Security against attacks in TOR networks

Vijaya Lakshmi.V
Computer Science & Engineering (P.G)
M.V.J College of Engineering
Bangalore, 560067, India
Vijayalakshmiv12@gmail.com

Abstract— **TOR(The Onion Router) network is a system that provide anonymity for its client by routing the client packets via various volunteering networks, such routing conceals the clients location & activity thus making traffic analysis or network surveillances impossible by the intruders, hence TOR is widely used network for purpose such as personal freedom, secure communication & confidential business. Yet there are vulnerable flaws in the nodes & routing that can be utilized by intruders to de-anonymize the network & monitoring the whole network, such attacks can be avoided by conducting simple measures making TOR more secure & attack resistant because "Prevention is always better than cure".**

Keywords- attacks; anonymize; tor security; tor vulnerability; security in tor; tor network;

I. INTRODUCTION

The Onion Router is one popular anonymizing network that makes uses of the concept of relay networking & encryption methods. It mainly consists of client & relay nodes which together form a virtual tunnel through which the user configures software such as his browser to establish connection via the tunnel. So as far as user ISP is concerned TOR is best project but how far is it secure if encryption fails, nodes gets compromised or trusted nodes collude? Such flaws can be utilized by an intruder resulting in revealing of confidential information.

Exit nodes: This is the most significant node that plays key role in forwarding user data to the external network where the data is usually in decrypted form, once an exit node is compromised the whole network gets deanonymised.

Ever since the TOR was introduced various attempts are made to deanonymize the network through software loop holes & its network policies. To overcome such threats we can adopt measures at client side & in the network preventing the intruder from attacking on the network, firstly we will define the possible attacks & their deployment then realize the defense mechanism we can adopt to avoid attacks & make security stronger.

II. BRIEF OVERVIEW OF TOR

A. *TOR Components*

Before analyzing the security issues let's understand basic terms & concept of TOR networking. The Tor comprises of the following components:

- *Entrance/Guard Node*: As the name says it is used as the entrance for user data via which it is encrypted & forwarded into the network.
- *Relay/Mix Node*: It forwards the data which received from entrance or other relay node to another relay or exit node, each relay node is chosen randomly.
- *Exit Node*: It marks the end of the network & exit nodes are chosen based on networks exit policy.
- *Client*: User whose data is injected into network through onion proxy.
- *Onion proxy*: It is the application used to hide user's identity.
- *Directory Server*: Stores router's information & public keys of all nodes
- *Cell*: Unit of transmission, fixed in size. Each cell is 512bytes.

- *Onion router*: A node that routes cells.

B. *Definition*

TOR is a circuit switched network which makes use of cryptography & TLS (Transport Layer Security) concepts to provide its client with pseudoanonymity. The cells are unlinkable thus preventing eavesdropping.

The entry node is the only node that knows the client's identity, the exit node is the only node that knows the destination's identity and the actual TCP data sent, and the middle node exchanges encrypted cells between the entry node and the exit node along a particular circuit.

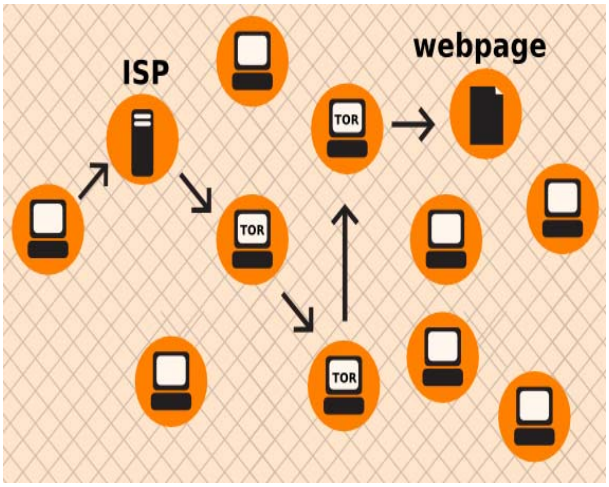

Figure 1: TOR working

C. *Encryption*

The cell is encrypted in form of layers, for each hope a layer is decrypted/removed & cell is forwarded, so on at the end the last layer is decrypted & forwarded to destination. During the forwarding or hopping the actual content is invisible to the relay/mix nodes, thus the name onion routing.

Node to node communication is secured by TLS & public keys are used by communicating clients for establishing AES session keys for each hop.

