problem we face executing it is that the ransomware does not execute itself but it changes its behavior killing its main thread and it auto-delete itself maybe because it detects the virtual environment.

So start to patch that binary in order to study our sample:

We open the executable with the Olly debugger with the argument 321 and starting analyzing the code searching for some tricks used for vm detection.

Looking at the list of the intermodular calls we investigated on the *GetProcAddress* syscall and we found something interesting:

C File V	'iew	Debug Plugin	s Options Window Help Tools BreakPoint->	
2	D		2 4: 🔁 🚺 La E Me Ta Wi Ha Co Pa St Br Re Tr St 🔯 📙 ? 🔶 🕂 🛠 🛨 Str HPP 🕀 📃	
013A2C94		CC	INT3	*
013A2C95	гş	55	PUSH EBP	
013A2C96		8BEC	MOV EBP,ESP	
013A2C98		83EC 24	SUB ESP,24	
013A2C9B		56	PUSH ESI	
013A2C9C		57	PUSH EDI	
013A2C9D	•	68 E8943B01	PUSH 95X6D0L0.013B94E8 ProcNameOrOrdinal = "CloseHandle"	
013A2CA2		68 F4943B01	PUSH 99X6DDL0.013B94F4 FileName = "kernel32.dll"	
013A2CA7	÷.,	33FF	XOR EDI,EDI	
013A2CA9	•	FF15 EC913B01	CALL DWORD FIR [<&KERNEL32.LoadLibraryW]	
013A2CAF	•	50	PUSH EAX hModule	
013A2CB0		FF15 E8913B01	CALL DWORD FTR [<&KERNEL32.GetProcAddre GetProcAddress	
013A2CB6	۰.	A3 8C0E3C01	MOV DWORD FIR [13C0E8C], EAX	
013A2CBB	•	B8 79B5E236	MOV EAX, 36E2B579	
013A2CC0	٠.	8945 EC	MOV DWORD PIR [EBP-14], EAX	
013A2CC3	٠.	8945 F0	MOV DWORD PIR [EBP-10], EAX	
013A2CC6	٠.	C745 E0 F87A	MOV DWORD FIR [EBP-20], 6F7AF8	
013A2CCD	٠.	C745 E4 91554	MOV DWORD PIR [EBP-1C],9A485591	
013A2CD4	>	8D2424	LEA ESP, DWORD PTR [ESP]	
013A2CD7	_	0F31	RDISC	
013A2CD9	٠.	55	POS REP	
013A2CDA	٠.	50		
UI3A2CDB	÷.,	8945 14		
UI3A2CDE	٠.	PELS FUSISBU	CALL DWORD FIR [< KRENEL32.GetFrocessne CetFrocessneap	
012A2CE4	•	0521		
DISAZCE0		FESE CODESEO		
013A2CE0		8845 88		
01322CF1		3345 FC		
013A2CF4		8945 F8	MOV DWORD PTR (FBP-81 FBX	
013A2CF7		3345 F0	XOR FAX, DWORD PTR (FRP-10)	
013A2CFA		8175 F8 79B5F	XXR DWORD PTR [FRP=8] 3/F2/B5/9	
013A2D01		6A 00		
013A2D03		FF15 8C0E3C01	CALL DWORD FTR [13C0E8C]	
013A2D09		FF75 E0	PUSH DWORD PTR (EBP-201	
013A2D0C		8F45 DC	POP DWORD PTR (EBP-24)	
013A2D0F		0F31	RDTSC	
013A2D11		8D36	LEA ESI, DWORD FTR [ESI]	
013A2D13		3345 E4	XOR EAX, DWORD PIR [EBP-1C]	

This ransomware uses the RDTSC anti-vm technique:

"The Time Stamp Counter (TSC) is a 64-bit register present on all x86 processors since the Pentium. It counts the number of cycles since reset". (Wikipedia)

If the code is being emulated then, there will be a change in the time stamp between. The Result in stored in EDX:EAX format.

Now the time difference in a real host machine would be usually less than 100, but if the code is emulated the difference will be huge.

Filling those instructions with NOP and patching the executable let us successfully launch the ransomware.

Now we wait until it decodes itself and, when it will contact the domains to take the RSA key (it means it decoded itself and loaded in memory), we will suspend the process in order to dump it from the memory for our further analysis.

We will use a useful tool for dump a process loaded into the memory: Process Dump.

