# Polyglot – the fake CTB-locker

SL securelist.com/blog/research/76182/polyglot-the-fake-ctb-locker/



Cryptor malware programs currently pose a very real cybersecurity threat to users and companies. Clearly, organizing effective security requires the use of security solutions that incorporate a broad range of technologies capable of preventing a cryptor program from landing on a potential victim's computer or reacting quickly to stop an ongoing data encryption process and roll back any malicious changes. However, what can be done if an infection does occur and important data has been encrypted? (Infection can occur on nodes that, for whatever reason, were not protected by a security solution, or if the solution was disabled by an administrator.) In this case, the victim's only hope is that the attackers made some mistakes when implementing the cryptographic algorithm, or used a weak encryption algorithm.

# A brief description

The cryptor dubbed Polyglot emerged in late August. According to the information available to us, it is distributed in spam emails that contain a link to a malicious RAR archive. The archive contains the cryptor's executable code.

Here are some examples of the links used:

hXXp://bank-info.gq/downloads/reshenie\_suda.rar

hXXp://bank-info.gq/downloads/dogovor.rar

When the infected file is launched, nothing appears to happen. However, the cryptor copies itself under random names to a dozen or so places, writes itself to the autostart folder and to TaskScheduler. When the installation is complete, file encryption starts. The user's files do not appear to change (their names remain the same), but the user is no longer able to open them.

Your documents, scripts, photos, databases and other important files have been encrypted with strongest encryption algorithm AES-256 and unique key, generated for this computer.			
Private decryption key is stored on a secret Internet server and <b>nobody</b> can decrypt your files until you pay and obtain the decryption key.			
You only have 96 hours to submit the payment. If you do not send money within provided time, all your files will be permanently crypted and no one will be able to recovery them.			
Press 'View' to view the list of files that have been encrypted. Press 'Next' to connect to the secret server and follow instructions.			
WARNING! DO NOT TRY TO GET RED OF THE PROGRAMM YOURSELF ANY ACTION TAKEN WILL RESULT IN DECRYPTION KEY BEING DESTROYED. YOU WILL LOSE YOUR FILES FOREVER. ONLY WAY TO KEEP YOUR FILES IS TO FOLLOW THE INSTRUCTION.			
View 95:23:44 Next			

When encryption is complete, the cryptor changes the desktop wallpaper, (interestingly, the wallpaper image is unique to each victim) and displays the ransom message.

The cryptor's main window

	Management Classes and a fill
	Your personal files are encrypted !!!
Your documents this computer.	s, photos, databases and other important files have been encrypted with strongest encryption algorithm AES-256 and unique key, generated for
Private decrypti	on key is stored on a secret internet server and nobody can decrypt your files until you pay and obtain the decryption key.
If you see the m have the last ch	ain locker window follow the instructions on the locker. Otherwise, it's seems that you or you antivirus deleted the locker program. Now you ance to decrypt your files;
Open site http:// to the secret ser	rd7v7mhidgrulwqg.onion.link or http://rd7v7mhidgrulwqg.torlink.co or http://rd7v7mhidgrulwqg.onion.to in your browser. They are public gates rver,
If you have prot 1.Download Tor 2.In the Tor Bro Note that this s 3.Copy and pas DE75E4-B645C BYKYAC-QKYC 013F32-E2611D	Intern strift gates, use direct connection: Browner from https://prorpidc.com/ wer open the (rdf/v/Imidigruikog.onion ever is available uit for Browser only, Retry in 1 hour if site is not reachable. Is the following public key in the input form on server. Avoid misprints. I+11FBF-BSG-SG-BFC28FT/SWM BSB-BYCHAD-TCTBUS-BFSBFC-AIPSUNP O-TDFEC-0004-F1-SAT14324530
These instruction	ons are also saved to the file named ReadMeFilesDecrypt!!!.txt in Documents folder. You can open it and use copy-paste for address and key.

New desktop wallpaper with the "open key" block unique to each victim computer

The user is offered the chance to decrypt several files for free.

