TOR BROWSER

The anonymity of the user's identity is a crucial factor considered by the individuals who wish to surf deep dark web. Over the past 19 years the Tor browser, which was originated from the Mozilla Firefox browser, has earned a reputation for being able to hide user identity when comes to surfing through the internet. Tor simply follows mechanisms to hide the users' originated IP (Internet Protocol) address when surfing internet and sending emails. With this outcome it has become nearly impossible to trace back the owners without having extraordinary mechanisms. Due to the simplicity and easy installation, made anyone to use it easily without any instruction. Hypothetically, the Tor browser has been the mediator behind many of the illegal manifestations occurred throughout the world and the things to be happen in the future.

***Keywords**: Tor; Onion Servers; Security; Privacy; Attack; Vulnerabilities; Networks, De-anonymization exploits, coderandomization, privacy-oriented software, Tor Browser.*

2021/10/30

A Case Study – Tor Browser Vulnerability History

Secure Software Engineering - IE4042 Assignment

IT18029314

Hirushi N. Atapattu

Table of Contents

CHAPTER 01:DOMAIN & HISTORICAL ANALYSIS

Product Overview

Description

The Tor browser which also known as the “Onion Router” is a free and open source network that allows a user to continue web browsing while staying anonymous, which makes user identification, trace back the original user or location tracking impossible. Another advantage of this product is that it provides the protection to the user from traffic analysis and network spying.

The “Onion Routing” is the primary principle which the Tor browser is following. The development credits are given to Paul Syverson, Michael G. Reed and David Goldshlag a mathematician and a computer scientist respectively at the United States Naval Research Laboratory in mid 1990s. The main objective of this Tor project was to protect the United State government’s Intelligence communications that happen online by encrypting the application layer of the communication protocol stack, like the segments of an onion, they're stacked one within the other. Tor directs Internet traffic via a free, global volunteer overlay network with over 6,000 relays to hide a user's location and use from network monitoring and traffic analysis. On September 20, 2002, the alpha version of Tor was released, created by Syerson with computer scientists Roger Dingledine with Nick Mathewson and dubbed The Onion Routing Project (later shortened to "Tor" as an abbreviation for the previous name). A year later, the first official disclosure took place.

Tor’s goal is to enable unrestricted and anonymous communication over the Internet. Tor enables anybody to services through the internet that may be blocked by repressive regimes, lets whistleblowers to contact with authorities anonymously, and provides a method for legal communication between companies and individuals who want to keep their private discussions secret.

However, just as a vehicle that is used to transport your kids to school may be used as a bank robbery getaway car, the Tor browser could be used to either aid crimes or perpetrate crimes.

Although Tor was originally created by the US intelligence in 1990s, it’s not really currently dominated by the US state. Tor is, in reality, not ruled by a centralized entity and is available to development by anybody with the technical capacity to test and enhance it. Tor gets global feedback from privacy-focused specialists for this reason alone, ensuring that it stays relevant and successful.

Ironically, the US government not just developed Tor but also is working on ways to deanonymize its users.

Overview of findings

Various Tor exploits and vulnerabilities have been identified and utilized all throughout course of its history. Attacks on Tor are a topic of ongoing academic research, which the Tor Project itself encourages the keen individuals.

Product Assets

Tor contains a lot of assets that might be vulnerable to security assaults. Because the project serves as a foundation for many browsers, it poses a significant danger if it is hacked. User data, such as user profiles, cookies, preferences, and browser history, is one of Tor's most valuable assets. It's necessary to keep certain assets safe in order to preserve user confidence, particularly because most users trust their browsers implicitly. Most people believe that emails and rogue websites are the sole sources of risk. The project might be jeopardized if this asset is not safeguarded. Tor's flexibility to create extensions is another of its primary advantages. While the Tor community does not manage the enhancements, who are concerned for how they may damage the system. This allows third-party applications to have an impact on the browser.

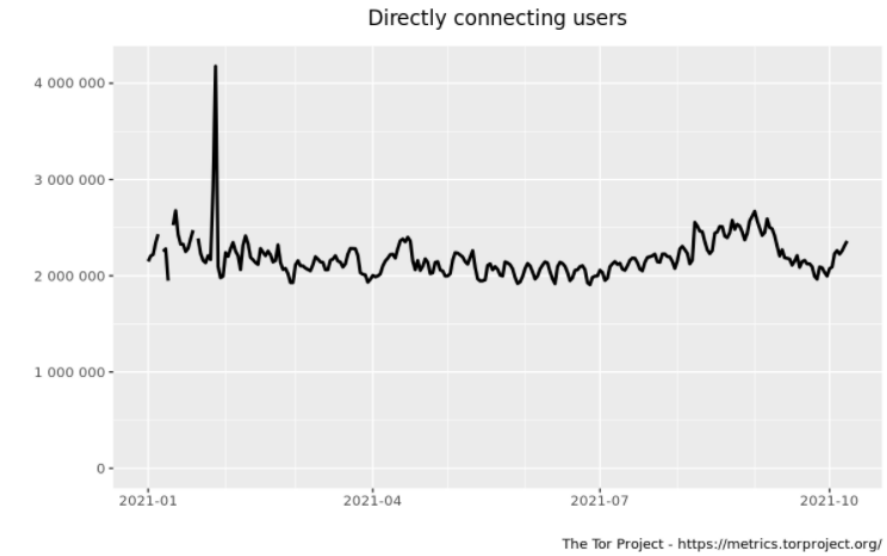
Because many of their access critical browser functions, extensions might pose a danger to the system if they aren't managed appropriately. Allowing these rights to extensions may lead to DoSs, cross-site scripting, and even the download of malicious software onto a user's computer. Trusting a strong extension resulted in a significant security risk in the very first vulnerability study below.

Tor protects its users from web-based threats since it allows web apps to deliver deeper functionality. Tor, on the other hand, is just as vulnerable to website fingerprinting, cross-site scripting, and other web-based vulnerabilities as any other browser. A browser serves as a conduit between the user's surroundings and all available online services. For a hacker, this knowledge is priceless. As a result, it's critical that Tor puts the browser and its functions in a sandbox, preventing access to the system it's running on. If an attacker managed to break out of the sandbox, it would be up to the operating system to address security threats.

Furthermore, if a user's browser directory is saved on a server, the server might be hacked, and everything in the cache and history could fall into the wrong hands. This isn't as much of a concern with a home computer, but if the device is linked to a corporate system, the cached data and history might include important information.

Tor's threat model distinguishes between two types of attackers: opportunistic and devoted opponents. An indiscriminate adversary is a hacker that doesn't go after a single person or company, instead luring them to websites that infect their devices or programmes that attempt to get undue rights. A devoted adversary, on the other hand, may choose to attack a specific person or company and is ready to seize equipment in order to retrieve data or passwords. They'll do whatever an opportunistic opponent may do.

Example Attacks



This graph shows the estimated number of directly-connecting clients; that is, it excludes clients connecting via bridges. These estimates are derived from the number of directory requests counted on directory authorities and mirrors. Relays resolve client IP addresses to country codes, so that graphs are available for most countries. Furthermore, it is possible to display indications of censorship events as obtained from an anomaly-based censorship-detection system (for more details, see this technical report). For further details see these questions and answers about user statistics.

Start date: 01/01/2021

End date: 10/12/2021

Figure 1: Tor users all around the world

Several research have been carried on onion networks over the years, with a particular emphasis upon the Tor network, which is the most widely used.

When it comes to onion network security, attacks can target three different network entities:

- Client: the attacker's goal in this case is to target a Tor network client;
- Server: the attacker's goal in this case is to identify a Tor network client;
- Network: throughout this case, the attacker is after the Tor network.

I shall describe possible attacks against such disparate entities, concentrating on the Tor onion network from here onwards. I also looked into attacks on generic mixed entities.

Attacks targeting the client

Several attempts have been made over the years to hack the Tor network and de-anonymize its users by linking their IP address to an outbound packet. These investigations often resulted in particular assaults that Tor developers were able to mitigate by updating the browser software. This section of the article details attacks targeted at causing harm to the Tor client.

- *Torben attack*

The Torben attack is used to identify a Tor client by attempting to manipulate web pages to compel the user to access content from unknown senders and exploiting low-latency qualities of anonymization connections to infer indicators of web pages that are transferred, allowing knowledge on the web pages the client visits through Tor to be retrieved.