∶:∖dump≯	pd32.	.exe	-pid	1748
n –	T	-		

Copyright @ 2015, Geoff McDonald http://www.split-code.com/ https://github.com/glmcdona/Process-Dump

dumping process 9sX6D0L0xLS_ anti-debug1_exe with pid 0x6d4...

dumping	'exe'	at	5F0000 to) fi	le '9	9sX6D0L0xLS_ anti-debug1_exe_hidden_5F0000.exe'
մատքողց	exe	dL	тоторов (,U I	ile .	<u>'ishobelexla_ anti-uebuyi_exe_ishobelexla_ anti-uebug1.exe_1010000.exe'</u>
dumping	'd11'	at	6F260000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_sensapi.dl1_6F260000.dl1'
dumping	'd11'	at	7090000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_MPR.d11_709C0000.d11'
dumping	'd11'	at	70A20000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_RASAPI32.d11_70A20000.d11'
dumping	'd11'	at	70DD0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_rasadh1p.d11_70DD0000.d11'
dumping	'd11'	at	71D20000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_wshqos.dll_71D20000.dll'
dumping	'd11'	at	71E90000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_rasman.dll_71E90000.dll'
dumping	'd11'	at	72220000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_oledlg.d11_72220000.d11'
dumping	'd11'	at	737C0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_WINNSI.DLL_737C0000.dll'
dumping	'd11'	at	737D0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_iph1papi.DLL_737D0000.d11'
dumping	'dll'	at	73970000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_DSR0LE.DLL_73970000.d11'
dumping	'd11'	at	73C20000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_rtutils.dl1_73C20000.dl1'
dumping	'd11'	at	73C50000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_NLAapi.dll_73C50000.dll'
dumping	'd11'	at	73E90000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_ntmarta.d11_73E90000.d11'
dumping	'd11'	at	73ED0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_wkscli.dll_73ED0000.dll'
dumping	'd11'	at	73EE0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_netutils.d11_73EE0000.d11'
dumping	'd11'	at	73EF0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_NETAPI32.d11_73EF0000.d11'
dumping	'd11'	at	74440000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_uxtheme.dl1_74440000.dl1'
dumping	'd11'	at	74670000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_comct132.d11_74670000.d11'
dumping	'd11'	at	74CD0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_wshtcpip_DLL_?4CD0000.dll'
dumping	'd11'	at	74F60000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_rsaenh.dl1_74F60000.dl1'
dumping	'd11'	at	75040000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_dnsapi.DLL_75040000.d11'
dumping	'd11'	at	75170000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_wship6.d11_75170000.d11'
dumping	'd11'	at	75180000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_mswsock.dll_75180000.dll'
dumping	'dll'	at	751C0000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_CRYPTSP.dl1_751C0000.dll'
dumping	'd11'	at	75360000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_srvcli.dll_?5360000.dll'
dumping	(dii)	at	75620000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_SspirLi.dll_75620000.dll'
dumping	(dii)	at	75670000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_CRYPTBASE.dll_75690000.dll'
dumping	(dii)	at	75740000	to	tile	'9sX6U0L0xLS_ anti-debug1_exe_protapi.dll_?5740000.dll'
dumping	(d11)	at	75780000	to	file	'9sX6D0L0xLS_ anti-debug1_exe_MSHSN1.dll_757B0000.dll'
dumping	(d11)	at	75810000	to	tile	'YSX5DU0L0xLS_ anti-debug1_exe_REKNELBHSE.dll_75810000.dll'
dumbina	(dii)	at	75890000	to	tile	YSX6D0L0xLS_ anti-debug1_exe_CRYP132.dll_75870000.dll
dumping	(dit.	at	75H40000	to	tile	'YSX6D0L0XLS_ anti-debug1_exe_SHLWHF1.d11_75H40000.d11'
dumping	(dii)	at	75880000	to	tile	'YSX6D0L0xLS_ anti-debug1_exe_LPK.dll_75HH0000.dll'
aumping		at	75880000	τo	tile	'YSK6D0L0XLS_ anti-debugi_exe_kernei32.dli_75H80000.dli
dumping	, all.	at	75870000	τo	tile	'YSKEDØLØXLS_ ANTI-debugi_exe_HD0HF132.dll_/SEY0000.dll'
dumping	, all.	at	75630000	τo	f 11e	'YSKBUOLOXLS_ ANTI-GEDUGI_EXE_NSLIF.GIL.75L30000.dll'
dumping	, 411,	at	75000000	τo	f 11e	'YSABUQUEXLS_ anti-debugi_exe_VLEHUI32.all_/SU00000.all'
dumping	, , , , , , ,	at	75070000	to	rile c:le	JOSKODULUXLS_ ANTI-CEDUGI_EXE_RFGRI4.CII_75D70000.CII
dumping	, 411,	at	75240000	to	rile	7SADDULUXLS_ AILT-UEDUUL_EXE_GDI32.UII_7SE40000.UII
dumping	, , , , , , ,	at	75270000	to	f 11e	75ADDULUXLS_ ANTI-GEDUGI_EXE_USFID.AII_75E70000.AII
dumping	, 411,	at	73730000	to	rile	75ADD0L0XLS_ ANTI-CEDUQI_EXE_USEAS2.CII_75F30000.CII
dumping	, 411,	ai -+	70110000	τυ •	r 11e	75ADDULUXLS_ dHtl=UEUUU1_EXE_SHELU22.UI1_76II0000.UI1
dumping	, 411,	at	76066666	to	r 11e	Jost Dalayle _ anti-uelugi_exe_urindi.uii_foldeded.uii
dumping	, 411,	at	76 26 00000	to	r 110	JOSTO DE DE LA CALLA DE LA CALA DE LA CALLA DE LA CALLA DE LA CALLA DE LA CALLA DE LA CALA
dumping	, 411,	at	76 880000	to	file	Jost Delevis _ anti-debugi_exe_wide 12. ati_70700000.uti
dumping	, 411,	at	76 FR0000	to	f i 10	/9s/CD0L0ALS_ anti-debug1_exe_WS2_22.DUL_/OFD0000.011
dumping	'411'	at	770F0000	to	file	1988/DOLOVIS anti-debugi exe instutil dil 770F0000 dil'
dumping	1111	at	77490000	to	file	'9886D0L0XLS_anti-debugi_exe_leve2 dll 22490000 dll'
dumping	1111	at	775 F0000	to	file	'9886D0L0XLS_anti-debug1_exe_htdl1.dl1 225F0000.dl1'
dumping	1111	at	77740000	to	file	'9886D0L0XLS_anti-debugi_exe_Ns1.dll 2724000.dll'
dumping	1111	at	77750000	to	file	'9886D0L0XLS_anti-debugi_exe_IMM32.DLL 77250000.dll'
dumping	'd11'	at	77800000	to	file	'9s86D0L0xLS_anti-debugi_exe_sechost.dll 77800000.dll'

