Tor and Onion Services

Rushit Shah
Concordia Institute for Information Systems Engineering (CIISE)
Concordia University
Montreal, Canada
rushit.shah@mail.concordia.ca

Nabhanyu Halgeri
Concordia Institute for Information Systems Engineering (CIISE)
Concordia University
Montreal, Canada
n_halger@live.concordia.ca

Alka Singh Rathour
Concordia Institute for Information Systems Engineering (CIISE)
Concordia University
Montreal, Canada
al_ratho@live.concordia.ca

***Abstract*— The Onion Router, also known as TOR, is a free and open-source software program that routes internet traffic through a series of volunteer operated servers to provide anonymity and privacy to its users. The servers that are used are known as "nodes" or "relays." With the help of Tor, it becomes difficult for anyone to track a user's online activities. However, despite the security and privacy Tor provides, several attacks have taken place on Tor in recent years. In this paper, we discuss the different types of attacks that have been carried out, along with the strategy and motivation behind them. Further, an attack on Tor is implemented, and the vulnerable areas through which the Tor network can be compromised are analyzed.**

Keywords—tor, attacks, network, routers

I. INTRODUCTION

Smart Tor is a free and open source programmed that enables users to browse the internet anonymously and securely to give online anonymity. The acronym "Tor" stands for "The Onion Router. Tor uses a method of multi layered to masquerade the traffic and utilize its own encryption to encrypt the traffic as it travels through different nodes and servers to it final destination. The traffics is difficult to track since each node in the network only knows the previous node and the following node where it is going in the chain.

Tor also uses bridges which acts as nodes but works for access to the Tor network and get around censorship, users use relays called "Tor bridges." Relays known as bridges are not publicly mentioned in the Tor directory, making it more difficult to stop them. Without their internet service provider or government being able to tell that they are using Tor, users in blocked areas can utilize a bridge to connect to the Tor network. The hidden services of Tor helps in hosting of website and services with anonymity and have an ending of .onion, furthermore they are only available and reachable within tor network. Even though the Tor offers and has a high level of digital anonymity it is not impenetrable and invulnerable to attacks that could be exploited.

Onion Router, or Tor, is a free and open-source software project that promises to give its users internet anonymity and privacy. As part of a project to safeguard official communications, the United States Naval Research Laboratory (NRL) created it in the middle of the 1990s. Later, it was made publicly available in 2002. A decentralized network of computers that enables anonymous internet access, was formally introduced in 2004. Almost 7,000 volunteer-run machines, often referred to as nodes or relays, are included in the network and are dispersed across the globe. It is challenging to pinpoint the origin of the data back to the user because a user's data is routed through several of these nodes when they join the Tor network, it is difficult to trace the origin of the data back to the user.

Tor Onion Services, originally known as Hidden Services, were disclosed by the Tor project in 2004. It makes it difficult to pinpoint the location of the server or the owner by enabling the anonymous hosting of web services throughout the Tor network. Onion Services, a pseudo-top-level domain that can only be accessed through the Tor network, can be reached using an onion address.

II. WHAT IS TOR ONION SERVICE?

This Internet users look for methods to anonymize their network data due to growing privacy and security concerns. The Tor system was created by the Pentagon as a research based. In the information age, it has evolved into an instrument for preserving freedom of speech and privacy [5]. The Tor Project has a negative reputation despite having good intentions. The expansion of Tor and the anonymity it offers have made the network a haven for illegal activity known as the "Dark Web," much like any big, growing city draws criminals [6]. A well-known illustration of a hidden service is Silk Road, a drug-selling website that the FBI took down in 2013 [5]. Ross Ulbricht, the site's administrator, was detained on suspicion of being "Dread Pirate Roberts," the site's fictitious founder, and he was given a life term.

A crucial resource for people who respect their online privacy and security is the Tor network. Users are looking for solutions to prevent their personal information from being used by hackers and other harmful actors considering the rising frequency of internet security breaches and data dumps. Users can hide their online identities and make it difficult for others to follow their digital footprints by using the Tor network.

Despite its acceptance and usefulness, the Tor network has come under fire and developed a bad reputation because of its links to illegal activity on the "Black Web." Like any large, developing metropolis that draws criminals, the Tor network's anonymity has spawned an underground market for illicit activity [6]. It is important to remember that the majority of people who use the Tor network do so for legal reasons, such as to access prohibited information and preserve their online privacy.

We shall give a thorough analysis of the Tor network in this essay, covering its development, history, and services. We'll

look at the various network attacks that have been made as well as the tactics hackers have employed to infiltrate the Tor network. We'll also go through current studies on strengthening Tor's security and show how a traffic analysis attack may really be implemented on the network.
Users, developers, and researchers may improve the security and privacy of the Tor network by using the knowledge gathered from this study. We hope that by providing a greater knowledge of the challenges and opportunities that the Tor network presents, this article will assist users in making decisions that will protect their online privacy and security.

III. Onion Routing/ Service

In this part we talk about the onion service, which is a network built on a low-latency onion-routing architecture, where traffic is forwarded through arbitrarily chosen Onion Routers (ORs), encrypting data with numerous onion skins to preserve unlikability [7].
In this situation, an OR can also be referred to as a relay, node, or simply a router. Each stream can be telescopically silently channeled through the network, which means that each router only knows the relays that come before and after it [8]. The source of the stream is only known to the first relay, the entry point. The only relay that is aware of the client's location is the last relay, the exit node. Only encrypted information is exchanged between the onion router(s) [9]. Symmetric cryptography is used to layer-encrypt data, which is then unwrapped by a relay before being forwarded to the following relay in the chain [104]. Typically, a circuit has three relays.

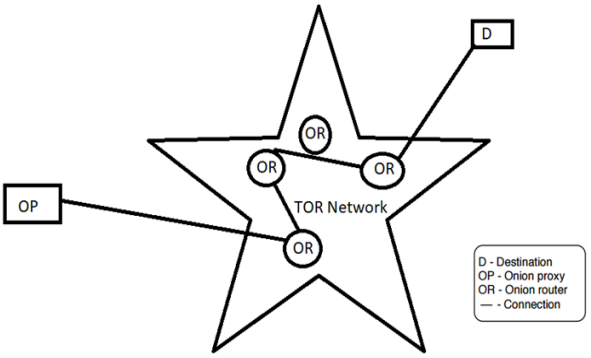


Figure 1: Onion Routing

The low-latency onion-routing architecture used by the Tor network allows traffic to be passed through arbitrary Onion Routers (ORs). To maintain unlikability, this routing method encrypts data using many onion skins.
A relay, node, or straightforward router can all be used to describe any one of the network's onion routers. Each router in the network is only aware of the relays that come before and behind it as the traffic stream is silently funneled through the system [8]. Just the first relay, which serves as the entrance point, is aware of the stream's source. Only the final relay, the exit node, is aware of the location of the client. The onion routers only communicate using encrypted data [9].
Symmetric cryptography is used to layer-encrypt the data, ensuring its confidentiality and anonymity, which is then unwrapped by a relay before being passed to the next relay in the chain [10]. A circuit in the Tor network typically consists of three relays, ensuring that the data is sent through the network in an anonymous and safe manner.
In essence, onion routing is a fundamental component of the Tor network, which guarantees anonymity and privacy to its users. Data is securely and anonymously routed through the network because to the low-latency onion-routing architecture of the network, in which each relay only knows about the relays that come before and after it. Layer-encrypting data with symmetric cryptography adds an extra layer of protection by preventing data from being intercepted by attackers.