- *Plug-in based attack*

These assaults are directed against the user's Internet browser, which he uses to traverse the network. External software connected further into browser (plug-in) is used by such threats, such as Flash, Java, and ActiveX Controls. These programs are standalone programs that operate with the permissions of the user here on operating system. Some of these technologies, such as Java or Adobe Flash, are run in appropriate virtual machines or frameworks, bypassing the Tor browser's proxy configuration settings, and interacting directly over the Internet network, circumventing the Tor ecosystem. Browser attacks can be carried out in a variety of ways: I be able to operate on the public Internet server approached by the client, through a malevolent server system encoding, for example, Adobe Flash content on the web page; (ii) by deploying an evil exit node, eavesdropping users' communications over non-encrypted channels and integrate malevolent plug-in relays; (iii) by deploying an evil exit node, intercepting users Due to the risk of exposing a client's identity via the use of browser plug-ins, as recommended by anonymizing browsers, such technologies, which are frequently deactivated by default, should be abandoned in order to interact anonymously and securely on the onion network.

- *P2P Information leakage*

This kind of attack is used to anonymize Tor clients by taking advantage of their connections to peer-to-peer networks. Indeed, using the BitTorrent protocol as an example, it is conceivable for a malevolent user to get the IP address of a client using Tor to interact with the torrent tracker. A torrent tracker is a communication service with whom the client must interact in order to get information about a list of peers who are willing to share the desired material. Peers' data is presented as a pair of IP addresses and listening ports.

In this instance, the attacker includes the fact that, but even though the information of trackers can be obtained anonymously through Tor, P2P interactions are often made insecurely, by talking directly with the peer. As a result, the attacker may use the Tor network's man-in-the-middle capability to change the content of the list provided by the torrent tracker by adding the IP address of a rogue torrent peer to the list. Because contact with such a peer is not established via Tor, the attacker may get the IP address of the client who initiated the tracker request.

- *Low-resource routing attacks*

To carry out such an assault, the adversary must enroll or hack several Tor routers with high bandwidth and uptime. By assuming such a compromise, the attacker may reduce the malicious node's resource needs by utilizing low-bandwidth connections, thereby abusing the node's ability to report false bandwidth numbers. The relay node does seem to have a high-bandwidth since this advertising is not validated by trustworthy directory servers, and its chances of being selected for a circuit are especially high. If both the entrance and exit nodes in a network are hacked, every data received may be recorded and analyzed to expose the client's IP address, for example via traffic correlation methods.

- *Unpopular port exploitation*

This attack takes use of the fact that Tor exit nodes often restrict the number of ports they may connect to on the open Network. Using a series of fraudulent entry nodes and malicious exit nodes, the attack tries to obtain the client's identity. The attacker's exit nodes allow communication on unpopular ports. It is also necessary for the attacker to have control over the service host that the client contacts. The attacker's goal is to persuade the client to establish a Tor circuit using an entrance and exit node controlled by the malicious user. Such a setup would enable an attacker to determine the Tor client's identity, for example, using traffic correlation methods.

To carry off the attack, the unauthorized attacker injects a script through into client's web page, causing the browser to interconnect to an Internet service listening on an unpopular port. The Tor network is used to create such a connection. This action will cause the client to establish a Tor circuit that will enable communication on the unpopular port provided. The likelihood of controlling the

circuit's exit node rises since the assailant possesses a set of exit nodes that allow such communication.

- *Induced Tor guard selection*

The only node that communicates directly with the client is the Tor entrance node. However, since the payload of Tor packets is encrypted, the entrance node cannot recover the actual content of transferred conversations without knowing the circuit nodes' decryption keys. As a result, although a single rogue guard node may not jeopardize communication, the attacker may be needed to control the Tor circuit's entrance node.

To persuade a Tor client to choose a particular rogue entry node, it is feasible to disable the client's communications with all public entry nodes except the attacker's. This operation may be carried out by modifying the victim's traffic capabilities, limiting connections to valid entry nodes at the network level using suitable rules established by network administrators or local Internet Service Providers, and so on.

- *Raptor*

RAPTOR (Routing Attacks on Privacy in Tor) is a set of attacks that the Autonomous System (AS) may use to deanonymize clients. One of the assaults is focused on traffic analysis of the network's asymmetric communications. Another attack uses traffic analysis to take advantage of the available churn in Internet routing and BGP routes. Finally, the final assault relies on Internet routing modification through BGP hijacking in order to locate users' Tor guard nodes.

Attacks targeting the server

The attacker's goal on these kind of threat is to target the concealed service in order to expose its identity or weaken it. As previously stated, the Tor network may be used to access services on both the open surface Internet and the Tor network (hidden services). In the latter instance, the customer is unaware of the service's identity. The following hypotheses may be needed in attacks aimed at revealing the secret service IP address: I The attacker must impersonate both a rogue client and a guard node; (ii) the concealed service must use a compromised guard node as an entry node. There are a variety of assaults against hidden services.

- *Off-path MitM*

A man-in-the-middle (MitM) attack against such a Tor hidden service is the basis for this assault. It is feasible to carry out a MitM attack by presuming that the hidden service's private key used to interact on the network is held by the attacker. The key point to remember in this instance is that the attacker does not have to be in the communication route in between client and the server.

- *Tor cell manipulation*

It is necessary to trace a targeted hidden service by manipulating Tor cells/packets. The request is "proxied" by the rendezvous point, which is implied to be controlled by the attacker, when the client sends a cell to a secret service to initiate communication. Such offer perceptually to the malicious actor the potential to perceive the request and apply slight alterations to the message/cell data , thereby also forwarding the message to the hidden service and all at the same sending a timestamp of the customized cell to a central server under the control of the attacker. The hidden service may not recognize the cell as intact, in which case it will send a dismantle message to the client. This message, which is sent from a hidden service to a client, is designed to pass from a hidden service entry node (controlled by the attacker) to the central server, which may send some cell information to the central server, such as the cell command (CELL DESTROY), the cell timestamp, the circuit ID, and the source IP address. The cell is targeted to accommodate the rendezvous point at this point, which may review the cell's timestamp to the central server once forwarding it to the client. Finally, through time correlation, the IP address of the secret service may be discovered from the central server.

- *Cell counting and padding*

The secret service is compelled to connect to a rogue rendezvous point during such attacks. The attacker sends a specially constructed Tor cell/packet to the hidden service's introduction point, indicating the rendezvous location. As a result, the message is sent to the hidden service, which is prompted to construct a Tor circuit in attempt to advance the (malicious) rendezvous point. When the rendezvous point gets the message (which contains some kind of cookie/token created by the client), it is programmed to transmit 50 padding cells to the concealed service via the same circuit. Padding cells, which are allowed by the protocol but ignored by the concealed service, make it easier to generate traffic signatures. The circuit is terminated/closed at this point by the rendezvous point. The attacker is intended to control the entrance node, which examines the flow of the circuits passing through it. If it gets a cell holding the circuit closure, it will check that it arrives after the cell containing the confirmation cookies, and that the number of prior cells is 3 up and 53 down the circuit. If these criteria are fulfilled, the attacker may infer that the guard node he controls was selected from the concealed service, allowing him to obtain the hidden service's IP address.

Attacks targeting the network

The Tor network itself is the target of the assault in this instance. It's essential to keep in mind that by targeting the whole network, the malicious actions may impact many nodes. As a result, rather of affecting a single node, the attack consequences may be spread across the whole network in this scenario.

- *Sniper*

The Sniper assault takes use of Tor's flow control mechanism by launching a DoS attack on a target Tor relay, thereby terminating the Tor function on the computer. This is accomplished by pushing a node to buffer huge quantities of data until it becomes overwhelmed and shuts down. To impair network capabilities and enhance the chances that a client would select an attacker's node, the adversary may attack a large number of nodes. Two assaults are outlined in the paper: I the attacker prevents the victim from reading from the TCP connection containing the attack circuit, causing the TCP window on the victim's departing connection to close and the victim to buffer up to 1000 cells; (ii) the attacker causes cells to be continuously sent to the victim (exceeding the 1000 cell limit and consuming the victim's memory resources), either by ignoring the package window at the packaging end of the circuit, or by sending SENDME messages from the delivery destination.

- *Bridge discovery*

The goal in this instance is to retrieve data from Tor bridge nodes. Such information isn't accessible to the general public. Two alternative methods to bridge discovery are considered: on the one hand, Tor bridges may be enumerated using mass emails and HTTPS servers via Tor. On the other hand, a malicious Tor intermediate router/node may be used to abuse Tor's proportional bandwidth routing algorithms for bridge detection reasons.