The highlighted file will be our unpacked sample of Zepto Ransomware.

Now we have our fresh and unpacked sample of Zepto Ransomware and we need to produce the .idb files of the two ransomware used for the comparison in the bindiff software, ida Pro will do this easily.

So let's compare our two ransomware and look at the results:



The first result returned by the tool are pretty pie graphs where we have the numbers of Functions, Calls, Basic Blocks and Jumps.

In green there are the matched elements (included also the changed), in red we have the new Zepto ransomware elements that aren't present in the Locky ransomware and in gray we have the Locky elements that aren't present in Zepto Ransomware.

Overall, the two binary have a similarity coefficient of **0.86** that is high for two different families of a ransomware.

As we can see from the lower part of the image there is a table representing our results, the Zepto ransomware has more functions, calls, basic blocks, jumps and instructions than locky ransomware.

And interesting enough are the results shown in Secondary Call Graph window saying that the **99.9%** on 821 functions of locky ransomware are matched with the Zepto ransomware and 15 functions changed (**1.8%**), impressive is that just 1 (**0.1%**) function unmatched. On the left window, we can see that 149 functions are unmatched (**15.4%**), it means there are added functions to the new version of that ransomware.

In the overall instructions of Locky ransomware (24,947) we have the **96.9%** of identical codes and just **3.1%** of different instructions.

How much changed Zepto Ransomware and how many new features it has? Well, answering exactly can't be that easy, but we can give you some good statistical numbers.

We can tell you that the Zepto ransomware has, of course, more overall instructions than locky ransomware, it has 32,292 and over that there are 8,110 new instructions, so **25.1%** new code.

It means that for sure that Zepto ransomware will have some new behavior than locky ransomware, but in most aspects, it will act as locky ransomware, but also with little improvements it will still avoid the av engine.

It looks like the author of the ransomware take the previous code of the locky ransomware and added new features and changed some code to evade signature-based detection.

