Check Point
SOFTWARE TECHNOLOGIES LTD.

WE SECURE THE FUTURE

The Future of Ransomware – ZeroNet Protocol

Amit Dori
Threat Intelligence and Research
March 2016

Introduction

Ransomware has become a formidable threat in today's cybersecurity world. Security vendors constantly try to stay one step ahead of the attackers and disrupt their operations. The Check Point Research Team recently found a new avenue which could potentially be used for the worst by ransomware developers.

ZeroNet is a new peer-to-peer (P2P) network whose aim is to protect private users from censorship and has the potential to revolutionize the way we use the World Wide Web. Any ZeroNet site remains available as long as one peer is still seeding it. While this is a productive new technology, we believe it will not be long before it is misused by malicious actors.

Ransomware modus operandi

Ransomware targets both business and private users, and encrypts files on devices and networks. There are many different types of ransomware, which use different methods to achieve their malicious intent.

A typical ransomware operation:

1. A machine is infected with ransomware.
2. The ransomware generates an AES encryption key and a RSA public-private key-set.
3. The ransomware establishes a connection to the C&C server informs it of the infection and stores the private RSA key within the server.
4. The ransomware scans the victim's machine to locate files and encrypt them using the AES key.
5. The ransomware encrypts the AES key with the public RSA key and notifies the victim that his files have been encrypted.

The notification usually includes a quick explanation about ransomware and what the user must do if he wants to retrieve his files:

1. Open a Bitcoin account and load it with the specified amount of Bitcoins.
2. Pay the Bitcoins to a hardcoded Bitcoin address.
3. Post proof of payment on a payment-gateway (usually a compromised site / a domain purchased by the attackers).

After receiving payment, the attacker "pushes" the decryption software to the victim. Attackers usually maintain a good "customer service" i.e. follow through on the decryption process, as failure to do so impacts payment by future victims.

The Ransomware Achilles heel – payment-gateways

There are several ways to track and identify the perpetrators behind ransomware attacks:

- The malware's code and data often contains relevant information about the attackers.
- The payment, which is usually in the form of Bitcoins (BTC).
- The payment-gateways that act as a communication channel between the victim and the attacker.

A skilled reverse-engineer can easily extract the data and analyze the operation of most ransomware. However, most ransomware use a DGA (Domain Generation Algorithm) to contact their C&C, which makes it very hard for security vendors to blacklist current and future C&Cs. Unless the malware writer made a grave mistake while developing the ransomware, analyzing the code will not help to catch him.

Following the money would be a smart move; just like in any other heist, the attackers eventually receive the payment. In the past, it was possible to track the various payment methods, meaning the attackers had to risk exposure in various ways.
Currently, ransomware developers have adopted Bitcoin as their preferred currency. It is easy to follow all transactions made within the Bitcoin network, but it's nearly impossible to follow the money trail – there's just too much data to handle.
Each legitimate transaction is followed by a massive amount of transactions designed to mislead and disrupt any search. The attackers usually use a Bitcoin Mixing Service to make sure no one can follow the money.

We are left with the payment-gateways as the only practical way to track and deal with the attackers. To create a payment-gateway, attackers simply hack or purchase a domain and upload their .php files. When the ransomware infects a device, it provides the victims with links to the payment-gateway.

It has become clear to security vendors that the easiest and simplest way to actively combat the ransomware threat is to attack the attackers' vulnerable spot: the payment-gateway.
Payment-gateways are shut down on a daily basis. Shutting down a single payment-gateway may seem to have little impact, but shutting down lots of them over time affects even the

massive ransomware campaigns, forcing the attackers to redo the whole process of setting up payment-gateways.

Introducing ZeroNet

Attackers constantly search for new technologies and methods to continue their operation in a more effective way. Interfere with their ability to host payment-gateways and receive ransom, and they will naturally look for other ways.
One such way is ZeroNet, a distributed P2P network.

What is ZeroNet?[1]

ZeroNet uses Bitcoin cryptography and BitTorrent technology to build a decentralized censorship-resistant network.
Users can incorporate their static or dynamic websites into ZeroNet, and site visitors can choose to also serve i.e. seed data to the website. Websites will remain online even if they are being served by only one peer.
When a site is updated by its owner, all nodes serving that site (previous visitors) receive only the incremental updates done to the site content.