- *DoS*

Denial of service (DoS) attacks are used to disrupt a network's operation component or service is not available to the users, or its availability is being reduced. CellFlood is a denial-of-service attack against with the Tor network. This attack takes use of the fact that using a private key to execute 1024-bit operations is approximately 20 times slower on contemporary servers than using the public key. As a result, processing a Tor cell takes 4 times as long/as heavy as creating one. This method may lead to a malicious client flooding a specific node with specially generated cells in order to take over all of the target's computational resources, resulting in a denial of service.

Vulnerability History

CVE 2016-9079 (The Zero Day Exploit)

An anonymous user sent an exploit to the Tor project's bug tracker there at end of November, claiming to have discovered it on the internet. Only the Tor Browser could run the released code, however it also operated in Mozilla Firefox versions 41 to 50. (Windows and Mac OS X systems).

By exploiting an import address table of the XUL.dll library, that is not protected by the toolkit, the malicious code sent as JavaScript was able to successfully circumvent Microsoft's Enhanced Mitigation Experience Toolkit (EMET). The hack exploits the SVG file parser's Use-After-Free flaw. The shellcode seems to be similar to the code employed by the FBI against Freedom Hosting customers in 2013. (tracking down IP addresses of users of child pornography websites). It was referring to the IP address in this case: 5.39.27.226's job was to authenticate the person - acquire the computer's name, the network interface's MAC address, and the IP address - just as it had done three years before.

Highlights:

- UAF stands for University of Alaska Fairbanks (Use After Free).
- This flaw affects the reading and display of Scalable Vector Graphics (SVG) files.
- Capable of gathering Tor browser users' identities (IP and Mac addresses) and sending them to the attacker's central server.
- Mozilla's Thunderbird email client and the Firefox Extended Assistance Release version are affected.
- This exploit has been used on Windows computers, but it can also be used on Mac and Linux systems.

Recommendations:

- Tor browser version 6.0.7 has been released, which addresses the CVE-2016-9079 zero-day exploit issue.
- Firefox ESR version 45.5.1.
- Download Mozilla's Thunderbird 45.5.1 upgrade.

We suggest that you check your Firefox and Tor browser versions. If you don't have the latest versions, you should obtain them. You'll be able to get rid of the Tor browser vulnerability with the new version.

CVSS Score	**5.0**
Confidentiality Impact	A considerable disclosure happened(**Partial**)
Integrity Impact	No impact to the integrity detected(**None**)
Availability Impact	Low impact to the availability detected(**None**)
Access Complexity	Low knowledge enough to exploit(**Low**)
Auth	The vulnerability may be exploited without requiring authentication(**Not required**)
Gained Access	**None**

Figure 2: CVSS Scores & details

CVE 2018-16983 (Restriction Bypass)

The text/html;/json Content-Type value in NoScript Classic before 5.1.8.7, as used in Tor Browser 7.x as well as other products, enables attackers to evade script blocking.

CVSS Score	**7.5**
Confidentiality Impact	A considerable information disclosure happened (**Partial**)
Integrity Impact	Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited (**Partial**)
Availability Impact	Reduced performance or interruptions in resource availability (Partial)
Access Complexity	Low knowledge is enough to exploit(**Low**)
Auth	The vulnerability may be exploited without requiring authentication(**Not required**)
Gained Access	**None**

Figure 3:CVSS Scores & Details

CVE 2019-12383 (UI Language Detector)

This vulnerability was reported in October 2017 and was successfully fixed in August 2018. Tor Browser respects privacy and security extremely seriously, thus the developers paid special attention to masking the user's IP address and attempting to prevent as many fingerprinting assaults as appropriate. They particularly said that attackers/websites should not be able to find information about their customers including time zones, UI language, or local ports.

But the attackers found a vulnerability where when you first visit a website using a non-English Tor Browser, it will ask whether you want websites to recognize your UI language.

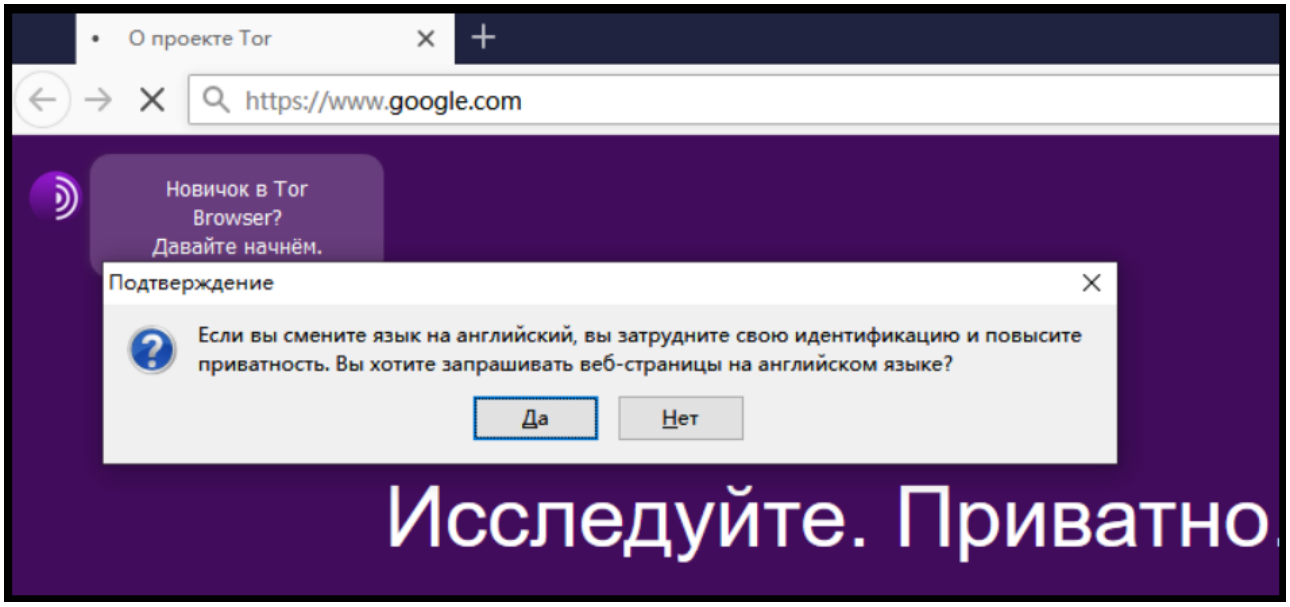


Figure 4: UI language recognition

The UI language is still identifiable even if you pick "don't transmit." Use "button" with default text: input type="submit">.

Varying languages have different default text and widths, which JavaScript can detect: "getComputedStyle(button element, null) is a function that returns the computed style for a button element. getPropertyValue("width");"

CVSS Score	**4.3**
Confidentiality Impact	A considerable disclosure happened(**Partial**)
Integrity Impact	No impact to the integrity detected(**None**)
Availability Impact	Low impact to the availability detected(**None**)
Access Complexity	The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit(**Medium**)
Auth	The vulnerability may be exploited without requiring authentication(**Not required**)
Gained Access	**None**

Figure 5: CVSS Scores & Details

CVE 2019-13075 (Info Disclosure)

There is a vulnerability in Tor Browser 8.5.3 that allows information to be exposed. Because content in that language is contained in the title property of a LINK element for a non-HTML page, it enables remote attackers to determine the browser's language through vectors utilizing an IFRAME element. This is due to a behavior in Firefox prior to version 68.

The flaw was revealed on June 30th, 2019. Since June 30, 2019, this vulnerability has been designated as CVE-2019-13075. It is feasible to carry out the assault from afar. The exploit does not need any kind of authentication. The victim must engage with the exploiter in order for it to be successful. The technical specifics are unclear, and there is no known exploit.

CVSS Score	**5.0**
Confidentiality Impact	A considerable disclosure happened(**Partial**)
Integrity Impact	No impact to the integrity detected(**None**)
Availability Impact	Low impact to the availability detected(**None**)
Access Complexity	Low knowledge is enough to exploit(**Low**)
Auth	The vulnerability may be exploited without requiring authentication(**Not required**)
Gained Access	**None**

Figure 6: CVSS Scores & Details

CVE-2021-28089 (Directory Exploitation)

The updates address two denial-of-service problems, which are detailed below. The problem of authority-only is one of them. The other problem affects all Tor instances, although it has the most serious consequences for directory authorities and relays. Once these packages are ready, we urge that everyone update to one of these versions.

Tor 0.4.5.7 resolves two critical denial-of-service issues in previous Tor versions.

One of these flaws (TROVE-2021-001) would enable an attacker to compel a Tor instance to spend a large amount of CPU by sending directory data to it. This is simplest to attack against authorities since anybody may submit to those, but directory caches might also use this weakness to download from relays or clients. The third flaw (TROVE-2021-002) only affects directory authorities, and it allows an attacker to remotely crash the authority by failing to make an assertion. Patches were sent to authority operators in order to assist maintain network stability.

