Deanonymization in the onion network - a practical study

Ivaylo Ivanov
Scientific Writing - TU Wien

Abstract

As dangers to privacy and anonymity online increase in number and complexity, anonymity networks such as Tor are becoming more popular than ever. This paper aims to present a practical study to how deanonymization in the onion network is done. First, it will give the reader an overview to how onion routing works and introduce some key concepts. Afterwards, it will present theoretical techniques and real-life showcases for deanonymizing Tor users. Finally, advancements in the fields of Tor security and performance are introduced as well as several practical guidelines that Tor users can follow in order to avoid exposure of their identity.

1 Introduction

The Internet has made an unimaginable impact on the human lifestyle. It is now possible for people from different continents to connect and communicate in the matter of seconds without the delay that the classic post services provide, individuals and companies can sell their products to customers from around the world and activists have an enormous and far-reaching platform to find new supporters. This convenience comes at a cost. In an era where data is one of the most valuable goods that one can sell or obtain, internet surveillance and data gathering are on the rise - from malicious actors in the local coffee shop and your local internet service provider to large social networking sites and government entities.

Undoubtedly, one of the most influential technologies for protecting your anonymity is the Tor project [1]. In theory, Tor should be able to protect users against a specific way of internet surveillance in the face of traffic analysis in an easy and user-friendly manner, which is why it is used by millions around the globe for different purposes - from whistleblowers and investigative journalists who are working in oppressive states to illegal online shops. But is Tor alone enough for protecting your privacy and can you be deanonymized despite using the browser with the onion logo? The aim of this paper is to present attacks and techniques that can be, and in some cases have been, used against Tor users with the sole purpose to deanonymize them, what scientific advancements are being made in the field and practical guidelines that users can follow to avoid deanonymization.

2 A deeper look into onion routing

The information in this section is mainly based on the "Tor: The Second-Generation Onion Router" paper [1]. If any other sources have been used, they have been cited as such.

2.1 How Tor works

One of the most common techniques for internet surveillance is traffic analysis [2] which mainly consists of monitoring who is communicating with whom in a network and drawing patterns and conclusions about the users' behaviour, interests, personality and, in some cases, even personal data like real name, phone number and home address. The worst thing is that traffic analysis may happen despite encryption schemes employed by the communicating parties - the number of network packages and their source and destination are always visible to a watching third party. Traffic analysis may also be conducted by a service provider. For example - online providers may use your location or the number of times you visited their website to impose discrimination like pricing of products or limiting the functionality of the platform.

Avoiding this common and easy to use internet surveillance method is the aim of the Tor project. Tor (The Onion Router) is a Chaum Mix [3] anonymity system designed for low latencies. In order for the user to connect to the desired destination reliably, a couple of steps need to take place. This section will look into how a connection over the Tor network is normally made and what are the different components that are needed for establishing a connection.

The traffic generated from a user goes through a so-called circuit, which consists of multiple Tor nodes (relays), which are more often than not run by members of the Tor community.

[1] https://www.fsf.org/news/2010-free-software-awards-announced

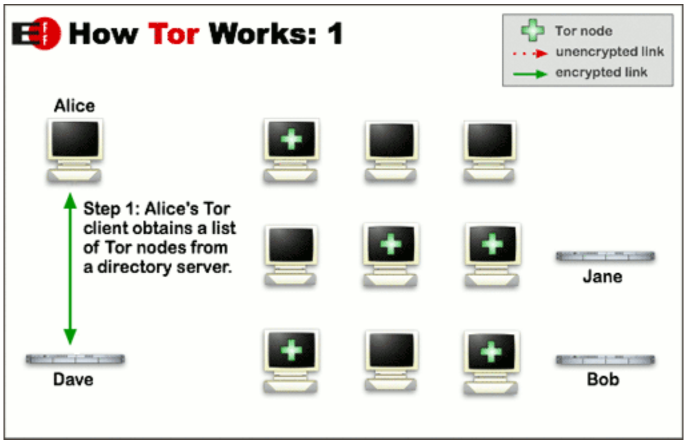
The circuit is built incrementally, starting from the client, and each relay in the circuit negotiates session details, such as encryption, with the next relay. Each circuit consists of a three types of relays:

- guard relay - this type of relay acts as an entry to the tor network. This is the only relay that the client directly connects to. It is the only relay that knows the source of the request that goes through the Tor network
- middle relay - the middle relay is the relay that is neither a guard nor an exit relay and is responsible for connecting the other two types. It has no notion of the real source and the real destination of the request of the client.
- exit relay - the relay where the request exits the Tor network. This relay only knows the destination of the request.

In order to build a circuit, the clients needs to choose the guard relay to use. This is done using a so-called directory server, which is a public centralized and trusted entity where all Tor relays are listed with their respective type. All the IP addresses of the Tor relays are thus public. Choosing a guard relay is the first step of constructing the connection and is shown in 1.