How does it work?

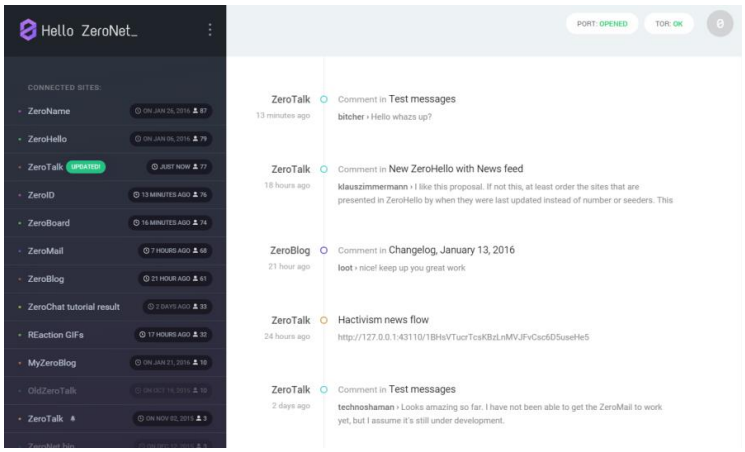
Each site is represented by a public Bitcoin address. A site can be opened by visiting: http://127.0.0.1:43110/{zeronet_site_address}
(e.g. http://127.0.0.1:43110/1EU1tbG9oC1A8jz2ouVwGZyQ5asrNsE4Vr).
ZeroNet then uses the BitTorrent network to find peers that are seeding the site and will download the site content (HTML, CSS, JS...) from these peers. Each visited site is also served by the user who visited it.
Every site contains a list in a Sha512 hash of all the files used, and a signature is generated using the site owner's private key.

"If the site owner modifies the site, then he/she signs a new list and publishes it to the peers. After the peers have verified the files list integrity (using the signature), they download the modified files and publish the new content to other peers.

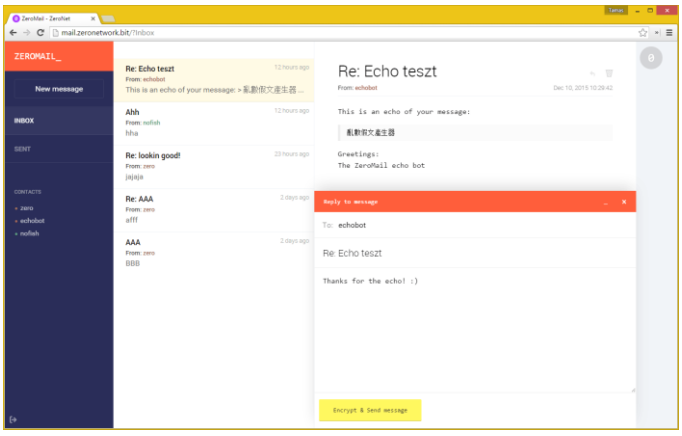
[1] http://zeronet.readthedocs.org/en/latest/

In order to become a site owner, all you need to do is generate a Bitcoin key set (public-private) and use the private key to sign files and publish updates." ***The ZeroNet Official Website.***

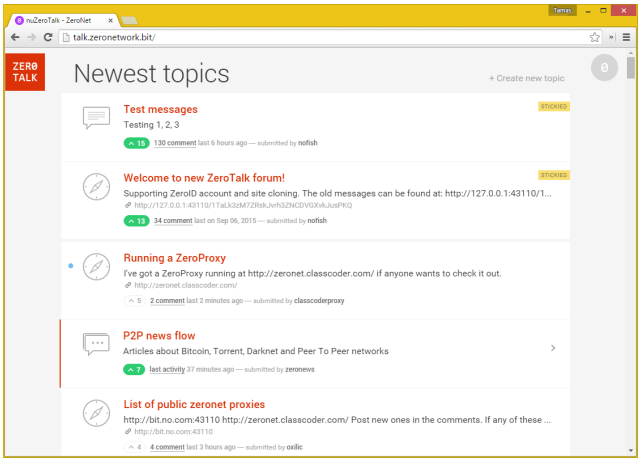
As the site's creator is the only one in possession of the private key, s/he is the only one who can perform changes to the site. In addition, the site owner can stay anonymous by using the Tor network. Because the site has a Bitcoin address, it can be used for Bitcoin payments.