Let's investigate on some changed functions and try to extract some big difference. Looking at the list of the matched functions, we can easily identify the functions that changed for this new version of the ransomware thank at the **similarity coefficient** computed by bindiff tool:

02	207820 Matched I	Functions												
						-	🔀 🎲 🗹 Show structura	I changes 🗹 SI	how only	y instructio	ons ch	anged	Show	identical
	Similarity A	Confidence	Address	Primary Name	Туре	Address	Secondary Name	Type	E	asic Bloc	ks		Jumps	
困	0.38	0.54	0010203E	sub 10203E	Normal	00403670	sub 403670	Normal	11	19	2	29	15	15 🔺
25	0.42	0.62	00111A5C	sub 111A5C	Normal	0040364D	sub 40364D	Normal	0	1	0			=
25	0.42	0.70	001087FC	sub 1087FC	Normal	00402045	sub 402045	Normal	23	23	7	51	16	28
2	0.50	0.99	001171C0	RtlUnwind	Imported	0040FF92	RtIUnwind	Thunk		1				
25	0,51	0,91	00103804	sub 103804	Normal	00407F65	sub 407F65	Normal	21	21	25	54	6	61
120	0.89	0,99	00104B84	sub 104B84	Normal	00404641	sub 404641	Normal	11	71	7	28	95	21
不	0,93	0,94	00106B9F	sub_106B9F	Normal	004062F7	sub_4062F7	Normal	0	3	0	0	3	0
120	0,94	0,95	0010904E	sub_10904E	Normal	004034C1	sub_4034C1	Normal	0	3	0	0	3	0
125	0,95	0,95	00115CD7	sub_115CD7	Normal	0041121B	sub_41121B	Normal	0	4	0	0	3	0
ふ	0.95	0.95	00115CF0	sub 115CF0	Normal	00411250	sub 411250	Normal	0	4	0	0	3	0
125	0.95	0.96	00103637	sub 103637	Normal	00408787	sub 4087B7	Normal	0	3	0	0	3	0
125	0,97	0,97	001113B8	sub_1113B8	Normal	00402020	sub_402020	Normal	0	3	0	0	3	0
-	0.97	0.97	00102019	sub 102019	Normal	0040FF70	sub 40FF70	Normal	0		- 0	0		0
困	0.98	0.99	00102281	sub 102281	Normal	00402288	sub 402288	Normal	0	78	0	0	121	0
12	0,99	0,00	00104700	sub_104703	Normal	00404307	sub 404307	Normal	0	- 26	- 0	0	51	•
1.5	1.00	0.00												
	1.00	0,99	0011/0CC	EnterCriticalSection	Imported	004120A8	EnterCriticalSection	Imported						
4	1.00	0,99	001170CC 001161B0	sub_1161B0	Normal	004120A8 004114FE	sub_4114FE	Imported Normal	0	2	0	0	1	0
黛	1,00	0,99	001170CC 001161B0 00103238	sub_1161B0 sub_103238	Normal Normal	004120A8 004114FE 004030C2	sub_4114FE sub_4030C2	Normal Normal	0	2 16	0	0	1 20	0
	1,00 1,00 1,00 1,00	0,99 0,95 0,98 0,99	001170CC 001161B0 00103238 0010AF32	sub_1161B0 sub_116238 sub_103238	Normal Normal Normal	004120A8 004114FE 004030C2 0040A694	EnterCriticalSection sub_4114FE sub_4030C2 sub_40A694	Normal Normal Normal	0	2 16 4	0	0 0 0	1 20 5	0
	1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,98 0,99 0,99	001170CC 001161B0 00103238 0010AF32 00106592	EnterCriticalSection sub_1161B0 sub_103238 sub_10AF32 sub_106592	Normal Normal Normal Normal Normal	004120A8 004114FE 004030C2 0040A694 00405950	EnterCriticalSection sub_4114FE sub_4030C2 sub_40A694 sub_405950	Normal Normal Normal Normal Normal	0 0 0 0 0 0	2 16 4 17	0	0 0 0 0 0	1 20 5 22	0
	1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,98 0,99 0,99 0,99	001170CC 001161B0 00103238 0010AF32 00106592 00106592	EnterCriticalSection sub_1161B0 sub_103238 sub_10AF32 sub_106592 sub_103199	Normal Normal Normal Normal Normal	004120A8 004114FE 004030C2 0040A694 00405950 00403023	EnterCriticalSection sub_4114FE sub_4030C2 sub_40A694 sub_405950 sub_403023	Normal Normal Normal Normal Normal	0 0 0 0 0 0 0 0 0 0	2 16 4 17 3	000000000000000000000000000000000000000	0 0 0 0	1 20 5 22 3	
	1,00 1,00 1,00 1,00 1,00 1,00	0,95 0,95 0,98 0,99 0,99 0,99 0,95	001170CC 001161B0 00103238 0010AF32 00106592 00103199 00115EB7	EnterCriticalSection sub_1161B0 sub_103238 sub_103238 sub_103238 sub_106592 sub_103199 sub_115EB7	Normal Normal Normal Normal Normal Normal	004120A8 004114FE 004030C2 0040A694 00405950 00403023 00411357	EnterCriticalSection sub_4114FE sub_4030C2 sub_4030C2 sub_405950 sub_405950 sub_403023 sub_411357	Imported Normal Normal Normal Normal Normal		2 16 4 17 3 2	000000000000000000000000000000000000000	0 0 0 0 0	1 20 5 22 3 1	