Figure 1: Querying a Tor directory server to find a full list of Tor relays

Source: torproject.org

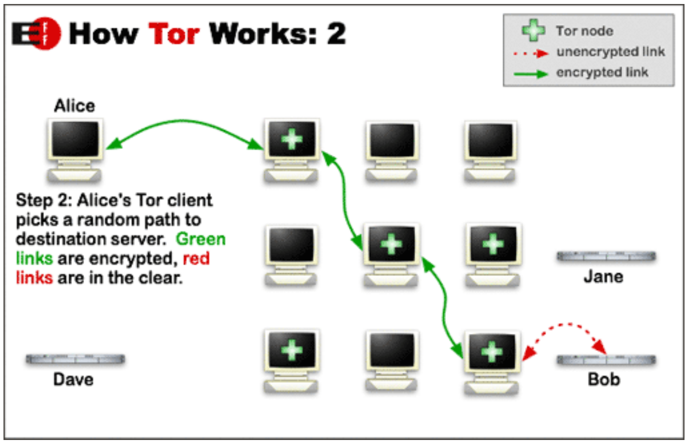

How Tor Works: 1
Tor node
unencrypted link
encrypted link
Alice
Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.
Jane
Dave
Bob

This is not the only function of the directory servers. As specified in the Tor Directory Protocol [4], the directory protocol are also responsible for reaching a consensus about the Tor relay list every hour. The network consensus document gets generated automatically by the directory servers and includes different relay descriptors [2] for each router that can later be used from the client for determining the circuit. Some examples are speed, whether the node is online or not and whether the node is misconfigured or not. The consensus documented gets downloaded and used by each Tor client and relay. This is also a way to monitor and control the presence of malicious Tor relays, but it is not perfect, as shown in a later section of this paper.

We mentioned previously that each Tor circuit is built incrementally. When the guard relay has been chosen, the clients connect to it using an encrypted channel. Afterwards, the client chooses the next hop of the mix (the middle relay), tells the guard node to connect to it and to negotiate new encryption keys to use for the communication. The same operation happens when the middle relay connects to the exit relay. After this operation, the exit relay connects with the destination of the request. Communication with this destination is normally not encrypted by Tor itself and may be in the clear, depending on the application-layer protocol used. This whole process can be seen in 2.

Figure 2: Establishing a three-hop Tor circuit

Source: torproject.org


How Tor Works: 2
Tor node
unencrypted link
encrypted link
Alice
Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.
Jane
Dave
Bob

As we already mentioned, the Tor circuit needs to be rotated from time to time in order to ensure the variety of the mix and so that the anonymity of the client is always preserved. Of course, this introduces an additional latency because the whole building process needs to be done for the new circuit as well. Rotating the circuit may help to overcome one of the limitations of the Tor network which is discussed in section 3.3.

2.2 Tor Hidden Services and Bridges

As we already mentioned in the previous subsection, a lot of the relays are run by the Tor community. Additionally, it is also possible to run the so-called Tor hidden services. Tor hidden services make it possible to run different services available only in the Tor network and without the need to re-

[2] https://metrics.torproject.org/collector.html#relay-descriptors

veal the IP address of the provider. As Tor works as a SOCKS proxy, the service can be any TCP service. Hidden services can also be hosted by anyone, regardless of whether or not they have a public IP address or are hidden behind NAT.

In order for clients to connect to a Tor hidden service, they need to go through a rendezvous process, which is where the client presents itself to the Tor hidden service and preparations for the initial connection are made. The rendezvous process will be covered in detail in Section 3.3.

As we mentioned previously, the addresses of all Tor relays are publicly available from directory services. This brings to the situation where ISPs or countries with active censorship are blocking requests to these addresses. In order to escape from this censorship, Tor bridges were created. Tor bridges work like normal Tor relays with the only difference that they are not listed in a directory server [5] and local ISPs cannot block them for this reason. This makes it possible for clients in countries with active censorship to use the Tor network without issues. The client only needs to get an address of a trusted Tor bridge and use it for setting up the connection.

2.3 Tor Limitations

The paper will focus on two limitations of Tor, which are relevant for the discussed deanonymization techniques. One of them is caused by the design of the protocol and the other is caused by how the Tor network is built.

Because of Tor's design and way of work, it is possible to attack Tor circuits with end-to-end attacks and Tor does not claim complete protection from them. Nothing stops a malicious party to observe both ends of the communication and conduct correlation attacks or to run a large number of relays in each layer in order to observe the whole circuit.

This issue can be tackled if the network of Tor relays is made large enough so that it would become infeasible for a malicious party to conduct the attacks. This is possible thanks to the Tor community and to the fact that relays can be started by anyone. After a look into the Tor metrics website [3] one can observe a positive trend which shows that the number of Tor relays has been steadily increasing. The graphic is visible in 3.

Unfortunately, the ease with which one can start a Tor relay, means that a lot of relays are running in technically inadequate environments, which hurt the speed and the bandwidth of the Tor network. This is one of the reasons why communicating over Tor may be a lot slower than communicating over the network normally. This is the second limitation of the Tor network which improves progressively each year. The trend can be seen on 4 and is again courtesy of the Tor metrics website.

Figure 3: Number of Tor relays and bridges

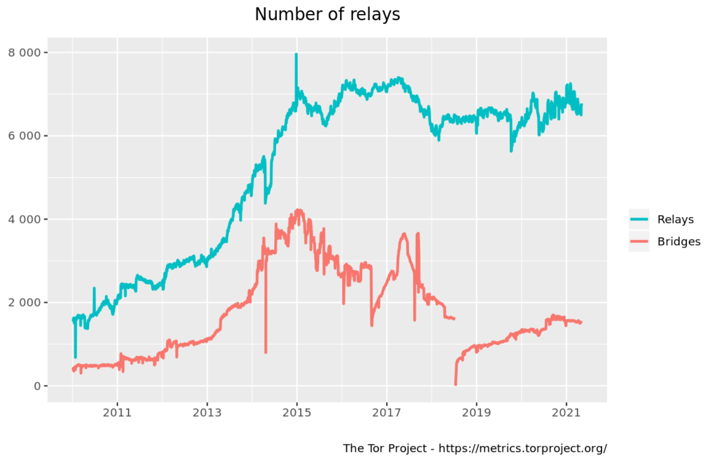
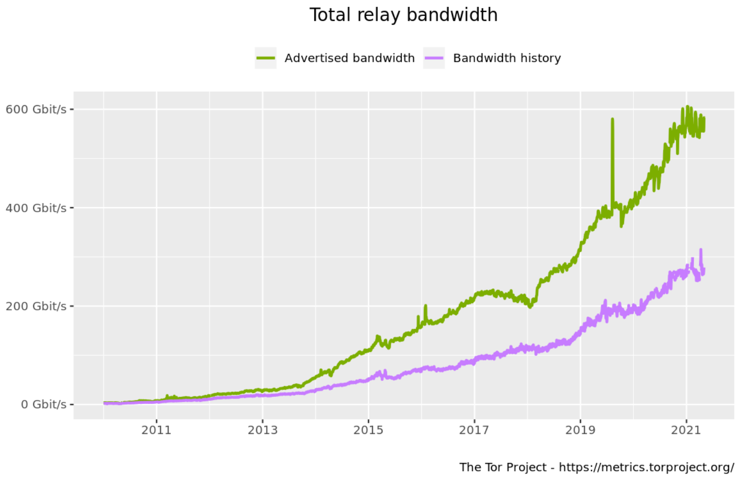