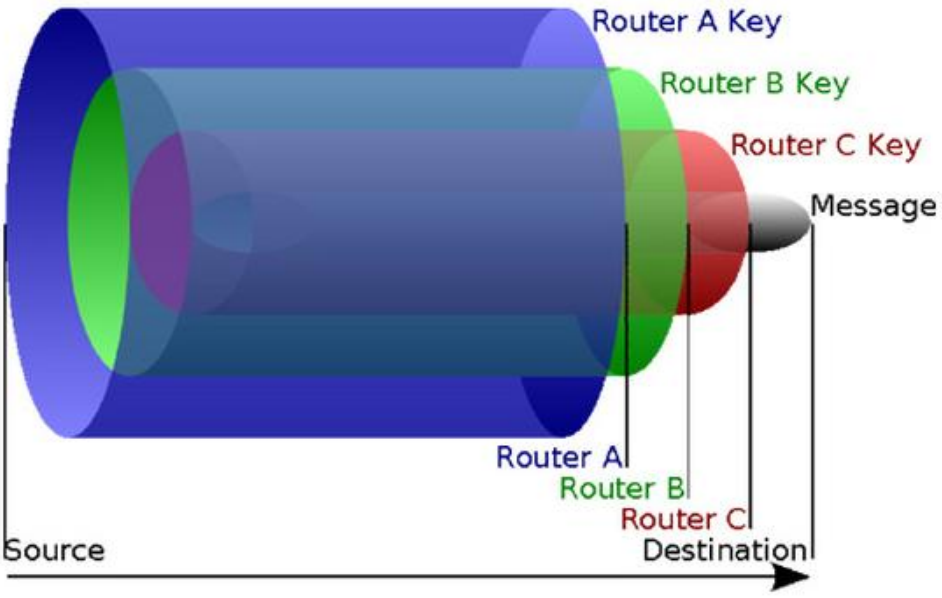

Figure 2: cell encryption in TOR

III. PROBLEM DEFINED: ATTACKS

This part defines the various possible attacks & how the nodes get compromised.

A. *Server attack*: This is the basic form of attack where the intruder compromises some server with high bandwidth & resources then it introduces it into the TOR system. Since the compromises server has high resources its chances of being chosen as guard or exit is high, these are the nodes where data is free from any encryption & internal contents can be read.

B. *Illusion attack*: The intruder compromises any random server because prime goal of intruder here is to seize the network working, intruder reports falsely to the TOR about its bandwidth & resources, since directory servers never actually verify the node capacity instead stores the information is given by the node, when most of the cells are hopped to compromised node, the server cannot handle the traffic & fails dropping all cells.

C. *Application attacks*: It is purely a skilled hacker attack, firstly hacker will analyses the system thoroughly for software flaws in system setting & network. Using this loop holes the hacker surpasses the security checks or may breach the security system, this type of attack can be performed at any node.

D. *Circuit attack*: This attack is done at & only at entrance & exit node, intruder compromises either entrance or exit & extends the attack to its peer, in order to compromise nodes before attack it can perform illusion attack &spread the attacks to peers.

E. *Timing attack*: Intruder correlates his timing & flow characteristics with the client via entrance, theoretically it can be proven but there is no practical evidence about it.

F. *Cell based attack*: The intruder must first be monitoring the entry & the exit nodes & must also have knowledge of who are using the network. The attacker then selects a random signal selecting an appropriate time and changes the cell counter of destination correlating to the signal. In this way, intruder embeds a signal into the destination. The signal will be transmitted along with the actual message to the entry onion router connecting to destination. Intruder at the entry onion router will record the variation of the received cells and recognize the embedded signal. If the same pattern of signal is recognized, the intruder confirms the communication relationship between source & destination.

G. *Browser attacks*: To browse the Internet anonymously using Tor, a user must use an HTTP proxy such as Privoxy so that traffic will be diverted through Tor rather than sent directly over the Internet. This is especially important because browsers will not automatically send DNS queries through a SOCKS proxy. However, pieces of software that plug into the browser, such as Flash, Java, and ActiveX Controls, do not necessarily use the browser's proxy for their network traffic. Thus, when any of these programs are downloaded and subsequently executed by the web browser, any Internet connections that the programs make will not go through Tor first. Instead, they will establish direct TCP connections, compromising the user's anonymity. This attack allows a website to identify its visitors but does not allow a third party to identify Tor users visiting a given website. These active content systems are well-known problems in anonymous web-browsing.