The homepage of ZeroNet Lists all added sites Enables the user to: Update, Pause, Resume, Delete a site



ZeroMail – a tool within ZeroNet to send P2P encrypted messages



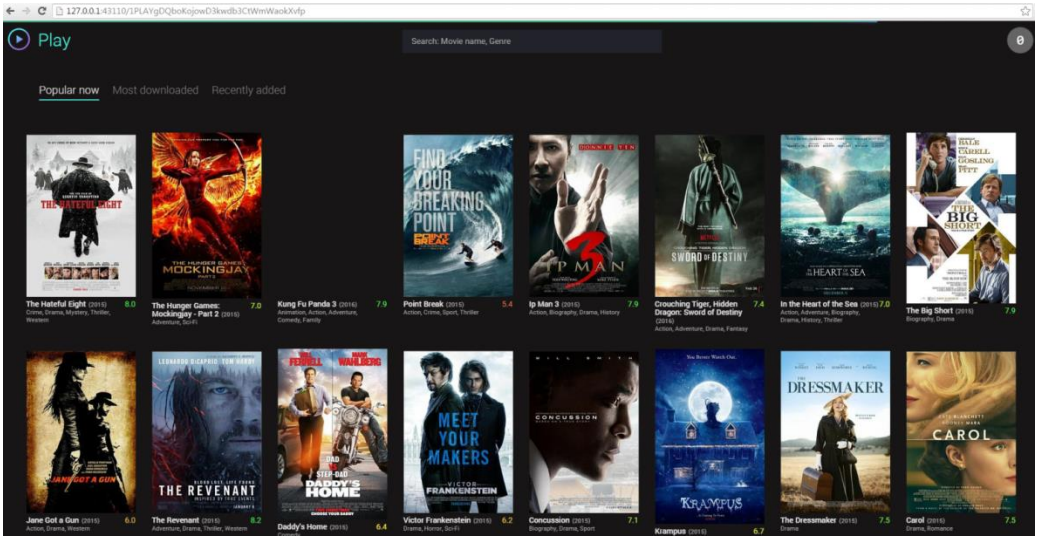
ZeroTalk – a sample site within ZeroNet that acts as a Decentralized, p2p forum demo

Potential threat posed by ZeroNet

Once ZeroNet emerged, it didn't take too long for the first "outlaw" to start using it for illegal activity. Movie piracy is a growing problem, as movies are leaked to P2P networks on a daily basis. Most of these networks are shut down one after the other due to copyrights violation.

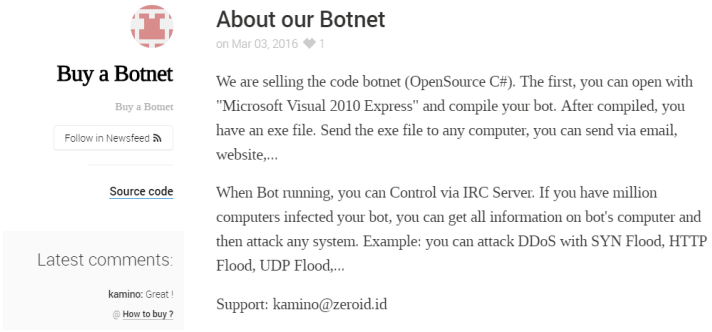
PopcornTime, a very easy-to-install P2P client for the everyday user, has all the latest movies and TV shows.
Since its launch, PopcornTime has faced legal threats several times, which led to the service's being shut down and eventually resurrected by third-party developers[2].

ZeroNet offers a new way to keep PopcornTime and similar services alive. The best example is a new ZeroNet site called **Play** (accessed via its ZeroNet URL). Play offers the same capabilities as any other P2P site, except for one important fact: as long as there is at least one user seeding Play, it will never be shut down.