CVSS Score	**7.5**
Confidentiality Impact	No impact to the confidentiality(**None**)
Integrity Impact	No impact to the integrity detected(**None**)
Availability Impact	High impact to the availability detected(**High**)
Access Complexity	Low knowledge is enough to exploit(**Low**)
Auth	The vulnerability may be exploited without requiring authentication(**Not required**)
Gained Access	**Yes**

Figure 7: CVSS Score & Details

CVE 2021-34550 (Privacy Exposure)

Tor (The Onion Router) is a protected anonymous browser that is used by those who wish to keep their surfing activities private. While it is often linked with the dark web and illegal activity, it has a large following among those who respect internet privacy, along with organizations that want to preserve the anonymity of their users, such as state groups that deal with victims of domestic violence. However, Tor has lately been shown to have various flaws that potentially expose privacy, including the CVE-2021-34550 issue.

This vulnerability affects Tor users who have not yet updated to version 0.4.6.5 or higher (also referred as TROVE 2021-006). This is true for Tor versions running on Windows, Linux, Android, and other operating systems.

Recommendations:

- Download the most current version of the Tor browser to effectively defend your network against such a Tor browser vulnerability.
- There really are no acknowledged workarounds for this phase of the repair process. The update must be installed as quickly as practical on all impacted systems and devices to resume using Tor safely.
- Keep track of new security flaws.

CVSS Score	7.5
Confidentiality Impact	No impact to the confidentiality detected(None)
Integrity Impact	No impact to the integrity detected(None)
Availability Impact	High impact to the availability detected(High)
Access Complexity	Low knowledge is enough to exploit(Low)
Auth	The vulnerability may be exploited without requiring authentication(Not required)
Gained Access	Yes

Figure 8: CVSS Score & Details

CHAPTER 02:DESEIGN ANALYSIS

Architecture Overview

Tor may be used to conceal a client's identity when browsing the open web (including websites accessible through a standard browser) or using secret Tor network services. In the first case, each connection includes multiple public relay nodes, as shown in Figure 2: I the client, (ii) the server, (iii) a Tor entry node (or guard node), (iv) a Tor exit node, and (v) a collection of Tor intermediate nodes larger than one. Because a single intermediate node is often used, we'll look at that situation next.

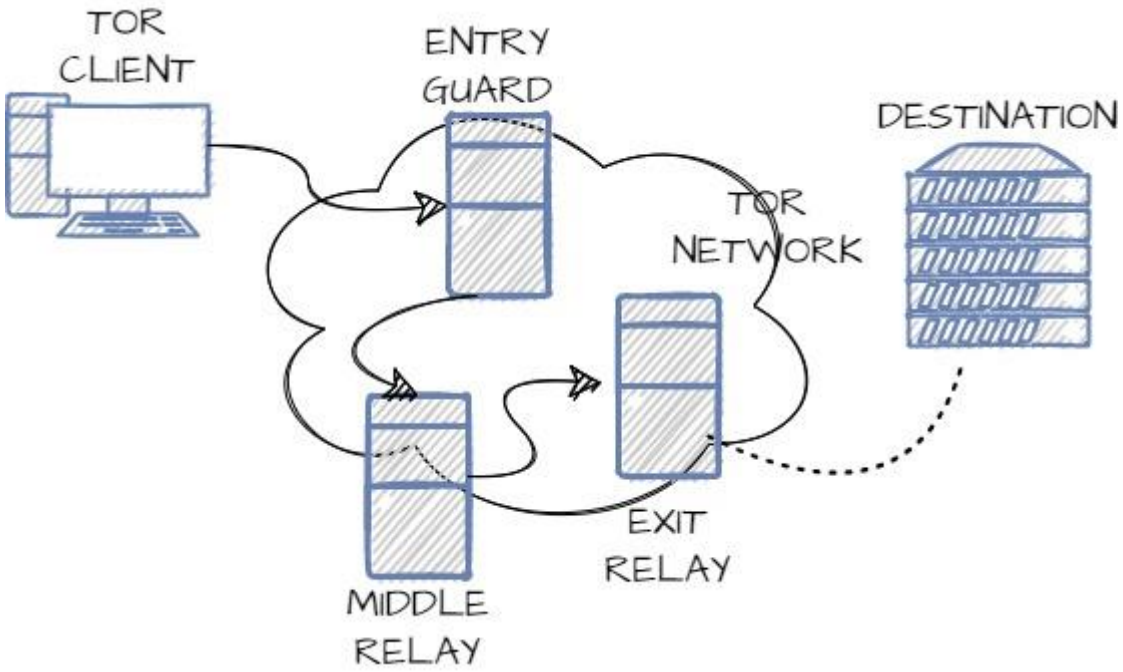


Figure 9: How Tor Browser works?

Tor also allows bridge nodes, which are extra guard nodes that, unlike public relay nodes, do not reveal their identity or IP address.

The Tor network may also be expanded by any host that wishes to join the network. Although this feature makes Tor a very scalable network, it also increases the susceptibility to man-in-the-middle attacks by design, particularly if a malevolent user controls the network's exit node and interactions with both the server really aren't encrypted.

The Tor nodes participating in the communication are selected by the client, and they form a Tor circuit, which is built by the client at the start of the connection by interacting with the circuit's network nodes and exchanging a different encryption key with each node. The client encrypts

messages using an onion encryption technique. Because each node on the circuit is the only one capable of decrypting the message it gets, each node only knows the identification of its previous and successor.

Furthermore, although the entry/guard node has been the only one directly interacting with the client, just the exit node has access to the original communication (to be sent to the outer Internet network server), which may be encrypted if encryption techniques are used (such as SSL).

Before moving on, a few additional explanations of how Tor works, and the nomenclature used with Tor are required. You'll see why breaching Tor is almost difficult without tremendous resources as we go along, although there are several parts of Tor that may have been compromised.

The Tor relay network is managed entirely by volunteers. Anyone, even you, may set up a server to act as one of the hundreds or even thousands of relays utilized by Tor users. Your server could simply "remove the outer packet and send the inside envelope" to the next location if you volunteered.

Bear it in mind if you're looking at IP addresses as part of a Tor inquiry. It's possible that the IP address you think is your target is instead an unwitting volunteer operating a Tor relay.

The number of users is perhaps the most illuminating feature of Tor, since more users increase anonymity, making it harder to discover one person among many. As per the Tor Metric Portal, over 750,000 Tor users utilize over 6000 relays throughout the globe.

Furthermore, even if the Tor circuit could be broken, securing international cooperation to identify Tor users adds another layer of legal and political concerns. To summaries, whenever a victim gets a harassing e-mail that looks to have originated in Italy, the suspect was not physically present in Italy.

Tor is a worldwide overlay platform of relays that aids in the accomplishment of user Internet traffic privacy and anonymity.

The Tor network establishes a virtual circuit for each transmission that consists of at least three subsequent, randomly chosen relays. The Tor client at the source system downloads knowledge about the relays from a directory server. The Diffie–Hellman key exchange protocol is used to exchange encryption keys with the specified relays. The data packets are encrypted several times at the source node, one for each relay node using that relay's encryption key before being sent to their destination. As a result, the entering node decrypts the outermost layer of encryption, while the exit node decrypts the innermost layer of encryption. To determine the next hop address for a received packet, each relay node decrypts the message using its unique decryption key. As a result, every Tor node knows about relay nodes that are just one hop distant from it. The inner-most layer of encryption is decrypted at the exit node, and the unencrypted data packet is sent to its ultimate destination. As a result, the

privacy of consumers' data is maintained until the very last step. Data between the last hop and the destination is likewise encrypted when using https through Tor.

Furthermore, the Tor browser changes its routing every 10 minutes to protect users' privacy. Figure 9 shows data routing across the Tor network.

The Tor network, on the other hand, ensures anonymity by assuring that even the overlay's relay nodes are aware of the virtual circuit's predecessor and successor relay nodes. To increase the anonymity of the virtual circuits, each one is created with a different collection of relay nodes.

