UNIVERSITY OF APPLIED SCIENCES

Department of Computer Science

Seminar IT-Sicherheit

TOR

Term Paper

Submitted on :

17 November 2015

Submitted by:
Torben Zurhelle (inf101482)

Consultant:
Prof. Dr. Gerd Beuster
Fachhochschule Wedel
Feldstraße 143
22880 Wedel
Phone: (041 03) 80 48 - 38
E-mail: gb@fh-wedel.de

This article analyses the structure of Tor network and its effectiveness to provide anonymity for its users. To achieve this, the protocols which are used by the Tor network to provide anonymity to the client and the complete architecture of the network are described. In addition to that, the method and infrastructure servers use to communicate anonymously through the Tor network is explained. Finally we look at the main attacks the Tor network is vulnerable to.

Contents

List of Figures

Dictionary

DARPA	The Defense Advanced Research Projects Agency is a agency of the US Department of Defence, which does research for the US Forces. Their main accomplishments in the IT are the developement of GPS, TCP/IP and ARPANET, the first version of the Internet.
Entry node	The Tor node which the client uses to enters the Tor network when communicating through the Tor network.
Evil nodes	In this context evil nodes are Tor clients which want to collect data about users and intercept communications which are transmitted through the Tor network.
Exit node	The last Tor node in a Tor communication chain which has to reveal itself to the targeted server to send the request.
ISP	Internet Service Provider are companies which connect end users to the internet by supplying the hardware infrastructure.
Onion router	Software component of the Tor client in charge of building a route of Tor nodes, a message is sent through and preparing the message so that every Tor node only gets the information it requires for its task. (chapter 2.3)
Onion proxies	A software component of the Tor client which relays messages from other Tor clients, fetches directories and builds connections though the Tor network.
Tor button	A component of the Tor browser which blocks all browser plug-ins that build a separate connection to the targeted website to provide their services. These plug-ins cancel the effect of the Tor network as their connection is not build using the Tor protocols.
Tor client	This a computer which has the Tor software installed and uses a browser to communicate through the Tor Network.
Tor network	This is a sub network of the TCP/IP network consisting of directory servers and Tor nodes which provide the infrastructure for Tor communication.
Tor nodes	This are Tor users who allow their computers to be used as relays for Tor communications.

1 Introduction

Probably everybody who is interested in cyber-privacy and security has heard about the Tor network. This report will show to what extent the Tor network is able to provide anonymity in the internet. This will be achieved thought detailed description on the architecture and protocols used within Tor.

1.1 Definition

"The Tor Project" [RD04] is the organisation founded by the original creators of Tor to support and update the protocol and the core software components to use Tor. They have defined Tor as follows: "The Tor network is a group of volunteer-operated servers that allows people to improve their privacy and security on the Internet. Tor's users employ this network by connecting through a series of virtual tunnels rather than making a direct connection, thus allowing both organizations and individuals to share information over public networks without compromising their privacy. Along the same line, Tor is an effective censorship circumvention tool, allowing its users to reach otherwise blocked destinations or content." [RD04]

1.2 Why is Tor important?

Traditional online communication is designed to provide fast, quite often reliable communication through the network. The early networking research was done under the assumption that the network communication would only be done by a small select group of people, like universities and governments. Therefore the focus was not as much on privacy and security as it should have been. As a result of that each transmitted package has a header file which is always plain text and contains a lot of data like the IP-addresses of the sender and the receiver, the type of communication and so on.

Everybody who can read the package, intended by the sender or through illegal actions, gets a lot of information about both the sender and the reciver of the message [KR08]. Encryption can only provide anonymity to a certain extend, as only the message itself is encrypted, but not the header, providing the reader with a huge amount of information about the computers communicating. To prevent that, Tor hides both the sender and receiver throughout the communication [RD04].

1.3 History

The main concept of Tor networks was developed and designed at the United States naval research Laboratory in 1995. Researcher developed a secure sub-network on the internet to protect US intelligence communication. The project were continued 1997 by the US research facility DARPA under the codename Onion routing [LAHR10].

In December 2006 the research-education nonprofit organization responsible for maintaining Tor was founded and turned the project into a civil project which provides the required software to join the tor network. Since then have been funded through sponsors. The most famous known sponsors are the U.S. International Broadcasting Bureau, Internews, Human Rights Watch, the University of Cambridge, Google, and Netherlands-based Stichting NLnet [Dun08].

2 Architecture

This chapter describes the protocols used for the network communication. Starting with the base concept of the communication relays followed by the actual protocol of the communication, which implemented the onion routing.ending with the description of individual components of the Tor Network like directory server bridges and entry guards.

2.1 Relays

The simplest solution of enforcing anonymity in digital communication is to simply pass the communication through a relay. This is how other anonymity services, like the Anonymizer [Cot95] or Hola[OV11] do it. They provide relay servers where users can pass their communication through to hide themselves from the server they communicate with. This method of communication has one weak spot.