The following files are choosen for the free decryption:	
c/program files/echo mirage/License.txt	
c:/program files/far manager/documentation/eng/Bug.Report.txt	
c:/program files/far manager/documentation/eng/Far.FAQ.txt	
c:/program files/far manager/documentation/eng/Plugins.Install.txt	
c:/program files/far manager/documentation/eng/Plugins.Review.txt	
Do not rename or delete files. Only chosen files will be decrypted and nothing else.	
Press 'Decrypt' to decrypt them.	
Press 'Next' to connect to the secret server and decrypt all files.	
Press 'Back' to go the first page.	
Decrypt	
Ded. 05.21.10 Next	
Back 95:21:10 Next	

### The free trial decryption window

After this, the user is told to pay for file decryption in bitcoins. The cryptor contacts its C&C, which is located on the Tor network, for the ransom sum and the bitcoin address where it should be sent.



C&C communication window

From this moment on, the cryptor allows the user to check the ransom payment status on the C&C.

Server accepts payment in Bitcoin (BTC) only.			
1.Pay amount of 0.7 BTC (about of 319.98 USD) to address: 3NCshvW5GwWBa6c2iAyoGDxv72eZU9hskb			
This address within the Bitcoin payment system. It is created individually for you and charge you only the key to it. Do not pay to other addresses.			
2.Transaction will take about 15-30 minutes to confirm.			
If you paid press 'Retry'			
Decryption will start automatically. Do not: power off computer, run antivirus program, disable internet connection. Failures during key recovery and file decryption may lead to accidental damage on files.			
If you have no Bitcoins press 'Exchange'.			
Retry 95:20:10 Exchange			

#### Ransom payment details

If the ransom is not paid on time, the cryptor notifies the user that it's no longer possible to decrypt their files, and that it is about to 'self-delete'.

You did not made payments for your files.
May be you did, but money exchange takes a lot of time?
Locker will self-delete now.
All files will remain encrypted and you lose them forever.
Read instructions in the file ReadMeFilesDecrypt!!!.txt in the Documents folder, it is your LAST CHANCE.
Press 'Exit'
Exit Exit



### **Imitating CTB-Locker**

Initially, this cryptor caught our attention because it mimics all the features of another widespread cryptor – CTB-Locker (Trojan-Ransom.Win32.Onion). The graphical interface window, language switch, the sequence of actions for requesting the encryption key, the payment page, the desktop wallpapers – all of them are very similar to those used by CTB-Locker. The visual design has been copied very closely, while the messages in Polyglot's windows have been copied word for word.

#### The main graphical interface windows:



### List of encrypted files:

Polyglot

CTB-Loc	ker

Dura Dour and test		c'InseizInsei Juhhnara Irorai Juirinsoi rIniisniiks Ity 's'onty' noona I	
For FAO tyt	c/program mes/far manager/documentation/eng	C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\0TQQCIPW\	Document (623).pdf (
Phoins Install txt	c/program files/far manager/documentation/eng	C:\Users\user\AppData\Local\Microsoft\Windows\INetCookies\	12TDU2OK.TXT
Plugins Review.txt	c/program files/far manager/documentation/eng	C:\Users\user\AppData\Local\Microsoft\Windows\INetCookies\	NMD61CFI.TXT
TechInfo.txt	c:/program files/far manager/documentation/eng	C:\Users\user\AppData\Local\Microsoft\Windows\INetCookies\	YXYD7FR.TXT
Bug.Report.txt	c:/program files/far manager/documentation/rus	C:\Users\user\AppData\Local\Microsoft\Windows\Notifications\wpnidm\	179e8db5.JPG
Far.FAQ.txt	c:/program files/far manager/documentation/rus	C:\Users\user\AppData\Local\Microsoft\Windows\Notifications\wpnidm\	5a45b4d3.JPG
Plugins.Install.txt	c:/program files/far manager/documentation/rus	C:\Users\user\AppData\Local\Microsoft\Windows\Notifications\wpnidm\	696010c8.JPG
Plugins.Review.txt	c:/program files/far manager/documentation/rus	C:\Users\user\AppData\Local\Microsoft\Windows\Notifications\wpnidm\	7fe97f77.JPG
l'echInfo.txt	c:/program files/far manager/documentation/rus	C:\Users\user\AppData\Local\Microsoft\Windows\Notifications\wpnidm\	876eabe6.JPG
Brackets.lua	c:/program files/far manager/plugins/brackets	C:\Users\user\AppData\Local\Microsoft\Windows\Notifications\wpnidm\	bc61a82.JPG
Hotkey.lua	c:/program files/far manager/plugins/emenu	C:\Users\user\AppData\Local\Microsoft\Windows\Notifications\wpnidm\	f76d6757.JPG
HotkeyClipboard.lua	c:/program files/far manager/plugins/emenu	C:\Users\user\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#1001	IsExistProfile[1].1S
HotkeyProperties.lua	c:/program files/far manager/plugins/emenu	(MicrosoftEdge\Cache\UU44W/15\	
:atalog.xml	c:/program files/far manager/plugins/farcolorer/base	C:\Users\user\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001 \MicrosoftEdge\Cache\0044W715\	KMC16_0001_CRE_M/
tatalog-console.xml	c/program files/far manager/plugins/farcolorer/base/hrd	C:\Users\user\AppData\Uoca\Packages\Microsoft_MicrosoftEdge_8wekyb3d8bbwe\AC\#1001	
catalog-rgb.xml	c/program files/far manager/plugins/farcolorer/base/hrd	\MicrosoftEdge\Cache\0044W/715\	SC-Magazine-banner-1
:atalog-text.xml	c:/program files/far manager/plugins/farcolorer/base/hrd	C:\Users\user\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001	WR-latest[1].JS
plack ore	c/program files/far manager/plugins/farcolorer/base/hrd/css	[miclosoftcuge(cache(bottword))	

