IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.6, Issue.10, Oct. 2018 E-ISSN: 2347-2693

Traffic Analysis Attacks Over Networks of Anonymous Communication

Parul[1*] , Narinder Sharma[2]

[1,2]Dept. of Computer Science, N.C. College of Engineering, Israna, Panipat, Haryana, India

Corresponding Author: paruldudeja93@gmail.com

Available online at: www.ijcseonline.org

Accepted: 18/Oct/2018, Published: 31/Oct/2018

Abstract— The Infrastructure of Onion routing has major feature which is its flexibility and that is common to all kind of distributed systems. The Onion routing approach offers anonymous data transfer scheme worldwide. The traffic analysis method can be used to break anonymity of the anonymous network, for example TORs (The Onion Routing). Traffic confirmation attacks in low latency networks, mixing networks and in other similar networks are active fields of research. The main idea behind this research is traffic confirmation and analysis of attacks in anonymous communication. Traffic confirmation attacks are used in this research to make successful analysis of traffic of communicating parties over anonymous communication on Internet. It is described in detail that the nature of dropping the packets of Tor Protocol (the onion router) can put anonymity in danger and can harm it. In this paper advantage has been taken of forward compatibility feature by TOR to perform a new drop mark attack and also explained about different traffic confirmation attacks.

Keywords— TOR, Services, Security, Privacy, Attacks, Anonymity, Traffic Analysis

I. INTRODUCTION

Security and privacy have always been the most concerned issues and topics for the communication on the Internet from the beginning. Most of the Internet users want anonymous communication so that their private information can remain secured. As the volume of data which includes confidential information is increasing day by day data safety has become the major concern of users. Anonymity is in demand for various reasons and there are many techniques, tools and protocols available to fulfill the requirement. The Onion routing approach is one of the best techniques that are used worldwide for anonymous communication by various anonymous clients over the network. Messages in Onion routing are encrypted into several layers in form of wrapped, secured data packets which later get unwrapped at each layer of traversing over the network in anonymous data transfers.
These packets are transmitted through different onion routers (relays) in the network which is combination of packet hops. These routers are responsible for carrying and forwarding of packets over the network. Onion routing protocol is a part of TOR organization which is a kind of distributed low-latency anonymous communication network that was developed by combined research projects of the Naval Research Laboratory and the Free Haven Project. It is currently the largest anonymity network in existence with about 7000 relays nodes around the world to hide user's location, identity and usage from anyone conducting network surveillance or traffic analysis [1].

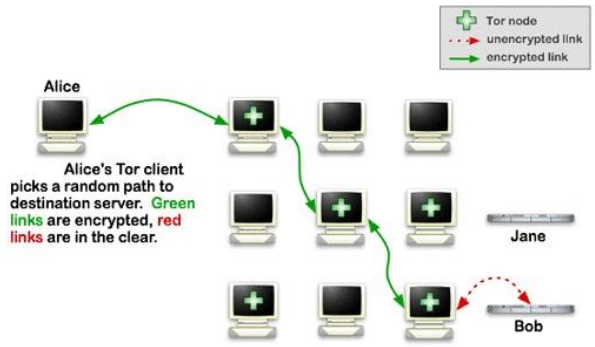

Figure 1: Explaining How Tor Works in anonymous communication.

It is very difficult to trace Internet activities of the user by using Tor services as the data packets are wrapped in several encrypted layers which cannot be decrypted by Tor easily. These activities include visiting any number of websites; posting or uploading data online in form of images, videos, text, files etc; instant messages and online ways of communication and many more. Proposed idea of Tor usage is to protect the privacy of its users and provide liberty and capability to have private communication structure in the network of users. Tor allows its users to communicate securely without being observed. The main idea behind this research is focused on Traffic Confirmation and Analysis of Attacks in anonymous communication.

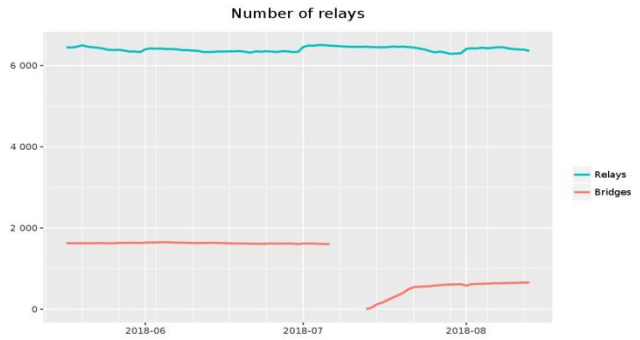

Figure 2: Graph statistics for number of Relays and Bridges in Tor Project of June-August 2018.