Lets take the common example: Alice wants to communicate with Bob. They use Ray's server as a relay (See Figure 2.1). If Mallory intersects one of the messages, for example the one sent from Bob, he would not be able to know that Bob is talking to Alice as it looks like Bob is talking to Ray. Since Ray is providing this service to many people at the same time, Mallory will have a hard time finding the outgoing message to Alice and connecting it to the incoming message from Bob.

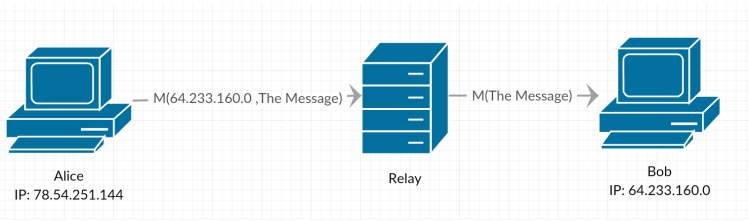

Figure 2.1: How relays work

The weakness of this method is Ray. Both parties have to trust Ray, that he is just passing the data on as he promised them. The problem is that both cannot be certain that he is not recording the communication, reading or selling it. As most relays are hosted by commercial companies, it makes it even more likely that they would do such things. So this method of "Secure" communication has same issues.

The Tor developer used this concept of relays as a basis. To raise the level of security they built a communication channel of at least 3 relays. This makes it impossible for any relay to know both communication partners.

2.2 Network Architecture

The idea of relays (see chapter 2.1) was the starting point of the development of the Tor network. Tor uses multiple relays for its communication, but instead of sending the packages through servers, it uses the computers of other users as relays. They are called Tor nodes. This method is more secure, as it uses a larger number of relays, which decreases the power of each relay. Additionally they are mostly privately owned by people with the same intention: communicating anonymously on the internet.

The structure of Tor communication is best shown by comparing it with the traditional way of communicating on the internet. Figure 2.2 shows the traditional method a browser uses to request data from a server. None of the message transmissions are shown as encrypted. This is because only 30 percent of the traditional internet communication is encrypted[KR08] and even then only the messages are encrypted(see chapter 1.2 paragraph 2), so they were neglected.

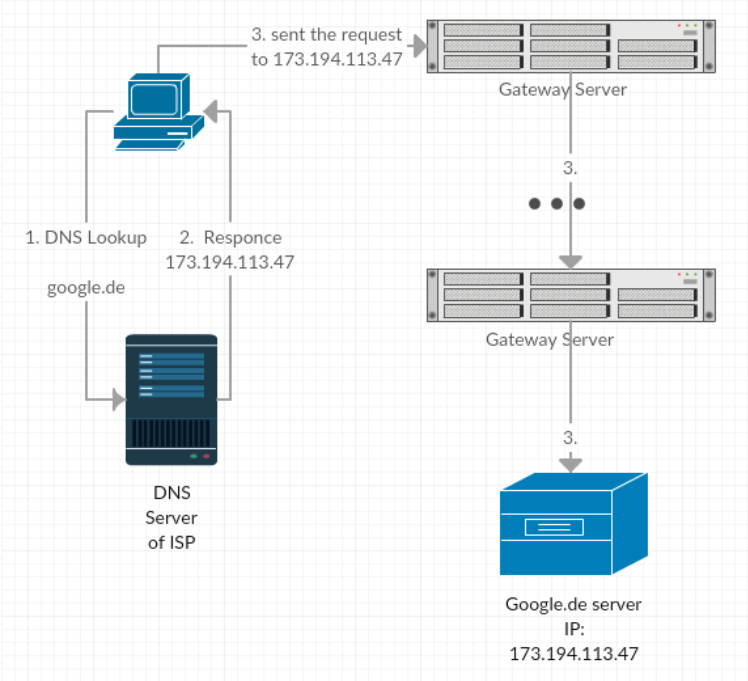

Figure 2.2: The standard procedure for requesting data from a website

The Tor network is an overlay network, as it is build on the TCP/IP Network. The figure 2.3 illustrates the way Tor browser request data from a web server. At the moment Tor browser is started, he registers at the directory server (See chapter 2.4for more details) to receive the list of all Tor-nodes how are currently active on the network(Tor service descriptor). The Tor service descriptor contains information about every active Tor-node like the IP-address and public key.[Mat15]

After the user requests information from a server, for example by entering the URL into the browser, the browser will contact a DNS server to get the IP address of the target server. This is similar to the traditional protocols. The difference is that the DNS server is selected by the Tor client and part of the Tor network. The Tor client has the list hardcoded. To prevent the DNS server from collecting information about an users activity, the Tor client will change the DNS server after a short, randomly generated time, so that the knowledge of the DNS request is spread out about several servers. [Mat15]