	1,00 1,00 1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,98 0,99 0,99 0,99 0,95 0,95	001170CC 001161B0 00103238 0010AF32 00106592 00103199 00115EB7 00111845	EnterCriticalSection sub_116180 sub_103238 sub_10AF32 sub_106592 sub_106592 sub_113EB7 sub_115EB7 sub_11845	Normal Normal Normal Normal Normal Normal Normal	004120A8 004114FE 004030C2 0040A694 00405950 00403023 00411357 00410165	EnterCriticalSection sub_4114FE sub_4030C2 sub_40A694 sub_406950 sub_403023 sub_411357 sub_410165	Imported Normal Normal Normal Normal Normal Normal		2 16 4 17 3 2 5	0 0 0 0 0	0 0 0 0 0 0 0	1 20 5 22 3 1 5	
	1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,99 0,99 0,99 0,99 0,95 0,99 0,99	00117/0CC 001161B0 00103238 0010AF32 00106592 00103199 00115EB7 00111845 00101A7E	EnterCriticalSection sub_103238 sub_103238 sub_104F32 sub_106592 sub_106592 sub_115EB7 sub_111845 sub_111845	Normal Normal Normal Normal Normal Normal Normal Normal	004120A8 004114FE 004030C2 0040A694 00405950 00403023 00411357 00410165 00401A85	EnterCriticalSection sub_4114FE sub_4030C2 sub_4030C2 sub_4030550 sub_403023 sub_411357 sub_410165 sub_410165	Imported Normal Normal Normal Normal Normal Normal Normal		2 16 4 17 3 2 5 3			1 20 5 22 3 1 5 3	
	1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,99 0,99 0,99 0,99 0,99 0,95 0,99 0,99	00117/0CC 001161B0 00103238 0010AF32 00106592 00103199 00115EB7 00111845 00101A7E 001004D2	EnterCriticalSection sub_116180 sub_103238 sub_10AF32 sub_106592 sub_1135E87 sub_111845 sub_101A7E sub_101A7E sub_101A7E	Normal Normal Normal Normal Normal Normal Normal Normal	004120A8 004114FE 004030C2 0040694 00405950 00403023 00411357 00410165 00401A85 00401A85	EnterCriticalSection sub_4114FE sub_4030C2 sub_405950 sub_405950 sub_403023 sub_411357 sub_401065 sub_401A85 sub_401A85	Normal Normal Normal Normal Normal Normal Normal Normal		2 16 4 17 3 2 5 3 23			1 20 5 22 3 1 5 3 38	
	1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,98 0,99 0,99 0,99 0,99 0,99 0,99 0,99	001161B0 001061B0 00104F32 00106592 00103199 00115EB7 001115EB7 00111845 00101A7E 0010CED2 0011724C	EnterCritical/Section sub_116180 sub_103238 sub_104732 sub_105432 sub_10569 sub_115E97 sub_111845 sub_101A7E sub_10CED2 HeapSetInformation	Normal Normal Normal Normal Normal Normal Normal Normal Normal	0041120A8 004114FE 004030C2 0040A694 00405950 00403023 00411357 00410165 00401A85 00400L22 00412EC	EnterCriticalSection sub_4114FE sub_4030C2 sub_4030C2 sub_40504 sub_40503 sub_411537 sub_410165 sub_401085 sub_4008CD2 HeapSetInformation	Imported Normal Normal Normal Normal Normal Normal Normal Imported		2 16 4 17 3 2 5 3 23			1 20 5 22 3 1 5 3 38	
	1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,99 0,99 0,99 0,99 0,99 0,99	00116180 00106180 00108328 00108592 00103199 00115E87 00111845 00101A7E 001047E 001047E 001047E 001047E	EnterCriticalSection sub_116180 sub_103238 sub_104732 sub_106592 sub_113199 sub_1115E87 sub_111845 sub_101A7E sub_101A7E HeapSetInformation sub_1121C0	Normal Normal Normal Normal Normal Normal Normal Normal Imported Normal	0041120A8 004114FE 004030C2 0040694 00405950 00403023 00411357 00410165 00401A85 00400ED2 004121EC 00410A86	EnterCriticalSection sub_4114FE sub_4030C2 sub_405950 sub_405950 sub_403023 sub_411357 sub_401065 sub_401A85 sub_401A85 sub_40LA85 sub_40A86	Imported Normal Normal Normal Normal Normal Normal Normal Imported Normal		2 16 4 17 3 2 5 3 23 8			1 20 5 22 3 1 5 3 38 10	