Figure 4: Tor total relay bandwidth



3 Deanonymization techniques and showcases

Because the main use-case is more often than not preserving anonymity, most of the attacks against Tor focus on identifying the communicating parties and their relationship. This process is known as deanonymization. In this section, we will go over some theoretical deanonymization techniques and some showcases about how Tor users got deanonymized in real life. For the purposes of describing the attacks, we will differentiate between them based on the following two factors:

- passive and active attacks - this attribute describes whether or not the attackers are passively observing the communication or actively trying to interfere

- single-end and end-to-end attacks - this attribute describes whether or not the attackers are attacking only one end of the Tor circuit or both

[3] https://metrics.torproject.org/

If possible, all of the attacks described in this section will be classified based on the attributes from above. Additionally, the reader has to keep in mind that even though the Tor relays undergo an admission process from a trusted central authority (as described in 2.1), the Tor relays are still semi-trusted, because their behaviour is not fully verified. This will be described in more detail in 4.1.

3.1 Bad Apple Attack

The Bad Apple Attack is one of the most well-known application-level attacks against Tor. This subsection will describe the attack based on the work by Le Blond et al. [6] In its core, the Bad Apple Attack can be viewed as an active correlation attack using Tor and another insecure application, in this case - BitTorrent. Before reading this section, it is recommended for the reader to know how BitTorrent works [7].

The attack consists of two parts - using the insecure application to reveal the source IP address of the Tor user and exploiting Tor itself to correlate the usage of an application running over Tor with the IP address revealed previously.

In order to execute the first type of attack, one must control a Tor exit relay for tracing the Tor users and a BitTorrent peer for receiving incoming connections. According to the study, a large number of Tor users use Tor only to connect to a centralized BitTorrent tracker and afterwards distribute content outside of Tor. This may be done for several reasons, for example increasing download speeds (the reader should keep in mind the overhead that onion routing brings and the total available bandwidth to the Tor network) or avoiding copyright infringement lawsuits which only use tracker subscriptions. BitTorrent traffic outside of Tor though makes it possible for a malicious party to trace the users' real IP addresses.

The second attack is using BitTorrent DHT (Distributed Hash Tables) tracking, which runs on UDP. Tor does not support UDP and even BitTorrent clients using Tor will fail to connect to the DHT tracker and revert to using the real public IP address and listening port of the user, which will be published to the DHT tracker. Additionally, each peer that wants to download specific content, receives a content identifier. This content identifier will also be published to the DHT tracker, alongside the IP-port pair of the user. The DHT information from above can be found from the BitTorrent client subscription to the centralized tracker and the BitTorrent handshake messages sent to the tracker. If the communication with the centralized tracker happens through Tor, a malicious exit relay can be used to identify the peer responsible for the content identifier and to collect all of the IP and port pairs for the users, connected to the peer. Then, the listening port is used to identify the user. The paper states in details why the listening port is a good identifier with a small number of false positives.

The Bad Apple Attack uses the described methods for tracing BitTorrent streams back to the source IP address in the same or different circuits (this paper will not go into details as to how streams in different circuits are attacked. An interested reader will find this in the in-depth description of the attack [6], which, as the researchers have shown, can be used to profile Tor users through traffic analysis [8] in real-life.

3.2 Deanonymizing users based on Bitcoin transactions

Bitcoin [9] is one of the most popular cryptocurrencies in the world right now (as per daily transaction rate according to the Blockchain Explorer[4]) and is used by many to execute online transactions in a pseudoanonymous fashion. A lot of services and websites support payments with the currency and it is especially popular with people running Tor hidden services. This section will show a passive, application-level correlation attack, discovered from Al Jawaheri et al. [10], that uses Bitcoin wallet addresses and website crawling in order to deanonymize users of Tor hidden services.

Before going into specifics, the paper will clarify what a Bitcoin wallet is. A Bitcoin wallet can be viewed as completely analogous to a regular wallet for storing money, with the only difference that it is digital and it stores Bitcoins. Each Bitcoin user needs a wallet in order to execute transactions and each wallet has a uniquely identifiable wallet address. Additionally, all transactions between two Bitcoin wallets are publicly accessible. This means, if a third party knows the addresses of the Bitcoin wallets of two other parties, it is possible to find out whether or not a transaction between these two parties has taken place. According to the sources, cited in the study, most of the transactions to Tor hidden services have been executed using Bitcoin. This is because anonymity is a top priority for Tor hidden service users and operators and Bitcoin brings a sense of anonymity (even though it has a pseudoanonymous model). Therefore, a lot of hidden services, like web shops, for example, actually have their Bitcoin wallet addresses listed on their websites. This made it possible for a scraper to be started, which scraped a list of Bitcoin wallet addresses from websites, running under onion addresses. Additionally, another scraper was started, which scraped Twitter [5] and the BitcoinTalk forums [6] for public Bitcoin addresses. Such addresses may be included in different posts, like asking for donations for charities. At the end, a wallet-closure analysis and a transaction analysis has been applied to the two datasets, revealing a set of users who have communicated with one or more of the hidden services, thus breaking their anonymity.

