ANONYMITY IS KING

Michael John Marcos & Anthony Joe Melgarejo
Trend Micro Inc., Philippines

Email {Michael_Marcos, Anthony_Melgarejo}@trendmicro.com

ABSTRACT

After a series of takedowns of command and control (C&C) servers related to notorious banking and ransom malware such as GameOver Zeus, CryptoLocker and Citadel, cybercriminals started to look for innovative ways to make their infrastructure difficult to locate. They began using the Deep Web network (e.g. Tor and I2P). Once a market for illegal drugs and stolen goods, the Deep Web has evolved into cybercriminals' armour – they use it to hide their actions. This paper discusses different Deep Web technologies and both the advantages and disadvantages of their use. The paper also takes a look at several notable pieces of malware that have used the Deep Web network, as well as the trends in the adoption and use of Deep Web by cybercriminals. Finally, it discusses technologies that can help researchers and administrators monitor malicious activities in the Deep Web network to aid their incident response activities, forensic investigation and solution creation.

1. INTRODUCTION

Cybercrime enterprises have become a norm as more and more organized crimes are adapted to the digital world. The exponential growth of online banking trojans, ransomware and cyber-espionage hacking tools in recent years is proof of a growing interest among cybercrime groups and threat actors in stolen information and monetary gain. These types of threats require a network infrastructure that operates 24/7 in order to execute activities successfully. The fact that the infrastructure must always be online is a drawback: servers in such an infrastructure are exposed and represent a single point of failure [1]. This downside limits the usability of a malicious server.

Law enforcement entities can make malicious servers inaccessible by doing any or all of the following [2]:

- DNS sinkholing of malicious server domains
- Revoking access to malicious server domains
- Banning the IP addresses of malicious servers
- Ordering a clean-up of compromised hosts by hosting providers
- De-peering the networks of malicious servers from the entire Autonomous Systems (AS).

Recent takedowns of malicious servers have driven threat actors and cybercriminals to move their operations to the Deep Web.

This paper provides an in-depth study of how threat actors abuse the anonymity of the Deep Web. The paper shows both the client-side and server-side perspectives of how malware uses the Deep Web to communicate with malicious servers.

The paper covers real-world malware cases where threat actors and cybercriminals have used the Deep Web for their operations. Data gathered through the *Trend Micro Smart Protection Network* shows Deep Web malware trends.

The last section of this paper discusses investigation techniques that can be used to monitor Deep Web activity in an infected machine.

2. MOTIVATIONS BEHIND THE SHIFT TO THE DEEP WEB

The primary advantage for users of the Dark Web is anonymity – its users can be confident that no tapping can be carried out on their communications. One disadvantage of using the Deep Web is increased network latency. However, the advantages of making malicious services reachable via the Deep Web outweigh this disadvantage.

2.1 Deep Web traffic is encrypted

Deep Web protocols enable layered encryption in the messages that transmit in the Deep Web network. Said protocols use a combination of encryption algorithms such as RSA,

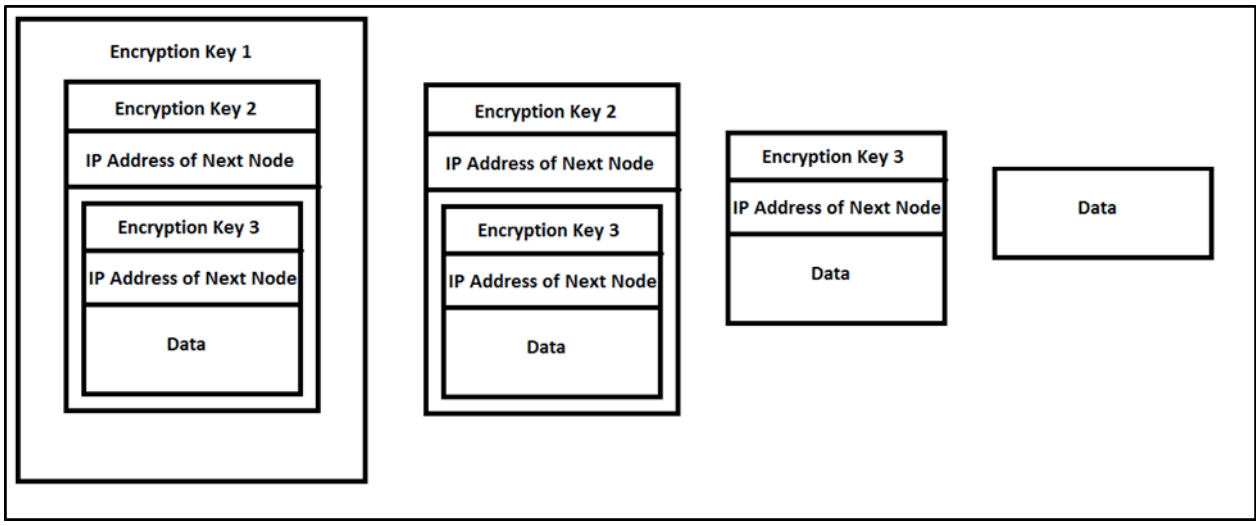

Figure 1: Layered encryption diagram shows several encryption keys: encryption key 1, encryption key 2 and encryption key 3, all representing the data that is transmitted within the Deep Web.

Diffie-Hellman, AES, etc. to encrypt messages. Since all messages – malicious or not – are treated equally, communications between an infected machine and a malicious server are concealed within legitimate Deep Web traffic. Most security appliances and network analysers find it difficult to distinguish between malicious and legitimate Deep Web traffic.

2.2 The Deep Web offers deception

Deep Web users can generate an infinite number of domains. Creating multiple instances of the Deep Web technology (e.g. Tor and I2P) is feasible provided the load and bandwidth can be handled by the malicious servers. The use by a small cybercrime group or an individual threat actor of multiple domains creates the illusion of a larger threat network infrastructure. Malware creators may design malware that connects to multiple Deep Web sites when in reality it is only connecting to one or two malicious servers. This creates additional work for security vendors who block domains used by Deep Web malware.

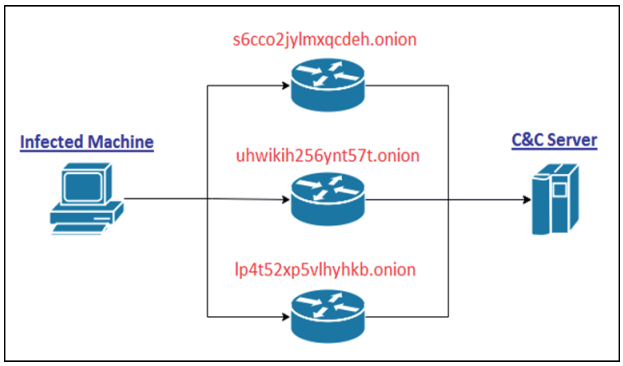

Figure 2: Deception technique with the use of the Deep Web. The data from an infected machine goes through several paths before it arrives at the malicious C&C server.

2.3 The Deep Web provides resilience

System administrators often sinkhole malicious domains to block communications to and from malicious servers. However, cybercriminals and threat actors can counteract this by using a Deep Web domain. An updated Deep Web domain is usually included in the configuration file that is pushed from the malicious server to all infected machines. The increased resiliency offered by the Deep Web makes every malware investigation a cat-and-mouse game between threat actors and researchers/law enforcement.