[2] http://www.theverge.com/2015/10/23/9600576/popcorn-time-history-timeline

In addition, one can find other fishy sites on ZeroNet, for example a site selling botnets:



Attackers may use the ZeroNet protocol to take ransomware attacks to the next level

ZeroNet offers a way to alter the payment system. Instead of payment-gateways, attackers can use ZeroNet sites hosted by the victims themselves. Meanwhile, the attackers stay anonymous by using the Tor network to transfer their communications.

To use ZeroNet effectively for this purpose, the attacker would use BitMessage[3]:

- BitMessage is a P2P communications protocol used to send encrypted messages to another person or to many subscribers.
- It is decentralized and trustless, meaning that you do not need to trust any entities such as root certificate authorities.
- It hides "non-content" data, like the message sender and receiver, from passive eavesdroppers.

The attack is typical of any ransomware, except for the ZeroNet part:

1. On one thread, the malware initiates its usual operation: scan the computer and encrypt all files using a newly generated AES encryption key. This key is then encrypted with a newly generated RSA public key.

2. On a different thread, the malware installs the ZeroNet bundle on the victim's PC and runs a script to do the following:
    - Register a new BitMessage address.
    - Send a BitMessage to a hardcoded address (the attacker's pre-registered BitMessage address) containing the RSA private key, and the newly generated site's public and private keys.
    - Create a new site with all of the information regarding the attack - just like any other ransomware message with the addition of the public-private keys of the newly generated site. The message tells the victim to purchase BTCs into the given account.

[3] https://bitmessage.org/wiki/Main_Page

The attacker owns a BitMessage address (which is hardcoded into the malware) at which the following events unfold:

1. When a new message arrives, the attacker adds the BTC account to his own and waits for the money to arrive.
2. The money is transferred to the attacker's BTC account. As the attacker holds the private key for the victim's account, once the money is transferred he can collect the payment and transfer it to his account or to a mixing service.
3. When that transaction is approved, the attacker updates the site (he has the private key) and uploads the encryption software with the correct RSA private key.

It is now possible to serve payment-gateways to the victim by the victim.

- The newly generated site serves as a BTC address so every victim gets a new BTC address in which to deposit the money. All the money is then transferred to the attacker's account.
- The site can only be taken down by actions of both the attacker and the victim.
- All the attacker has to do is to create a bot that waits for new messages, and have a script ready to add the BTC account as his own, withdraw the deposited money and deliver the decryption software.
- Once all of this has been done, the attacker deletes the victim's site from his ZeroNet database and waits for new victims.

Proof of Concept (PoC)

We ran our script on a VM with windows 7 32bit.

Assumptions:

- The machine was infected with a ransomware, which besides encrypting the machine, also called our script and set it in motion.
- The script was given the private RSA key as an argument.
- The script downloaded the ZeroBundle[4] (which contains all of the ZeroNet related programs and files) and BitMessage[5] (including the dedicated dameon[6]) to a dedicated location on the machine.
- BitMessage is configured properly in daemon mode with the daemon.py working as well.

Operation:

1. The script uses ZeroNet's own modules to generate a new site (which is basically a new Bitcoin key-set).

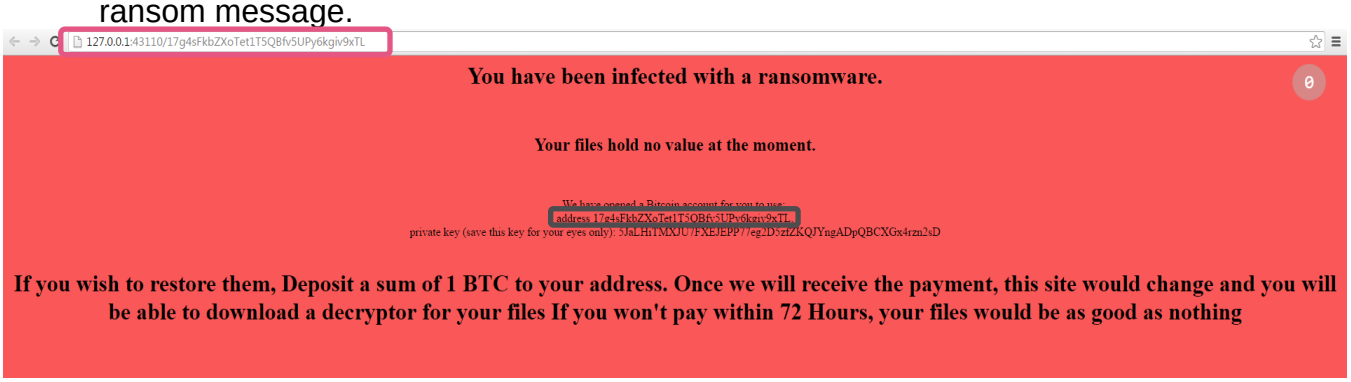
[4] https://github.com/HelloZeroNet/ZeroBundle/releases/download/0.1.1/ZeroBundle-v0.1.1.zip
[5] https://bitmessage.org/download/windows/Bitmessage.exe
[6] https://github.com/Dokument/PyBitmessage-Daemon


```
Generating new privatekey...
privateKey:     5JaLHiTMXJU7FXEJEPP77eg2D5zfZKQJYngADpQBCXGx4rzn2sD
Site address:   17g4sFkbZXoTet1T5QBfv5UPy6kgiv9xTL
------------------------------------------------------------------------
                Site created!
```