It is important to note that the attack is a side-channel attack and does not require any modifications to the network or usage of any non-public information. A similar approach to this attack has been used for deanonymization of a hidden

[4] https://www.blockchain.com/explorer

[5] https://twitter.com/

[6] https://bitcointalk.org/

service operator which will be described in a showcase in Section 3.6.

3.3 Circuit Fingerprinting

Circuit fingerprinting attacks are passive single-end attacks that allow deanonymization of both the client and the operator of a hidden service. This subsection will describe two attacks based on the work of Kwon et al. [11]

The first attack shows us whether or not a hidden service is running in a network and if a client is communicating with it. Remembering the Tor paper [1], we know that a hidden service sets up an introduction point (IP), where it is available for contact. When a client wants to connect to a hidden service, a rendezvous point (RP) gets set up, the client connects to it using a Tor circuit and calls the service through the IP and tells it the specific RP that has been set. The hidden service then builds a new circuit to the RP and communicates through it with the client. Thus, both the client and the hidden service operator should remain anonymous. The paper from Kwon et al. [11] argues that if a malicious party controls a guard relay, during the setup phase of the connection between the client and the hidden service, a specific pattern of requests may be observed, which will help fingerprinting the circuits between the client and the RP, the client and the IP, the hidden service and the IP and the hidden service and the RP. This makes it possible to determine whether or not a hidden service is running in the network. This paper will not go into details about the fingerprinting method. An interested reader will find the description in the paper from Kwon et al. [11]

The second attack from the paper is basic website fingerprinting that aims to deanonymize a user communicating with a hidden service. Previously, it was argued that website fingerprinting will not be effective in Tor, however the paper described how website fingerprinting for Tor hidden services is efficient and produces clean results because hidden services do not work using circuit multiplexing (when a client connects to multiple websites, a single circuit gets reused and the exit relay routes the requests. This is not the case with hidden services - for each client there is a different circuit). Each website is distinct in terms of content, design and programming. This allows a malicious entity to create a unique fingerprint of the website based on the size of the packets that get transmitted when connecting to the website, the duration of activity etc. This unique fingerprint may then be utilized to determine whether or not a client is visiting this website. When this technique is employed at a guard relay, this effectively breaks the user privacy as the adversary learns that the user has visited this hidden service.

3.4 Showcase: Harvard Bomb Threat

This showcase has been presented in the DEF CON 22 talk by Adrian Crenshaw [12]. Interested readers will find the information displayed here in a more detailed manner in addition to more showcases.

In December 2013, a fake bomb threat arrived at Harvard's student news paper and caused evacuation of the campus. The bomb threat came from a Guerilla Mail [7] temporary email address with the public IP address of a Tor exit relay (the addresses are listed in the directory server) in the headers (Guerilla Mail adds the originating IP to the headers in order to mark who sent the message). This showed the investigators that the user used Guerilla Mail and Tor to send the hoax bomb threat. They then decided to see whether or not somebody was using Tor at that time on the territory of the campus and found out that there was a connection to a Tor guard relay around the time when the bomb threat was sent. They found the person (Eldo Kim) who initiated the connection and after questioning, he admitted everything.

In conclusion, the deanonymization in this showcase happened because of a user error and a timing correlation attack. If Eldo used a Tor bridge, which are not publicly known, and was not the only person using Tor in a monitored network, he would not have been caught.

3.5 Showcase: Freedom Hosting

This showcase has been presented in the DEF CON 22 talk by Adrian Crenshaw [12]. Interested readers will find the information displayed here in a more detailed manner in addition to more showcases.

Freedom Hosting was a service which offered hosting of hidden services in the Tor network. Among other legitimate services, a lot of illegal services also got hosted on the Freedom Hosting machines and this inevitably attracted the attention of law enforcement agencies, in this case - the Federal Bureau of Investigation (FBI).

In July 2013, the FBI was able to compromise one of the servers of Freedom Hosting and managed to inject a malicious JavaScript code that exploited CVE-2013-1690 [8] which was present in the Mozilla Firefox version on which one of the latest Tor browser versions was based. The bug was already fixed in a newer version of the Tor browser, but not everybody has upgraded. The script made requests to FBI servers with identifying information like the public IP address and the MAC address of the clients. Thus, the FBI was able to deanonymize the users of the hidden service who were not doing updates in a timely manner.

To deanonymize the hidden service operator, the FBI used the server that they have seized - based on the IP address, they found out who the hosting provider was and managed to trace the payments records to a specific person - Eric Eoin Marques. When the police raided him, Marques dived for

[7] https://www.guerrillamail.com/

[8] https://www.mozilla.org/en-US/security/advisories/mfsa2013-53/

his laptop in order to turn it off, but failed to do so. This was later used as proof that he was the hidden service operator.

In conclusion, this was a single-end active attack that exploited a bug in the Tor browser in order to deanonymize Tor users. The deanonymization could have been avoided if the users updated the Tor browser in a timely manner. The deanonymization of the hidden service operator was done following public information and payment records of the hosting company. Had the operator not hosted illegal content of high interest from the authorities, used a privacy-oriented hosting provider and turned off his encrypted machine when not in use, the deanonymization may have been avoided.