2.4 The Deep Web provides high availability

Deep Web domains can be reused by another computer. This is helpful when malicious servers have breakdowns such as a power outage or denial of service (DoS). Should servers be replaced or the server location renewed, the Deep Web domain can be reused.

3. UNDERSTANDING THE DEEP WEB TECHNOLOGIES USED BY MALWARE

3.1 Tor (The Onion Router)

As shown in Figure 3, Tor anonymizes network traffic by routing it to three randomly selected 'nodes' or 'relays' ('B' in Figure 3) before connecting to the actual destination ('Remote Server' in Figure 3). The data is encrypted by the number of hops it makes before reaching the remote server. In each hop, the current node decrypts a layer of this encryption – similar to peeling the layers of an onion – to determine the next hop. The key used in decrypting the data is unique for each hop. Therefore, each node only knows the previous and next hops. Users are required to install the Tor client (e.g. *Tor Browser Bundle*) to route their network traffic to the Tor network. The Tor client contacts one of the Tor network's directory servers ('A' in Figure 3), which contains a listing of available nodes, to select the nodes that it will use to build a 'circuit' for communicating with the remote server. These available nodes are mostly run by volunteers. Note that the data is fully decrypted after the last node or the 'exit node' ('C' in Figure 3). Therefore, the use of more secure protocols (like HTTPS) is recommended to protect your information from being revealed to attackers monitoring the exit nodes or even actually running one.

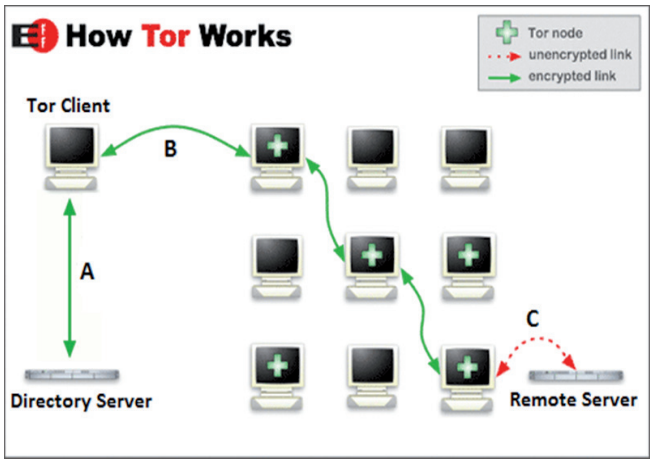

Figure 3: An illustration of how the Tor network works. The installed Tor client sits on the user's computer. Data transmitted within the Tor network is encrypted until it reaches its final remote server destination.

Tor hidden services

Tor allows its users to offer web services without revealing the users' location or IP address. These services are called hidden services ('HS' in Figure 4). For a hidden service to become reachable, it needs to publish its presence in the network. First, it randomly selects nodes to act as its introduction point and build circuits to them ('A' in Figure 4). It then sends its descriptor, consisting of its introduction points and public key (PK), to a distributed hash table (DB) ('B' in Figure 4). The descriptor is found by clients requesting XYZ.onion (XYZ is derived from PK) ('C' in Figure 4). After getting the descriptor, the client builds a circuit to one of the introduction points ('D' in Figure 4). The client will also select a random node as the rendezvous point (RP) ('E' in Figure 4), where the selection could have been made earlier. The client sends a one-time secret and the rendezvous point to the hidden service via the selected introduction point. The hidden service builds a circuit to the rendezvous point and sends the one-time secret (F in Figure 4). Once the rendezvous point has confirmed the one-time secret from the hidden service, the client and the hidden service can communicate anonymously with each other.

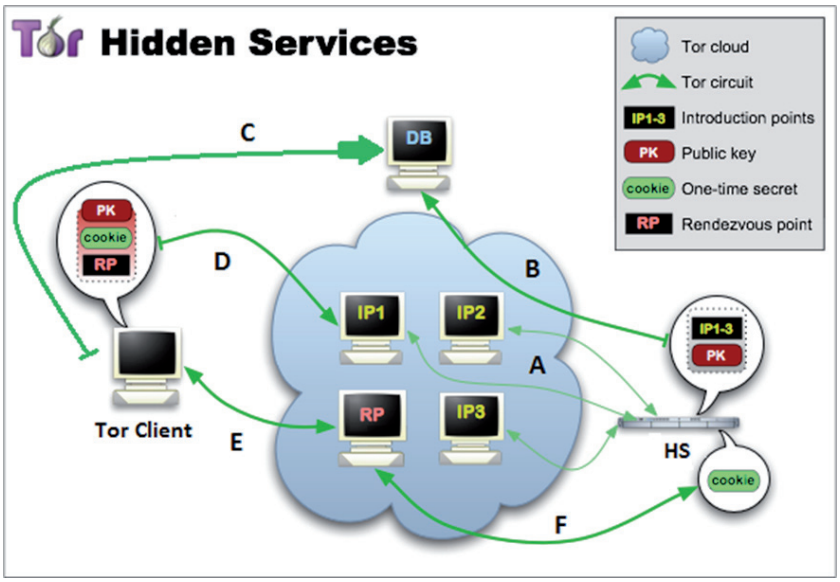

Figure 4: An illustration of how the Tor hidden services work. The client sends a one-time secret to the hidden service, which the rendezvous point confirms in order to set up the communication between the client and the hidden service.

Tor used by malware

The trade of illegal drugs (think Silk Road), firearms and child pornography in cyberspace has been linked with Tor from the beginning. However, cybercriminals and threat actors have jumped on the Tor bandwagon. The creators of the popular banking trojan ZBOT/ZeuS saw the advantages of using the Deep Web [3]. A variant of ZBOT/ZeuS known as TSPY_ZBOT.AAMV [4] included Tor as part of its C&C communication and botnet routines. Looking at the malware's code, there are three pairs of DOS and NT Headers. The first one is for the 32-bit executable itself (Figure 5), the second is for 64-bit process injection (Figure 6), while the last pair is the Tor executable (Figure 7). After installing itself on the system, it proceeds to inject its code into explorer.exe (see Tables 1 and 2).

The injected ZBOT code in explorer.exe then connects to http://egzh3ktnywjwabxb.onion/zs/mimi.jpg, an Onion site. ZBOT does this to download its configuration file, which is essential for its information- or credential-stealing routine. Since it connects to a C&C server that is hosted in a Deep Web domain, security researchers will have a hard time pinpointing the actual IP address of the C&C server.

It then uses process replacement by creating a suspended process of svchost.exe and overwriting its memory with the

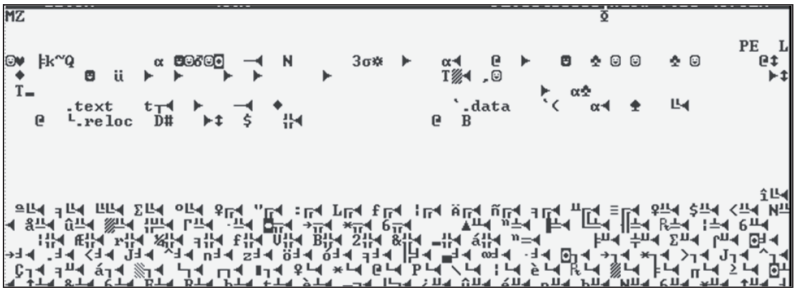

Figure 5: Screenshot of the DOS and NT header of the 32-bit executable of TSPY_ZBOT.AAMV.