2. It changes the content of index.html to display the ransom message.
3. The script uses the dedicated BitMessage daemon functions (with a slight modification) to send a BitMessage to the attacker (a hardcoded BitMessage address)

```
Message Status: msgqueued
```

4. ZeroNet protocol is called and the browser opens on the newly generated site i.e. the ransom message.

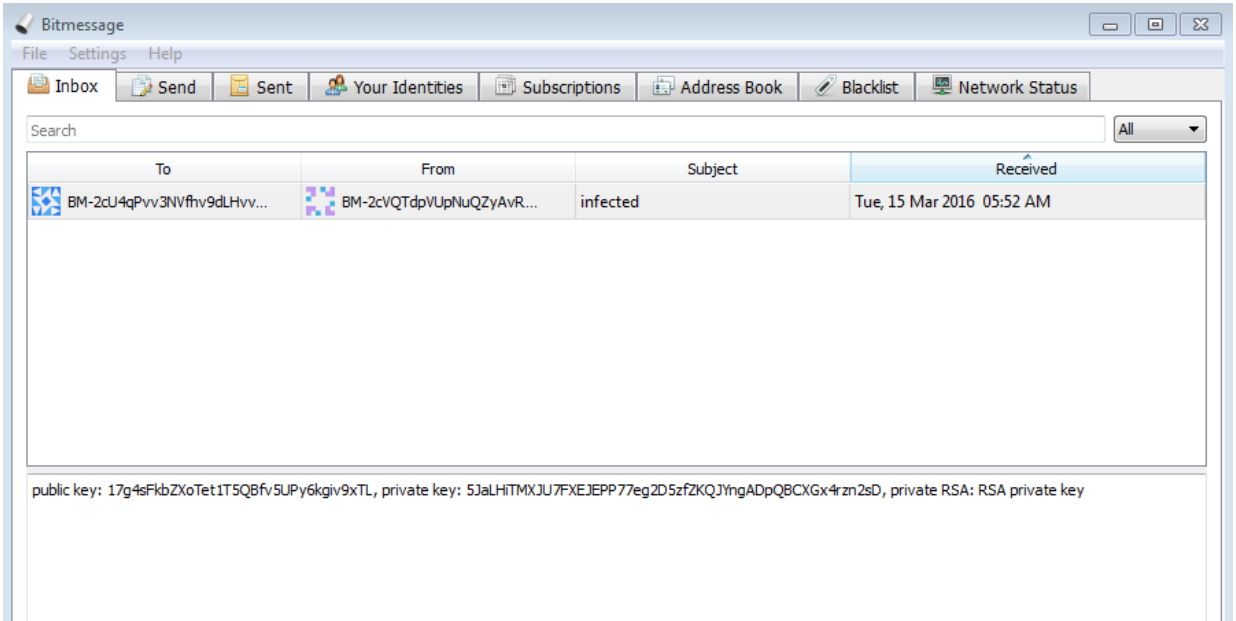


*Notice how the URL matches the public key generated

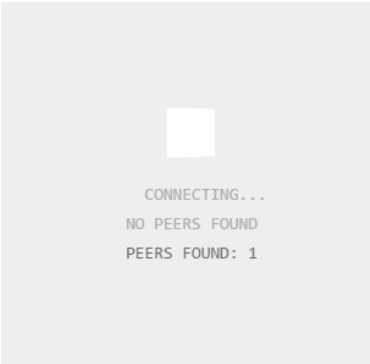
At this point, the ball is in the victim's court. If s/he pays, the attacker will see the transaction and take ownership of these Bitcoins. Afterward, the attacker will change the content of the site to contain the decryption tools and sign it with the private key.