IV. Attacks on Tor

The biggest network for anonymous contact is called Tor. Recent papers debate the efficacy of Tor and discuss the flaws in the Onion Router design. De-anonymizing attacks are increasingly utilizing these weaknesses. The need for hybrid attacks that can be used at the network layer, protocol layer, or application layer has increased as attacks have become more complicated and potent over time. We'll talk about public Tor attacks and group them into categories for additional investigation. Some ethical flaws have also been created by Tor's freedom and privacy principles. The network's cover attracts criminal activity, which has damaged its image.
A majority of Tor attacks concentrate on figuring out which client and server are utilizing the Tor network for communication [10]. De-anonymization is the procedure in question [11]. A circuit between the client and an exit node has been established in the Tor network, and the exit node is in communication with the host. A hidden service is being provided under a pseudonym, and the attacker wants to connect that pseudonym to the operator's real identity, either directly or through an intermediary step (such as a physical location or IP address) [12] [13]. The attacker also wants to confirm that the client and the server are communicating.
Users using Tor may potentially be the victim of social engineering attacks like phishing emails or texts, which deceive recipients into disclosing their identity or jeopardizing their security. For instance, a hacker could send a message that appears to be from a reliable source, like a friend or a Tor developer, and encourage the user to divulge personal information or download a malicious file [6]. This kind of attack might be challenging to recognize and has the potential to jeopardize the security of the entire Tor network.
The security and privacy of the Tor network are being constantly enhanced by researchers and developers in order to allay these worries. This entails creating fresh methods for encrypting and safeguarding data as well as finding and fixing Tor software flaws [7]. Also, initiatives are being made to inform Tor users of the dangers of the network and to offer advice on the best ways to be anonymous and private online.
Attacks on the Tor network continue to be a major worry despite these measures. Attackers will probably keep coming up with new and more advanced ways to compromise the network's security as its popularity

continues to rise. In order to maintain the anonymity and privacy of Tor users, it is crucial to keep looking into and creating new methods, as well as to be always on the lookout for new threats to the network.

Attacks can be divided into seven categories according to their strategy and objective -

- Correlation Attacks End-to-end - Passive Attack
- Congestion Attacks End-to-end - Active Attack
- Timing Attacks End-to-end - Active Attack
- Fingerprinting Attacks Single-end - Passive Attack [14]
- Denial of Service Attacks Single-end - Active Attack
- Supportive Attacks
- Revealing Hidden Services Attacks

A. Correlation attacks

The correlation attack uses the packet timing to link the network flows. The packet flows, packet sizes and its timing help to correlate the network flows which can help in breaking the anonymity in anonymous communication. As a result, the attacker can conduct attacks over TOR as he finds the correlation between the entry and exit nodes to confirm that a communication is taking place between the client and the server.

[28] helps to clearly understand the way in which attackers use the flow correlation method. If we consider a Tor network with M ingress rows and N egress rows, the relation between them cannot be determined because of the use of encryption and onion routing. This helps a user to stay anonymous and carry out its activities. But to de-anonymize the network's activity, attacker inspects the packet contents and identifies the associated flow pairs. This is done by comparing the characteristics of the traffic. After linking the associated network flows, the adversary can surpass the anonymity and exploit the network.

A statistical correlation metrics is used currently for the flow correlation attacks. Previously, the correlation metrics that were used by flow correlation algorithms are-

a) Mutual information

This metric works on comparing the dependency of two random variables. For instance, to use this metrics, the traffic features of an egress Tor flow are considered which is dependent on its corresponding ingress flow. The traffic features of target flows is studied which is later reconstructed and compared. To make a reliable decision and outcome by following this method, it requires a long vector of features.

b) Pearson Correlation

One of the advantages of this metric as compared to the previous one is that it does not require an empirical distribution of variables. Hence, the metric can be worked with a shorter length of data. The Pearson correlation is a linear correlation between the random variables.

c) Cosine Similarity

In this metric, the angular similarity of two random variables is measured. In terms of the requirement of data, the cosine similarity is like Pearson correlation as it does not require a creation of empirical distribution of variables and can instead be directly applied on two random variables.

d) Spearman Correlation

This metric measures statistical dependence between the rankings of two variables.

The main aim of the attacker is to increase the fraction of Tor connections which is being intercepted. There are two popular ways which are used by the attacker to get the desired result. One is to run many Tor relays. When these relays are run on the network, the traffic features of the Tor connection are recorded. If the adversary has the access to malicious relays, it increases the chance of intercepting both the ends in a Tor connection. The other method is by taking into account the autonomous systems or internet exchange points. A specific number of autonomous systems and IXPs intercept a significant fraction of Tor traffic. This helps in performing the correlation attack on Tor.

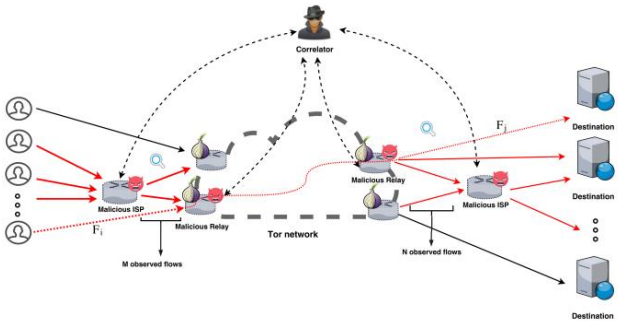


Fig 2: The flow correlation attack setting on Tor [28]

- HTTP Based Application-Level Attack-

The attack is based on the problem of low latency applications based on TCP streams and not on specific web browsing on Tor. The assumption of this attack is that the attacker can control multiple routers as well as the entry and exit routers of the circuit. The following is possible because the tor operates in a voluntary manner. The HTTP's vulnerability is used for man-in-the -middle attacks.

The client has a greater chance to select their entry and exit routers in the circuit as the resource claims of their routers are exaggerated. A forged web page or a targeted web page attack can then be carried out successfully if the client issues a HTTP request. The overall idea is to let the client's browser initiate malicious connections. This helps in creating a distinctive traffic pattern. The client's identity is exposed by the detection of the entry router. The entry router is not required to carry out the attack if the adversary is capable of sniffing the packets that are transmitted via the link between the client and the entry router.