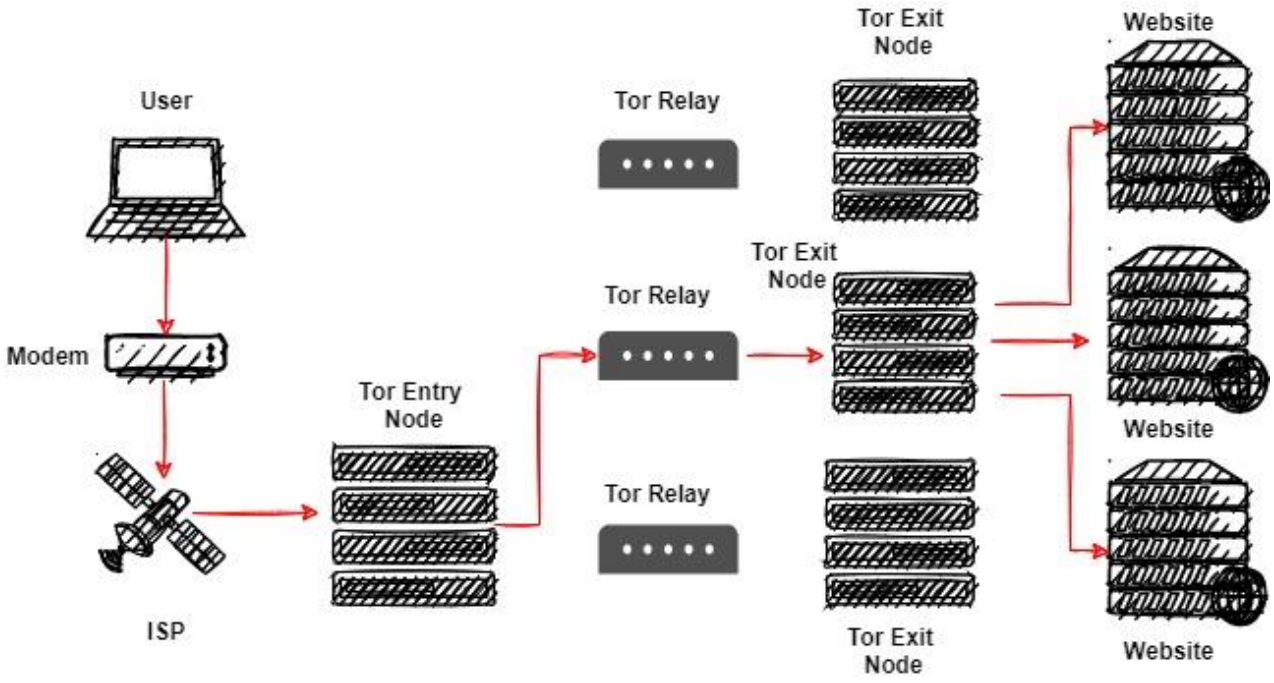


Figure 10: Simplified Inner Tor Connection

Data Collection Methodology

We put up a Tor router on a 1 Gb/s network connection to better understand real-world Tor use. During December 2007 and January 2008, this router was added to the presently deployed network. We were able to capture a big volume of Tor traffic in a short amount of time thanks to this arrangement. Our node was continuously in the top 5% of routers in terms of bandwidth among the approximately 1,500 routers identified as Running by the directory servers at any one moment while it was up and running.

We recognize that while gathering data from an anonymous network, major privacy considerations must be addressed. In most circumstances, the information we record — which contains at most 20 bytes of application-level data — cannot be used to connect a sender with a recipient since Tor is intended to resist traffic analysis from any one Tor router. When deciding what information to record and what information was too sensitive to keep, we carefully evaluated the privacy concerns. In the end, we decided to record data from two sources:

To begin, we modified the Tor router to keep track of circuits made via our node and cells routed through our node. Second, we only collected enough data from the exit traffic that was routed via our node to capture up to the application-level protocol headers.

It was necessary to run the router twice, with two different exit policies, in order to maximize the number of entry and exit connections that our router observed: (1) running with an open exit policy (the default exit policy) allowed our router to observe a large number of exit connections, and (2) banning all exit traffic granted the router to observe a large number of clients.

Logging of Entrance/Middle Traffic: We operated our network with a strictly limited exit policy to gather data about Tor clients. From January 15 through January 30, 2008, we operated our Tor router in this setup for 15 days. Minor changes were made to the router's code to allow for more logging. The time the cell was received, the previous hop's IP address and TCP port number, the next hop's IP address and TCP port number, and the circuit identity associated with the cell are all recorded for every cell routed via our node.

Exit Traffic Logging: From December 15–19, 2007, we operated the Tor router with the default exit policy for four days to gather statistics on traffic leaving the Tor network. The default policy is the most popular for routers that accept exit traffic. Our router routed about 709 GB of TCP traffic leaving the Tor network during this period. We used tcp dump on the very same physical system as our Tor router to collect data on traffic exiting the network. Using the "snap length" option, Tcp dump was set to collect just the first 150 bytes of a packet (-s). This limit was chosen to allow us to collect up to application-level headers for protocol identification. Because an Ethernet packet is 14 bytes long, an IP header is 20 bytes long, and a TCP header with no options is 20 bytes long, we could only collect 96 bytes of application header data. To identify application-layer protocols, we employed ethereal, another tool for protocol analysis and stateful packet inspection. We filtered away packets having an origin or destination IP address of every live router broadcast during our collecting period as a post-processing step. Only exit traffic remained.

Pros of using Tor browser

- Tor protects you from anti-spies by preventing others from monitoring your online activities.
- Tor works to make all users seem identical so that no one can identify you based on the features of your browsing or devices.
- Numerous-layer encryption: To keep you anonymous, your traffic on the Tor network is diverted and encrypted multiple times.

- Free admission: The Tor browser provides you access to internet sites that your network has prohibited.

- Your internet activity is hidden. When you stop using it, your browsing history and cookies are instantly reset.

Cons of using Tor browser

- The Tor network is a useful tool for concealing online activities. Tor, on the other hand, is not without flaws and mistakes. There's always the possibility that attackers may take control of and monitor many nodes. It is feasible to discover the identity of a user if the route's entrance point and exit node are controlled by a single player.

- Vulnerabilities in the Tor Browser, as well as the NoScript and HTTPS Anytime plugins, will never be totally removed. In the past, security researchers have been successful in smuggling harmful software into the Tor Browser by exploiting holes in extensions.

- Furthermore, anonymity on the Tor network may be assured only unless you do not leave clues of your identities in the form of cookies, other add-ons, or BitTorrent activity.

Threat model

After analyzing the attackers' motives and aims, I'll examine how they accomplish them. The threat modeling of persistent CSRF attacks is shown in the diagrams below, which show the vulnerabilities that may be exploited by attackers. The level-0 Data Flow Diagram (DFD) of stored CSRF attacks is shown in Figure 3.

First and foremost, the attacker must investigate the target website to discover which features may be useful to him (e.g. financial transaction). On the target website, the attacker will need a legitimate user account. The procedures are shown in Figure 3, with the numbers indicating the navigation stages.

After finding the vulnerabilities, the intruder must carry out various tests to ensure that the malicious operations are effective. As fast as it meets the requirements, the attacker develops a malicious link that the user is persuaded to click in order to carry out the CSRF attack.