The main part of the Tor protocol is the routing. The onion-router uses the Tor service descriptor to build a route through more then three Tor nodes. It uses this route to prepare the message to be

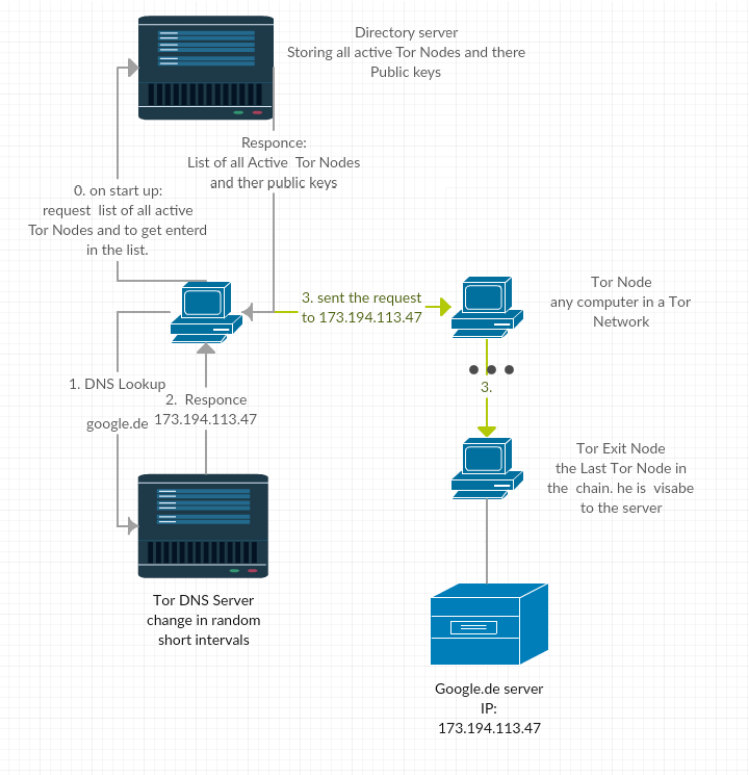

Figure 2.3: How Tor communicates with a server

sent using onion-routing (see Chapter 2.3). Finally the packages are sent to the first Tor node on the routing list, which will send them to the next node and so on. Every Tor node in the network only knows the node they received the message from (as tit has to send a reply back to it) and the node they have to send the message to.

The communication within the Tor network is encrypted, which prevents any tracking of the path. The last node in the chain has to communicate with the target server. This Tor node is called the exit node. The target server is not part of the Tor network, so the exit node has to transmit the packages to the Server using a common communication protocol like http, https, ftp, ftps and et cetera. The exit node will identify itself to the target server as requester of the information. For the server it is as if it had requested the data and the real requester is hidden. [Hen11]

The exit-nodes are not chosen from the whole Tor service descriptor. They could get problems , for example by requesting information which is illegal in its country. Therefore the owners of an exit node has to know how to deal with lawsuits. To prevent the exit nodes from raising to much attention and to prevent attacks, which try to reveal the path of the sent package, the Tor client is configured to frequently generate new random routes, which have different subsets of nodes from the Tor service descriptor.

2.3 Routing (Onion Routing)

The routing mechanism, which Tor uses, was invented by David Chaum under the name Chaum Mixes. The adoption for the Tor network were named Onion routing. The Onion Router is implemented on a TCP/IP Network. This implementation provides real-time communication without the common issue of packet-based networks, which is the waste of constant allocation of storage space. Each Tor node maintains a TLS connection to every other Tor node in the network. One part of a Tor node is a sub process which is called onion proxies. Its task is to fetch directories, establish connections across the network and handle connections from user applications. A onion proxy accepts TCP streams and multiplexes them across the network.

The Tor nodes stores two keys, the long term identity key and the onion key. The identity key's task is to sign TLS certificates and to encrypt the node log. This is a file containing data, like the summary of keys, addresses, bandwidths, and exit policies, the node has used. The Tor nodes, which are also directory servers, also use this key to sign the Tor service descriptor. The second key is called the onion key.It is used to decrypt requests by other Tor nodes to build a connection. This key is rotated periodically and independently between the nodes to prevent a compromisation of the key [Hen11].