3.6 Showcase: Silk Road

This showcase has been presented in the DEF CON 22 talk by Adrian Crenshaw [12]. Interested readers will find the information displayed here in a more detailed manner in addition to more showcases.

The last showcase in this section is about the owner and founder of an online black marketplace known as "The Silk Road". "The Silk Road" operated in a fashion similar to conventional online marketplaces but offered illegal products and services ranging from weapons and drugs to fake personal documents, stolen bank information, hacking and hitman services. Because of the nature of its content and the alleged 1.2 billion dollars in revenue, the hidden service gained the attention of the authorities (again FBI) who managed to track down the hidden service operator, going by the name "Dread Pirate Roberts", after two years of investigation.

The first thing the authorities did was to search for the first time the hidden service was mentioned in the public internet. They finally managed to find a post on a drug-oriented forum, posted by a user with the handle "altoid", in which was described how to access the website. Additionally, they found a post on BitcoinTalk forums posted by the same user and in the same advertising fashion. This was when they decided to investigate the post history of the user in these forums and found a request for "an IT pro in the Bitcoin community", where a real Gmail [9] address, belonging to Ross Ulbricht, was listed. After further investigation, the FBI saw similarities between the economic views of Ross Ulbricht , which he posted on his social media account, and the ones of "Dread Pirate Roberts", which were available on "The Silk Road". Afterwards, the FBI found another post, this time on StackOverflow [10], where a user with the name "Ross Ulbricht" (afterwards changed to "frosty") asked for help with a piece of PHP code for connecting to a Tor hidden service. This was not enough information to warrant a court order, but certainly made Ross look like a suspect and FBI started monitoring him closely. They discovered that connections to the servers, hosting the hidden servers, were made in a coffee shop in a close proximity near where Ross lived. The connections occurred at the same time when he was using his personal Gmail account. Eventually, the FBI managed to seize one of the machines and found that one of the public SSH keys belonged to "frosty" and a portion of the StackOverflow code mentioned above. In July 2013, the US Customs intercepted an order of nine fake identity documents, all bearing Ulbricht's picture and all having different names. This information was enough to bring Ross for questioning, where he denied ordering the IDs, but told the authorities that hypothetically anyone could order such documents from "The Silk Road". The FBI then questioned his roommates who knew him as "Josh" and read the personal messages of "Dread Pirate Roberts", which showed interest in fake identity documents. Eventually, the FBI managed to arrest Ross in a public library right after he had entered the password to his computer, which revealed that he was in fact "Dread Pirate Roberts".

In conclusion, the deanonymization in this showcase was possible not because of missing Tor usage but despite it. Even though Ross used hidden services for the website, he leaked information on other channels, which made him a suspect and subsequently landed him in prison. Had he not used the same identities on all websites, not connected to the host from a place near his home, while simultaneously using his personal email account and not talked about interests, he would not have been caught.

4 Avoiding deanonymization

As the reader may have already seen, Tor offers excellent protection of ones privacy and anonymity when used correctly. All the showcases from above and most of the described attacks show that the deanonymization of Tor users happened not because of design and implementation failures in Tor itself, but because of information leaks on other channels or correlation attacks. Nevertheless, the "Bad Apple Attack" and the circuit fingerprinting attacks describe what may happen in case the attackers have started malicious Tor relays. In this section, the paper will analyze what scientific advancements are being done in the field and specifically how to detect malicious relays, how Tor bandwidth usage can be decreased so that Tor can become suitable for everyday usage and general guidelines for how to avoid being deanonymized by information leaks and correlation attacks.

4.1 Detecting spoiled onions

This subsection will go into detail on how malicious Tor relays have been detected and classified in the past. In the paper

[9] https://mail.google.com/

[10] https://stackoverflow.com/

from Winter et al. [13], a way of detecting malicious Tor relays which are running active attacks against the connections of the Tor circuit is described. As already mentioned in 2.1, directory servers are responsible for assigning different descriptors to the Tor relays. This makes it possible to mark bad exit relays, which should be avoided by the clients. According to the paper from Winter et al. [13] in order for a relay to be marked as malicious, a suspected relay needs to be communicated to the project, the reported attack gets reproduced and if it can be verified, then two of the nine directory administrators manually blacklist the relay using the `AuthDirBadExit` descriptor. This does not render the relays useless as they can continue functioning as guard and middle relays. This manual reporting and verification process is cumbersome. For this reason, the paper presents two tools - the first is called `exitmap`[11] and can do active scanning and detect Man-in-the-Middle attacks against HTTPS, XMPP, IMAPS, SSH and DNS as well as TLS prevention using `sslstrip`[12], carried out by exit relays, and the second - `HoneyConnector`[13] that can be used for establishing bait connections over a Tor circuit with IMAP and FTP credentials in them to identify sniffing exit relays. The implementation of the scanners is out of the scope of this paper and is left to the interested reader.

After monitoring all (at the time) ~950 Tor exit relays over a period of several months in 2014, the researchers were able to identify and cluster ~65 malicious exit relays which were directly reported to the Tor Project. The overlap between the datasets obtained from `exitmap` and `HoneyConnector` consisted only of two hosts - one in Taiwan that sniffed IMAP credentials and injected HTML code and another in Hong Kong that ran `sslstrip` and sniffed FTP credentials.

Using `exitmap` the researchers detected several Russian exit relays that HTTPS MitM attacks and shared the same root certificate authority. They are thus suspected to be run by the same person. In addition, they were also running in the same IP network and had the same Tor version running.