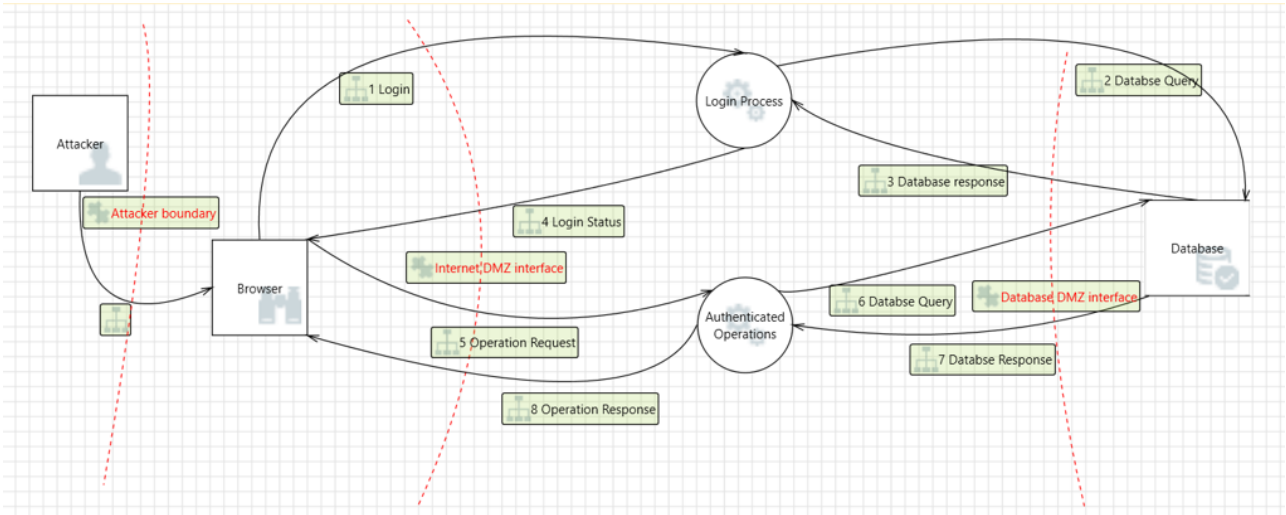


Figure 11:Tor Website fingerprinting threat model

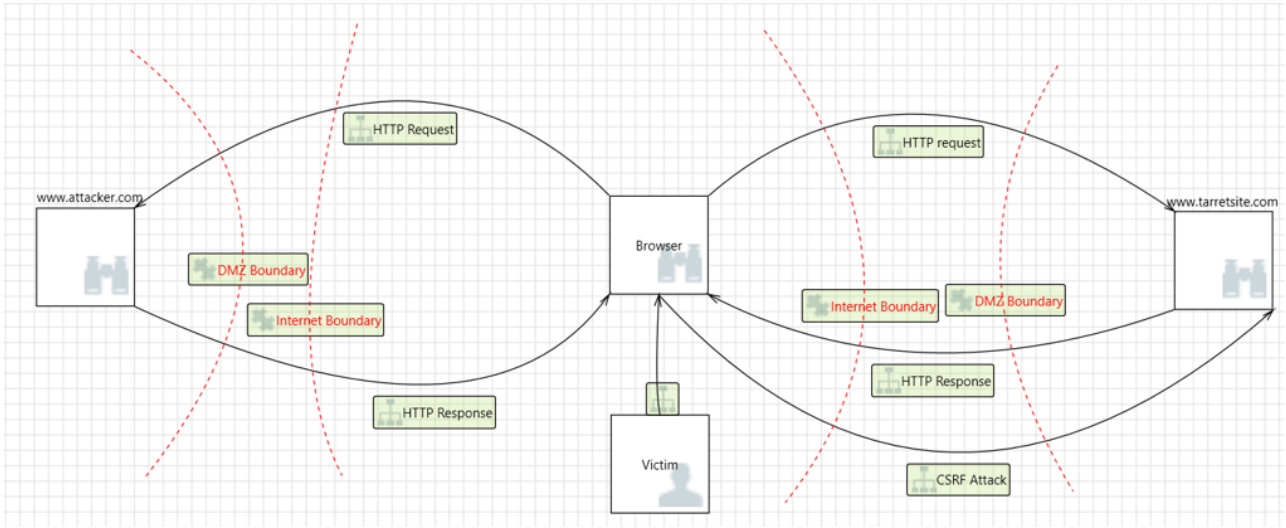


Figure 12:Reflected model of CSRF attack:

Assets to Threat Model Tracking

The sandboxes mentioned in the architectural overview act as a barrier between the web page and the rest of the browser. Accessing the assets outlined in chapter one, such as cookies, the clipboard, and user history, necessitates communicating with the browser kernel, which keeps track of each sandbox's unique privilege information, describing what it can (and can't) access. Tor attempts to reduce the likelihood of an exploit against the system by putting this sandbox on the browser's edge. Each sandbox has its own rendering engine and file permissions. If the TabProcess wants to do more than merely change the state of the DOM, it must connect with the browser kernel API through a Channel, which decides which actions it is permitted to do and executes them.

Tor requires a means to manage input and output from the browser process to the outside world as a result of the sandboxed tabs explained before. To guarantee the security of another asset, user data,

network requests, mouse and keyboard inputs, and other external communication messages must be sent to the proper sandbox. Inter-Process Communication (IPC) is how Tor handles this. In a nutshell, this mechanism is responsible for transferring messages from external inputs to the appropriate website sandbox. It prevents messages from being delivered to TabProcesses that aren't supposed to be receiving them, as well as the leakage of user data. Figure 4 depicts the communication between these two processes.

Because the I/O handler is separated from the rendering process, any errors in the I/O process will not cause the rendering engine to hang or impair performance. The IPC process further decreases the risk of an exploit by prohibiting messages from being viewed by TabProcesses other than the one for which they were intended. If this wasn't in place, a maliciously created website may exploit a weakness in the rendering engine to collect data from the user's keyboard, network traffic, or mouse clicks in another tab. Because to the unique treatment of I/O messages, the chances of such exploit happening are reduced.

Tor users may install third-party extensions to extend the capabilities of Tor. Tor can protect the rest of its system by separating extensions into three sandboxed processes: Content Script, Extension Core, and Native Binary. While Tor cannot reasonably control the content of an extension or the specific functionality it may provide, it can protect the rest of its system by separating extensions into three sandboxed processes: Content Script, Extension Core, and Native Binary. Tor grants various permissions and degrees of access to the system to the three processes. Native Binaries are the only processes that have full user rights on the host computer, and they can only receive messages from the Extension Core. Although the Extension Core has TabProcess capabilities, it can only communicate with the DOM / web content through XMLHttpRequests and Content Scripts. Content Scripts can change the DOM of a single web page, but they can only communicate with other processes through the Extension Core.

Because functionality is divided across three processes, each with its own set of permissions, each process's capabilities are severely constrained. Content Scripts, the weakest link in this design, are unable to interface directly with Native Binaries, the sole process that permits arbitrary code execution. An attacker must first exploit the Content Script to send a malicious message to the Extension Core before successfully executing code on a user's system. They must next locate another attack in the Extension Core in order to transmit the malicious message to the Native Binary. To execute code on the host system, the attacker must then exploit a vulnerability in the Native Binary.

A hacker must be able to attack vulnerabilities in three different processes in order for an extension to effectively escape the sandbox.

In its sandboxed design, Tor's assets are well-organized. Security was built in from the start. Some significant components of this approach are highlighted in the threat model above. Tor makes it exceedingly difficult for an exploit to break out of its sandbox and get access to or impact other processes using a mix of defense-in-depth tactics and strict adherence to the principle of least privilege.

CHAPTER 03:CODE INSPECTION ANALYSIS

Inspection Selection

For the inspection section I selected the source files main.c, risky_options.c, subsysmgr.c, tor_main.c, config.h, statefile.h which are included in the tor browser source code folder. The main.c file contains the main functionalities of the tunneling and coupling of different nodes and relays of the onion network. It is a good place to search for the vulnerabilities. susbsysmgr.c file is the important code part on hiding user's identity to which took most of my attention as we all attached to the onion routing with solemn goal of acquiring anonymity.

From here onward I will explain the vulnerabilities that were extracted from the files that could harm the CIA triad.

Code Inspection Results

media.webaudio.enabled

In Firefox 24 and Tor Browser Bundle, the Web Audio capability is disabled. Firefox 25 enabled it, and it is currently enabled by default. A considerable number of possible vulnerabilities were discovered in this component after examining security-relevant defects in Firefox.

Deactivate at the Low or Medium security level as a recommendation.

media.audio_data.enabled

The Audio API was a test API that was replaced by the Web Audio API. It was enabled in Firefox 24 and Tor Browser Bundle; however, it is now deactivated in Firefox 28.

Deactivate at the Low security level, as a recommendation.

layout.css.flexbox.enabled

This preference has been true by default since Firefox 22, and it was removed in Firefox 28. The revision that removes the preference is located at https://hg.mozilla.org/mozilla-central/rev/1a09d295 aa1c, and it is simple enough that it could be re-added or copied to other styles.

gfx.downloadable_fonts.enabled

By default, Tor Browser Bundle can download, parse, and utilize web fonts in the.ttf,.otf, and .woff formats. Mozilla did a security review of downloaded fonts, and their main worry was much like ours: that the font parsing subsystems would be vulnerable to an attacker. Firefox incorporates the OpenType Sanitizer to counteract this problem.

The OTS Sanitizer looks to be successful in preventing problems that can be exploited. However, no program is flawless, and Font Parsing on Windows is causing a lot of anxiety.

Disable at the High security level, as a recommendation. Normally, I would advise deactivating them at the Low or Medium security levels, however the Tor Browser Bundle team has said that they favor remote fonts over local fonts for the purpose of user fingerprinting.

gfx.font_rendering.graphite.enabled

The Graphite Font Shaping feature is capability used to reproduce complicated letters better precisely in South-East Asian languages. The functionality has been set automatically from about Firefox version.

At least one security-relevant flaw in the recent year was detected in graphite parsing, along with three in the previous two years. The library is not updated by Mozilla, and although Mozilla states they fuzz it, it is just not apparent how frequently with relation to new releases, or how completely. It was submitted to a security evaluation by Mozilla.