### Window for the trial decryption of 5 random files:

Polyglot

CTB-Locker

Test decryption.	Test decryption.		
The following files are choosen for the free decryption: c:program files/echo mirage/License txt	The following files are chosen for the free decryption:		
c:/program files/far manager/documentation/eng/Bug Report txt c:/program files/far manager/documentation/eng/Far FAQ btt c:/program files/far manager/documentation/eng/Plagins Install txt c:/program files/far manager/documentation/eng/Plagins Review txt	C:\ffiles\img13.JPG C:\Users\user\AppData\Loca\\Microsoft\Internet Explorer\brndlog.IXT C:\Users\user\AppData\Loca\\Microsoft\OneDrive\17.3.6517.0809\fabric.mdl2.JS C:\Users\user\AppData\Loca\\Microsoft\OneDrive\17.3.6517.0809\ThirdPartyNotices.TXT		
Do not rename or delete files. Only chosen files will be decrypted and nothing else.	C:\Users\user\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb\wrVBD9YP87.TXT		
Press 'Decrypt' to decrypt them. Press 'Next' to connect to the secret server and decrypt all files.	Do not rename or delete files. Only chosen files will be decrypted and nothing else.		
Press 'Back' to go the first page.	Press 'Decrypt' to decrypt them. Press 'Next' to connect to the secret server and decrypt all files. Press 'Back' to go to the first page.		
Decrypt	Decrypt		
Back 95:21:10 Next	<pre>&lt;<back 95:55:27="" next="">&gt;</back></pre>		

The private key request window:



#### The desktop wallpapers:

Polyglot

CTB-Locker



#### The 'connection failed' error message:

Polyglot

**CTB-Locker** 



### Offline decryption instructions:

## Polyglot

#### CTB-Locker

al 🗐 📼 🖬 🗖 🗖	
Online occi ypuon.	Offline decryption.
Find another computer with active to the Internet and perform the following steps on it:	Find another computer with actve connection to the Internet and perform the following steps on it:
<ol> <li>Open site <u>http://rd7v?mhidgmalwag.onion.lnk</u> or <u>http://rd7v?mhidgmalwag.torfink.co</u> or <u>http://rd7v?mhidgmalwag.onion.to</u> in your browser. They are public gates to the secret server. Go step 3</li> </ol>	Open <a href="http://pf5dahldauhrjxfd.onion.cab">http://pf5dahldauhrjxfd.tor2web.org</a> in your browser. They are public gates to the secret server. If you have problems with aates, use direct connection:
If you have problems with gates, use direct connection: 1. Download Tor Browser from <u>http://torproject.org/</u> 2.1. ist. Tore Downser to a tor gail ist and have provided and the second secon	1. Download Tor Browser from <u>http://torproject.org</u>
2. In the 1 or provset open the tory/managurung onion Note that this server is available via Tor Browser only. Retry in 1 hour if site is not reachable. 3. Cover (Child) and naste the following multi-low in the inert form on server. Avoid missering.	<ol> <li>In the Tor Browser open the <a href="http://pf5dahldauhrixfd.onion/">http://pf5dahldauhrixfd.onion/</a> Note that this server is available via Tor Browser only.</li> </ol>
DE754-B654C41F45B00-B574-B656C45F45B656C4F723E7XSM BXTXAC-QXYCES-BXXAD-TCTBMS-EF35FE-MESMNP 013F32-E2611D-D17BEC-000941-F3A11A-324530	Copy and paste the following public key in the input form on server. Avoid missprints. BX5JDXW-VAE2SED-EW2BBVS-JCRB24K-S4OH5AJ-75PBENO-5E2DQXX-HSESESH IAY3QGK-YMQ5J03-LCIE7RN-TEAKY5V-T47ASVM-HRIFSXQ-PKBQVI3-IELDXKE
4.Follow the instructions on the server.	5LGJHHE-YPZSZXB-EDR62SC-HDCV6PA-OL4Q7CZ-ICPWIZU-UFMZYRA-GCA3YKU
	Follow the instructions on the server.
Back 95:16:44	<pre><sback 93:58:21<="" pre=""></sback></pre>

