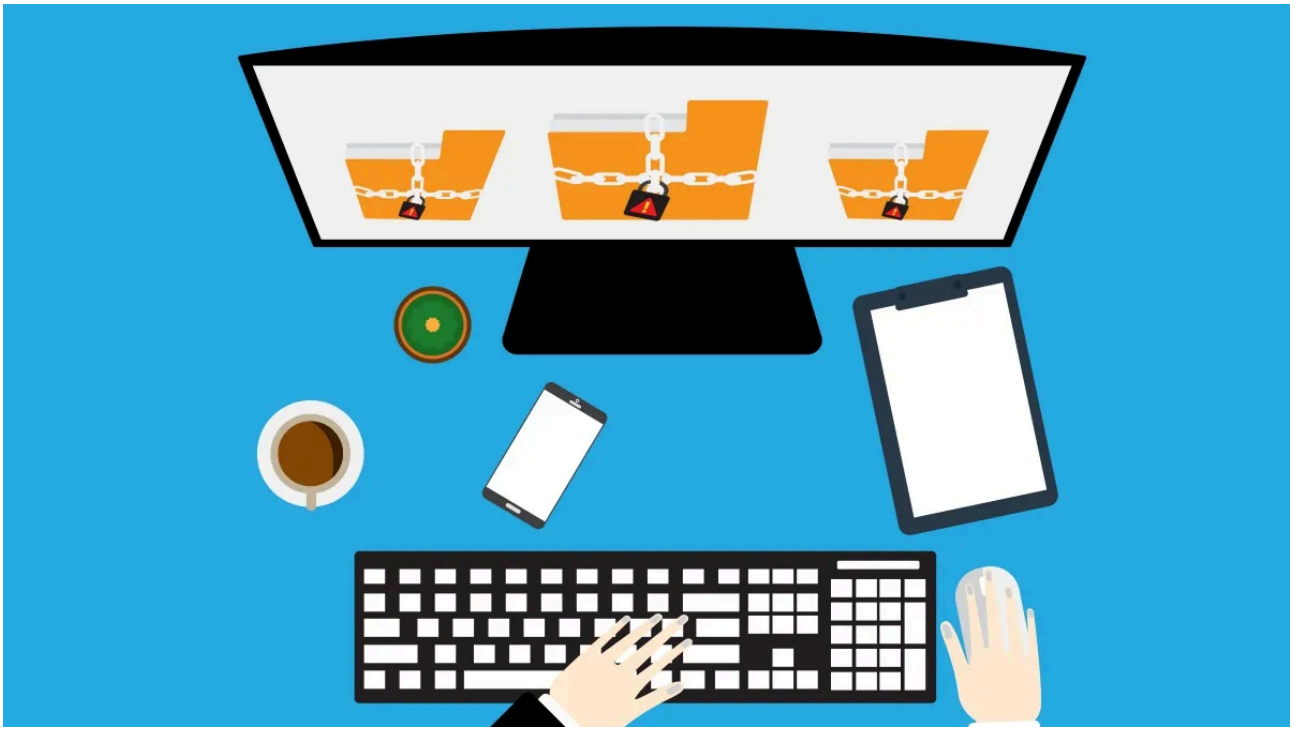
New Locky variant - Zepto Ransomware Appears On The Scene

By Pierluigi Paganini

Published: 2016-07-07 · Archived: 2026-04-05 19:01:24 UTC



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|XXXX].js” where X are some combination of letters (a-f) and numbers (0-9).