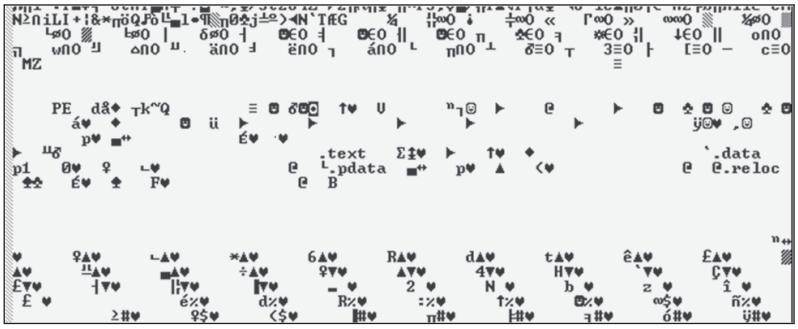

Figure 6: Screenshot of the DOS and NT header referring to the 64-bit executable used for the process injection routine of TSPY_ZBOT.AAMV.

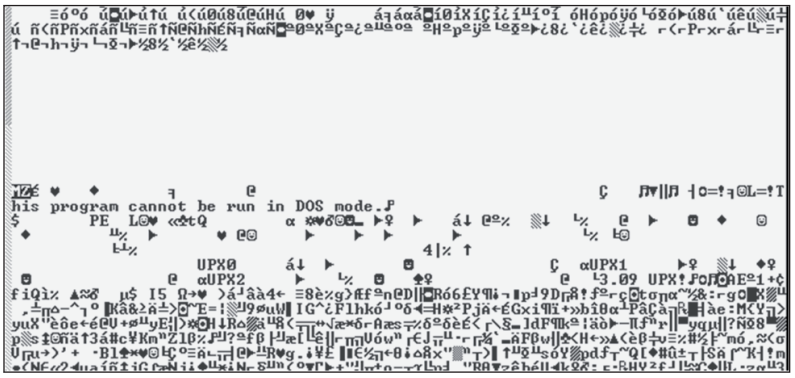

Figure 7: Screenshot of the DOS and NT header referring to the embedded Tor executable of TSPY_ZBOT.AAMV.

Process Path	Operation	Info
C:_virus\11.exe	create file	C:\Documents and Settings\ \Application Data\Mubyx\fekya.exe

Table 1: TSPY_ZBOT.AAMV drops its copy in the %AppData% folder.

Process Path		Operation	Info
C:\Documents and Settings'	\Application Data\Mubyx\fekya.exe	remote thread...	C:\WINDOWS\explorer.exe

Table 2: TSPY_ZBOT.AAMV injects its code into explorer.exe.

Protocol	Host	URL	Body	Caching	Content-Type	Process
HTTP	checkip.dyndns.org	/	105	no-cache	text/html	explorer:1228
HTTP	egzh3ktnywjwabxb.onion	/zs/mimi.jpg	512	no-cac...	text/html; c...	explorer:1228

Table 3: TSPY_ZBOT.AAMV, while injected in explorer.exe, checks the infected machine's IP and downloads a configuration file from the Onion site.

Figure 8: A screenshot of the Onion address used by cybercriminals who deployed TSPY_ZBOT.AAMV.

Tor executable. It executes the Tor hidden service with the following command line parameters:

--HiddenServiceDir "%appdata%\tor\hidden_service"

--HiddenServicePort "1080 127.0.0.1:23318"

--HiddenServicePort "5900 127.0.0.1:26824"

--HiddenServiceDir defines the location of the hidden service's configuration. The location is a folder containing two files: hostname and private_key. In this scenario, the file of interest is the hostname, which contains the text vv7sx5v3iv6rwgz6.onion. This is the Onion address used by cybercriminals to access the infected system's hidden service (see Figure 8). This Onion address is sent to the malicious C&C server to notify the cybercriminals of the newly created hidden service.

--HiddenServicePort defines the virtual port, the IP address and the redirection port to which the connections to the virtual port are redirected. In this scenario, the virtual ports are ports 1080 and 5900. Note that the virtual ports are ports to which users think the hidden service connections are flowing. In reality, the connections are redirected to ports 23318 and 26824. Note that redirection ports are generated randomly.

The hidden service allows cybercriminals to control the infected system remotely. Note that ZBOT is capable of virtual network computing (VNC). This is evident since explorer.exe listens to the redirection ports specified in the hidden service configuration (Figure 9).

Process	Protocol	Local Address	Remot...	State
explorer.exe:1228	TCP	WinXP:23318	WinXP:0	LISTENING
explorer.exe:1228	TCP	WinXP:26824	WinXP:0	LISTENING

Figure 9: The explorer.exe process listening to the redirection ports.

3.2 Tor2web

While Tor requires its users to install an application – such as *Tor Browser* – which serves as the client that connects to the Tor network, Tor2web allows users access to Tor hidden services using normal web browsers. From a cybercriminal's point of view, use of Tor may mean they first need to find ways to install the Tor client in the target system, usually through either of the following methods:

- Downloading Tor (or its required modules) as part of the malware's installation routine
- Embedding Tor (or its modules) into the malware itself.

To combat these, security software and scanners may be configured to detect Tor running in suspicious processes or in processes where it does not need to run. Moreover, system

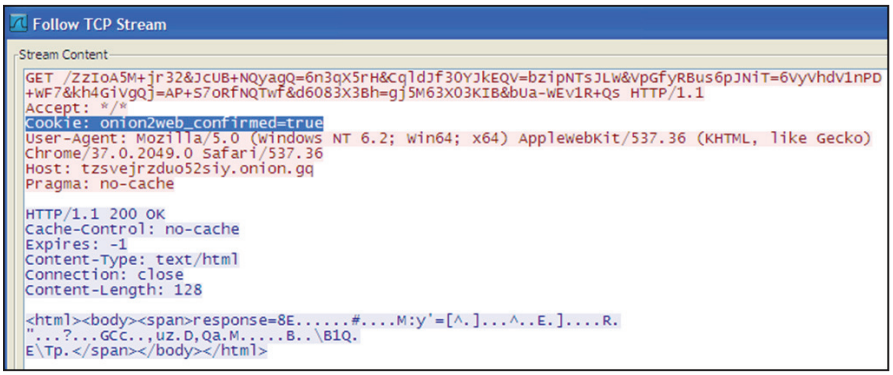

Figure 10: Encoded communication of a CTBLocker variant with its C&C server.