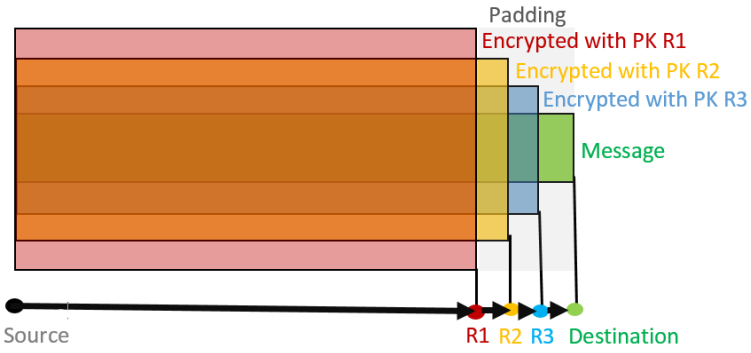

Figure 2.4: Showing the change of the transmitted data through the onion routing process (R are the relays the message is passing through and PK means primary key)

Onion routers use the network of onion proxies to sent their messages. They generate a virtual path through several different nodes (at least 3) (see chapter 2.1). The message, the onion router has to send, and a cover (a cover is a kind message header and footer, which is descried later in this chapter) is then encrypted with the public key of the last Tor node in the virtual path. The resulting message receives a new cover and is encrypted with the public key of the second last Tor node on the virtual path and so on. The onion router knows the key from the Tor service descriptor. Then the message is sent to the first node in the on the virtual path. This node decrypts the multiple encrypted to receive the last added cover and the message which is still several times encrypted with the other covers. This cover contains the information the node needs to relay the message to the next node or send a reply to the previous node. This information has an expiration time which states how long the node should store the data to be used again (t_n) and the address of the node, where the message has to be sent next(Y) [RD05]. Two keys are also sent, one key to encrypt the replay message (K_{bx}) and the other to encrypt the message on the reply with(K_{fx}). As Tor allows different encryption algorithms to be used, they have to be defined in the message together with the related keys (algorithm for K_{bx} is F_{bx} and for K_{fx} is F_{fx}) [Mat15].

There is one big and obvious flaw in Onion Routing. Since every node removes a cover, the message is shrinking over the course of transmission. This property makes the message trackable and allows people intersecting the message determine how many relays it has already passed and still needs to

pass. This is prevented with right padding which means, that the relay adds random bits to the right side of the message to keep the size constant.

To simplify the previous explanation here is the function of the Onion Routing(O) from x to z over y, where PK_y is the function for the encryption with the public key of y.

$$O_{x,y,z} = t_n, Y, F_{fx}, K_{fx}, F_{bx}, K_{bx}, PK_y(O_{x,y,z}), Pad \tag{2.1}$$

Figure 2.5 illustrates how equation 2.1 is implemented for a message which is sent through 3 Tor nodes to a target server.

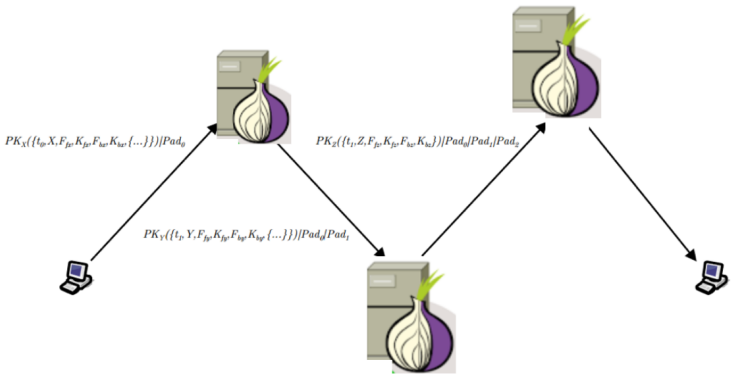

Figure 2.5: Implementation of Onion Routing by sending messages through 3 Tor nodes to a server

2.4 Directory Server

One of the main components of a Tor network are the directory servers. They are a small group of redundant, well known onion routers. They track changes in the network topology and node states. The directory server acts as a HTTP server. Tor nodes have the IP addresses and the keys of every directory server hardcoded. When a Tor node is started, it registers at every directory server to get the Tor service descriptor. Afterwards it will periodically continue requesting the directory servers for an updated version of the Tor service descriptor. The directory server regularly builds the Tor service descriptor from the received information and its own opinion of the structure of the network. This list contains the IP address, public key, the bandwidth and more information about each active node in the Tor network. The Tor node needs them to build its routing lists. Every time the Tor service descriptor is built, it is signed and sent to all active nodes so that they can update their version. To maintain security and prevent intrusions, the directory only accepts data from nodes where it already knows the public key. In the event of a communication attempt by an unrecognised key,the messages are sent to the directory server administrator for approval. After receiving approval, the node id and public key is saved on the server and the directory server will allow the next request on its own [Mat15].