Recommendation: For South-East Asian or other types of application, deactivate at the Medium or High security level. For other places, deactivate at the Low security level

gfx.font_rendering.opentype_svg.enabled

SVG support in OpenType fonts is a feature that allows you to use SVG within font files to create colorful, animated, or more dynamic glyphs. This functionality was deactivated in Firefox ESR 24 but is now accessible in (at least) Firefox 29. iSEC was unable to locate any security reviews or security-related problems for this feature. Because it does not seem to be supported in any other browsers, iSEC does not foresee widespread use of this feature on the Internet. A rival option, SVG fonts, is implemented in Chrome, Safari, and Opera.

Block at the Minimal security level, as a recommendation.

javascript.options.asmjs

The ASM.js functionality in Firefox is controlled by this option. If you disable this function, JavaScript will still be executed, but not using the more efficient ASM.js engine. There have been a few problems discovered in the ASM.js source, however due to the confined environment, exploitation may need extra trickery, since many popular exploit methods may not be applicable.

Recommendation: At the Medium security level, disable.

dom.indexeddb.enabled

For user fingerprinting considerations, the IndexedDB functionality is presently unavailable in Tor Browser Bundle. In response to these concerns, iSEC would want to express concerns about its security since the functionality has a history of security flaws. Despite the fact that Mozilla has undertaken a security audit, its extensive feature set and API suggest a huge and complicated codebase with potential vulnerabilities.

Recommendation: Continue to set the security level to 'None' or 'Low'.

jar: protocol

The jar: protocol handlers are a Firefox-specific function that is primarily utilised on Intranet sites and is virtually ignored on the rest of the Internet. It is a great candidate for blocking because to its uncommon character, modest sophistication, and lack of popular usage.

Recommendation: Use the provided patch to disable at the Low security level.

Tor main () is the Tor main program's entry point (\ tor-0.3.1.8 \ src \ or \ main. c). In actuality, main () (tor-0.3.1.8 source) or tor main.c are the files that call it.

The main () function in the source file (test *. c) that provides unit testing may be connected to main. c, which is why they are separated.

There is no name conflict since the latter does not have a main () function with the same name. Let's have a look at the major connection between the three first (). It stores the integer value produced by tor main () to the local variable r, then processes it depending on r's value:

If 0=r=255, the caller (often the CRT starting procedure that initialises the Tor process Runtime Environment) passes the value of r to main; otherwise, 1.

Furthermore, at the start of tor main (), an integer variable result is defined and set to 0. The inner workings of tor main () will be revealed.

The result is then set to the appropriate value before being returned to the program().

The main () command specifies the two arguments you'll get: argc (number of parameters when running the tor command) and argv (list/array of specified parameters).

Under some conditions, Tor main () will utilise these two arguments. When tor init () is invoked, for example, it is handed over, and one of tor init's major jobs is to parse the command lines supplied in argv. Launch the Tor system according with user's desire using these parameters.

Command line arguments are used in the methods for transmitting the number and strings of the Tor. The first code issue compilation block, which is relevant to the Windows platform, appears after tor main () initialises its own return value. The content of a condition is written as code that can be executed.

For a 32-bit Windows platform, and it doesn't specify "heapenableterminationontermination on once the heap data is damaged," which sets the enumerated constant value to 1. (The heapenableterminationonissue assumption is that the OS Heap Manager identifies faults in every heap that the process uses.)

The WER service [Windows Error Report] will be called by the heap manager, and the process will be terminated. The procedure cannot be deactivated after this function has been enabled.)

HeapSetInformation () is called if this method has been defined, and heapenableterminationonconfigurationuption is supplied in as the second argument to enable this function.

The heap handle to be set is the first argument of HeapSetInformation (), which is normally returned by HeapCreate () or GetProcessHeap;

The heap handle is NULL since tor main () did not generate one before. The "low fragment Heap" is selected automatically starting with Windows Vista.

(Low-fragmentation heap, LFH) is used or created by the programme. (v = vs.85) http://msdn.microsoft.com/en-us/library/windows/desktop/aa366750

Note that the OS will continue to execute the programme even though HeapSetInformation () is not invoked. As a result, the return value of HeapSetInformation () should be 1.

The key is to use SetProcessDEPPolicy () to permanently activate the Tor master process's Data Execution Protection (DEP) function.

Tor is a network programme that transmits and receives data periodically across the internet. Remote code execution might be caused by any encoding flaws in the software.

Buffer overflow/stack overflow threats may be mitigated by include this call.

SetProcessDEPPolicy () is an API function in Kernel32.dll that utilizes "dynamic connection upon loading" to overlay its memory of the Tor process (see "system prerequisites" in the related MSDN documentation). As a result, you must first retrieve the handle of this module using GetModuleHandleA (), (The program that utilizes "dynamic link during runtime" will use LoadLibrary/Ex ()).

With etProcAddress (), you may acquire the address of the SetProcessDEPPolicy () function, assign it to the typedef type definition, and if the declared function pointer "setdeppolicy" can be resolved to the function's address, you can immediately use setdeppolicy to "attempt" to open DEP.

Using GetProcAddress () to get the address of SetProcessDEPPolicy () on previous versions of Windows will fail to parse.

Examine the pointer. It should not, and it cannot, initiate Data Protection (DEP) for the Tor main process if it is nil. Furthermore, if the system is capable of resolving

SetProcessDEPPolicy () address, but failing to call it will not harm you, so don't worry about handling the problem; simply call it-even if you can't.

Tor will continue to operate even if DEP is enabled. This is entirely dependent on secure coding awareness. Pass 0x3 to the DWORD argument dwFlags of SetProcessDEPPolicy, as described in the MSDN page (). It shows that both PROCESS DEP ENABLE (0x00000001) and PROCESS DEP DISABLE ATL THUNK EMULATION (0x00000002) are enabled.

The source code annotations are consistent. Finally, the PROCESS DEP ENABLE (0x00000001) bit indicates that the DEP cannot be deactivated throughout the Tor process' lifetime. When the initial condition compilation block is completed, it utilizes all of Windows' extra security features for apps.

After the platform's unique code block, can the Tor process dump stack information after a crash for further debugging and analysis? To setup the backtracing handler, the procedure Configure backtrace handler () is used (tor-0.3.1.8 src common backtrace. c).

Configure backtrace handler () takes a reference to a character constant that may be used to access the Tor application's version information using the get version () helper procedure.

To relinquish the static traffic, Configure backtrace handler () first runs the macro tor free () (tor-0.3.1.8 src common util. h — 83).

Tor free () may safely release the NULL pointer and make the relevant memory address NULL since the given global variable bt version is a NULL pointer.

It then uses tor asprintf () to report the relevant program launch information to the console/shell, and then runs install bt handler () (in the same source file) and passes the return value to tor main, depending on whether the version information is retrieved ().

Install bt handler () returns -1 if the callback function is not registered; otherwise, 0 is returned, and tor main () is informed to "Hook "!

Take a look at the backtrace. c. We can see that install bt handler () only returns 0 if the USE BACKTRACE option is not set during building.

The Backtracking handler is not registered by the caller. Otherwise, install bt handler () will be aimed at signals such as crashes (SIGSEGV, SIGILL, SIGFPE, SIGBUS, SIGSYS, SIGIO), and crash handler will be installed (),

To create STACK tracing information, the latter uses backtrace ().

Only configure backtrace handler () and install bt handler () will be executed when the programme begins and runs properly; when the program starts and runs abnormally, configure backtrace handler () and install bt handler () will be called.

To crash or receive one of the previous six signals, crash handler () and backtrace () are invoked (the latter two are CallBack). The last abort () call logic in crash handler () may be used to ensure that the crash handler () operation must only be cancelled if it crashes.

Project Specific Checklist

Inspection Summary

Tor Browser is a PET that lets users to mask content and URLs of requested web sites from external observers, allowing for private and anonymous online viewing. However, WF attacks, which let an external, passive attacker to discover the identity of the requested web sites between the Tor client and the guard node, jeopardize this protection.

In this thesis, we demonstrate that a Tor-provided configuration advice for Tor Browser users' privacy and anonymity may be less secure than the default option, particularly when considering known website fingerprinting techniques. We demonstrated that, contrary to popular belief, removing JavaScript in Tor Browser may render users more susceptible to website fingerprinting attacks under surveillance and censorship situations.