File pos	Mem pos	ID	Text
A 000000000001	000000000001	0	onion.gq
A 000000000019	000000000019	0	onion2web_confirmed=true
A 00000000003D	00000000003D	0	onion.lt
A 000000000055	000000000055	0	disclaimer_accepted=true
A 000000000079	000000000079	0	tor2web.fi
A 000000000091	000000000091	0	disclaimer_accepted=true
A 0000000000B5	0000000000B5	0	tor2web.org
A 0000000000CD	0000000000CD	0	disclaimer_accepted=true
A 0000000000F1	0000000000F1	0	tor2web.blutmagie.de
A 000000000109	000000000109	0	disclaimer_accepted=true
A 00000000012D	00000000012D	0	onion.cab
A 000000000145	000000000145	0	onion_cab_iKnowShit=1
A 000000000165	000000000165	0	Mozilla/5.0 (Windows NT 6.2; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2049.0 Safari/537.36

Figure 11: Tor2web proxy list of CTBLocker variant TROJ_CRYPCTB.SME.

configuration may impose strict policies on the downloading and installing of Tor on computers. Network and system administrators may even choose ultimately to prohibit the use of Tor, if it is not deemed essential to their users' daily operations.

Cybercriminals may revert to the use of Tor2web should they prioritize convenience and ease of use. This also means that cybercriminals are trading off their security and anonymity [5]. Using Tor2web is simple: users replace the .onion of the hidden service address with a Tor2web gateway such as .tor2web.org. Some examples of Tor2web nodes we have seen managed by different individuals or organizations are the following:

- *.tor2web.fi
- *.tor2web.blutmagie.de
- *.onion.sh

Tor is installed in the Tor2web node and is responsible for routing the user's network traffic to the Tor network.

With Tor2web, cybercriminals and threat actors are able to bypass security guidelines pertaining to Tor. The following are some reasons why using Tor2web is an advantage for malicious cyber activity:

- Virtual Tor: Tor is not present in the system since it is not downloaded or installed onto the system. Therefore, no suspicious Tor-related file activity will be detected.
- Outbound and inbound network traffic to and from the Tor2web node is normal network traffic: The data will only be encrypted (for the Tor network) upon reaching the Tor2web proxy. Also, the data – received from the hidden service – sent back by Tor2web is already decrypted. Therefore, network appliances will not detect suspicious Tor network traffic.

Tor2web used by malware

The Deep Web is not only utilized by malware such as banking trojans and botnets. It is also used by malware with a more destructive nature such as crypto-ransomware. The Deep Web makes crypto-ransomware C&C servers, payment sites and support portals resilient.

Curve, Tor, Bitcoin-Locker, also known as CTBLocker or CRYPCTB, is the first of its kind to use the Deep Web. The name CTBLocker is taken from its characteristics: it uses the Elliptic Curve Diffie-Hellman (ECDH) [6] algorithm in its encryption routine; it uses Tor for its C&C and payment sites; and it prefers bitcoin as the currency for paying its ransom. After its file encryption routine, it contacts its C&C server to send information for decrypting files. The server will reply with the bitcoin address and price of decryption.

Figure 10 shows that the CTBLocker's C&C server is a Tor hidden service using the Onion address tzsvejrzduo52siy.onion. It connects to an onion.gq domain which is a Tor2web proxy. A CTBLocker variant known as TROJ_CRYPCTB.SME [7] eliminates the need to have the Tor client's code embedded, which is seen in the older variants. Further examination of its code reveals five other Tor2web proxies it tries to connect to as its failsafe routine (Figure 11).

In the list, there are corresponding cookies for each Tor2web proxy and at the bottom is the user-agent that is used in the packet header for C&C communication. This list tells us that the increasing popularity and availability of Tor2web proxies will present a new set of difficulties for security researchers. Constant updates in network traffic monitoring rules will be required to block access to malicious C&C servers.

The reason for there being several Tor2web proxies as a failsafe is because the owners of these proxies may choose to

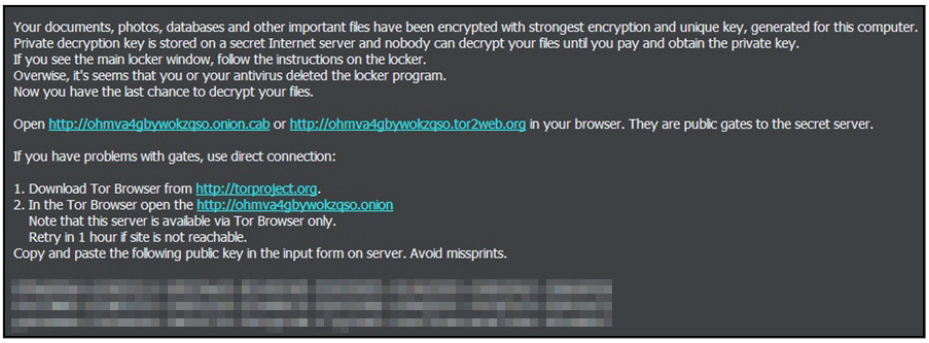

Figure 12: Screenshot of the ransom note shown by a CTBLocker variant.

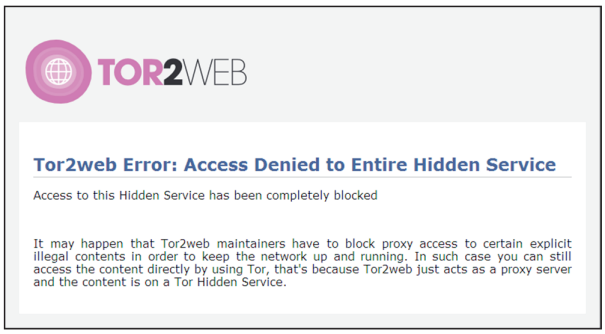

Figure 13: The error message displayed when access to a hidden service via Tor2web proxy is blocked.

Figure 14: Screenshot of the payment site index of a CTBLocker variant, accessed via Tor2web proxy.

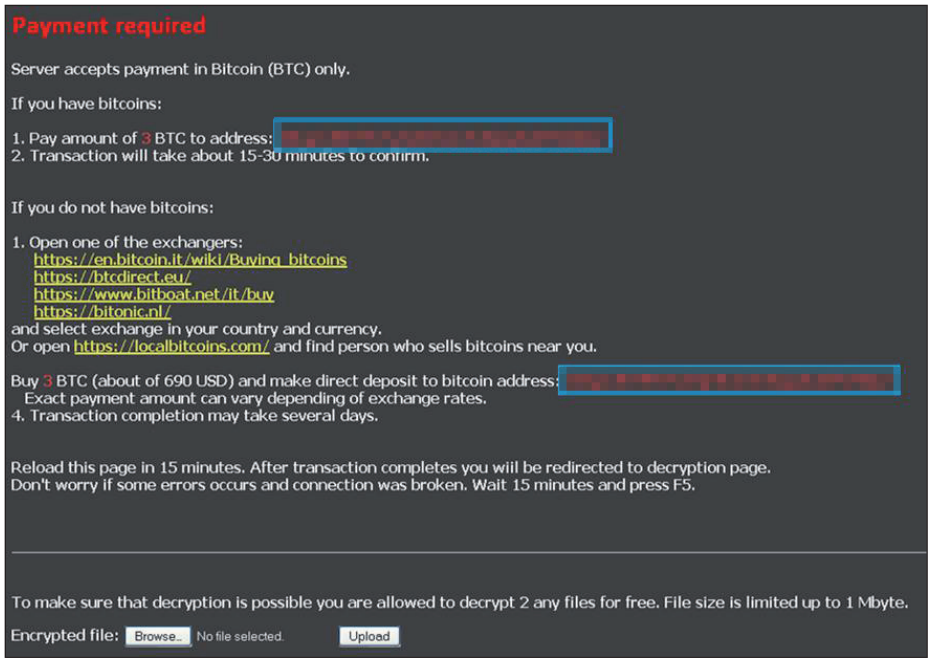

Figure 15: Screenshot of the payment page of a CTBLocker variant, accessed via Tor2web proxy.

block clients' access to the C&C servers. However, note that Tor users can easily change the onion domain or even create multiple onion domains to host the same C&C server [2].