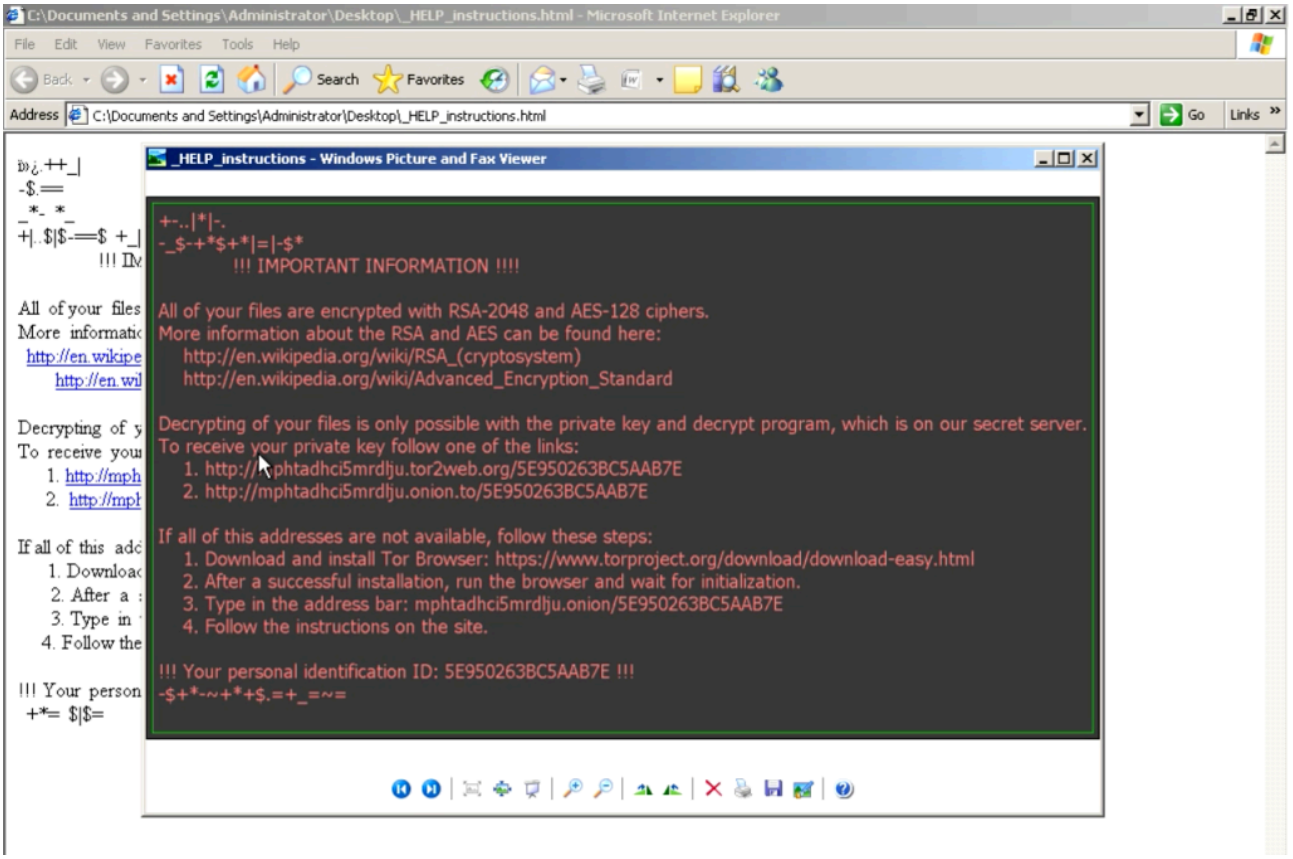
Once the js file is executed through wscript it downloads the main payload binary from the C2 Servers.
Many of them have a list of hardcoded domains for download the binary, other variations use just a few domains.
That is done through HTTP GET requests to define C2 domains and the server functionalities 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:



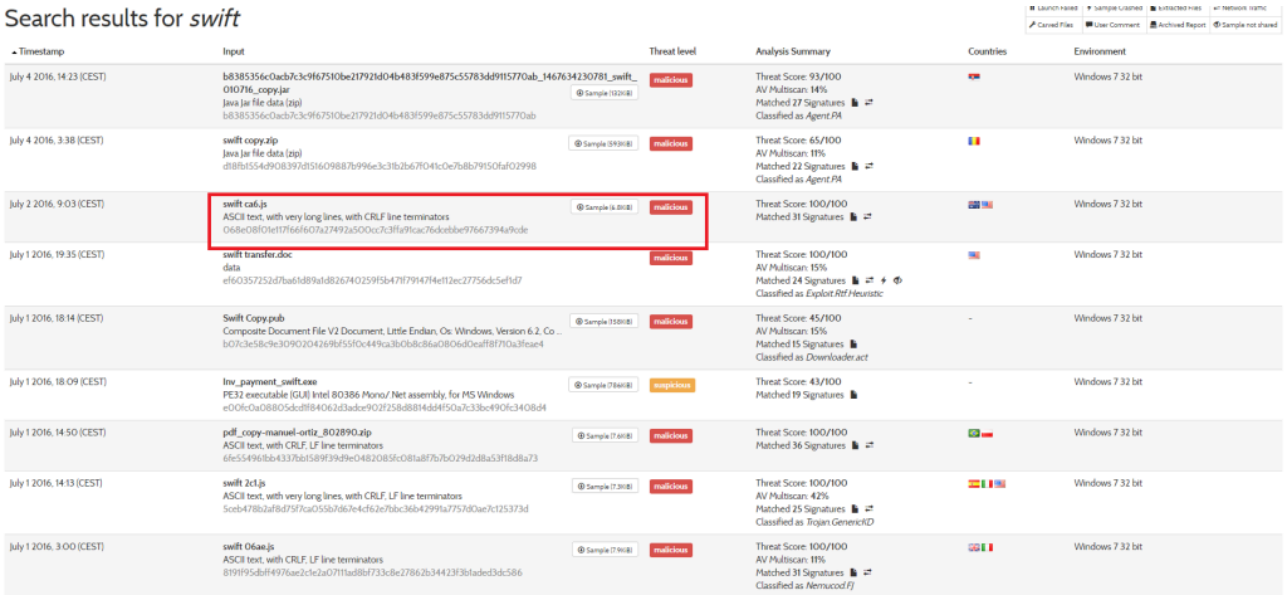
Cisco Talos researchers tracked all the attachment they found and on 137,731 spam emails and discovered

that there were 3,305 unique samples.They collected them here.