H. *Privacy Attacks*: If TOR used as directed by the Tor Project can provide an effective layer of privacy, however users are do not following the basic guidelines due to which information gets leaked from their browsers that could help intruder identify them. Most users do not utilise the Tor button and thus browser information in the form of user agent strings was leaked potentially providing geo-location and host PC configuration information to be identified by a determined attacker. In this way an attacker may potentially be able to tailor attacks to specific versions of browsers in use. Also using HTTP requests through search engines provides a hint to user location because sites like Google provide country specific searches. The Tor button and other such user agent switching plug-ins in certain web browsers will attempt to prevent the divulgence of geo-location information and OS details by "spoofing" details of defined user agent strings. Mitigation and countermeasures to geo-locating via user search terms would ideally be user education and also using only specific HTTPS search engines for this task.

I. *Circular attack*: Here the intruder makes use of network as a normal user and destines the packet in such a way it forms a circular circuit inside the network thus consuming the bandwidth to transfer same packet again & again. The onion router may never realize it is under attack since for every hop the node can know only its previous node address never realizing that the packet actually came from the destination to which the node forwarded it.

J. *Watermarking attacks*: This attack can be performed very effectively in TOR since TOR is a low latency network & a unique signal is embedded to identify the receiver to link with sender. The intruder partitions the flow duration into small intervals and adjusts inter packet delay between packets for manipulating the packet count within an interval. The population of packets within the intervals reveal the watermarking bit encoded in them.

IV. DEFENCE MECHANISM

To prevent & keep in check about the attacks following measures in network & client side TOR proxy:

A. *In Network*

Uptime verification: By sending periodic messages to the nodes we can keep track of the server availability & unavailability period, if the server has been unavailable for longer duration there is possibility of compromising activity & must be investigated.

Bandwidth scanning: The bandwidth od routers are measured & corrected in the directory server information. There are 2 types of scanning, namely:

- Centralised
- De-centralised

a) Centralised: In this method group of nodes are formed based on their characteristics, then local bandwidth(L) of a node is recorded & divided by its actual capacity(C) obtaining the aggregate value of stream passing via the node, S=L/C.

b) De-centralised: Here node based scanning is done, simply nodes monitor each other & detect collusion.

B. *In Client software*

- TOR Options: There are keys provided for customizing TOR settings one such key is TORButtons, following are the different types of TORButtons
- TORPlugins: Disable this option.
- Dynamic content Isolation: This setting option disables javascripts, doesn't allow meta-refresh tags & blocks pop ups.
- Hooking javascripts: This option masks the time zone user agent & operating system. Since time zone varies from place to place the intruder can gain the knowledge of user location by knowing the time data was sent.
- No automatic TOR updates: The TOR updates are never made from SSL websites, thus if a user receive any updates it might be a malicious program or addition of malicious code to existing software, and also disable suggestions on Google.

Apart from it one can block history reads & writes, don't save sessions & history, manage cookies & clear every time you use the internet, it's always good to mask user agent. Use Firefox web browser extensions which let you customize browser & its component with XML & javascripting.

V. CONCLUSION

As time evolves TOR evolves, so does the attack mechanism, so it is always safe to enhance security measures in the TOR & find the solution for the existing flaws inside system. Before an attack can occur one needs to understand possibilities of attacks & take counter measures to prevent because "prevention is always better than cure".

ACKNOWLEDGMENT

The author would like to thank her guide Mrs. Sreedevi .N, H.O.D of I.S.E, all the faculty members of computer science department & Principal of M.V.J.C.E, Bangalore, India for their constant support & guidance.

REFERENCES

[1] TheOnionRouter/TorFAQ.November2006.from: http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ
[2] Blackhat,(2007)"*Securing the Tor Network*" from: https://www.blackhat.com/presentations/bh-usa-07/Perry/Whitepaper/bh-usa-07-perry-WP.pdf
[3] Cassandra Security, (2009) "*Onion Routing and Darknets*" from: http://cassandrasecurity.com/index.php?s=onion+routing
[4] Torproject.org(2010) from: http://gitweb.torproject.org
[5] Vulnerabilities of TOR from: http://freehaven..net/~arma/slides-25c3.pdf
[6] watermarking TOR from: http://dj.eas.aus.edu/snac/document