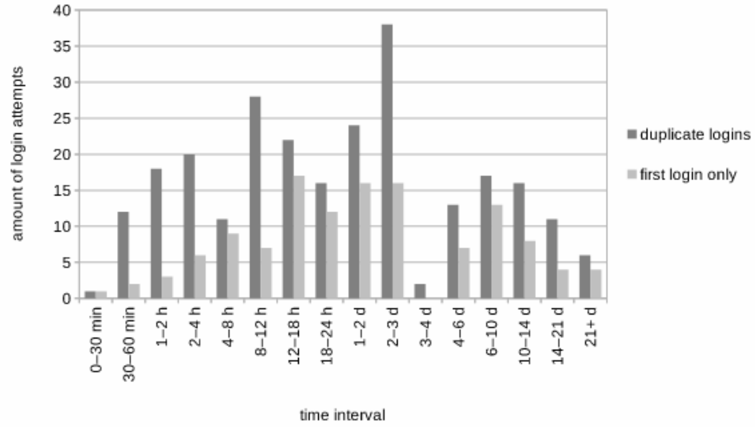
Using a controlled host and `HoneyConnector`, the researchers were able to detect exit relays that sniffed credentials and group them in two groups: "International Sniffer Group" and "Indian Sniffer Group". All the hosts from the latter were running behind IP addresses of the same Indian ISP, with the same bandwidth and the same Tor version running on different versions of Microsoft Windows (7 and Vista). The probability of getting redirected to these nodes was very low because of frequently rotating IP addresses (bad uptime statistics) and low bandwidth. It is more interesting to see who, how and after how much time reused the bait credentials that were being submitted over `HoneyConnector`. The graphic in Figure 5 shows how much time after submitting the honeypot credentials the first login attempt was made, as well as if there were any duplicate logins afterwards. It is visible that most of the connections occurred after a significant amount of time, which in combination with the client fingerprinting and the human errors described in the paper leads to the conclusion that most of the login attempts were carried out manually. In addition, several exit relay operators managed to deanonymize themselves by accessing the controlled host not through Tor but by using the Google Chrome browser, which revealed their public IP address.

The scanners described above cannot scale arbitrarily and thus another more generalized defense mechanism needed to be implemented. Such a defense mechanism is also proposed in the paper and includes a patch of the deprecated Torbutton browser extension. An interested reader will find more information in the paper.

What the findings in the paper entail for the normal Tor users is a question that Philipp Winter answers in a blog post about the topic [14]. He argues that the malicious relays are a fraction of the whole number of relays at that time and it is safe to argue that it this is even truer nowadays when keeping in mind the data in Figure 3. Additionally, the data from the paper suggests that regular nodes offer more bandwidth than malicious ones and are thus more likely to be picked in the circuit initialization. Finally, even if the client chooses a malicious relay as an exit of the circuit, many of the MitM attacks described in the paper will trigger a warning window in the Tor browser, which requires manual input from the user before connecting.

Figure 5: Time interval between connection with honey credentials and login attempts

Source: Spoiled Onions: Exposing Malicious Tor Exit Relays by Winter et al. [13]



4.2 Securely scaling Tor

As the reader may remember from 2.1, running Tor relays around the globe is done by volunteers. Tor benefits a lot from this, as it makes the network truly decentralized, but there may

[11] https://github.com/NullHypothesis/exitmap
[12] https://tools.kali.org/information-gathering/sslstrip
[13] https://github.com/mmulazzani/HoneyConnector

also be disadvantages such as low network bandwidth [15]. This is viewed as one of the main pain points in adopting Tor for everyday usage [15] and may lead to users unintentionally leaking information and deanonymizing themselves in a manner, similar to the one described in 3.6. Additionally, low network bandwidth in the Tor network makes it possible for DoS attacks like the ones described by Jansen et al. [16] to be conducted. These issues can be tackled by the recent scientific research by Komlo et al. [17], which proposes a way to improve Tor network performance without any negative effect on the anonymity and privacy of the users.

As described in 2.1, Tor clients and relays need to download the consensus document from the directory servers in order to keep the list with Tor relays up to date. This approach protects the users against path-based attacks, because malicious adversaries cannot manipulate the users' relay selection and each user has the same view of the network [17]. The protocol proposed in the paper removes the necessity for each Tor client to download the consensus document while still maintaining these security properties. This reduces the bandwidth consumed with service information and thus improves Tor's speed.

For the improvements to work according to the paper [17], the researchers propose two data structures for encoding network directory information: the authenticated network directory document is called ENDIVE (*Efficient Network Directory with Individually Verifiable Entries*) and the entries in ENDIVE are called SNIPs (*Separable Network Index Proof*), where each SNIP represents a relay and the ENDIVE is a complete set of all SNIPs that gets agreed upon once every epoch (similar to the way a network consensus is reached).

It is important to note that only relays need to download ENDIVEs - first the entire ENDIVE at bootstrap and afterwards only the changes once every epoch. Clients do not need to have the whole map of the network and thus only download a network parameters document once per epoch.

The researchers also differentiate between traditional relay status entries[14] and SNIPs [17]. A SNIP contains the same information as a relay entry with three additional fields:

- an index range - described in the paper as "a range of integer values whose size is proportional to the desired probability of selecting this relay"
- an authentication tag - calculated from a directory server over the content of a SNIP. Because of this tag, clients can validate SNIPs and build circuits without having the signed ENDIVE.
- two timestamps - one that indicates when the SNIP was generated and one that indicates when the SNIP expires.

This paper will not go into details about the fields. An interested reader will find more information in the paper by Komlo et al. [17].