OUR ANALYSIS

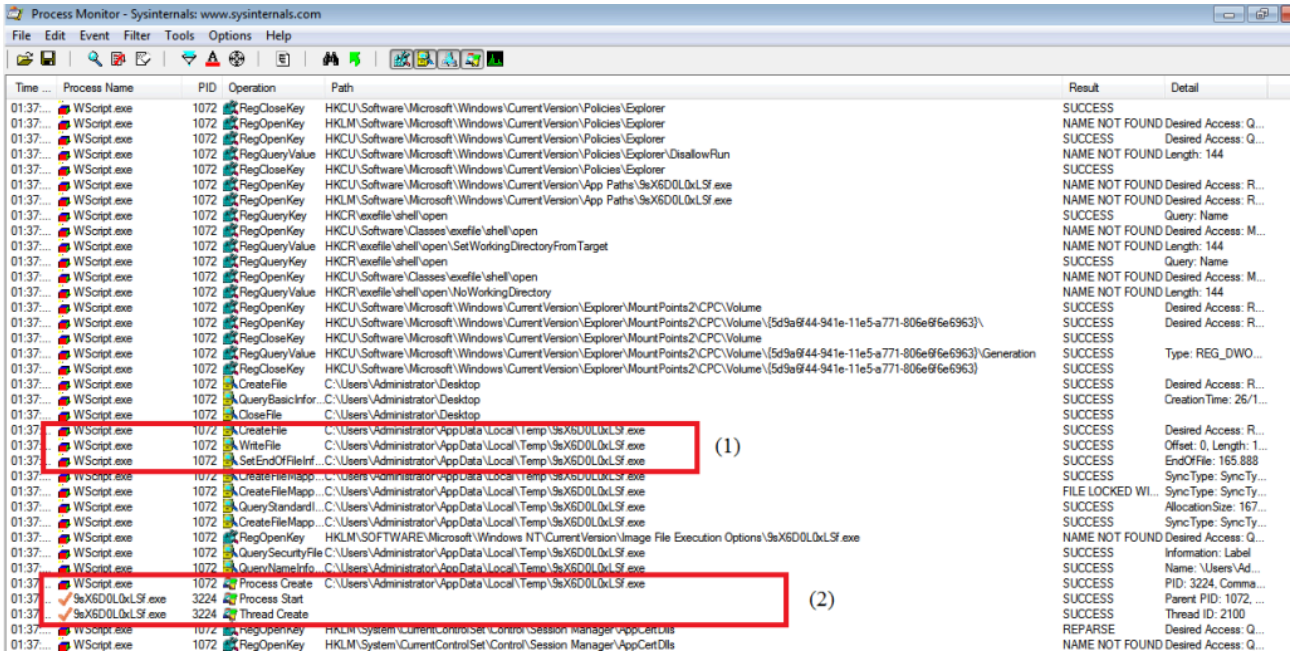
Our main contribution will be to find the actual code that differs from the previous version of the Locky Ransomware. We hope to help in detecting variation on some core features (as encryption routine, files enumeration, drive enumeration…), and to allow experts to distinguish the Locky ransomware family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

We will do this through bindiff 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.