To avoid this attack, it is necessary to minimize the chance of choosing malicious routers in the circuit. This can be done by increasing the total number of Tor routers. The circuit construction algorithm is also evolved to select only fully trusted and dedicated routers through strict authorization and authentication. Additional ways to avoid the attacks is by using HTTPS and web browser plugins.

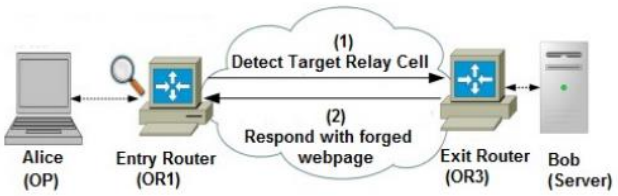


Fig 3: Forged webpage injection attack [29]

- Bad Apple Attack-

The Bad Apple Attack is a kind of application-level attack that requires the installation of a malicious program on the client's computer in order to obtain the client's IP address and use Tor for communication. Since it is not Tor-specific, the method used to retrieve the IP address is not covered in depth in the paper. After the malicious application has the IP address, it can use the Tor network to relay it to a malicious server. The attacker has to have access to the client's circuits' exit nodes in order to be able to correlate traffic from the malicious application with other traffic. The exit node can correlate communication from the malicious application with other network traffic because Tor combines numerous streams, possibly from various apps, in a single circuit. For instance, the attacker can link the client to a specific website if the malicious exit node first detects traffic from the malicious program and then detects traffic from that website on the same circuit.

- Replay Attack-

The replay attack is carried out by the attacker by creating a duplicate cell by selecting a cell in the entry node. This cell is then sent to the second node in the circuit. Once the circuit has been established, the duplicated cell is selected, making it a relay cell. The encryption of Tor layer with the Advanced Encryption Standard can be decrypted by using these relay cells. The cells are detected at the exit nodes and notifies the attacker about the entry nodes. The decryption error takes place as the decryption and encryption counter goes out of sync by duplicated cells. The attack is confirmed by the adversary by matching the timing of error generated and duplicated cells sent. Once confirmed, the communication between the client and server can be accessed by the attacker.

- Cell based Counter Attack-

The cell-based counterattack works on the method of manipulating the timing of relay cells and the exit and entry counter cells. The attacker obtains the right to embed a signal in the client or server traffic. With the help of this attack, it can be confirmed that there lies a communication between the client and the server and further attacks can thus be implemented by the adversary. The attacker needs to be really careful while sending each symbol and needs to calculate the timings accordingly because, if the timing is not appropriate, the attacker can lose the connection between the client and the server. If the timing for the symbols to be sent is too short, the cells can get combined by other relays in the circuit. On the other hand, if attacker waits too long to send the symbols, it increases the latency which might force the user to create a new circuit. Along with this, the attacker also needs to choose the number of cells which needs to be sent in such a way that the combined cells can be recognized as the symbols. This is necessary because natural congestion or any delay that is caused in the network, results in the insertion of cell pattern at the middle onion routers.

B. Congestion Attacks

The congestion attack focuses on monitoring the connection between two nodes as well as creating new paths through other nodes through which all the available capacity is consumed. When the attacker clogs one of the target paths, a change in the observed speed of the victim's connection is observed.

An attack carried out in 2005, with the help of congestion as well as timing analysis, was able to reveal all the routers that were involved in the Tor circuit. This was carried out by measuring the load of each node in the network and then congesting the nodes. By this, they were able to discover which nodes were responsible for the participation in a particular circuit.

[1] conducted the congestion attack on Tor. To carry out this attack, three features were necessary. First, when routing requests, tor routers do not introduce any artificial delays. This makes it easy for an adversary to observe changes that take place in request latency. Second, the Tor router addresses are officially known and thereby easily obtained from the directory servers. Third, the Tor server implementation that was used during this attack did not restrict the arbitrary length parts.

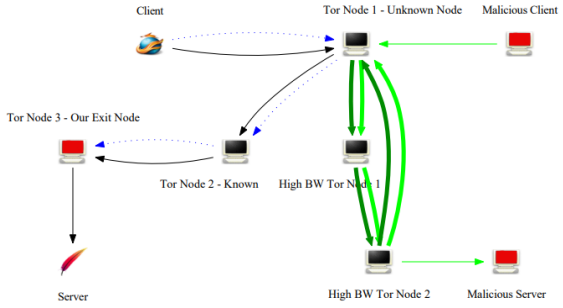


Fig 4: Congestion attack setup [25]

The above figure provides details about the steps of the attack that was conducted. The adversary ensures that a repeated request performance is carried out by the initiator at known intervals. Then, the adversary observes the patterns of arrival times for these requests. Finally, the adversary performs the clogging attack, thereby changing the pattern. With the help of this attack, the adversary determines the entry node.

To carry out the congestion attack, the attacker uses the method of looping back to the circuit. For this, the attacker creates a long circuit which consists of the inclusion of that router repeatedly on the path. The router will have no idea that the circuit is looped back to itself. As a result, during the time of packet scheduling, Tor will treat this long circuit attack as many different circuits. The attacker then carries out the attack and transmits the data when the circuit is long

enough. The main hindrance for the attacker is time. For an attack to be carried out successfully, it is necessary to have a sufficient long circuit length. The length of the circuit created also depends upon the delay and bandwidth because if the delay is large and the bandwidth is small, it will take a longer time for the attacker to create the circuit which is of sufficient length to serve the purpose. To check if the victim's circuit latency depends on the attack, the attacker can vary the strength of the attack. However, the attacker should not make the congestion attack too powerful especially for the targets which have low bandwidth as the routers in this case will be overwhelmed by the amount of traffic it is encountered resulting in the routers being knocked out. If the routers stop working, the attacker will stop receiving any requests.

- Attack by modulating traffic-

The modulating traffic attack consists of a corrupt server and one or more corrupted Onion Routers. To execute this attack, the attacker needs to make the client connect with the corrupted server. The attacker only has the partial view of the Tor network so in order to start the attack, the adversary uses unencrypted HTTP traffic. The corrupted ORs are then connected with legitimate ORs. These corrupted ORs fill the connection with probe traffic to check if they are on a path from target client to corrupt server and record the latency of the connection. If the latency pattern matches the pattern that the corrupt server sends to the client, it is assumed that there are chances of the OR to be a part of the circuit from the client to the corrupt server. This technique is utilized to discover all the ORs that are on the path from client to server. This attack is also used to check if the origination of two different circuits arises from the same client. This attack works better when the length of circuits is greater or a larger pool of ORs are used in the Tor network.

C. Timing Attacks

This attack employs traffic analysis techniques to connect otherwise unrelated data streams back to the initiator. The attacker can connect to specific Tor nodes and measure message latencies. The attacker can analyze the data using traffic-analysis techniques by estimating the traffic load of a Tor node against known traffic patterns. A malicious server sends data to the victim in a pattern, which is then observed by creating a connection through candidate onion routers and performing traffic analysis in a variant of this attack.