After the brief familiarization with the new data structures, this paper will describe how paths with oblivious route selection ("Telescoping Walking Onions") are built, according to the "Walking Onions" paper [17]. Let R_n be the last relay in the client's current circuit and R_{n+1} be the next relay the circuit will be expanded to. Instead of selecting the next relay directly, as done in vanilla onion routing, the client selects a random integer i in a range which is available in ENDIVE. Then, the client sends the random integer, alongside with the client's half of the circuit extension handshake message, to R_n. The relay R_n is responsible for querying the ENDIVE for a SNIP, which contains the index i in its ranges. After such a SNIP has been found, R_n initiates a circuit extension request and relays the client's handshake to the relay R_{n+1}, described by the SNIP. After R_{n+1} responds, R_n relays the response to the client, alongside with additional service information for verification. The client then verifies the SNIP of new node, checks if the election process was honest (if i is in the index range of SNIP) and uses the public keys in the SNIP to authenticate the handshake response from R_{n+1}.

In addition to "Telescoping Walking Onions", another circuit extension protocol is presented. "Single-Pass Walking Onions" is out of the scope of this paper, as it loosens forward secrecy for path selection. An interested reader will find more information in the paper by Komlo et al. [17].

The performance improvements of the circuit extension protocols are visible in Figure 6 (the vertical line represents the size of the Tor network at the time of writing; the letter in the brackets represents the authentication method, more information in the paper [17]). It is visible that vanilla circuit extension uses a lot more bandwidth than the improved variants. According to the paper [17], at the network scale at the time of writing and at 10 times the network scale at the time of writing, relays with Walking Onions use 4-6× and 24-41× less bandwidth than the ones using vanilla onion routing respectively. It is also worth noting that the client bandwidth usage is almost constant with Walking Onions, regardless of the protocol variant and the authentication method used.

The consequences this paper entails for Tor users are obvious: massive improvement in Tor's speed and bandwidth consumption for service information without disregarding the existing security and privacy model of Tor. At the time of writing, active work is being done in order to implement Walking Onions in the Tor specifications[15].

4.3 General guidelines

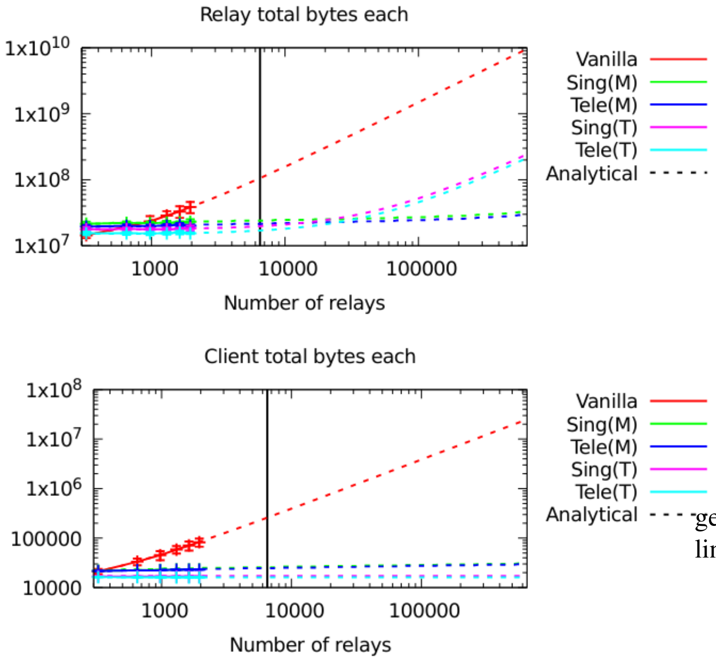
Apart from the analysis and technical advancements performed in the previous sections, Tor users themselves can

[14] https://stem.torproject.org/api/descriptor/router_status_entry.html

[15] https://lists.torproject.org/pipermail/tor-dev/2020-March/014178.html

Figure 6: Per-epoch total bytes usage from relays and clients

Source: Walking Onions: Scaling Anonymity Networks while Protecting Users by Komlo et al. [17]



follow some general guidelines in order to protect their anonymity and privacy online. These guidelines will be briefly presented in this subsection and are based on the previous sections of this paper and the talk from Adrian Crenshaw at DEF CON 22 [12].

As the reader may have already seen from Section 3 of this paper, deanonymization mainly happens not because of Tor usage, but despite it with the main way for deanonymizing the user being a correlation attack or an information leak. Thus, the following list represents several precautions that Tor users should take to avoid having their identity exposed:

- do not participate in illegal activities. This point is (hopefully) self-explanatory
- use bridges whenever possible. As the reader may remember from 2.1, bridges are not listed in the Tor directory servers and connections to them are hard to be blocked by a central entity (like the infamous "Great Firewall of China"[16])
- do not be the only person using Tor in a monitored network. This can be used to perform a correlation attack, as presented in 3.4. The user should instead opt for unmonitored networks, use a VPN to a trusted site and run all traffic over the tunnel or just use a bridge as mentioned in the previous point in the list
- do not leak personal information and keep online identities separate (use different usernames and emails, access different accounts from different locations). Using Tor is pointless if the users leak personal information that can later be used to correlate them to their usage. The showcase presented in 3.6 is a prime example of how not to handle your online identity with focus on anonymity
- do not forget that Bitcoin is pseudoanonymous. Bitcoin is a really popular way for doing transactions among Tor users. The user should not forget that all of the transactions are public and that the address of a Bitcoin wallet can be used for mapping browsing and shopping activity. This has also been presented in 3.2
- if the user has an encrypted machine, the user should leave it in a powered-down state when not in use. Thus, the information on the machine cannot be accessed even if physical access has been obtained (presented in 3.5)

In addition to the guidelines from above, which are more general, users should also follow these tool-specific guidelines:

- always have the latest version of the Tor browser installed. This will protect the users from malicious entities exploiting zero-day vulnerabilities like the one presented in 3.5 and may also bring performance improvements like the "Walking Onions" protocol.
- encrypt wherever and whenever possible. Non-encrypted traffic will most certainly be a subject to sniffing, spoofing and injection attacks which may be used for deanonymization of Tor users. Tor users should not forget that the traffic between the exit relay and the destination is not encrypted by default (as seen in 2.1) and another method of encryption (such as HTTPS) is needed
- do not ignore the warnings of the Tor browser. The browser has been configured with sensible defaults with the users' anonymity and privacy in mind and the users should not ignore any warnings that it gives them. Such warnings may include updates and HTTPS errors among others
- only open links from the Tor browser in the Tor browser itself and use Tor consistently. Doing otherwise may link your real IP address to your Tor usage, as presented in 4.1

The users can cover all of the tool-specific guidelines above by using Tails OS [17], which is a Live USB OS specifically designed for Tor usage. When a user wants to use Tor, he/she just boots a machine from a USB stick with Tails on it and all of the traffic from that machine gets routed through Tor.

[16] https://blog.torproject.org/closer-look-great-firewall-china

[17] https://tails.boum.org/

The only available browser is the Tor browser with defaults, focused on the privacy and anonymity of the users. After the machine gets turned off, no trails are left by Tails OS.

5 Conclusion

Tor, the low-latency anonymity network, has become one of the most important and popular tools for keeping ones anonymity and privacy online with users ranging from whistleblowers to journalists and political activists. As such, there is a lot of interest in deanonymizing Tor users from a lot of different parties and even though such attempts have been made, most of the deanonymization cases are a result of incorrect Tor usage and side-channel information leaks or correlation attacks.

Interest in Tor is not only one-sided. Scientific advancements and improvements have been made in order to make Tor more secure, faster and more suitable for day-to-day use. With large academic interest in the topic and fast responses from the authors of the Tor project itself, it is safe to say that Tor development will not stop any time soon.

Apart from these improvements, each user can follow a number of guidelines to protect themselves from deanonymization. These guidelines range from more general ones like changing geographic locations and keeping online identities separate to using operating systems specifically built for Tor.

List of Figures

References

[1] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *USENIX Symposium on Operating System Design and Implementation (OSDI)*, 2004. https://www.usenix.org/legacy/publications/library/proceedings/sec04/tech/full_papers/dingledine/dingledine_html/index.html.

[2] Jay Stanley and Barry Steinhardt. Even Bigger, Even Weaker: The Emerging Surveillance Society: Where Are We Now? https://www.aclu.org/sites/default/files/field_document/bigger_weaker.pdf, 2007.

[3] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), 1981.

[4] Tor Directory Protocol, version 3. https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt.

[5] Tor: Bridges. https://2019.www.torproject.org/docs/bridges.

[6] Stevens Le Blond, Pere Manils, Chaabane Abdelberi, Mohamed Ali Kaafar, and Claude Castelluccia. One Bad Apple Spoils the Bunch:Exploiting P2P Applications to Trace and Profile Tor Users. In *USENIX Symposium on Operating System Design and Implementation (OSDI)*, 2011. https://www.usenix.org/legacy/events/leet11/tech/full_papers/LeBlond.pdf.

[7] Bram Cohen. The BitTorrent Protocol Specification. https://www.bittorrent.org/beps/bep_0003.html, 2008.

[8] Piotr Zieliński and Steven J. Murdoch. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In Borisov N. and Golle P., editors, *Privacy Enhancing Technologies. PET 2007. Lecture Notes in Computer Science, vol 4776.*, 2007. https://link.springer.com/chapter/10.1007/978-3-540-75551-7_11.

[9] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf, 2008.

[10] Husam Basil Al Jawaheri, Mashael Al Sabah, Yazan Boshmaf, and Aiman Erbad. Deanonymizing Tor hidden service users through Bitcoin transactions analysis. *Elsevier Computers & Security*, 89, 2020.

[11] Albert Kwon, Mashael AlSabah, David Lazar, Mark Dacier, and Srinivas Devadas. Circuit fingerprinting attacks: Passive deanonymization of tor hidden services. In *24th USENIX Security Symposium*, 2015. https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-kwon.pdf.

[12] Adrian Crenshaw. DEF CON 22 - Adrian Crenshaw - Dropping Docs on Darknets: How People Got Caught. https://www.youtube.com/watch?v=eQ2OZKitRwc, 2014.

[13] Philipp Winter, Richard Köwer, Martin Mulazzani, Markus Huber, Sebastian Schrittwieser, Stefan Lindskog, and Edgar Weippl. Spoiled Onions:Exposing Malicious Tor Exit Relays. In *Privacy Enhancing Technologies*, 2014.

[14] Philipp Winter. What the "Spoiled Onions" paper means for Tor users. https://blog.torproject.org/what-spoiled-onions-paper-means-tor-users, 2014.

[15] Mike Perry. Tor's Open Research Topics: 2018 Edition. https://blog.torproject.org/tors-open-research-topics-2018-edition, 2018.

[16] Rob Jansen, Tavish Vaida, and Micah Sherr. Point Break: A Study of Bandwidth Denial-of-Service Attacks against Tor. In *28th USENIX Security Symposium*, 2019. https://www.usenix.org/system/files/sec19-jansen.pdf.

[17] Chelsea H. Komlo, Nick Mathewson, and Ian Goldberg. Walking Onions: Scaling Anonymity Networks while Protecting Users. In *29th USENIX Security Symposium*, 2020. https://www.usenix.org/system/files/sec20-komlo.pdf.