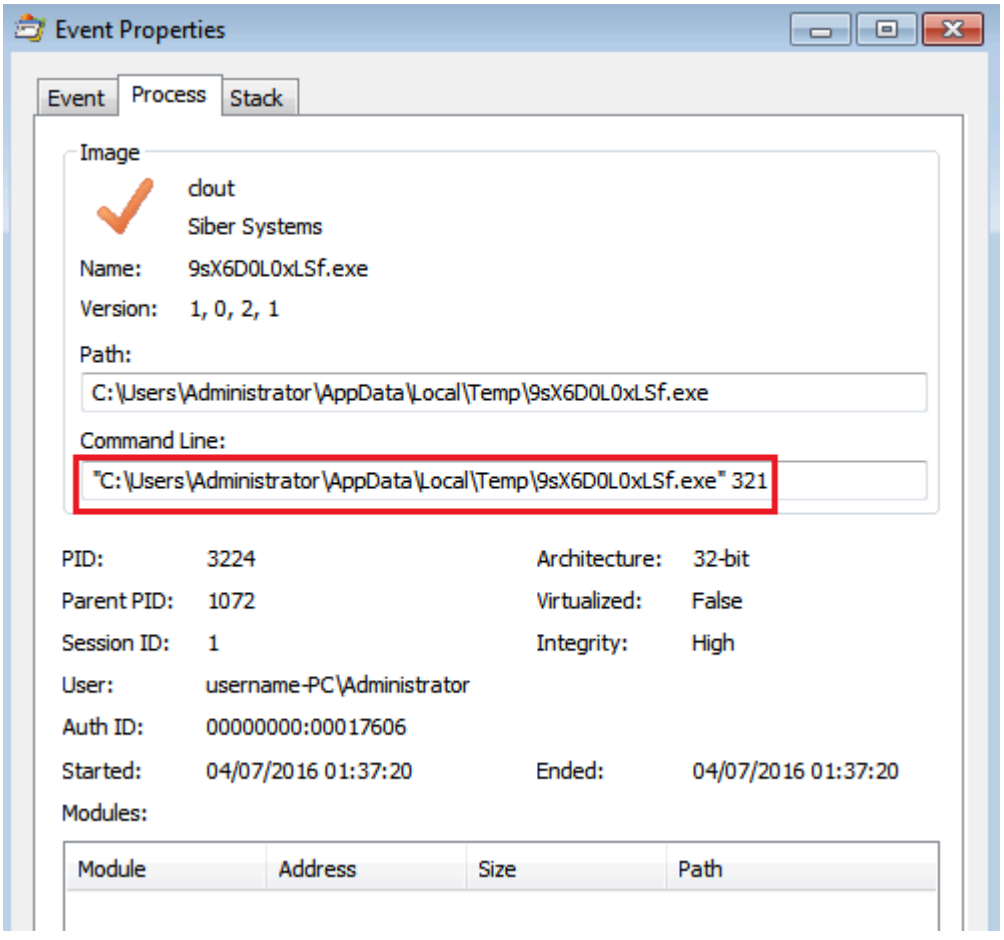


We grabbed the most recent one in order to study the most recent variant.
The file name is "swift ca6.js"
SHA256:068e08f01e117f66f607a27492a500cc7c3ffa91cac76dcebbe97667394a9cde.
As we can see, the file has the same name pattern discovered by Cisco Talos researchers.

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.



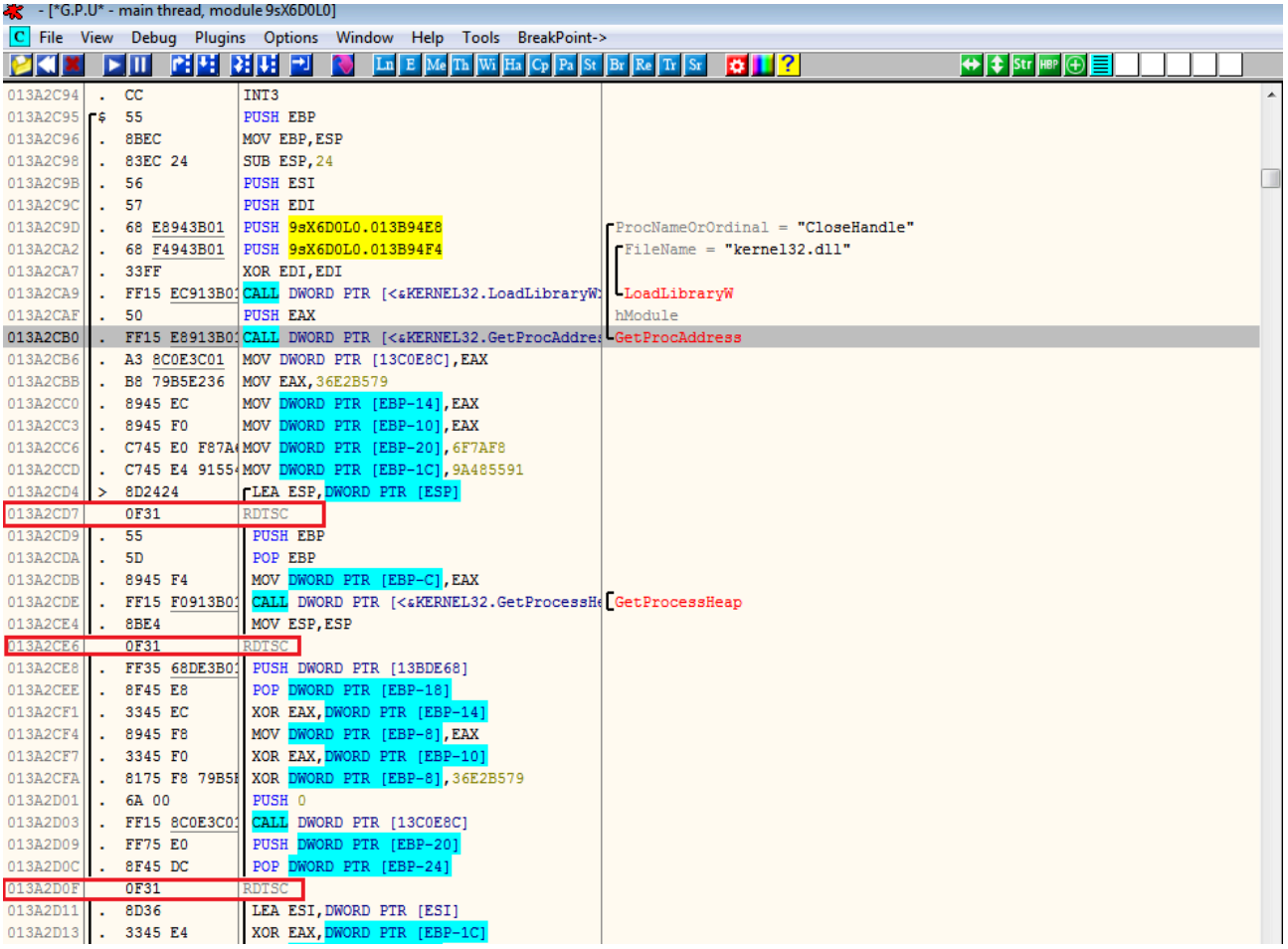
In the above image, we can see in (1) that the script creates the binary file and create a process launching it (2). We found interesting that the js downloader calls the binary with an argument needed for decoding the main payload like the Locky ransomware and most weird is that it uses the same argument for the decryption routine:



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:

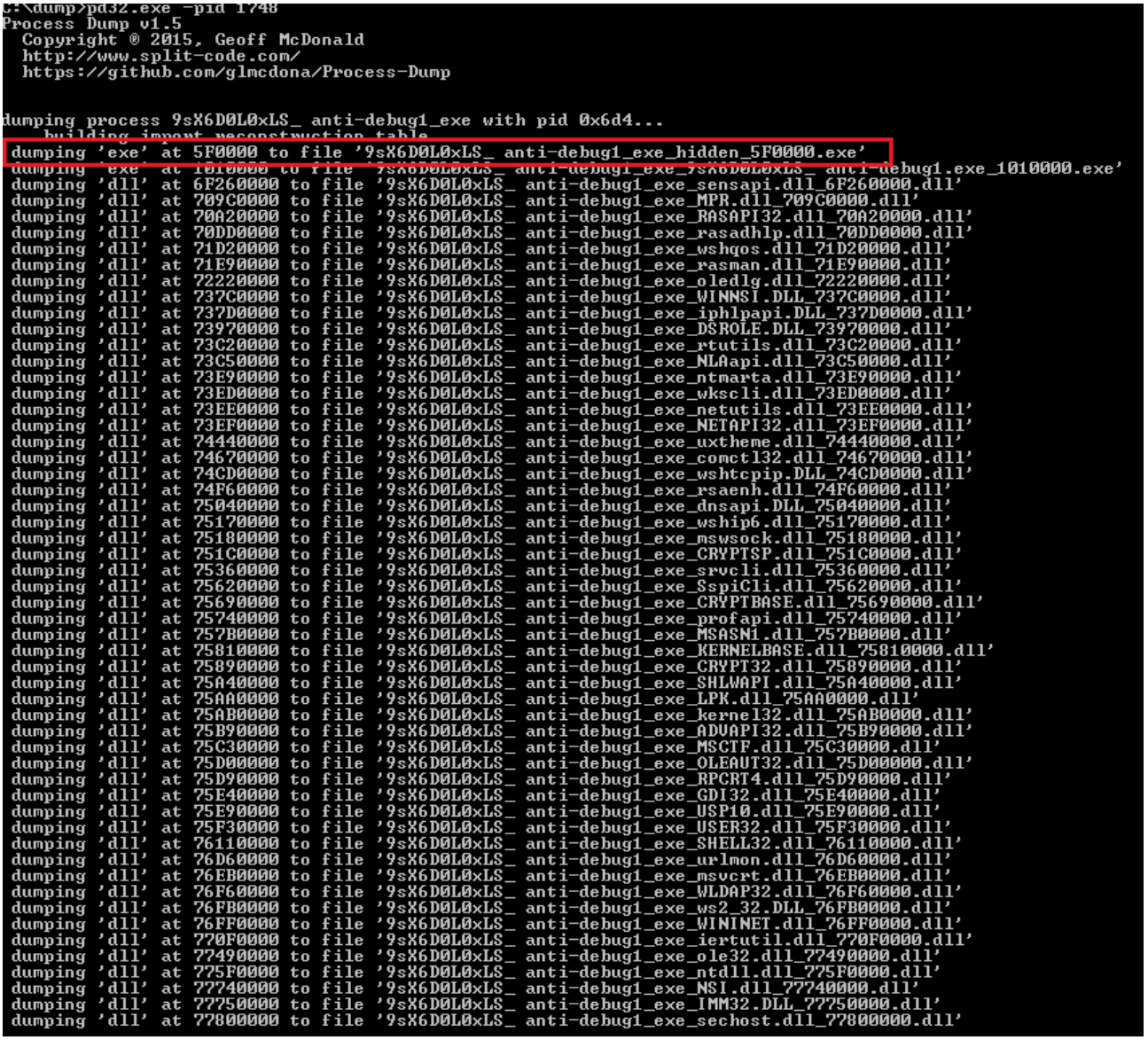


This ransomware uses the RDTSC anti-vm technique:

“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.