	1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00 1,00	0,99 0,95 0,98 0,99 0,99 0,99 0,99 0,99 0,99 0,99	00116180 00116180 00104532 00106592 00103238 00106592 0010582 001015E87 001015E87 001015E87 00101726 0010724C 0011724C 00102E02 00102E03	Enter/initialsection sub_116180 sub_103238 sub_104732 sub_104732 sub_10592 sub_10592 sub_11687 sub_111845 sub_10177E sub_1077E sub_1077E sub_102802 sub_102803	Normal Normal Normal Normal Normal Normal Normal Normal Imported Normal Normal	0041120A8 004114FE 004030C2 00405950 00405950 00405950 004015550 0040141357 00411357 0041165 00401A85 00410A85 00411EC 00410A86 0040883	EnterCriticalisection sub_4114FE sub_4030C2 sub_4030C2 sub_40595 sub_40595 sub_401367 sub_41367 sub_41367 sub_410185 sub_401A85 sub_40A85 sub_40A85 sub_40A883	Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal		2 16 4 17 3 2 5 3 23 8 1		0 0 0 0 0 0 0 0 0 0 0	1 20 5 22 3 1 5 3 38 10	
	1,00 1,00	0,99 0,95 0,98 0,99 0,99 0,99 0,99 0,99 0,99 0,99	001161B0 001161B0 00103238 00106592 00103528 00103592 00115EB7 001115EB7 001011845 001011724C 0011724C 0011724C 0011724C 0010BD83 00101718C	EnterCriticalsection sub_116180 sub_106732 sub_106732 sub_106592 sub_113199 sub_116592 sub_111845 sub_10175 sub_10175 sub_10075 sub_10075 sub_100083 setFilePointer	Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal	0041120A8 004114FE 004030C2 0040A694 00405950 00403023 00411357 004110165 004010485 0040BCD2 004121EC 00410A86 0040AB83 0041248	EnterCriticalisection sub_4114F2 sub_404694 sub_404694 sub_409690 sub_403023 sub_411357 sub_40185 sub_40185 sub_40185 sub_40185 sub_40185 sub_40803 sub_40893 setFilePointer	Imported Normal Normal Normal Normal Normal Normal Imported Normal Normal		2 16 4 17 3 2 5 3 23 8 1		0 0 0 0 0 0 0 0 0 0 0	1 20 5 22 3 1 5 3 38 10	
	1,00 1,00	0,99 0,98 0,98 0,99 0,99 0,99 0,99 0,99	00116180 00116180 00106182 00106592 00103238 00106592 0010587 001115887 00101845 00101845 00101845 00107240 00117240 00117240 0010280 0010883 001017180 0010896	Enter/Infl disection sub_10329 sub_10329 sub_104532 sub_106592 sub_105592 sub_115697 sub_101476 sub	Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal	0041120A8 004114FE 004030C2 0040A694 00405950 00403023 00410165 00401A85 00400ECD2 00410165 00401A85 00408CD2 004121EC 00410A88 0040A883 00412148	Entitic/InfCallsection sub_4030C2 sub_403052 sub_4030550 sub_4030550 sub_4030550 sub_403055 sub_404555 sub_404655 sub_404652 sub_404652 sub_404655 sub_404655 sub_404655 sub_404665 sub_404665	Imported Normal Normal Normal Normal Normal Normal Imported Normal Normal Normal		2 16 4 17 3 2 5 3 23 8 1 1			1 20 5 22 3 1 5 3 38 38	
	1.00 1.00	0,99 0,95 0,98 0,99 0,99 0,99 0,99 0,99 0,99 0,99	00117/0CC 00116180 0010AF32 0010AF32 00106592 0013199 00115EB7 001115EB7 00111A7E 0010CED2 0011724C 0011724C 0011724C 0011724C 00118DB33 00110BE98 00108E95	EnterCriticalisection sub_116180 sub_116180 sub_106732 sub_106592 sub_106592 sub_111845 sub_1107E sub_1017E sub_1017E sub_1017E sub_1017E sub_1017E sub_101676 sub_101676 sub_108665	Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal Normal	0041120A8 004114FE 004030C2 0040A654 00405950 00403023 00411357 00401A85 004001A85 00400C2 004121EC 00410A88 0040AB83 00412148 0040AC96 004083CE	EnterCriticalisection sub_4114F2 sub_4104654 sub_400654 sub_4036950 sub_4036950 sub_411357 sub_40185 sub_40185 sub_40185 sub_40185 sub_40185 sub_40185 sub_40185 sub_40805 sub_40805 sub_408305 sub_408305	Imported Normal Normal Normal Normal Normal Normal Normal Normal Imported Normal Normal		2 16 4 17 3 2 5 3 23 23 8 1 1			1 20 5 22 3 1 5 3 38 10	