- A cell counter-based attack against Tor-

In a normal case of the attack, the attacker requires both ends of the circuit to carry out this attack, but a variation of this attack allows the attacker to carry out this attack by just knowing the exit onion router. In this case, the attacker carries out a man in the middle attack.

The attack works by hiding a secret signal in the traffic's cell counter, which is then recognized by another malicious node in the network to confirm the communicating parties. The secret signal could be a bit sequence. The signal injection can be performed on either the exit or the entry onion router. The injection operates by varying the number of queued relay cells on each onion router. For example, the attacker could specify three relay cells for '1' and one relay cell for '0'. Some variation may occur during the transfer of these cells from one onion router to another due to network congestion and latencies.

- Browser based attacks-

It describes a time-based attack that takes advantage of web browser behavior when it interacts with tampered HTTP traffic. This attack requires two malicious onion routers: an entry onion router and an exit onion router. The entry onion router looks for time-based patterns in the user's traffic, whereas the exit onion router modifies the HTTP traffic to include HTML or JavaScript code that generates calls to a malicious web server in recognizable time patterns.

This attack does not require the malicious entry and exit onion routers to be in the same circuit. If a user leaves their browser open after receiving code from a malicious exit onion router, the attacker will have control of an entry onion router in a new circuit created by the onion proxy at some point in the future. This allows the attacker to connect the user's new circuit activities to their previous activities on the compromised circuit, potentially jeopardizing their anonymity.

- Indirect Rate Reduction Attack-

The main motive of this attack is to extract the information about the clients which are communicating with a predefined server through the Tor network. To carry out this attack, the communication with the entry node of the circuit of the clients has to be intercepted. Therefore, the clients chosen by the adversary needs to be monitored beforehand by the attacker. The attacker uses a technique of congestion control behavior of TCP. The congestion control behavior states that the congestion window of the device scales down on receiving three same ACK packets. The attacker identifies the exit nodes that have the chances to connect to the server as well as lot of different ports. These ports receive three packets by the adversary. The packets have wrong sequence number, but the IP address of these packets are spoofed so it seems that they have come from the server. The exit node will send three ACK packets which will scale down the congestion window of the server.

The adversary must make sure that enough time is given to the server congestion window to recover between the iterations of attack. It is possible for the adversary to clearly determine the clients that are communicating with the server after carrying out the attack multiple times. The adversary has to wait several minutes before trying to repeat the attack by sending fake packets as some time is required for the execution of iterations that are sufficient enough for the connections to last long.

D. Fingerprinting Attacks on Tor

A website fingerprinting attack can identify a website even if its traffic is encrypted using Tor or a VPN. This is accomplished by analyzing packet information such as

packet length, packet count, and timing. A website can be identified using this information without having to inspect the encrypted payload.

In Tor, there are two ways to collect traffic data. The first method entails an attacker establishing an entry node and capturing traffic that passes through it. However, because Tor chooses nodes at random, a victim is unlikely to connect to the attacker's node. The second method involves a network operator, such as an ISP, as the attacker. This attacker can intercept traffic packets between a victim and a Tor entry node.

- Closed and Open world tests-

Two different schemes can be used to evaluate fingerprinting attacks. The first type of test is a closed-world test in which the victim can only access a limited number of websites that the attacker attempts to detect. For example, the attacker could choose 100 websites and research their distinguishing features. In this scheme, the victim can only access these 100 websites, and the attacker wants to know which ones they are.

The second method of evaluation is an open-world test. In this case, the victim can access any website on the Internet, and the attacker must determine whether the website is one of the monitored sites. If the website is monitored, the attacker must also determine which of the 100 monitored websites it is. This scheme is more difficult to implement because the attacker must identify a potentially unknown website among a large number of unmonitored sites.

- Website fingerprinting-

In this attack, the adversary determines the size of the file which is being downloaded by counting the total size of packets on each port. When a user visits a webpage, the information is displayed by the browser by sending a request to download different files. These files are downloaded through a separate TCP connection using different ports. Each webpage requires a different file, and every file is of different size. The information related to size of file creates a unique fingerprint that is used to identify the webpage. To extract the information, the attacker builds a collection of fingerprints to monitor the user's browsing behavior. This collection is built by comparing recorded fingerprints against their collection. The attackers use the data related to the number of incoming and outgoing packets to estimate the file size as it is difficult for them to detect the precise size due to the fixed data cells of 512 bytes which is used by Tor. The fingerprint is represented by a vector that counts the occurrences of subsequent incoming packets.

E. DoS attacks on Tor

The attacker carries out a DOS attack on TOR by flooding it with traffic or requests which prevents legitimate users from accessing the tor network.

Various research papers have conducted experiments to understand the DoS attacks which are taken place on the Tor and the repercussions to it.

Earlier during the lack of upper bound on the length of tor circuits in older tor enabled the attacker to perform the bandwidth amplification DoS attack. This attack was carried out by creating cyclic, arbitrary length circuits through high bandwidth tor relays. This led to a congestion which had an effect on the latency of legitimate circuits which can be used to determine the guard relay on circuit.

A DoS attack was carried out by asymmetric, amplification packet-spinning. The goal was to keep the legitimate relays busy with expensive cryptographic operations which results in the legitimate clients choosing attacker's relays. To carry out this attack, the attacker makes use of malicious relays to create a circular Tor circuit that starts and ends at a malicious relay.

A selective DoS attack can take place by the attacker which increases the probability to choose the attacker's relays as guard and exit relays. This helps the attacker to de-anonymize a large fraction of Tor circuits.

Tor's end-to-end reliable data transport can be exploited through which the memory is consumed by filling up the application layer packet queues by implementing the sniper attack which is a memory-based DoS attack. By this method, the attacker is capable to sequentially disable 20 exit relays in a span of 29 minutes, making the Tor network unusable while remaining undetected.

From an attacker's perspective, the goal is to either disrupt the tor network entirely or disrupt a portion of it that affects the entire subpopulation of Tor users. To disrupt a portion of Tor, the attacks are carried out against the bridge infrastructure and the set of unpublished relays of Tor that permits the users to participate who are otherwise prevented to access the Tor network directly. An attack is considered successful if it entirely prevents the users to access Tor or if the performance is degraded to such an extent that the anonymity service becomes too burdensome to use. In a current scenario, even the degrading performance is seen as an impediment as the delay in the performance as compared to the current levels can lead to a lot of customers to abandon the network.

- Unbalancing Load-

A bandwidth DoS attacks can be used over TorFlow to disrupt Tor services. The TorFlow is used to scan Tor relays to measure their relative performance. The relay weights which are produced by TorFlow can be disrupted by an adversary. The attacker launches a bandwidth DoS attack that clogs the TorFlow scanner's link. The TorFlow scanners are identified by the adversary through their IP address. Once the scanner's link is clogged, the latency and packet loss increase. As a result, the scanners take more time to download files through Tor relays. The scanner will then term the performance of relay as bad thereby reducing the accuracy generated for the relay weights. This process leads to a disruption in process of loading balance.