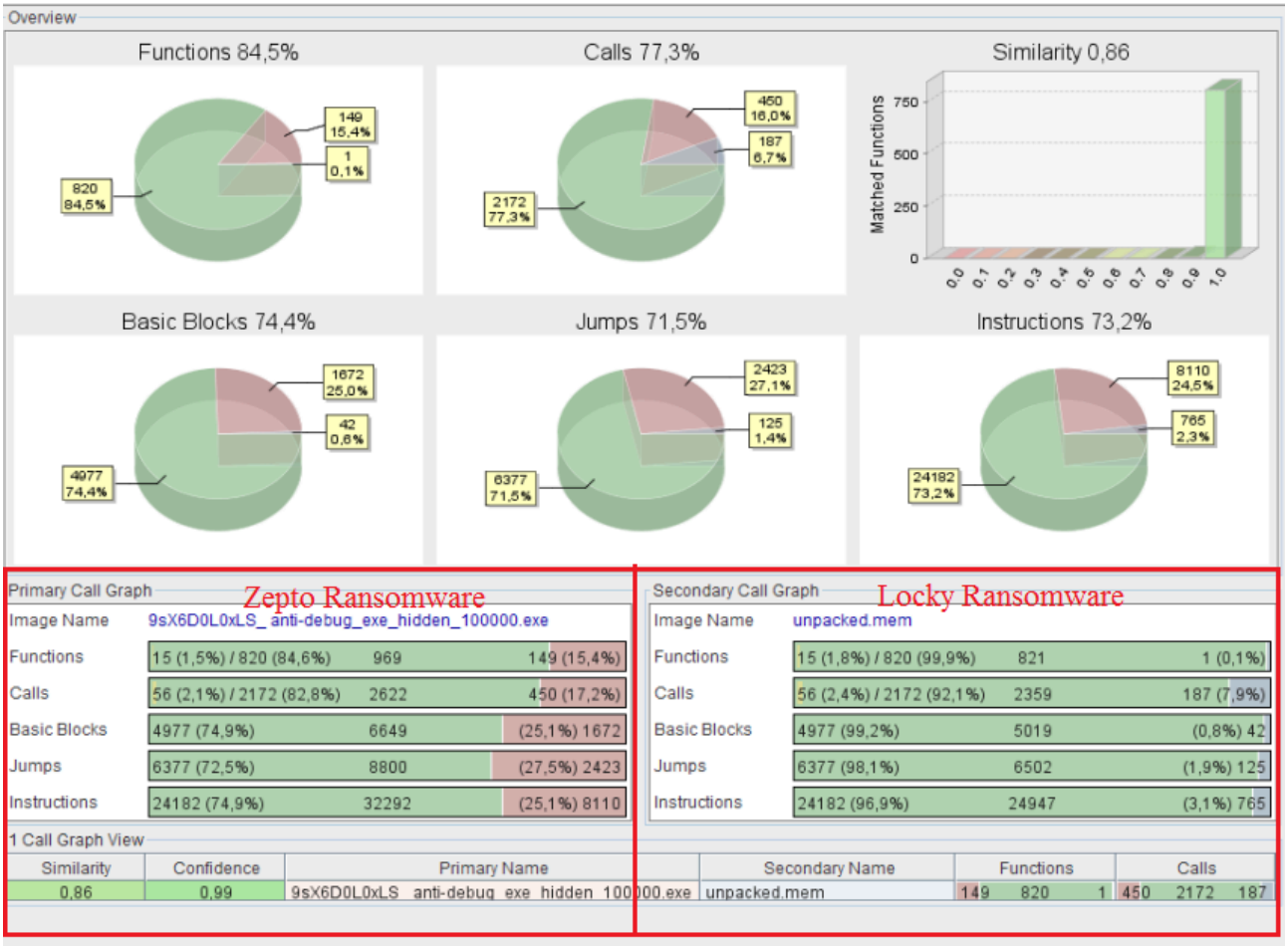
We will use a useful tool for dump a process loaded into the memory: [Process Dump](#).



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.

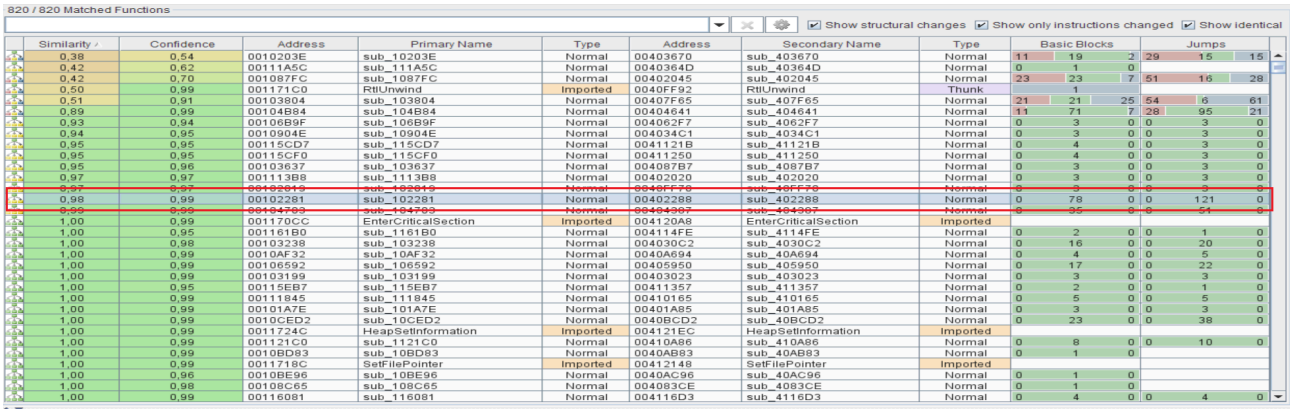
As we can see from the lower part of the image there is a table representing our results, the Zepto ransomware has more functions, calls, basic blocks, jumps and instructions than locky ransomware.
And interesting enough are the results shown in Secondary Call Graph window saying that the **99.9%** on 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:



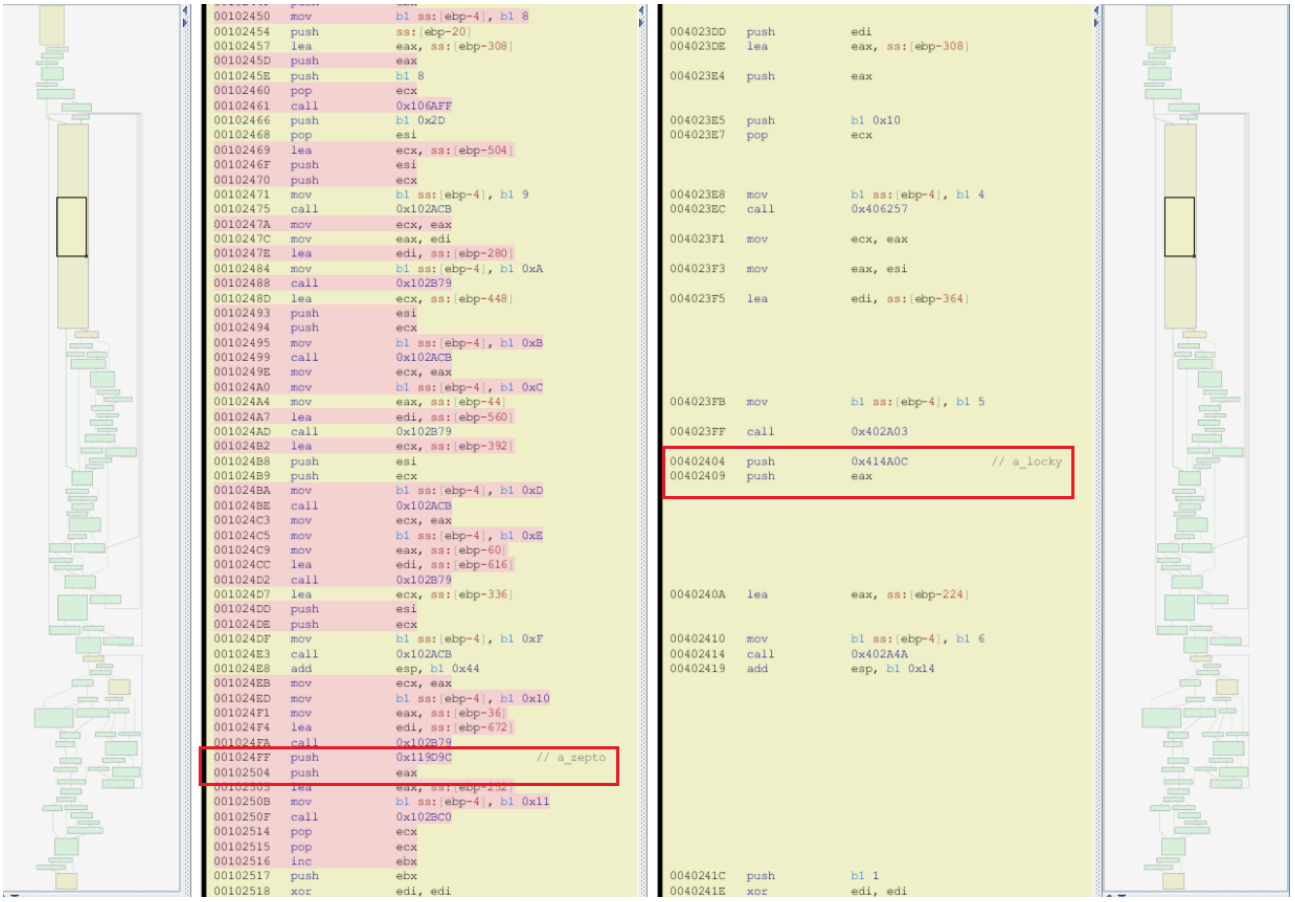
820 / 820 Matched Functions

Show structural changes | Show only instructions changed | Show identical

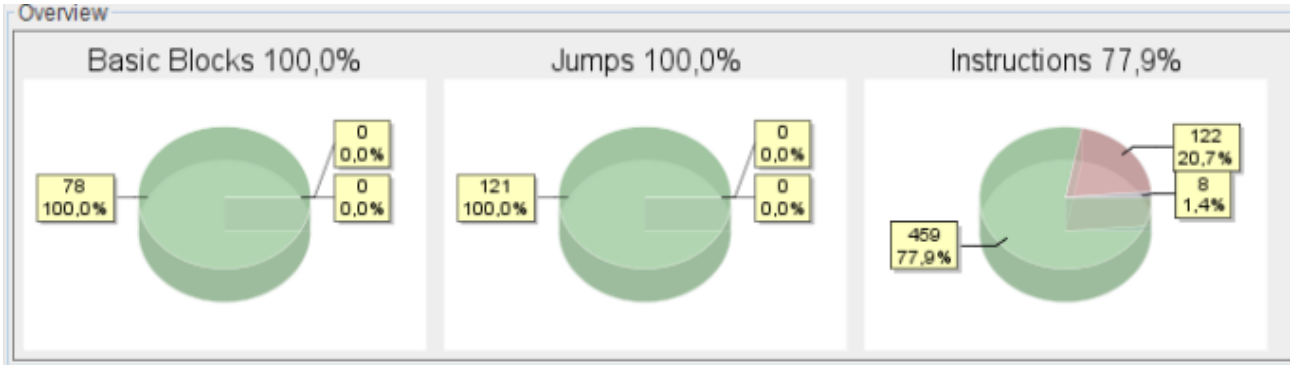
Similarity	Confidence	Address	Primary Name	Type	Address	Secondary Name	Type	Basic Blocks			Jumps		
0,38	0,54	0010203E	sub_10203E	Normal	00403670	sub_403670	Normal	11	19	2	29	15	15
0,42	0,62	00111A5C	sub_111A5C	Normal	0040364D	sub_40364D	Normal	0	1	0			
0,42	0,70	001087FC	sub_1087FC	Normal	00402045	sub_402045	Normal	23	23	7	51	16	28
0,50	0,99	001171C0	RtlUnwind	Imported	0040FF92	RtlUnwind	Thunk		1				
0,51	0,91	00103804	sub_103804	Normal	00407F65	sub_407F65	Normal	21	21	25	54	6	61
0,89	0,99	00104B84	sub_104B84	Normal	00404641	sub_404641	Normal	11	71	7	28	95	21
0,93	0,94	00106B9F	sub_106B9F	Normal	004062F7	sub_4062F7	Normal	0	3	0	0	3	0
0,94	0,95	0010904E	sub_10904E	Normal	004034C1	sub_4034C1	Normal	0	3	0	0	3	0
0,95	0,95	00115CD7	sub_115CD7	Normal	0041121B	sub_41121B	Normal	0	4	0	0	3	0
0,95	0,95	00115CF0	sub_115CF0	Normal	00411250	sub_411250	Normal	0	4	0	0	3	0
0,95	0,96	00103637	sub_103637	Normal	004087B7	sub_4087B7	Normal	0	3	0	0	3	0
0,97	0,97	001113B8	sub_1113B8	Normal	00402020	sub_402020	Normal	0	3	0	0	3	0
~~0,97~~	~~0,97~~	~~88182818~~	~~sub_182818~~	~~Normal~~	~~8048FF78~~	~~sub_48FF78~~	~~Normal~~	~~0~~	~~3~~	~~0~~	~~0~~	~~3~~	~~0~~
0,98	0,99	00102281	sub_102281	Normal	00402288	sub_402288	Normal	0	78	0	0	121	0
~~0,99~~	~~0,99~~	~~88184783~~	~~sub_184783~~	~~Normal~~	~~88404287~~	~~sub_404287~~	~~Normal~~	~~0~~	~~35~~	~~0~~	~~0~~	~~51~~	~~0~~
1,00	0,99	001170CC	EnterCriticalSection	Imported	004120A8	EnterCriticalSection	Imported						
1,00	0,95	001161B0	sub_1161B0	Normal	004114FE	sub_4114FE	Normal	0	2	0	0	1	0
1,00	0,98	00103238	sub_103238	Normal	004030C2	sub_4030C2	Normal	0	16	0	0	20	0