We can realize from the above image that on the 820 matched functions, just 15 functions are changed, and 805 functions are identical.

It means that **98.1%** of the Locky ransomware functions are **identical** to the Zepto ransomware.

For that, we can confirm that the Zepto ransomware is just an **extension** of the Locky ransomware **adding it new features**.

Analyzing the changed functions the most notable discover was on the encryption routine function used to encrypt the files because it has the same CFG and changes are made just in adding the final extension of the files:

4	00102450		bl.cc.[obc-4] bl.9				
le l	00102450	nuch	br 33.[ebp-4], br 0	00402200	push	add.	8
	00102434	pusn	ss:[ebp-20]	00402300	pusn	edi	
	00102457	iea	eax, ss:(ebp=308)	00402305	iea	eax, ss:[ebp-308]	
	00102455	push	eax	00400074	and the second sec		
	0010245E	pusn	DI 8	00402364	pusn	eax	
	00102460	pop	ecx				
	00102461	call	UXIUGAFF	00400005		11.0.10	
	00102466	push	DI UX2D	004023E5	push	DI UXIU	
	00102468	pop	esi	004023E7	pop	ecx	
	00102469	lea	ecx, ss: ebp-504				
	0010246F	push	esi				
	00102470	push	ecx				
	00102471	mov	b1 ss:[ebp-4], b1 9	004023E8	mov	bl ss:[ebp=4], bl 4	
	00102475	call	0x102ACB	004023EC	call	0x406257	
	0010247A	mov	ecx, eax				
	0010247C	mov	eax, edi	004023F1	mov	ecx, eax	
L	0010247E	lea	edi, ss:[ebp=280]				
	00102484	mov	b1 ss:[ebp-4], b1 0xA	004023F3	mov	eax, esi	
	00102488	call	0x102B79				
	0010248D	lea	ecx, ss:[ebp-448]	004023F5	lea	edi, ss:[ebp-364]	
	00102493	push	e31				
	00102494	push	ecx				
	00102495	mov	DI SS:[eDp-4], DI UXB				
	00102499	call	UX10ZACB				
	00102495	mov	ecx, eax				
	00102480	mov	eav est (ebp-4), bi oxo	00402358	most	bl est[abr-4] bl 5	
	00102484	loo	eda, ss:[ebp=44]	004023FB	mov	bi sst[ebp-4], bi s	
	001024A7	call	0v102079	00402355	call	0×402803	
	001024B2	lea	ecx, sst[ebp=392]	00402511	CBII	04102405	
	001024B8	push	esi	00402404	push	0x41480C // a locky	
	001024B9	push	ecx	00402409	push	eax	
	001024BA	mov	b1 ss:[ebp-4], b1 0xD				
	001024BE	call	0x102ACB				
	001024C3	mov	ecx, eax				
	001024C5	mov	b1 ss:[ebp-4], b1 0xE				
	001024C9	mov	eax, ss:[ebp-60]				
	001024CC	lea	edi, ss:[ebp-616]				
	001024D2	call	0x102B79				
	001024D7	lea	ecx, ss:[ebp-336]	0040240A	lea	eax, ss:[ebp-224]	
	001024DD	push	esi				
	001024DE	push	ecx				
	001024DF	mov	b1 ss:[ebp-4], b1 0xF	00402410	mov	b1 ss:[ebp-4], b1 6	
	001024E3	call	0x102ACB	00402414	call	0x402A4A	
	001024E8	add	esp, bl 0x44	00402419	add	esp, bl 0x14	
	001024EB	mov	ecx, eax				
	001024ED	mov	bl ss:[ebp-4], bl 0x10				
	001024F1	mov	eax, ss:[ebp-36]				
	001024F4	lea	ed1, SS:[eDp=672]				
	001024FA	call	0x102B/9				
	001024FF	push	OXII9D9C // a_zepto				
	00102504	push	edx				
	0010250B	nov	b1 ss:[ebp-4], b1 0x11				
	00102508	call	0×102800				
	00102514	DOD	ecx				
	00102515	pop	ecx				
	00102516	inc	ebx				
	00102517	push	ebx	0040241C	push	b1 1	
	00102518	xor	edi, edi	0040241E	xor	edi, edi	