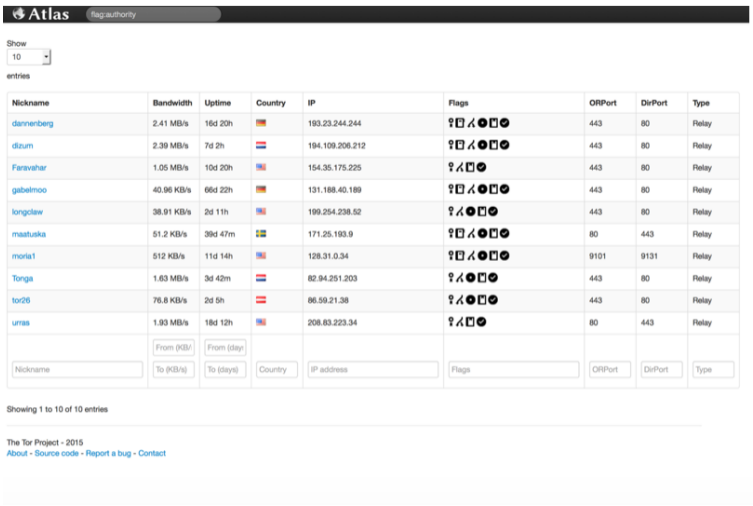

Figure 2.6: The list of all active directory servers[RD04]

Currently there are 10 directory servers set up(see Figure 2.6). All servers must be redundant and synchronised with each other, to be usable. To prevent manipulation in a directory server, the Tor nodes should only trust data which is approved and signed by multiple directory servers. In the unlikely case that all directory server are offline all active nodes have to use the last Tor service descriptor and new nodes have to wait until one server is back online. The structure of the directory server is modelled after mixminion, a protocol invented for the type III anonymous remailer. The protocol is defined as follows: ” First, we make the simplifying assumption that all participants agree on the set of directory servers. Second, while mixminion needs to predict node behavior ”(Tor only needs a threshold consensus of the current state of the network) ” Third, we assume that we can fall back to the human administrators to discover and resolve problems when a consensus directory cannot be reached” [GDM02].

2.5 Bridges

Tor provides anonymity and untraceable communication but these two factors are good reasons for certain governments to prevent users from accessing the Tor network by blocking the connections to the Tor nodes, which are easy as they just need to block all active nodes in the Tor network. They just have to start a Tor node and use the Tor service descriptor they receive from a directory server. If a Tor client cannot connect to any other Tor node, he cannot use the Tor network.

This issue was solved by Tor through bridges. Bridges are just unpublished relays. They are treated differently from other Tor nodes in only one point:instead of being included in the Tor service descriptor, they are stored in a secret, unpublished list. They do not show up on the Tor service descriptor, so it is hard for governments or ISP's to block them. There are two official methods to receive bridges: one method is through the Tor bridges website [asn15a]. On the website the user has to pass a captcha to receive three IP addresses and public keys of bridges. This low number makes it difficult to obtain the IP- addresses of large numbers of bridges. The other method is to write an encrypted E-mail from a trusted E-mail service, like Gmail, with the header "get bridges" to `bridges@torproject.org` which will result in a reply email containing three IP addresses of bridges and their public keys. This information can then be entered into the Tor client. The user should enter several bridges into their client to prevent traceability [asn15a].

2.6 Entry Guards

Entry guards are a fairly new component of the Tor network, as they were introduced in May 2006 to increase the anonymity of Tor users. This technological concept was so strong that only 4 days after the release just 206 clients were using a version which did not include this technology.

Entry guards solve the problem of evil Tor nodes. Tor has a huge problem with their relays. Since relays are donated, the intentions of the people providing them is not easy to identify. The issue is when one provider runs both the entry and the exit point of a communication, the provider can identify the IP address of the client and the targeted server through attacks like message tagging or traffic confirmation attack. This scenario is not as unlikely as it sounds, since nodes with a higher bandwidth are selected preferentially for the communication and the route through the Tor network changes often.

The task of entry guards is to reduce the possibility of compromisation and reaching an evil node in the network. The Tor client generates a list containing Tor nodes (guard list). At the time of selection of a guard for the list , an expiry date is randomly generated for this guard, which lies between 30 and 60 days. When a guard expires, it is dropped from the guard list and a new Tor node is chosen. The guard list is used every time the Tor client is building a connection. Instead of choosing an Tor node from the Tor service descriptor, the client will now randomly choose a Tor node from the guard list [NB08]. This improves the anonymity of the clients as long as there are no evil nodes in the guards list. The integrity is maintained until the next guard expires. On the other hand, the probability of communicating insecurely is higher, as long as there is at least one evil node in the guards list. Nevertheless research from Overlier and Syverson proves that even with this risk the guards list produces more secure connections, than would be generated without it [OS06]. This paper was the reason that the entry guards were included into the Tor client.

The core part of the entry guard protocol is the selection of the entry guards. The first part of the protocol is called guard flag algorithm. The guard flag algorithm is executed inside the directory server to mark Tor nodes inside the Tor service descriptor, which are suitable as entry guards. An Tor node should fulfil at least two criteria to be flagged with a guard flag.