The similarities do not stop there. Even the encryption algorithms used by the cybercriminals have clearly been chosen to imitate those used in CTB-Locker.

	Polyglot	CTB-Locker
Algorithms used for file encryption	File content is packed into a ZIP archive and then encrypted with AES-256.	File content is compressed with Zlib and then encrypted with AES- 256.
Algorithms used while working with the keys	ECDH (elliptic curve Diffie-Hellman), curve25519, SHA256.	ECDH (elliptic curve Diffie-Hellman), curve25519, SHA256.
Extensions of encrypted files	File extensions are not changed.	File extensions are changed, depending on version: – .ctbl – .ctb2 – 7 random lower-case Latin symbols
Demo decryption	5 files are decrypted for free as a demo. Their decryption keys and file names are saved in the registry.	5 files are decrypted for free as a demo. Their decryption keys are only stored in the RAM memory while the process is running.
C&C location	C&C is in the Tor network, communication is via a public tor2web service.	C&C is in the Tor network, communication is via a Tor client integrated into the Trojan, or (in some versions of CTB-Locker) via a public tor2web service.
Traffic protection / obfuscation	Bitwise NOT operation.	AES encryption.

That said, we should note the following: a detailed analysis has revealed that Polyglot was developed independently from CTB-Locker; in other words, no shared code has been detected in the two Trojans (except the publicly available DLL code). Perhaps the creators of Polyglot wanted to disorient the victims and researchers, and created a near carbon copy of CTB-Locker from scratch to make it look like a CTB-Locker attack and that there was no hope of getting files decrypted for free.

### **C&C** communication

The Trojan contacts the C&C server located on Tor via a public tor2web service, using the HTTP protocol.

Prior to each of the below data requests, a POST request is sent with the just one parameter: "live=1".

### Request 1.

At the start of operation, the Trojan reports the successful infection to the C&C. The following data is sent to the C&C:

```
{
"ip":"xxx.xxx.xxx.xxx",
                          //ip address of the infected computer
"method":"register",
                       //action type. "register" = Trojan informs C&C of new infection
                                                 //Infected computer's ID
"uid":"xxxxxxxx-xxxx-xxxx-xxxx,
"version":"10f",
                 //Trojan version contained in its body
"info":"Microsoft (build xxxx), 64-bit",
                                        //OS version on the infected computer
"description":" ", //Always a whitespace (" ")
"start_time":"14740xxxxx",
                              //Trojan's start time
"end_time":"0", //Encryption finish time. 0 = no encryption has run yet
"user_id":"5"
                //Number hardwired in the sample
}
```

This data block is passed through a bitwise NOT operation, encoded into Base64 and sent to the C&C in a POST request.