1,00	0,99	0010AF32	sub_10AF32	Normal	0040A694	sub_40A694	Normal	0	4	0	0	5	0
1,00	0,99	00106592	sub_106592	Normal	00405950	sub_405950	Normal	0	17	0	0	22	0
1,00	0,99	00103199	sub_103199	Normal	00403023	sub_403023	Normal	0	3	0	0	3	0
1,00	0,95	00115EB7	sub_115EB7	Normal	00411357	sub_411357	Normal	0	2	0	0	1	0
1,00	0,99	00111845	sub_111845	Normal	00410165	sub_410165	Normal	0	5	0	0	5	0
1,00	0,99	00101A7E	sub_101A7E	Normal	00401A85	sub_401A85	Normal	0	3	0	0	3	0
1,00	0,99	0010CED2	sub_10CED2	Normal	0040BCD2	sub_40BCD2	Normal	0	23	0	0	38	0
1,00	0,99	0011724C	HeapSetInformation	Imported	004121EC	HeapSetInformation	Imported						
1,00	0,99	001121C0	sub_1121C0	Normal	00410A86	sub_410A86	Normal	0	8	0	0	10	0
1,00	0,99	0010BD83	sub_10BD83	Normal	0040AB83	sub_40AB83	Normal	0	1	0			
1,00	0,99	0011718C	SetFilePointer	Imported	00412148	SetFilePointer	Imported						
1,00	0,96	0010BE96	sub_10BE96	Normal	0040AC96	sub_40AC96	Normal	0	1	0			
1,00	0,98	00108C65	sub_108C65	Normal	004083CE	sub_4083CE	Normal	0	1	0			
1,00	0,99	00116081	sub_116081	Normal	004116D3	sub_4116D3	Normal	0	4	0	0	4	0

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:



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

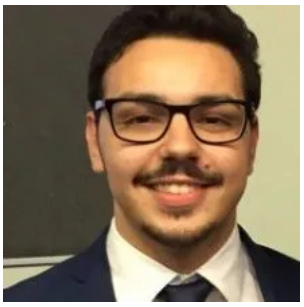
"If Zepto sticks with this attack vector it may never become a serious threat. However, it's very likely Zepto moves into exploit kits as time goes on," Williams said. "A move by Zepto to malvertising, for example, could get bad very fast," he said.

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

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



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.

[adrotate banner=”9″]

Edited by Pierluigi Paganini

(Security Affairs – Locky Ransomware, Zepto ransomware)

Source: http://securityaffairs.co/wordpress/49094/malware/zepto-ransomware.html