One of the easiest methods used by the adversaries to carry out this attack is to brute force the TorFlow scanner by flooding it with bandwidth at constant rate.

To increase the impact on the final set of relay weights generated, the attacker can use another method where a certain amount of bandwidth is used to flood the victim and later pausing the attack for some time. This process is repeated simultaneously which will produce an output where the scanner records normal download times for some

relays and reduced times for other set of relays when attack is paused.
The adversary also uses another set of technique by determining the best set of performing relays and targeting them at a specific set of time. The adversary measures the fastest relays and targets them with bandwidth DoS attacks which creates a greater impact in performance.

F. Supportive Attacks

The attacks here do not aim to directly de-anonymize the Tor users but are helpful in a way to carry out the attack of de-anonymization at a later point of time.

- Sybil Attack-

The number of active Tor relays, which are nodes that assist in directing internet traffic through the Tor network to safeguard users' privacy and anonymity, suddenly increased in June 2010. Later, it was found that someone had installed a large number of Tor relays on PlanetLab equipment. Although it may appear innocent, this can be exploited to attack the Tor network. The attacker can enhance their consensus weight and potentially gain control of a sizable chunk of the network by setting up a lot of relays. They can now monitor and intercept traffic, jeopardizing the security and privacy of Tor users.

A Sybil attack is when an attacker generates numerous virtual identities or nodes to control a network more effectively than they could with just one. The success of many attacks against Tor depends on the volume of traffic an attacker can see, also referred to as their consensus weight. An attacker's consensus weight increases with the number of nodes they control, making it simpler for them to carry out attacks like fingerprinting and correlation attacks. In essence, a Sybil attack facilitates information gathering and compromises user security and privacy on the network.

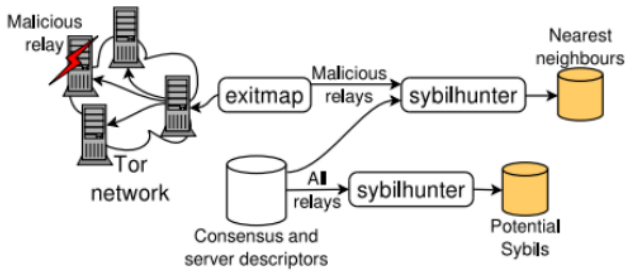


Fig 5 : Sybil Attack setup [29]

The Sybil attack not only facilitates other attacks but also jeopardizes users' access to and privacy within the Tor network. The efficiency of the Tor network depends on the relays' dependability because failing relays can decrease user experience and limit anonymity. If users experience problems brought on by faulty relays, they may stop using Tor completely, which would reduce the number of users and the level of anonymity. By purposefully affecting the dependability of anonymous communications and raising the possibility of user anonymity being compromised, attackers can exploit this by installing their own malicious relays. The Tor network's functionality and security are thus seriously compromised by the attack.

- Packet Size Analysis Attack-

A technique was published in 2011 that distinguished the Tor traffic from non-Tor traffic just by intercepting and analyzing the traffic passively. On examination of this Tor traffic, researchers found out that the third packet transmitted is around 140 bytes and the size of 5th packet is around 920 bytes. With the help of these simple patterns, it is possible to classify around 98% of the actual Tor traffic. Additionally, many packet size are larger than 512 bytes which is the size of Tor cell. This experiment was conducted in a controlled condition but can also be used in the real world by the attackers to exploit the Tor network. Just by analyzing the traffic it was possible for the adversary to differentiate between Tor and non-Tor traffic.

- Tor Authentication Protocol

A paper outlining the attack on Tor Authentication Protocol (TAP) was published by a researcher named Y Zhang in 2009. The attack was carried out when a user runs multiple concurrent sessions of TAP. The TAP plays a crucial role in Tor's security as it negotiates the session keys between user and onion routers (ORs) in a circuit. The vulnerability takes place when the OR A is malicious and the user connects to it and negotiates a session key, then negotiates a session key with OR B which is non-malicious. The attacker can now interleave the messages from both the sessions to create a different session key between user and OR B. This attack does not directly harm the user, but it goes against the original purpose of TAP protocol.

G. Revealing Hidden Services Attack

- First Node Attack

When the attacker's relay tries to connect to the hidden service's server, it would immediately reveal the location of the hidden service to that node.

To become the first node in the circuit from a hidden server, there is a requirement of a malicious node and a client by the attacker that connects to the hidden service. A timing pattern is sent by the client in the communication. If the malicious node is on the circuit, it can detect the timing pattern. By knowing the IP addresses of all nodes until the rendezvous point, the malicious node can determine if it is present on the circuit or not. The timing analysis can reveal the location of a malicious node if it is present in the first or second node after the hidden service. If a malicious node is not in the correct position, the attack is carried out again until the malicious node becomes the circuit's first node. The anonymity of the hidden service is compromised and can reveal its location with this attack.

- Clock Skew Attack

Another way of identifying the location of hidden services is by clock skew attack. In this attack, a list of potential servers is selected. By repeatedly making requests to the hidden servers, the temperature of the server increases which results in its clock to turn at a slightly different rate. The timestamps which are received by the hidden service can be analyzed and its clock skew can be determined. By comparing this clock skew with the skew of the candidate servers, it is possible to locate the hidden service.

H. Entry and exit onion router selection attack

- Compromising Anonymity using Packet Spinning

The following attack uses looping circuits and malicious onion routers. It describes a Tor anonymity attack that makes use of looping circuits and malicious onion routers. The attack creates circuit loops to prevent other onion routers from being selected, resulting in a denial-of-service attack. This increases the likelihood of malicious onion routers being selected in circuits, allowing for additional attacks. The attack is based on the assumption that circular circuits are not detectable and that legitimate onion routers spend time performing cryptographic calculations.

- Low resource routing attacks

This attack consists of an adversary which exploits flaws in the routing mechanisms of the Tor network to compromise the anonymity of users with limited resources. These attacks typically involve supplying false resource information to directories or exploiting the victim's circuit creation process in order to identify patterns in the network's routing algorithm. The goal is to increase the likelihood of the adversary being chosen as the entry and exit nodes in the user's circuit, thereby jeopardizing their anonymity. These attacks are especially dangerous for users with limited resources because they may not have access to the network's more secure options.

V. IMPLEMENTATION

A. Tor client, server and client+ server Attacks

Attacks can still be made against Tor and Tor browsers. Majority of attacks on Tor is exploiting vulnerabilities to traffic analysis, confirmation attacks, and probable guard discovery these are the attacks that have always been under the limelight among the top researchers and users. However, several other lesser-known attacks exist and in recent years they are some of the most infectious and top-rated attacks. Since the beginning of Tor, research on onion networks has revealed that adversaries are capable of assaulting three different things:

Client: To identify it, a Tor network client is chosen.
Server: An attempt is made to identify or weaken the Tor onion (hidden) service.
Network: Multiple malicious Tor nodes are typically used to target the larger Tor network.
Additionally, there are generic assaults that target several Tor entities; often, both the client and server are attacked simultaneously.