The most important one is that these Tor nodes should have a great up time, as the Tor client can only work with active notes. In the case that a guard list only has one active node, all inactive nodes will be dropped instantaneously from the guards list and new nodes will be chosen. This action should be performed as few times as possible, to keep a healthy guards list in action as long as possible [TE06].

The second criteria is the node bandwidth. To reduce bottlenecks within the network, the directory server only flags nodes which have a bandwidth above a certain threshold.this is generally the median of the bandwidth of all active notes, but at least a bandwidth which is above 250 KB/s [TE06]. The issue of using the current bandwidth as a measure is that it is constantly changing and is therefore unreliable. To counteract this, the bandwith is translated into weighted bandwidth units (WBU). WBU is a points system which analyses the bandwidth of a node over a longer time period.

The issue with these criteria is that they have to be finely tuned. If the criteria are to sensitive, then there are not enough nodes in the guards list to ensure a regular changing of the entry node and so bottlenecks could raise, as the amount of communication gets divided through too few nodes. On the other hand a long guards list raises the possibility of evil nodes in the guards list [OS06].

The directory server will rarely give nodes the guard flag which are marked as possible exit nodes. These nodes already have a special role within the Tor network and their bandwidth should not be wasted for tasks, the majority of nodes are also able to perform [TE06]. If a client has to select a new guard, he checks the Tor service descriptor node by node, to find nodes which have guard flags.From these list all nodes gets removed which, are already in the guard list, have the same /16 IP Block as the client or any of the guards already in the guard list get. Finally the guards list will be filled up by randomly selected guards which have passed the previously mentioned checks [Mat15].

In general, exit bandwidth is protected by assuring that relays with the exit flag are chosen in the exit position more often than in other roles. In particular guards that are also exits will find themselves used more often as exits and less often as guards. This design choice has implications,which we will discuss later on [TE06].

3

Tor and the DarkNet

Tor is not only used to hide your normal Internet activity. 3.5% of the total traffic on the Tor network is on hidden services on the DarkNet. [asn15b] Dark-Net is the technical term of the part of the Internet which is not accessible through search engines. This definition is quite general. This report only talks about the parts of the DarkNet which are not accessible with normal browsers as the hosts are so obsessed with privacy and anonymity that the servers enforce the protection of Tor networks.

3.1 How to travel through the DarkNet

The previous chapter only discussed the case where the user wanted privacy. But what if the service provider wanted to hide its servers on the internet for privacy or anonymity reasons. For these cases Tor provides a new type of URL which ends on ".onion". The user finds the addresses on the normal internet. For example WikiLeaks uses an onion site to communicate and receive documents from their sources. To make this possible, they published the following URL on their website: `http://suw74isz7wqzpmgu.onion/`.This URL is only reachable through the Tor network.

3.2 .onion sites

In the previous Chapters only describe the event that the client want to hide himself from the Server and to hide the how conversation from the outside world. What if the server wants to hide himself. URL and other Top-Level Domain Names use the DNS infrastructure which makes them locatable and publish the identity of the Host[JA15] . Tor provides a solution for this problems the ".onion"domains. The .onion domains have end to end encryption as well as secure, anonymised services, which obscure the identity and location of the server from the client. Tor Onion sites are designed to work without central controlling authorities. This is achieved through non-central routing and service publication. Therefore .onion names cannot be registered, transferred, assigned or revoked. The owner of a .onion side proves his ownership just through access to the combination of the private and public asymmetric encryption keys that were used in the algorithm to generate the .onion name[JA15].

.onion domain names are self-authenticating, as they are derived from the public key used by the the server during the establishment of the connection with the user. This fact means that the 16 character long cryptographic section of the .onion name is not meaningful to humans [JA15]. Nevertheless they still have to follow DNS syntax to be expressed as DNS implementations.

3.3 Implementation of .onion sites

The first step people have to do when setting up a .onion side is to calculate a set of public and private asymmetric cryptographic encryption keys. The hidden service can use the public key generated and a list of randomly chosen Tor nodes from the service descriptor. These nodes are used by the service as introduction points. From the introduction point and the public key the hidden service descriptor is made. The hidden service descriptor is then signed with the private key to prevent manipulation. Finally the hidden service descriptor is uploaded to a distributed hash table. The "distributed hash table" is a sub service of the directory servers.

User can learn about .onion sites through the normal internet. Organisations like WikiLeaks [oni] or DuckDuckGo [oni06] publish their hidden services on their normal websites. Other services are published through forums and onion lists. Onion lists are websites on the normal internet, which just contain a list of different onion sites. Most of the time they are all groups of similar websites. The user enters the onion domain name in his browser which has to be connected to the Tor network. The Tor client inside the browser connects to the distributed hash tables to look up the hash of the website, which contains of the first 16 characters of the domain name followed by ".onion". The server which stores the hash replies with the public key and a list of entry points for the requested service [Unk14].