Contents of the sent request

Parameters of the POST request:

signature – CRC32 from the sent data ver – Trojan version gcdata – data, with contents as described above.

```
Wireshark · Follow TCP Stream (tcp.stream eq 12) · network
                                                                               _ 🗆 ×
POST / HTTP/1.1
Host: rd7v7mhidgrulwqg.onion.link
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Content-Length: 373
Content-Type: application/x-www-form-urlencoded
signature=242242339&ver=10f&gcdata=hN2Wi93F3cbK0c3K0cvRzszG3dPdkpgL15Cb3cXdjZ
qYloyLmo3d092K1pvdxd3HmZzPncyamtLLns3L0submcnSx8%2bZxtLGyZ7Hx8nPmZzPzc
%2fd092Jmo2MlpCR3cXdzs
%2bZ3dPdlpGZkN3F3bKWnI2QjJCZi9%2bolpGbkIiM38jf36yajYmWnJrfr56clN
%2f039edipaTm9%2fIyc
%2f01tPfycvSnZaL3dPdm5qMnI2Wj4uWkJHdxd3f3dPdjIuejYugi5aSmt3F3c7LycrGyMjOycnd0
92akZugi5aSmt3F3c%2fd092KjJqNoJab3cXdyt2C
HTTP/1.1 200 OK
X-Check-Tor: false
Date: Thu, 15 Sep 2016 17:57:38 GMT
Content-Type: text/html; charset=utf-8
X-Onion-Url: rd7v7mhidgrulwqg.onion
Age: 0
X-Cache: MISS
Transfer-Encoding: chunked
Connection: keep-alive
Accept-Ranges: bytes
0044
 {"signature":"2327609319","data":"hN2LmoeL3cXdsLTd092ckJua3cXdz92C"}
Ø
1 client pkt(s), 2 server pkt(s), 1 turn(s).
                                   •
Entire conversation (895 bytes)
                                          Show data as ASCII
                                                              -
                                                                          Stream 12 🗧
Find:
                                                                             Find Next
                          Hide this stream
                                           Print
                                                                  Close
                                                     Save as..
                                                                              Help
```

Request 1 and the reply received from the C&C

# Request 2.

When the Trojan has finished encrypting the user's data, it sends another request to the C&C. The content of the request is identical to that of request 1 except the field "end\_time", which now shows the time encryption was completed.

# Request 3.

This is sent to the C&C to request the bitcoin address for payment and the ransom sum to be paid.

```
{
"method":"getbtcpay"
"uid":"xxxxxxxx-xxxx-xxxx-xxxx-xxxxx"
}
```

The C&C replies to this request with the following data:

```
{
    "code":"0",
    "text":"OK",
    "address":"xxxxxxx", //bitcoin address (may vary)
    "btc":0.7, //amount to be paid in BTC (may vary)
    "usd":319.98 //amount to be paid in USD (may vary)
}
```

# Request 4.

This is sent to request a file decryption key from the C&C.

#### Request 5.

The Trojan reports that data decryption has been completed and states the number of decrypted files to the C&C.

### Description of the encryption algorithm

During our analysis of the malicious code, it became evident that the Trojan encrypts files in three stages, creating intermediate files:

- First, the original file is placed in a password-protected ZIP archive. The archive has the same name as the original file plus the extension "a19";
- Polyglot encrypts the password-protected archive with the AES-256-ECB algorithm. The resulting file again uses the name of the original file, but the extension is now changed to "ap19";
- The Trojan deletes the original file and the file with the extension "a19". The extension of the resulting file is changed from "ap19" to that of the original file.



Flowchart of the search and file encryption actions performed by Polyglot

A separate AES key is generated for each file, and is nothing more than a 'shared secret' generated according to the Diffie-Hellman protocol on an elliptic curve. However, first things first.

Before encrypting any files, the Trojan generates two random sequences, each 32 bytes long. The SHA256 digests of each sequence become the private keys **s\_ec\_priv\_1** and **s\_ec\_priv\_2**. Then, the Bernstein elliptic curve (Curve25519) is used to obtain public keys **s\_ec\_pub\_1** and **s\_ec\_pub\_2** (respectively) from each private key.

The Trojan creates the structure **decryption\_info** and writes the following to it: a random sequence used as the basis for creating the key **s\_ec\_priv\_1**, the string **machine\_guid** taken from the registry, and a few zero bytes.

```
struct decryption_info
{
     char s_rand_str_1[32];
     char machine_guid[36];
     char zeroes[12];
};
```

Using the private key **s\_ec\_priv\_2** and the cybercriminal's public key **mal\_pub\_key** produces the shared secret **mal\_shared\_secret** = ECDH(*s\_ec\_priv\_2*, *mal\_pub\_key*). The structure *decryption\_info* is encrypted with algorithm *AES-256-ECB* using a key that is the *SHA256* digest of this secret. For convenience, we shall call the obtained 80 bytes of the encrypted structure **encrypted\_info**.