B. Induced Tor Guard Selection (client-side Tor threat)

Tor clients can be persuaded to use a malicious Tor guard (entry) node by, among other things, changing the target's traffic capabilities, obstructing connections to trustworthy entry nodes on the network, and so on. End-to-end correlation and other assaults are considerably aided by this.

C. P2P Information Leakage (client-side Tor threat)

Peer-to-peer connections are taken advantage of to obtain the client's IP address. For instance, when clients use the BitTorrent protocol to connect over Tor and communicate with the torrent tracker, opponents can obtain the IP address of those clients. The IP address and listening port of peers who can share the requested resource are retrieved by torrent trackers, tracker lists can be retrieved anonymously over Tor, the P2P connection itself cannot, making it vulnerable to MitM attacks that could lead to a list containing the IP address of a malicious torrent peer. This indicates that it is possible to identify the client whose IP address sent the tracker request (through Tor).

D. Raptor Attacks (client-side Tor threat)

Tor traffic is carried by autonomous systems (ASes), which have advanced eavesdropping capabilities. To deanonymize clients, an AS or a group of cooperating ASes can do time analyses between the client and the first relay and between the last relay and the destination. Three methods are used to accomplish this: altering BGP routingarchive.org so that more ASes can analyse a user's traffic; modifying BGP announcements so that ASes are chosen on paths to and from relay nodes; and conducting timing analysis on unidirectional traffic at both communication ends.

E. Cell Counting and Padding [15] (server side Tor threat)

An onion (hidden) service is compelled to connect to a malicious rendezvous point in this attack. A signature is added to the communication by a specifically created set of Tor padding cells (of a particular number) and a cookie or token produced by the client. It is possible to determine which guard node was selected by the onion service and thus its IP address if an entry node controlled by an attacker monitor recognizes these signatures.

F. Off-path MitM Attacks [16] [15] (server side Tor threat)

Man-in-the-middle attacks can be launched against the targeted service by an adversary who has gained access to (or assumed ownership of) the onion service's private key. The only way to prevent this attack is to stop using them.

Onion address and generate a new one because there is no revocation mechanism for onion services.

G. *Cell Traffic Analysis [17] (server + client side Tor threat)*

Network traffic analysis-capable adversaries insert packets (particular, repeating traffic in the TCP connection) server-side and attempt to observe these packets client-side using statistical correlation. As a result, Tor client traffic can be recognized if a client is linked to a malicious server and the adversary has control over a lot of entry (guard) nodes, one of which is picked in a particular Tor circuit.

H. *Abstract of our Attack Implementations*

We have homed on to two different types of attacks, first implementation being a DDOs attack which can be launched with publicly available hosts and DNS like cloud flair and another variant where we have a hidden services created with onion service and to launch a DDOS attack.
Furthermore, the second implementation attack that we try to demonstrate is which is being actively employed by threat actors and cyber criminals over Tor client distribution and injecting an obscure and specialised malware which changes the crypto wallet addresses.

I. *Implementation Attack – 1*

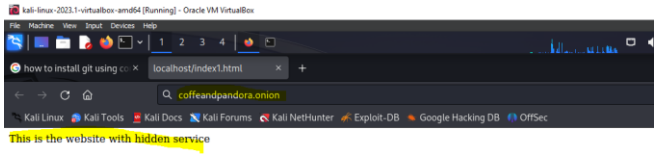
The first one that we try to demonstrate is a DDOS attacks that can be launched through crating a hidden/ shared hosting service. The set up has been configured with Debian and Ubuntu installations. In addition, we are using public nameserver like 1.1.1.1 (from Cloudflare) or 8.8.8.8 (from Google) this will help us to create the shared hosting service through Cloudflare. For creating of the hidden services, we generate a skeleton configuration that will allow us to generate an address, we can name it accordingly to our mode of attack. In this case we are naming the address to be "coffeandpandora".

To create a hidden service with our address name we use a docker container to initiate the process. Once the container is downloaded, we can then generate our site skeleton and it will yield the following output.

$docker run -it --rm -v $(pwd)/web:/web coffeandpandora /hiddentorser generate ^ coffeandpandora

[+] Generating the address with mask: ^coffeandpandora
[+] Found matching domain after 137072 tries: coffeandpandora.onion
[+] Generating nginx configuration for site coffeandpandora.onion
[+] Creating www folder.
[+] Generating index.html template.

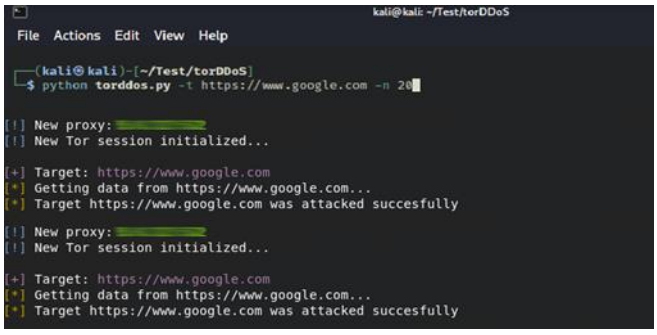
Once we have our skeleton ready, we can deploy and run it by running the docker container.
With docker run – d – restart=always –name hide\ - v $(pwd)/web:/web coffeandpandor/hiddentorser



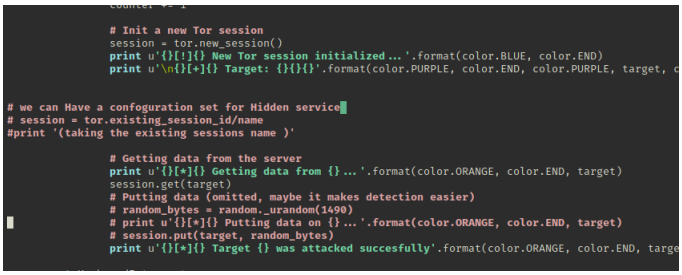
Screenshot 1: Hosted website running on hidden service of onion locally.

To execute the DDOS attack we have customised the payload to come from TOR network.
The modus of attack that we use is by one VM generating and running a shared hosting service and another one running a hidden running service. These would be attacking the victim system (This is implementation is done on internal networks and have not fully exposed to the onion services for safety and security concerns)
A brief description on how the DDos works -
1. Initial timestamp and increment the counter.
2. Init a new Tor session, this can be our shared or hidden services. It is more feasible to use hidden services as these are more obscure and more difficult to render. The shared will be blocked by the original hosting services like google and Cloudflare, even though Cloudflare hosting services has been used many times for carrying out DDos attacks in the past.
3. Sending of random_bytes = random._urandom(1490) to go back and increment even more to cause traffic overflow and lead to DDOS on the website or system that is targeted.