The Tor Client uses the ISAAC[Jen94] cryptographic random number generator, which is the number generation algorithm always used in Tor, to select a random Tor node from Tor service descriptor(rendezvous point)[Mat15]. The Tor client creates an invitation message and adds the address of the rendezvous point and a self generated one time secret [Unk14]. The invitation message is now encrypted with the public key of the target service, and sent to one on the introduction points of the target service. The introduction point will forward the message to the target service [Unk14].

Since the message is encrypted, it is ensured that only the system which set up the Tor domain name can read the message. Then the receiver decrypts the message. Using the newly gained information, the service can now send a message to the rendezvous point to identify itself. The identification is done by using the just received one time secret. As the rendezvous point is sending the onetime secret to the Tor-client, he knows that the connection to the right server has been established [Unk14]. Neither the Tor client nor service know the location, IP address or any other information about each other. This is because both connections to the rendezvous point are established through the Tor network, so they are routed through at least 3 Tor nodes and are end to end encrypted as described in chapter 2.2.

4 Weaknesses

Over the years Tor has proven not to be as secure as it has promised to be and several attacks were developed to weaken the Tor network. The main attackers and opponents of the Tor network are also some of its main supporters and helped develop the first versions. The US government on one hand supports the project but on the other hand it also tries to find weaknesses through the NSA to collect more data about users and the secret information they want to hide. There are also other threats which are attacking the Tor network, like other governments, criminal organisations, organisations trying to prevent exchange through the DarkNet and organisations trying to listen in on the communication to make money from it. This chapter will describe some of the strongest attacks against Tor.

4.1 Browser plug-ins

The browser coming with the Tor client software package does not come with browser plug-ins and Tor advices their user not to install them. Many browser plug-ins like Flash, Java, and ActiveX controls build up a second connection to the server to provide their services. These connections do not use the Tor protocol so they are direct unencrypted connections, which destroy the anonymity gained with using the Tor network [RD04]. Tor's solution is to include a Tor button, which is a tool in the browser to block the communication of plug-ins. Until March 2014 it was still possible to turn this feature on and off in the browser [Mat15]. Through this way the most effective attacks happened, like Practical Onion Hacking from FortConsult [crt06] and the FBI's operation Torpedo which used the popular hacker tool Metasploit to build infected Flash code and published it on popular DarkNet porn sides to collect the identity of thousands of users.[Pou14]

4.2 Fingerprint Attacks

Fingerprint attacks try to identify a users movement through the internet. The attackers place scripts on web services which try to collect data from requests that could identify a user. This information which identify Tor users is called fingerprint.

One example of such a fingerprint is the system time. The time of the users system can be used to track users, since most protocols require timestamps from the client. Through these timestamps the systemclocks settings can be calculated. Systhemclocks are not running synchrone, they vary in terms of nanoseconds.

The attacker collects these fingerprints and can determine the users activity on the website. As the first paper about this kind of attack [AKD05] was published a couple of months ago, there is no fix implemented jet. The experts from the Tor project [RD04] have analysed the threat and raised the concerns that the threat might not be defended against, as the main source of fingerprints is the browser canvas. This is the reason the attack is also called Canvas Fingerprint attack. The browser

canvas draws the graphical website, without this tool most websites would be unusable or atleast would be quite strangely represented in the browser.

4.3 Memex

The previous attack types targeted the user of the Tor network, but there are also exploits which target the DarkNet infrastructure. DARPA, the research facility where the basis of the service provided by the tor project was developed, is currently working on Memex [Cla14]. This is a search engine like Google, Yahoo and Bing which differs in the fact that it covers around 85 percent of internet. Traditional search engine only cover about 0,004 percent [KR08]. Memex also searches through the DarkNet and indexes this information for US government agencies who use this data to hunt criminals. DARPA has published code fragments on git[WtMpt15] which shows that they are able to find, index and search through Tor sides. The exact functionality of Memex is not known. The development team made it clear that they are not targeting to identify Tor users as they themselves built to provide anonymity in the Internet. Last year over 50 percent of all arrest warrants related to human and drug trafficking contained the term Memex in their evidence list [Whi14].

5 Conclusion

The Tor project developed a network which should provide anonymity for its users. The technologies which are used in encrypting and hiding the communication trough the use of multiple relays, Onion Routing and directory servers are concepts which have been designed to provide a high extent of anonymity for the client and hiding of the communication through their architecture.

Tor does not only provide anonymity, but has also developed a mechanism whereby servers can provide their information and services anonymously on the internet. This is a problem for a lot of governments as the criminal activity in the DarkNet is high. Therefore government and other organisations are constantly trying to develop attacks against the network. The majority of these attacks only works if the client uses tools with penetrate the Tor network.