We get a recognition rate of 42 percent when JavaScript is removed, compared to 35 percent when it is enabled, utilizing just previously suggested classifiers and publicly accessible information on a set of blocked URLs. Our findings suggest that a reasonably resourced attacker may effectively discover a Tor browser user's ultimate destination. This strengthens the argument against Tor PET's claims of privacy and security guarantee, which promises to improve users' civil rights such as privacy and freedom of expression all around the globe). Our findings suggest that this Tor suggestion may wind up harming people who use Tor to circumvent censorship. It simplifies the attacker's classification effort since he doesn't have to account for the randomization of web sites while creating classifiers to identify visited web pages.

We built up an environment to monitor traffic for a collection of genuine restricted web sites by evaluating the Tor specification, source code, Tor browser design papers, and suggestions. Then, to clean and interpret our data, we constructed a Python traffic analysis framework. We also completed considerable code refactoring for a nine-year-old Tor dissector that we're working on with the Wireshark development team to include into their Wireshark baseline code. We also altered the Tor source code to leak the Tor client's keys. While this enables for the decryption of web traffic (which isn't affected by website fingerprinting attacks), we feel it might be beneficial as a Tor protocol instructional tool. We next utilized Dyer's Framework to categories and assess how well an attacker can differentiate between different web page instances.

In light of our findings, we feel it is necessary to examine the hazards associated with Tor's JavaScript settings suggestion. Our findings suggest that there is a trade-off in this setting not just between security and usability, but also between security and privacy. That is, although deactivating JavaScript protects users against well-known security and privacy breaches connected to active content, it also renders them more susceptible to current website fingerprinting attacks. As a viable alternative to deactivating JavaScript, we strongly advocate Tor's alternative design requirement to sandbox JavaScript.

We've also gained fresh insight into the additional danger presented by Tor Browser's JavaScript settings, which should prompt Tor browser developers and users to listen closely to this option. We also attempted to present a more realistic (although early) study of website fingerprinting attacks in Tor by taking into account a feature that is often kept off by normally in other website fingerprinting attacks research. Finally, we are making the tools we built available to the public in order to foster continuing study and conversation in this vital topic.

CHAPTER 04:IS TOR LEGAL?

Tor (The Onion Router) is a powerful tool for anonymous internet browsing and access to the dark web. To quickly address a question that many people have, using the Tor browser is legal practically everywhere in the globe. However, it's always a good idea to double-check if this is the situation in the nation where you're now staying.

In general, if you use Tor to view a YouTube video or do a Google search, you won't have to worry about the cops apprehending you. However, you are still liable to the law if you're using Tor browser to profit from its greater anonymity while engaging in criminal acts. You might be in serious danger if you're discovered. Tor is a fascinating phenomenon from a legal standpoint. As far as the Internet stays 'global and open,' anonymity will be a component of cyberspace, whether via Tor or another network. Anonymity, on the other hand, may be a mixed gift, and Tor presents a slew of legal issues.

Edward Snowden and Tor

Tor is often associated with illegal activity. Because many criminals on the dark web utilize Tor, this is the situation. Users may be more truly anonymous thanks to the browser, which makes it a perfect tool for them. What is deemed criminal or unlawful varies greatly from nation to country. Cybercrime is no exception. For example, practically everywhere in the globe, selling heavy narcotics and arms online is prohibited. Giving a negative view about a tyrant or leaking governmental information, on the other hand, may be a criminal in several nations.

Edward Snowden's case is an illustration of the grey area that internet actions might fall into. Snowden disclosed classified information concerning the NSA and its spying to the general public in 2013. As a result, the US government has issued a warrant for his arrest. According to American law, Edward Snowden's actions (discovering state secrets) are unlawful. Many feel, however, that Snowden was correct in his actions. After all, don't all people have a right to know who is following their every move and eavesdropping in on their interactions? This is a moral quandary in which privacy is crucial.

Edward Snowden, for example, utilized Tor to leak classified NSA papers to the general public. Others should use the Tor browser every time they go online, he said. So according to him, this is among the few methods to remain anonymous while exercising your freedom to freely publish. Governments and important developments, such as the National Security Agency in the United States, won't be able to track you there – for the most part. Snowden utilized the browser to engage in illegal activities. His appeal to others, on the other hand, was not criminal: he simply urged that more people secure their online anonymity by using the Tor browser. Although there are some

debates about how secure the Tor browser is, the reality remains that it makes it far more difficult for governments, websites, and other groups to eavesdrop on individuals online.

Tor is now used by law enforcement in criminal investigations

Tor is becoming a commonly used framework for national law enforcement. For law enforcement's purposes, the Tor project describes three key activities:

- ‘Online surveillance: Tor enables authorities to browse dubious websites and services without leaving traces. Investigations may be impeded if the system administrator of an illicit gaming site sees repeated links from official or security agencies IP addresses in use records.

- Sting operations: Anonymity also permits law enforcement to conduct internet "undercover" operations. If conversations involve IP ranges from police addresses, the cover is exposed, regardless matter how excellent an undercover officer's "street cred" is.

- True anonymous tip lines: While digital anonymous caller lines are popular, they are significantly less practical without anonymity software. Although a name or email address is not connected to material, sophisticated sources realize that server logs may rapidly identify them. As a consequence, tip line websites that do not allow anonymity are reducing their tip sources.

Tor is also used for generic investigation, intelligence gathering, and infiltration, as seen by the current takedown of Silk Road 2.0, which was based on the Tor network.

National law enforcement's use of Tor raises a variety of interesting legal issues, such as whether there are any restrictions on law enforcement using Tor to collect evidence, and whether there are any restrictions on law enforcement processing data available via Tor or within Tor if we consider it publicly available data.

The legal limits that are normally defined in national legislation for law enforcement actions might vary substantially from one country to the next. This is particularly true in the case of digital evidence collection, which poses difficulties for domestic procedural law. Many of these issues might be addressed by using Tor to gather evidence. In certain legal systems, for example, the fact that now the agency gathering evidence via Tor is anonymous may raise issues about 'deception' in criminal proceedings, or otherwise obstruct the use of these evidence in court.

Tor and Other Unlawful Activities

Apart from Snowden's case, there are numerous other examples of Tor being used for illegal purposes, most of which are less well-known or controversial. The most well-known example is the Silk Road. Silk Road was a website that sold illegal weapons, drugs, and other items. The website

has already been traced directly back and taken down, but similar marketplaces are bound to exist on the dark web. Even if Tor isn't illegal in and of itself, it allows users to access these platforms and pages anonymously. As a result, you could argue that Tor obliquely supports criminality by making criminals less easily identifiable.

When something as significant as the closure of Silk Road occurs, heated debates about whether or not the Tor browser is required tend to erupt. This then leads to more general discussions about online anonymity. People will have to choose between fighting cybercrime and being concerned about their online privacy. It's a difficult balance to strike, and it's often not just a legal, but also an ethical issue.

Tor is frequently associated with criminal activities on the dark web as a result of high-profile cases like Silk Road. Tor allows customers to create websites that are only obtainable by other Tor users, which hurts Tor's case. Illegal marketplaces are simple to set up and accessible to the 'right' audience right away. The Tor browser encourages such activities, even if that isn't why the platform was created in the first place.

Privacy and Tor

Although criminals regularly use Tor, this does not entail that the browser is criminal by definition. Tor, on the other hand, aids in the creation of an online ecosystem that prioritizes privacy and freedom. With Tor, you may browse without being watched by others (such as hackers, governments, and your workplace). It protects your right to release as well as your freedom of expression. Tor is a fantastic project when it comes to internet safety and anonymity.

Despite this, Tor also has to contend with a number of security concerns. Tor's security has been circumvented and even breached by a number of government agencies, including the CIA and FBI. Investigators were able to pinpoint down people who were linked to illicit activities in this manner. Furthermore, a flaw in Tor's architecture allowed Linux and MacOS IP addresses to be exposed in 2017, therefore nullifying the browser's anonymity.

Always keep in mind that even if you use the Tor browser, you and your internet information may be exposed in some manner. This should not be an issue for you as long as you follow the (local) law. In most cases, using the Tor browser isn't an issue in and of itself. Even said, you could always utilize Tor in conjunction with a VPN connection to guarantee you're operating with the greatest possible online security. Your web traffic will be protected in two ways, and you will have an additional layer of encryption to protect your privacy and anonymity.

Video Link:

https://mysliit-my.sharepoint.com/:v:/g/personal/it18029314_my_sliit_lk/EYMKcrL7i0BGh_oiYGfZfcsB0DJN0blwPU_tzTDmYNdQIw?e=rX5yxo

Github Link: https://github.com/Nisasala/SSEAssignment_IT18029314/tree/main