The execution – We can use cloud flare or have our own hidden service website that can be used to configure the DDOs attack. In this case we have used the cloud flair for demo purposes as there are more steps involved in setting up of the DDOs attack with an obscure hidden service. Ultimately, we have planned and provision to run the attack with hidden services which we can further work on developing. Below is the screen shot of running the attack. In the screen shot below shows the arrangements made for running DOS through hidden services [19].



Screenshot 2: Test of DDoS



Screenshot 3: Code snippet of DDos

J. Implementation Attack – II

Another Recent attack that has surfaced recently is a Tor browser combined with a malicious malware payload for distribution. The modus operandi of the use's social media platforms such as YouTube and many others to distribute and propagate the malware. With the ongoing conflict of Ukraine and Russia, the tor nodes which are originated from Russia are being blocked resulting in a net loss of 150K user with the censorship on the Tor in Russia.
This has led to threat actors to capitalize on this and distributing out copies of rigged Tor browsers with claims of being secure and legit. The installers that are distributed carry names like 'torbrowser_ru.exe,' and additionally contain the language packs allowing users to select their preferred language as it appears in a real Tor browser installer.
The archive extracts the malware in the background, executes it as a new process, and registers it for system auto starts while the default Tor browser is started in the front. Additionally, the malware conceals itself on the compromised system by using an uTorrent icon.

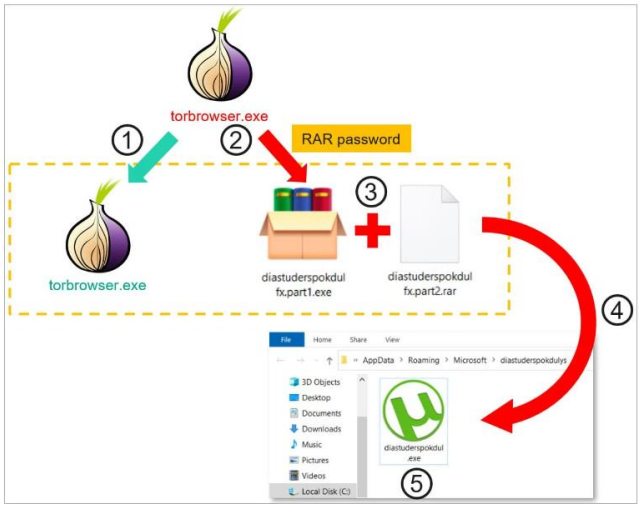
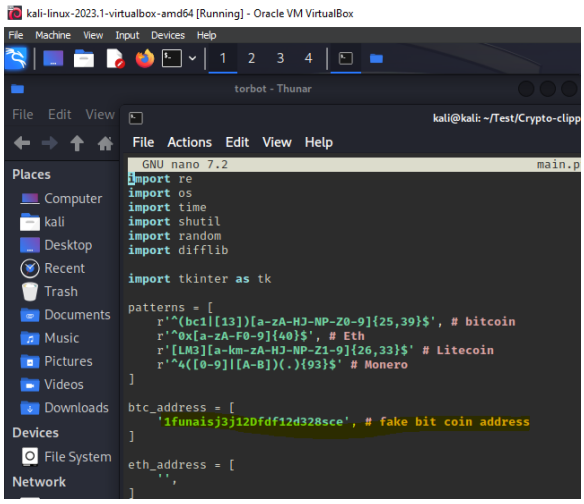


Figure [20] – Tor client, source: Kaspersky

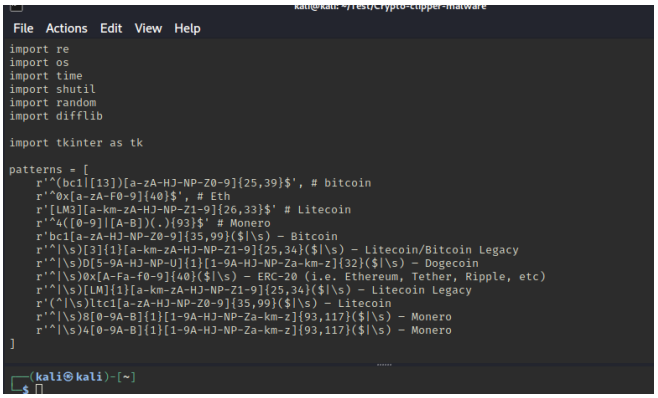
From 2022 to 2023 March 16,000 variants of these Tor installers have been detected in 52 countries, while the main target are Russia, China, and Eastern Europe, it has also been seen used in United States, Germany, France, the Netherlands, and UK.
Furthermore, our observations reveal that these attacks target people who use Tor for additionally anonymity for carrying payment in Bitcoin. This allows threat actors to have a malware with high obscurity and customizability, which they have successfully exploited by using a malware technique called clipboard hijacking. It is common and second nature for users to copy bitcoin addresses to the clipboard before pasting them into a webpage for payment because they are lengthy and difficult to enter. Using regular expressions, the malware scans the clipboard for recognisable crypto wallet addresses which it replaces it with a related cryptocurrency address controlled by the threat actors which is a huge list of 26K to 30k addresses. The threat actor's address will be pasted when the user pastes the cryptocurrency address, allowing the attackers to steal the sent transaction.

The following regular expression were found with our research inside the sample of the malware-
bc1[a-zA-HJ-NP-Z0-9]{35,99}($|\s) – Bitcoin
(^|\s)[3]{1}[a-km-zA-HJ-NP-Z1-9]{25,34}($|\s) – Litecoin/Bitcoin Legacy
(^|\s)D[5-9A-HJ-NP-U]{1}[1-9A-HJ-NP-Za-km-z]{32}($|\s) – Dogecoin
(^|\s)0x[A-Fa-f0-9]{40}($|\s) – ERC-20 (i.e. Ethereum, Tether, Ripple, etc)
(^|\s)[LM]{1}[a-km-zA-HJ-NP-Z1-9]{25,34}($|\s) – Litecoin Legacy
((^|\s)ltc1[a-zA-HJ-NP-Z0-9]{35,99}($|\s) – Litecoin
(^|\s)8[0-9A-B]{1}[1-9A-HJ-NP-Za-km-z]{93,117}($|\s) – Monero
(^|\s)4[0-9A-B]{1}[1-9A-HJ-NP-Za-km-z]{93,117}($|\s) – Monero
This is a basic version of the malware; the more advanced version involves the malware contacting a c&c server in real time and having it sent and generate a similar crypto address which matches the last few characters of the original address which reduces the chances of being caught.