The Encryption routine implemented in the Zepto ransomware is similar to the Locky one. On the left and right sides we can realize that the CFG graphs are identical if we look just at branching instructions and calls, it changed just the instructions in the yellow basic blocks. In fact, the report for that function say us exactly that:



CONCLUSION

"If Zepto sticks with this attack vector it may never become a serious threat. However, it's very likely Zepto moves into exploit kits as time goes on," Williams said. "A move by Zepto to malvertising, for example, could get bad very fast," he said.

What we can say is that Zepto Ransomware isn't a new variant of the Locky Ransomware that uses some copycat, because there are too much identical code.

If an av engine tracked the main behaviors of Locky, ransomware as drive enumeration or encryption routine will still spot this threat as a Locky ransomware because, as we saw, this new version of Locky doesn't change the inner logic of the most crucial behaviors.

We can define Zepto ransomware as Locky Ransomware 2.0 and with a lot of probability, the authors of that new variant are the same behind Locky Ransomware.

REFERENCES

- <u>http://securityaffairs.co/wordpress/48725/malware/locky-ransomware-back.html</u>
- <u>http://www.iswatlab.eu/wp-</u>
 <u>content/uploads/2015/09/Technical_Report_Ransomware.pdf</u>
- <u>http://blog.talosintel.com/2016/06/gotta-be-swift-for-this-spam-campaign.html</u>
- <u>https://threatpost.com/locky-variant-zepto-debuts-with-big-spam-push/119017/</u>
- <u>http://resources.infosecinstitute.com/anti-debugging-and-anti-vm-techniques-</u> <u>and-anti-emulation/</u>



Written by the IT Security Expert Antonio Cocomazzi

Antonio Cocomazzi is an IT Security Expert specialized in the malware analysis field. Young and recently graduated, he conducts a 6 months research focused on Ransomware giving a full characterization of the recent families defining a new methodology for dissecting this kind of malware.

Edited by Pierluigi Paganini

(Security Affairs – Locky Ransomware, Zepto ransomware)

CybercrimeLockymalwareransomwareZepto

Share On



You might also like

There you can buy or download for free private and compromising data of your competitors. We public schemes, drawings, technologies, political and military secrets, accounting reports and clients databases. All this things were gathered from the largest worldwide companies, conglomerates and concerns with every activity. We gather data using vunlerability in their IT infrastructure. in their IT infrastructure.

Industrial spy team processes huge massives every day to devide you results. You can fid it in their portal:

http://

(Tor browser required)

We can save your time gaining your own goals or goals of your company.With our information you could refuse partnership with unscrupulous partner, reveal dirty secrets of your competitors and enemies and earn millions dollars using insider information.

"He who owns the information, owns the world"

Nathan Mayer Rothschild

The strange link between Industrial Spy and the Cuba ransomware operation

May 28, 2022 By Pierluigi Paganini

How does GhostTouchAttack work?



GhostTouch: how to remotely control touchscreens with EMI

May 27, 2022 By Pierluigi Paganini

Copyright 2021 Security Affairs by Pierluigi Paganini All Right Reserved. Back to top

- Home
- Cyber Crime
- Cyber warfare
- <u>APT</u>
- Data Breach
- Deep Web
- Digital ID
- Hacking
- Hacktivism
- Intelligence
- Internet of Things

- Laws and regulations
- <u>Malware</u>
- <u>Mobile</u>
- <u>Reports</u>
- <u>Security</u>
- Social Networks
- <u>Terrorism</u>
- ICS-SCADA
- EXTENDED COOKIE POLICY
- <u>Contact me</u>