In the final stage of its routine, similar to all ransomware, CTBLocker displays its ransom note. The note (shown in Figure 12) contains instructions for the victim on how to recover their encrypted files. In the note, the victim is asked to visit the payment site, which is hosted on the Tor network via Tor2web. If accessibility problems occur (Figure 13), the victim is instructed to install the *Tor Browser* to access the site. This way, access will not be blocked since the connection won't need to go through a third-party proxy. The threat actors are aware and prepared for these hindrances. Once the victim accesses the site, he will be asked to enter the public key indicated in the ransom note (Figure 14).

Upon sending the public key, the victim is redirected to the payment page (Figure 15). The page contains the bitcoin amount and wallet information for paying the ransom. Instructions for exchanging real currency for bitcoin are also included. The bottom of the page contains an offer to decrypt two files for free. This is done as proof of the legitimacy of the decryption – which is a trend observed in the crypto-ransomware variants found at the time of this research.

3.3 Invisible Internet Project (I2P)

The Invisible Internet Project (I2P) uses 'garlic routing', an extension of onion routing. When I2P is installed, the machine acts as a router for other I2P users. Its router information (called routerInfo) is sent to the network database (called netDB). The netDB information is maintained and distributed to a subset of routers called floodfill routers. To communicate via the I2P network, the client must first look up the available routers in the netDB. After selecting the routers, it builds two one-way tunnels known as outbound and inbound tunnels. The tunnel information (called leaseSets) is sent to the netDB. In garlic routing, the client router bundles other messages (called cloves) to form a garlic message. Each clove has its own delivery instruction, making it more difficult to monitor the network traffic since each clove has a different destination. Also, similar to onion routing, the messages in I2P are encrypted and each router only knows the previous hop and the next hop. Note that in I2P, only web services inside the I2P network (called eepSites) are allowed to be contacted by default. The eepSite also has its own pair of tunnels. The client again looks up the netDB for the eepSite's leaseSet to find its inbound tunnel. Once the garlic message reaches the server router, the cloves are sent to their actual destinations (Figure 16).

A delivery status message is sent back to the client router to inform it that the connection has been successful. The database store message, which contains the client's leaseSet, is used by the server router to determine how to reach the client's inbound tunnel. This is for optimization, eliminating the need for the server router to look up the netDB for the client's leaseSet.

I2P used by malware

After the takedown of the Zeus botnet [8], a new banking trojan took the spotlight. This malware, known as DYRE [9], steals credentials by setting hooks in web browsers (Figure 17) to perform man-in-the-middle attacks, and inserts web injects to modify the banking website's content displayed by the browser.

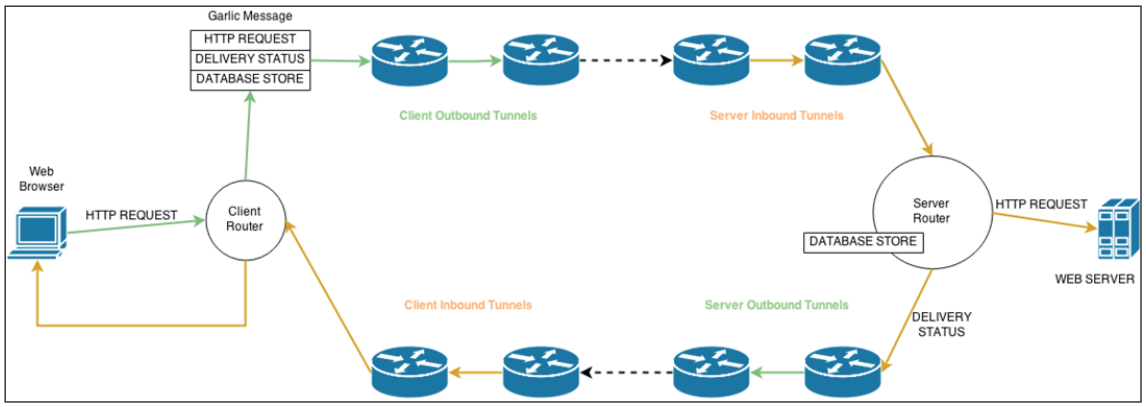

Figure 16: Diagram showing how I2P works.

File pos	Mem pos	ID	Text
0000005224FB	0000005224FB	0	5r3ywoac6
000000522593	000000522593	0	DisplayName
0000005225AF	0000005225AF	0	Software\Microsoft\Windows\CurrentVersion\Uninstall
000000522637	000000522637	0	SYSTEM\CurrentControlSet\services
000000522797	000000522797	0	Global\bdm2wosh32
0000005227BF	0000005227BF	0	D:P(A;;GA;;;SY)(A;;GA;;;BA)(A;;GA;;;WD)(A;;GA;;;RC)(A;;G
00000052286F	00000052286F	0	D:P(A;;GA;;;SY)(A;;GA;;;BA)(A;;GA;;;WD)(A;;GA;;;RC)S:(ML
000000522907	000000522907	0	chrome.exe
00000052291F	00000052291F	0	firefox.exe
000000522937	000000522937	0	iexplore.exe
000000522997	000000522997	0	C:\Program Files\Internet Explorer\iexplore.exe
00000053E11C	00000053E11C	0	C:\WINDOWS\system32\
00000053E1DC	00000053E1DC	0	Windows.WinINET.processorArchitecture="x86".publicKe

Figure 17: DYRE malware targets web browsers such as Google Chrome, Firefox and Internet Explorer.

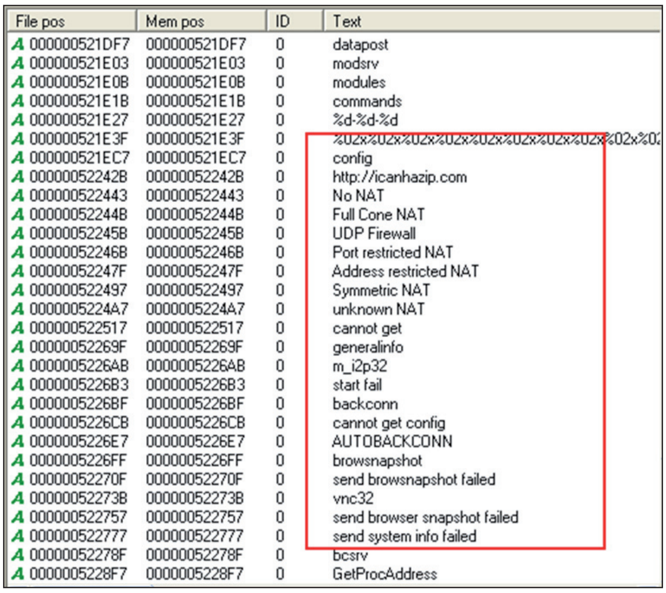

Figure 18: DYRE is capable of getting and sending NAT status and sending browser snapshots, among other things.