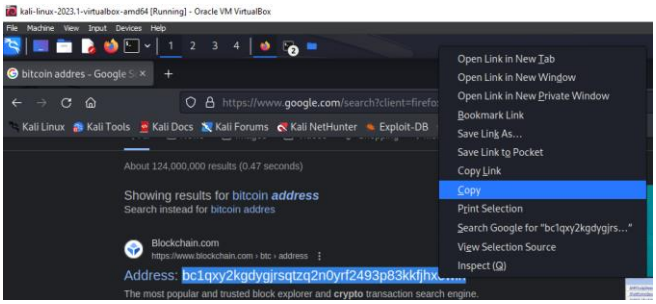
Below is a test version of the malware that is created, here it can be seen that the regular expression of bitcoin wallets like bitcoin, etheruim, Litecoin and monero respectively. Below that we can add the attacker-controlled address, note the address that is highlighted is used only for demo purpose, it is a 25-digit address whereas bit coin address is 26. When we copy any bitcoin address it changes the actual address to the attacker's address. This can be done not only for the crypto wallet but also for bank codes and name with e-transfer method.



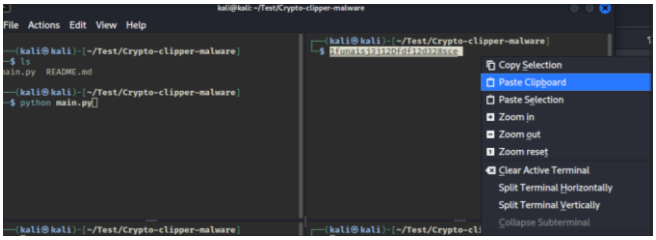
Screenshot 4: Code for Clip Malware



Screenshot 5 : Code for Clip Malware



Screenshot 6: Copying a bitcoin address



Screenshot 6: The Bitcoin address changed because of malware program.

The way to defend against this attack exists numerous ways.
1. Use caution when installing software or opening email attachments from sites you are unfamiliar with.
2. For all your accounts, use strong, one-of-a-kind passwords, and avoid copying and pasting them wherever feasible.
3.Whenever feasible, avoid copying and pasting confidential data, including passwords, credit card numbers, and other personal information.
4. Make use of a clipboard manager that will notify you when a program accesses the clipboard and will prevent unauthorized access.
5. Consider utilizing a virtual keyboard to protect yourself from viruses such as keyloggers and other threats that may be watching your keystrokes.

VI. Conclusion and Observation

Although Tor claims to be the pinnacle of anonymity and security, there are security misconfigurations, vulnerabilities and social engineering factors which always find a way to exploit the protocols and techniques.
In this paper, we discussed the ways in which attackers carry out vulnerabilities on Tor and the possible solutions which have been brought to mitigate the vulnerabilities by some extent. Later we have implemented some attacks on Tor to understand the extent at which Tor can be compromised and the possible loopholes which needs to be fixed for Tor to be free from future attacks.

REFERENCES

[1] *https://github.com/Attacks-on-Tor/Attacks-on-Tor.git -*

[2] *Tor. Who uses Tor? 2014. url: https://www.torproject.org/about/torusers.html.en*

[3] *R. Snader and N. Borisov. "Improving Security and Performance in the Tor Network through Tunable Path Selection". In: IEEE Transactions on Dependable and Secure Computing (2010).*

[4] *L. Overlier and P. Syverson. "Locating Hidden Servers". In: IEEE Symposium on Security and Privacy (2006).*

[5] *W. Nicol. A Beginner's Guide to Tor: How to navigate through the underground Internet. 2016. url: http://www.digitaltrends.com/computing/a-beginners-guide-to-tor-how-to-navigate-through-the-underground-internet*

[6] *Damon McCoy et al. "Shining light in dark places: Understanding the Tor network". In: Privacy Enhancing Technologies. Springer. 2008, pp. 63-76.*

[7] *L. Overlier and P. Syverson. "Locating Hidden Servers". In: IEEE Symposium on Security and Privacy (2006).*

[8] *M. Mulazzani, M. Huber, and E. Weippl. "Anonymity and Monitoring: How to Monitor the Infrastructure of an Anonymity System". In: IEEE Transactions on Systems, Man and Cybernetics.*

[9] *Juha Salo. "Recent Attacks On Tor". In: Aalto University.*

[10] *Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Tech. rep. DTIC Document.*

[11] *Yixin Sun et al. "RAPTOR: routing attacks on privacy in tor". In: 24th USENIX Security Symposium (USENIX Security 15)*

[12] *Ryan Pries et al. "A new replay attack against anonymous communication networks". In: Communications, 2008. ICC'08. IEEE International Conference on. IEEE. 2008.*

[13] *S. Murdoch. "Hot or Not: Revealing Hidden Services by their Clock Skew". In: Proceedings of the 13th ACM Conference on Computer and Communication Security.*

[14] *Sam DeFabbia-Kane. "Analyzing the effectiveness of passive correlation attacks on the tor anonymity network". PhD thesis. Wesleyan University,*Y. Zhang et al., "Home M2M networks: Architectures, standards, and QoS improvement," IEEE Commun. Mag., vol. 49, no. 4, pp. 44–52, Apr. 2011.

[15] Protocol-level Hidden Server Discovery Zhen Ling, Junzhou Luo, Kui Wuy and Xinwen Fuz.

[16] R Off-path Man-in-the-Middle Attack on Tor Hidden Services Amirali Sanatinia, Guevara Noubir Northeastern University.

[17] Low-Cost Traffic Analysis of Tor Steven J. Murdoch and George Danezis University of Cambridge, Computer Laboratory 15 JJ Thomson Avenue, Cambridge CB3 0FD United Kingdom

[18] Website://www.whonix.org/wiki/Speculative_Tor_Attacks#cite_note-15.

[19] bleepingcomputer.com/news/security/tor-and-i2p-networks-hit-by-wave-of-ongoing-ddos-attacks/

[20] source Kasperkeyn

[21] Jansen, Rob, Tavish Vaidya, and Micah Sherr. "Point Break: A Study of Bandwidth Denial-of-Service Attacks against Tor." USENIX security symposium. 2019.

[22] Karunanayake, Ishan, et al. "De-anonymisation attacks on Tor: A Survey." IEEE Communications Surveys & Tutorials 23.4 (2021): 2324-2350.

[23] Abe, Kota, and Shigeki Goto. "Fingerprinting attack on Tor anonymity using deep learning." Proceedings of the Asia-Pacific Advanced Network 42 (2016): 15-20.

[24] Bauer, Kevin, et al. "Low-resource routing attacks against Tor." Proceedings of the 2007 ACM workshop on Privacy in electronic society. 2007.

[25] Evans, Nathan S., Roger Dingledine, and Christian Grothoff. "A Practical Congestion Attack on Tor Using Long Paths." USENIX Security Symposium. 2009.

[26] Ling, Zhen, et al. "Protocol-level attacks against Tor." Computer Networks 57.4 (2013): 869-886.

[27] Salo, Juha. "Recent attacks on Tor." Aalto University (2010).

[28] Nasr, Milad, Alireza Bahramali, and Amir Houmansadr. "Deepcorr: Strong flow correlation attacks on tor using deep learning." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018.

[29] Evers, B., et al. "Thirteen Years of Tor Attacks." (2016).