Only when Polyglot obtains the *encrypted\_info* value does it proceed to generate the session key AES for the file. Using the above method, a new pair of keys is generated, **f\_priv\_key** and **f\_pub\_key**. Using **f\_priv\_key** and **s\_ec\_pub\_1** produces the shared secret **f\_shared\_secret** = ECDH(*f\_priv\_key*, *s\_ec\_pub\_1*).

The SHA256 digest of this secret will be the AES key with which the file is encrypted.

To specify that the file has already been encrypted and that it's possible to decrypt the file, the cybercriminals write the structure **file\_info** to the start of each encrypted file:

The elliptic curve, the Diffie-Hellman protocol, AES-256, a password-protected archive – it was almost flawless. But not quite, because the creator of Polyglot made a few mistakes during implementation. This gave us the opportunity to help the victims and restore files that had been encrypted by Polyglot.

### Mistakes made by the creators

As was mentioned earlier, all the created keys are based on a randomly generated array of characters. Therefore, the strength of the keys is determined by the generator's strength. And we were surprised to see the implementation of this generator:



A graphical representation of the random sequence generation procedure

Let's convert this function into pseudocode so it's easier to follow:



Please note that when another random byte is selected, the entire result of the function *rand()* is not used, just the remainder of dividing the result by 32. Only the cybercriminal knows why they decided to make the random string this much weaker – an exhaustive search of the entire set of the possible keys produced by such a pseudo-random number generator will only take a few minutes on a standard PC.

Taking advantage of this mistake, we were able to calculate the AES key for an encrypted file. Although there was a password-protected archive below the layer of symmetric encryption, we already knew that the cybercriminal had made another mistake.

Let's look at how the archive key is generated:

zip\_key[0] = machine\_guid[5]; zip\_key[1] = machine\_guid[15]; zip\_key[2] = machine\_guid[12]; zip\_key[3] = machine\_guid[20]; zip\_key[4] = 0;

We can see that the key length is only 4 bytes; moreover, these are specific bytes from the string MachineGuid, the unique ID assigned to the computer by the operating system. Furthermore, a slightly modified MachineGuid string is displayed in the requirements text displayed to the victim; this means that if we know the positions in which the 4 characters of the ZIP archive password are located, we can easily unpack the archive.

ReadMeFilesDecrypt!!!.txt — Блокнот		Freedo	1.0	
Вайо Прики Борилт Вай Сорико важ дайты заинфореани !!! Важ долучеть дотографии, бази данных и дру прикатышк клю хранится на секретном сервере Скли вы нарист гланно секо покора, следуйте отхройте сайт http://rd7v8hidgrulugg.onfon. Ссли у вак есть проблемы с доступом, искользу :	угие важные файлы были завифрованы силл в интернете и инсто ине сюжет расширо- инструкция на локере. В другои случая Unk or http://rdv/mbidguulug.colina yhte gmaxoe подключение: infon w только черев тог Browser, Boetopure e fon on a форм вода на серере, избегалт Maximum строково параметра Перечен: Incerest Machemod Incerest Colina (Colina (Colina Colina (Colina (Colina Colina (Colina Colina (Colina (Colina Colina Colina (Colina Colina (	нейшим шифрованием с уникальным кля вать ваши файли пока вы не оплатите , кажется, что вава антивирусная по co or http://rd7v7mhidgrulwigg.onfo onwrry weges 1 час, если сайт не до onwrry weges 1 час, если сайт не до onwrry weges 1 час, если сайт не до page of the second second second second second second onworks.	чом, сгенерированным д и не получите привати ограмма или ви удаллии г. со в вашен бриузере. ступен.	ля данного кс ый ключ. покера. Телк Они является

The MachineGuid string displayed in the requirements screen

#### Conclusion

Files that are encrypted by this cryptor can be decrypted using Kaspersky Lab's free anti-cryptor utility RannohDecryptor Version 1.9.3.0.

All Kaspersky Lab solutions detect this cryptor malware as: Trojan-Ransom.Win32.Polyglot PDM:Trojan.Win32.Generic

MD5

c8799816d792e0c35f2649fa565e4ecb - Trojan-Ransom.Win32.Polyglot.a