Aside from stealing credentials, DYRE is known to:

- Send NAT status
- Send the infected machine's system/general information
- Download an I2P module
- Create a back connection
- Download its updated configuration file (contains a list of targeted banks and web injects)
- Send a browser snapshot
- Download a VNC module
- Download a TV module.

TSPY_DYRE.YYYP [10] is known to apply various techniques for contacting its C&C servers, including I2P (Figure 19). First, it contacts the C&C server hard-coded in its binary to send a beacon with the following sample request format:

```
https://{C&C IP address}/{campaign ID}/{Computer name}_{Windows OS build}.{Unique ID}/5/spk/{Victim's public IP}/
```

It also uses a domain-generation algorithm (DGA), appending 34-character domain names to these top-level domains (TLDs): so, tk, cn, hk, in, to, ws and cc (Figure 20).

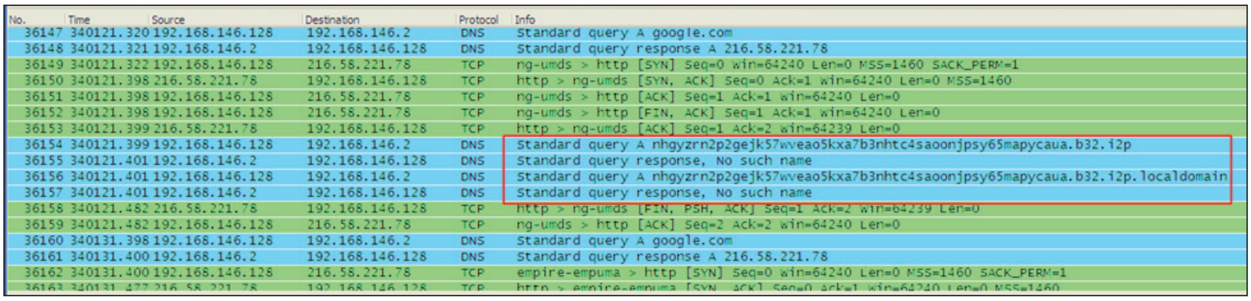

Figure 19: TSPY_DYRE.YYYP connects to its I2P C&C server.

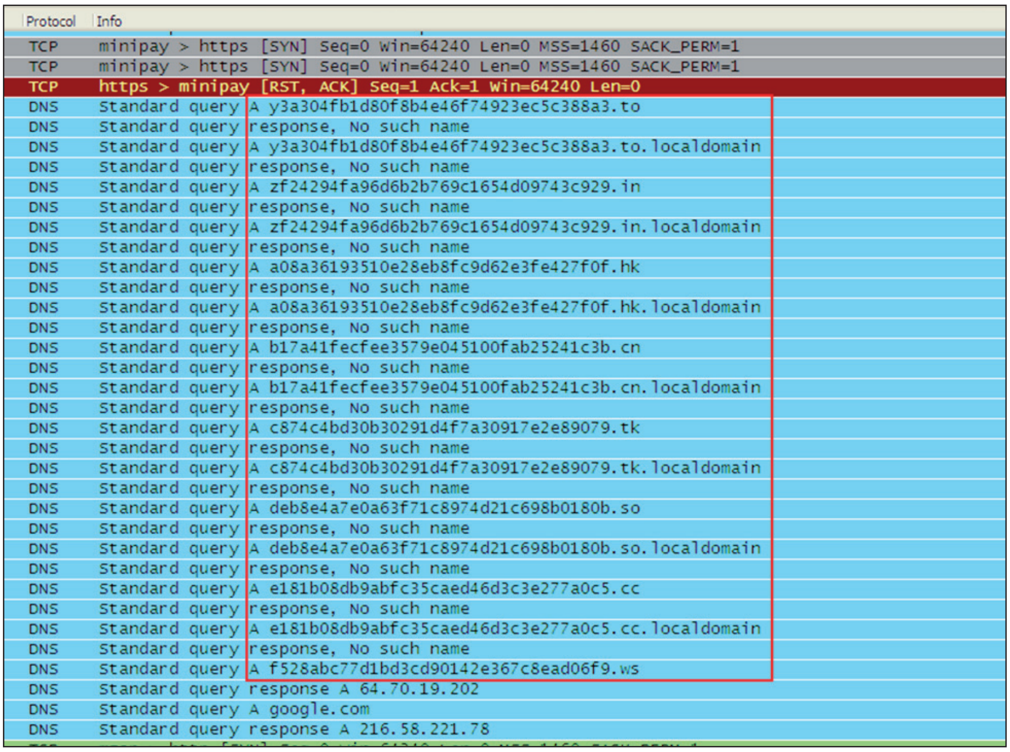

Figure 20: TSPY_DYRE.YYYP appends various TLDs to a 34-character domain name.

The aforementioned routines ensure that the malware maintains continuous communication with its C&C servers. This is essential for cybercriminal groups and threat actors since not only does it mean they can control the infected machines, but they also have more opportunity to receive stolen information. In addition, these routines protect their C&C servers from takedowns.

4. CYBERCRIMINALS' AND THREAT ACTORS' USE OF THE DEEP WEB NETWORK

The effectiveness of a malware attack relies on constant communication with infected machines. Cybercriminals and threat actors using the Deep Web find all kinds of ways to use the technology for their malicious schemes, allowing them to be flexible when planning their attacks. The anonymity that the Deep Web provides makes it one of the best alternative options to set up malicious servers.

Over the past three years, we've grouped cybercriminals' use of the Deep Web technology into three categories: as a ransomware support portal, as a botnet administration C&C server, and as a file server.

4.1 Ransomware support portal

One piece of malware that offers exceptional customer service is the ransomware known as BAT_CRYPVAULT [11], which arrives as an attachment to spammed email. After infection, the malware instructs the affected user via a ransom note to visit its website using a Tor browser in order to decrypt his/her files. Upon visiting the site, it asks for the VAULT.KEY file that is generated by the malware. This is used by the site to generate an account for the infected user. After uploading the key file, the user is redirected to the homepage of the ransomware's support portal (Figure 21).

TROJ_CRYPTESLA [12], which targets gaming-related files, is another piece of ransomware that has a support portal hosted in a .onion domain (Figure 22).

4.2 Botnet administration C&C server

Placing C&C servers on the Deep Web has become a viable option since it masks the infrastructure controlled by cybercriminals. It gives them an additional layer of protection against actions taken by law enforcement.

One example is a piece of mobile malware specifically targeting users of *Android* phones. ANDROIDOS_POSLEM connects to a C&C server in Tor. It arrives via a malvertisement, disguised as an *Adobe Flash Player* installer.