The Tor project tries to provide a product which allows anonymity of communication on the internet. Most of the times they succede with this task, but as Tor uses the infrastructure and hardware architecture of the Internet, which was not designed to provide anonymity,they reach sometimes the limits of the architecture which results in vulnerabilities, but " The Tor project" who manage the open source project are constantly working with the community to close them. The fact that the project was mostly developed and is still supported by the US government, which tries through the NSA and other agencies to track and collect data from the users in the internet, does not improve trust. On the other side Tor is an open source project and has therefore published all of its programming structure, which has been checked by independant sources.

Bibliography

[AKD05] David Lazar Marc Dacier Albert Kwon, Mashael AlSabah and Srinivas Devadas. Circuit fingerprinting attacks: Passive deanonymization of tor hidden services. *24th USENIX Security Symposium*, aug 2005.

[asn15a] asn. Bridge db, https://bridges.torproject.org/bridges, feb 2015.

[asn15b] asn. Some statistics about onions, https://blog.torproject.org/blog/some-statistics-about-onions, feb 2015.

[Cla14] Liat Clark. darpa-memex-human-trafficking. *wired*, feb 2014.

[Cot95] Lance Cottrell. Anonymizer, http://anonymizer.com, jan 1995.

[crt06] Andrew christensen and Fortconsult reasurch team. Practical onion hacking finding real addresses in the tor network. *Wired magazine*, oct 2006.

[Dun08] Meredith Hogban Dunn. *Tax Form Tor*. Department of the Treasury internal Revenue Service, nov 2008.

[GDM02] David Hopwood George Danezis, Roger Dingledine and Nick Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. *Defcon10*, jul 2002. Cambridge University <george.danezis@cl.cam.ac.uk>The Free Haven Project <arma,nickm@freehaven.net>.

[Hen11] Lance Henderson. *Tor And the Dark Art of Anonymity*. Amazon Distribution GmBH, Leipzig, 2011.

[JA15] A. Muffett J. Appelbaum. The .onion special-use domain name. Technical report, Internet Engineering Task Force The Tor Project, Inc., Facebook, oct 2015.

[Jen94] Bob Jenkins. Isaac: a fast cryptographic random number generator. Technical report, Oracle,Microsoft, apr 1994.

[KR08] James F. Kurose and Keith W. Ross. *Computernetzwerke - Der Top-Down-Ansatz*. Pearson Deutschland GmbH, München, 4. edition, 2008.

[LAHR10] Michael Ligh, Steven Adair, Blake Hartstein, and Matthew Richard. *Malware Analyst's Cookbook and DVD - Tools and Techniques for Fighting Malicious Code*. John Wiley and Sons, New York, 2010.

[Mat15] Nick Mathewson. Git reposetory of the tor project, https://gitweb.torproject.org/, oct 2015.

[NB08] Ian Goldberg Nikita Borisov. *Privacy Enhancing Technologies*. Springer, eighth edition, jul 2008. 8th International Symposium, PETS 2008 Leuven, Belgium, July 23-25, 2008 Proceedings.

[oni] Wikileaks:tor, https://wikileaks.org/wiki/wikileaks:tor.

[oni06] the-duckduckgo-hidden-service, https://duck.co/forum/thread/1762/is-the-duckduckgo-hidden-service-legitimate, may 2006.

[OS06] L. Overlier and P. Syverson. *Privacy Enhancing Technologies, Locating Hidden Server*. Springer, first edition, may 2006. In Proceedings of the 2006 IEEE Symposium on Security and Privacy.

[OV11] Derry Shribman Ofer Vilenski. Hola!, https://hola.org/, sep 2011.

[Pou14] Kevin Poulsen. The fbi used the web's favorite hacking tool to unmask tor users. *Wired*, dec 2014.

[RD04] The Tor Project inc. Roger Dingledine, Nick Mathewson. Tor overview, https://www.torproject.org/, aug 2004.

[RD05] Paul Syverson Roger Dingledine, Nick Mathewson. Tor: The second-generation onion router. Technical report, The Free Haven Project and Naval Research Lab, oct 2005.

[TE06] Mashael AlSabah Roger Dingledine Ian Goldberg Tariq Elahi, Kevin Bauer. Changing of the guards. *The Tor Project, Inc and University of Waterloo*, 2006.

[Unk14] Unknown. Tor rendezvous specification. Technical report, The Tor Project, https://gitweb.torproject.org/torspec.git/plain/rend-spec.txt, 2014.

[Whi14] Dr. Christopher White. Memex. *Darpa Broad Agency Announcement*, feb 2014.

[WtMpt15] Dr Chris White and the Memex project team. Darpa memex project vagrant vm. Technical report, Darpa, jan 2015.