Let's look at the attacker's side:

The attacker receives a new infection message:



The attacker now holds the public and private keys as well as the RSA key. Let's try to connect to that site from his side:

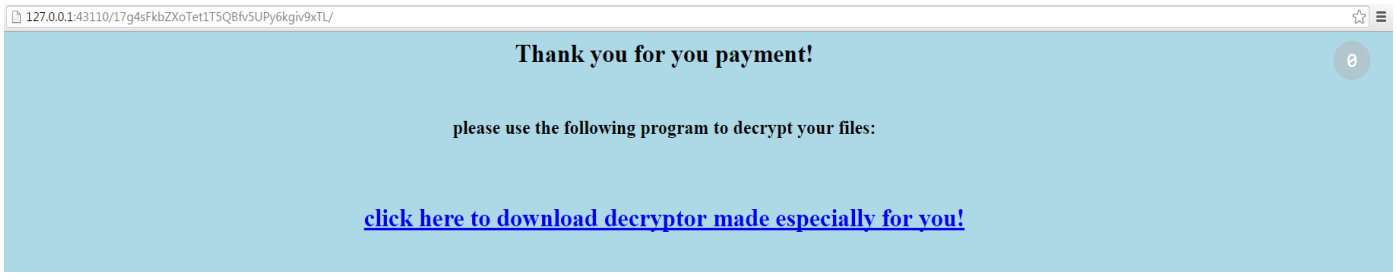


Assuming that a payment has arrived, the attacker changes the site content, and signs and publishes to his victim it (all of this takes place via ZeroNet's modules):

```
\ZeroBundle\ZeroNet>python zeronet.py sitesign 17g4sFkbZXoTet1T5QBfv5UPy6kgiv9xTL
Private key (input hidden):
Site:17g4sF..9xTL Opening site data directory: data/17g4sFkbZXoTet1T5QBfv5UPy6kgiv9xTL/...
Site:17g4sF..9xTL - [SKIPPED] content.json
Site:17g4sF..9xTL - index.html (SHA512: bfcf313a428e844345f55a3e4fe4dc4b63377d2cf2ee74abc7706ec1c156f068)
Site:17g4sF..9xTL Adding timestamp and sha512sums to new content.json...
Site:17g4sF..9xTL Verifying private key...
Site:17g4sF..9xTL Correct 17g4sFkbZXoTet1T5QBfv5UPy6kgiv9xTL in valid signers: ['17g4sFkbZXoTet1T5QBfv5UPy6kgiv9xTL']
Site:17g4sF..9xTL Signing content.json...
Site:17g4sF..9xTL Saving to content.json...
Site:17g4sF..9xTL File content.json signed!
```

```
\ZeroBundle\ZeroNet>"../Python/python.exe" zeronet.py sitePublish 17g4sFkbZXoTet1T5QBfv5UPy6kgiv9xTL
- {'to': 1, 'cmd': 'response', 'ok': 'Reloaded'}
- Sending sitePublish
- {'to': 2, 'cmd': 'response', 'ok': 'Successfuly published
- Done.
```

Once published, the victim views it as a new site:



Conclusion

Technological advancements are welcome but entail certain risks which should be addressed. As we can see in the case of ZeroNet, new technologies can be abused by malicious factors for the worst. ZeroNet could be potentially used not only for the benefit of the general public, but also for the benefit of ransomware writers, by serving as a payment-gateway.

We have presented a POC presenting how this could be done. It is our mission to anticipate such uses and block attack vectors even before they are created. We hope that with further research and development, security vendors will learn how to protect against such threats and keep users worldwide safe from such threats. We at Check Point strive to better understand potential threats in order to develop and implement protections that will keep users one step ahead.