Further investigation of the components and files of this mobile malware reveal that it is almost identical to Orbot [13], an app created by the makers of Tor to enable *Android* phones to connect to the Tor network. The only difference is an additional service, Slempo. This service contains all the

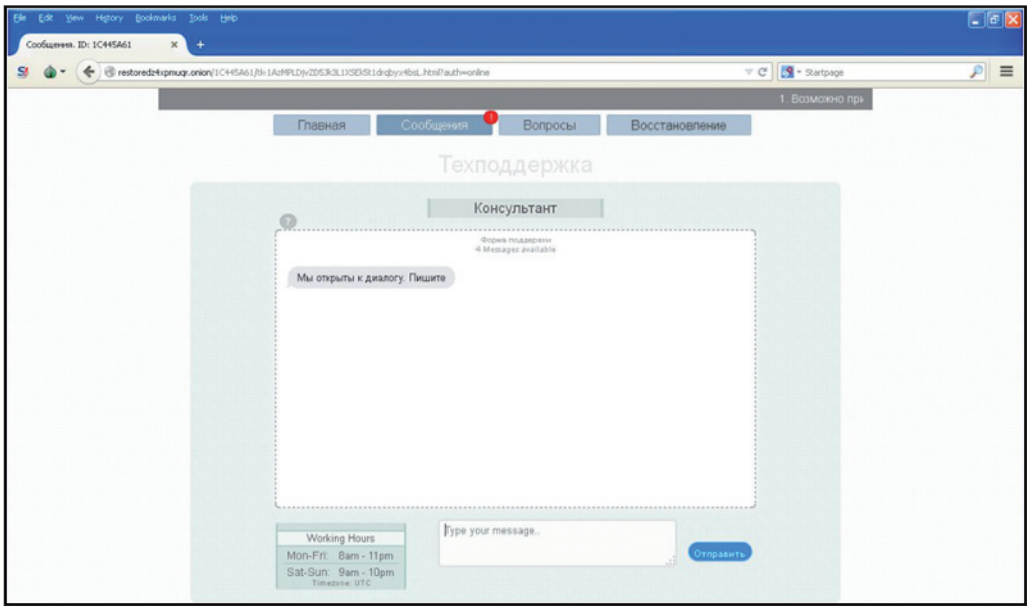

Figure 21: BAT_CRYPVAULT's support portal where users can send a message to the ransomware's support team.

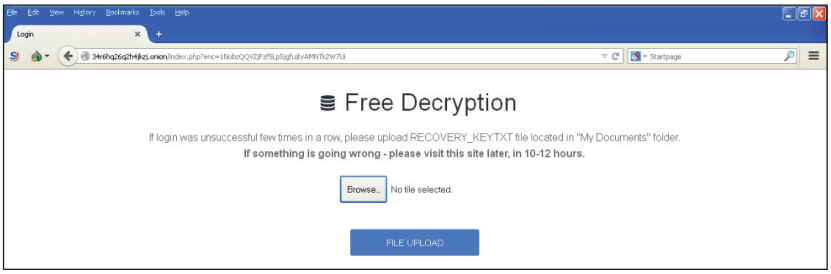

Figure 22: TROJ_CRYPTESLA's support portal, hosted in a .onion domain. This ransomware also lets users upload some of their encrypted files for free decryption in order to convince them of the legitimacy of the decryption service.

```
public static void sendCheckData(Context context)
{
    SharedPreferences sharedpreferences;
    JSONObject jsonobject;
    sharedpreferences = context.getSharedPreferences("AppPrefs", 0);
    jsonobject = new JSONObject();
    String s;
    jsonobject.put("type", "device check");
    jsonobject.put("phone number", Utils.getPhoneNumber(context));
    jsonobject.put("country", Utils.getCountry(context));
    jsonobject.put("imei", Utils.getCutIMEI(context));
    jsonobject.put("model", Utils.getModel());
    jsonobject.put("os", Utils.getOS());
    jsonobject.put("client number", "1");
    s = jsonobject.toString();
    try
    {
        if (send(context, "http://yuwurw46taaep6ip.onion/", s).getStatusLine().getStatusCode() != 200)
        {
            throw new Exception();
        }
        break MISSING_BLOCK_LABEL_143;
    }
    catch (Exception exception) { }
    Utils.sendMessage(sharedpreferences.getString("CONTROL_NUMBER", ""), s);
    return;
    JSONException jsonexception;
    jsonexception;
    jsonexception.printStackTrace();
    return;
}
```

*Figure 23: sendCheckData sends the infected phone's details to the C&C server that is hosted on a *.onion domain.*

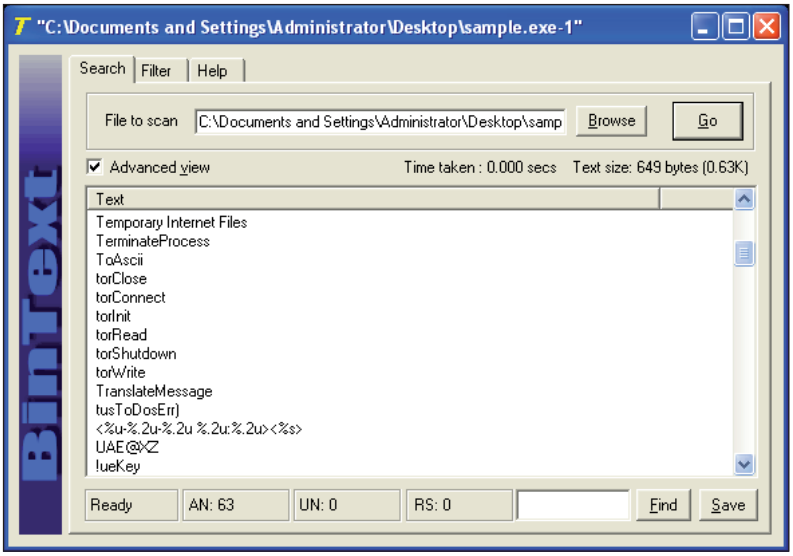

Figure 24: The commands torClose, torConnect, torInit, torRead, torShutdown and torWrite are all from the addons.bat file of this particular BKDR_BIFROSE variant. These commands all relate to Tor usage.

malware's information-stealing routines. Quick inspection of the resource folder of Slempo shows that the Tor binary and the Geo IP database used to track the location of the infected phone are concealed as *.mp3 files.

The sendCheckData function found in the TorSender class (Figure 23) sends the following information to the C&C server:

- Telephone number
- Country
- IMEI number
- Model
- Version

BKDR_BIFROSE [14], a backdoor implicated in targeted attacks that target the human resource personnel of government offices in Africa and Europe, has a variant that also connects to a C&C server that is in the Deep Web. This variant drops addons.dat, a component that enables the backdoor to use the Deep Web, in the %User Profile%\Application Data folder (Figure 24). The malware uses a process replacement technique to hide the component in an iexplore.exe process.

4.3 File server

As cybercriminals and threat actors have embraced the use of the Deep Web for their botnets, use of the same as malicious file servers has quickly followed suit. The downloader TROJ_HANCITOR, which downloads and executes BKDR_VAWTRAK [15], uses the Tor2Web proxy. This online banking trojan is hosted in a file server hosted on the Deep Web. The Tor2Web proxy is used by this malware to connect to a .onion site.

Analysis of the network traffic indicates that inbound and outbound connections of the malware with the C&C server pass through port number 443, which is used for secured communications.

Further investigation of point-of-sale (PoS) malware TSPY_POSLUSY and TSPY_FSYNA reveal that both also use the Deep Web. After validation of stolen information, both pieces of malware upload the information to a file server hosted in a *.onion domain.

5. DEEP WEB MALWARE: CURRENT TRENDS

Our continuous monitoring and investigation of malware trends reveals an increase in the number of malware families that use the Deep Web:

2012	2013	2014	2015
Skynet	Sefnit	Chewbacca	CryptoWall 3.0
	Atrax	BitCrypt	CTB Locker
	Zbot	Bifrose	Dyre
		Onionduke	VaultCrypt
		CryptoWall 2.0	TeslaCrypt
		LusyPOS	Babar
		Slempo	Chanitor
			Vawtrak

Table 4. Malware families using the Deep Web as observed from 2012 until the first quarter of 2015.

The popularity of hidden services in the criminal underground, like Silk Road, convinced a cybercriminal to create a Zeus variant that connects to Tor named Skynet. Nine months after the discovery of Skynet, three botnets were identified to be connecting to the Deep Web. These were released in 2013, with an interval of three to four months between the releases. The number of pieces of malware using Deep Web nearly tripled between 2013 and 2015.

Based on the data we've gathered, the early adopters of the Deep Web technology were threats actors that manage the following operations:

- Cyber extortion by ransomware
- Data theft via information stealers
- Cyber espionage through targeted attacks

Using the *Trend Micro Smart Protection Network*, we've noted that the use of Deep Web URLs by malware is steadily increasing over time (see Figure 25). We also noticed two spikes that occurred in the first quarters of 2013 and 2015. These were due to the outbreaks of Sefnit and CTBLocker, respectively.

6. DETECTING AND MONITORING DEEP WEB ACTIVITY

The Deep Web offers a protected platform for cybercriminals to support a variety of malicious activities. The services range from drug selling, human trafficking and online child pornography [17] to protection against law enforcement takedowns. As such, both security professionals and law

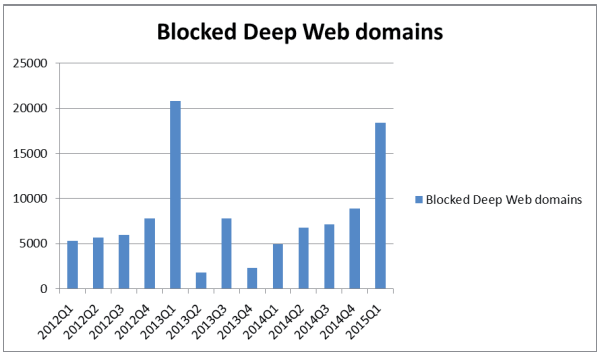

Figure 25: Deep Web domains blocked via Trend Micro Smart Protection Network. The spikes in the first quarter of 2013 and 2015 are due to two outbreaks: Sefnit in 2013 and CTBLocker in 2015.

enforcement need information to be able to detect and monitor Deep Web activities within their jurisdiction.

Monitoring the whole Deep Web infrastructure is challenging both technically and legally. To get around this, we focus on the information that we can gather on an infected machine. Thus, the scope of the analysis is limited to the endpoint machine.

For malware that connects directly to a Deep Web URL, good sources of information from which to extract forensic artifacts are the following:

- Command-line arguments
- Recently installed files and folders
- Prefetch (.pf) files
- Network traffic logs

The following are the recommended steps for forensic analysis of Deep Web malware:

- *Isolate the scope and timeline:* Analysts should assess the situation first and determine where and when the breach started. Checking the logs on different security appliances is a good place to start looking for indicators of compromise.

  Identify infected machines and disconnect these from the network. Analysts should create an image of the memory dump of infected machines prior to analysis to avoid tampering with the timestamps in the infected machines.
- *Identify malicious processes:* To identify malicious processes, analysts should check for anomalies in the following process information:
  - Parent process
  - Security identifiers (SIDs)
  - File location
  - Executable name
  - Start time information
- *Analyse files and folders created or modified by malicious processes:* Analysts should watch for files dropped or downloaded after identifying malicious processes. Deep Web malware embed in their code or downloads from a C&C server the files needed to connect to the Deep Web network. Reviewing the files

may provide an insight into how the malware uses the Deep Web network.

- *Cross-reference findings with other forensic artifacts:* To strengthen the findings gathered, correlation of timestamps between different artifacts is strongly recommended. Analysing the prefetch file of the malicious process shows how many times it was run, when it was first launched, and when it was last executed.

  Command-line arguments issued by the malicious process may also give details as to whether the malicious process used Deep Web as a client or a server.

- *Examine network connections:* Multiple network connections found in the logs may provide hints of a Deep Web connection. Tagging the IP addresses to the Deep Web network is viable in proving a Deep Web connection. Analysts can take advantage of websites that log the IP addresses of machines connected to the Deep Web network. Taking note of the network ports used is also important since most Deep Web malware use default ports configured to connect to the Deep Web network.

7. CONCLUSION

Threat actors and cybercriminals strive to make their servers continuously operational. They come up with ways to achieve resilience and the Deep Web seems to be the top choice at this time. We have showed that, over time, more and more malware creators are embracing the Deep Web technology. Tor, I2P and Tor2web have been incorporated by malware creators into their binary code. Embedding the components of Deep Web in the malware binary or downloading it in a remote location are some of the examples we found used to make Deep Web work in the affected system.

These installations often leave digital footprints, which can be used by security professionals to get a clear picture of the Deep Web activity. This paper has presented techniques that can assist security professionals in analysing Deep Web activity from the affected system. Using these techniques, we can correlate the information gathered with the results of malware analysis to get a more accurate idea of how the Deep Web was utilized by the malware. Continuous monitoring of how threat actors operate in the Deep Web is needed to be able to come up with real-time solutions and counter future threats that utilize the Deep Web network.

REFERENCES

[1] Casenove, M., Miraglia, A. Botnet Over Tor: The Illusion of Hiding (2014).

[2] Brown, D. Resilient Botnet Command and Control with Tor. Defcon (2010).

[3] Melgarejo, A. 64-bit ZBOT leverages TOR, Improves Evasion Techniques. TrendLabs Security Intelligence Blog (2014).

[4] Trend Micro, Inc. TSPY_ZBOT.AAMV technical description. Threat Encyclopedia (2014).

[5] Pellerano, G. Tor2web: Exposing the Darknet on Internet (2013).

[6] Wikipedia. Elliptic Curve Diffie-Hellman.

[7] Trend Micro, Inc. TROJ_CRYPCTB.SME technical description. Threat Encyclopedia (2015).

[8] Remorin, L. Gameover: ZeuS with P2P Functionality Disrupted. TrendLabs Security Intelligence Blog (2014).

[9] Marcos, M. New DYRE Variant Hijacks Microsoft Outlook, Expands Targeted Banks. TrendLabs Security Intelligence Blog (2015).

[10] Trend Micro, Inc. TSPY_DYRE.YYYP technical description. Threat Encyclopedia (2015).

[11] Trend Micro, Inc. BAT_CRYPVAULT.A technical description. Threat Encyclopedia (2015).

[12] Trend Micro, Inc. TROJ_CRYPTESLA.A technical description. Threat Encyclopedia (2015).

[13] Tor Project. Tor on Android.

[14] So, C. BIFROSE Now More Evasive Through Tor, Used for Targeted Attack. TrendLabs Security Intelligence Blog (2014).

[15] Trend Micro, Inc. Banking Malware Vawtrak Now Uses Malicious Macros, Abuses Windows Powershell. TrendLabs Security Intelligence Blog (2015).

[16] Wikipedia. Luhn Algorithm.

[17] Sharwood, S. Silent Circle Shutters Email Service (2013).