Traffic confirmation is a process of capturing and then investigating communication to reduce information in any kind of network (communication network) which can be executed even when the communication is securely transmitted in form of encryption. A traffic confirmation attack can happen only when the attacker controls the relays on both ends of a Tor circuit and then observe different parameters like traffic timing, volume etc. to conclude that the two relays are on the same circuit. If the IP address of the user is known to first relay in the circuit and destination is known to last relay she is accessing then together they can deanonymize her. As the volume of messages increases more number of messages is traded and stored which becomes fairly easy to guess the traffic.

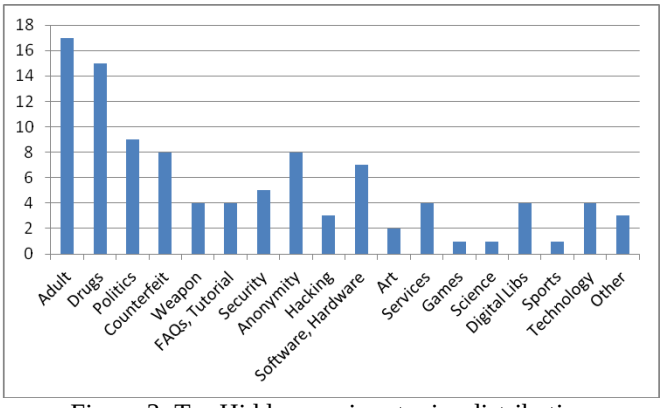

Figure 3: Tor Hidden services topics distribution

The application of traffic analysis can be done in the fields of military intelligence, counter intelligence, life analysis and is a matter of concern in computer security. Functions of Traffic analysis can be supported by devoted computer software programs. Figure 3 describes various Tor services over anonymous communication. Advanced traffic analysis techniques can include different forms of social network analysis [1]. These attacks are further categorized into two types: Passive, to detect stepping stones where the firewall only looks at the streams and Active, where the stream of data is modulated (often called watermark). Since an adversary is controlling the contents of the traffic and is probably encrypting it both types of identity rely on traffic data, usually to match the incoming and outgoing streams, the relationship between the packet intermediation times. Families of traffic analysis techniques used to attack are anonymous channels. Traffic confirmation attacks are used in this research to make it successful analysis of traffic.

A series of traffic analysis attacks has been used to reduce the security of the anonymous communication network. The long-term crossroads attacks (also known as disclosure attacks) answer the long term observation of input and output messages to detect communication divisions. Stream Traffic Analysis has been used to detect web requests and to provide answers through low-latency networks. In the end the attacker can interference the network or attempt to impact the way in which the honest nodes have chosen the path to anonymize their traffic of communication. Recently the attacks have focused on weak adversaries and that is easy to get the access of their activities and it has been shown that some forms of traffic analysis can be performed without any penetrate or access to the original data streams [2].

II. RELATED WORK

Congestion attacks are a kind of attack that takes advantage of the weakness in Tor's circuit construction protocol and analyze the traffic. To execute a better version of this attack the idea was introduced by Murdoch and Dennis's [3] in their Congestion attack. These attacks have a well-known technique to achieve guard relay of Tor users and onion services. Basically this attack consist of two steps: first step include applying clogging attack variant in which sending lots of data to flood a relay is performed and in second step of this attack resides in observation whether the effects of attacks at first step is visible on the channel between the adversary and the target one i.e., target relay or not. Attack for traffic confirmation in this paper is variant version of the attack introduced by Murdoch and Dennis's Congestion attack. Description of path selection in order to select guard relay in anonymous communication during the process of Traffic analysis is descried later in this paper.

III. BACKGROUND OF TRAFFIC ANALYSIS ATTACKS

Traffic Analysis is a bit more complicated. It is very subtle and difficult to figure out that if this happens then we had a way to hide information on a message and the hacker has still seen the information that it would be a traffic analysis attack. It is an attack used to find out communication patterns involved in the process related to it. These attacks can be applied on messages even when messages are encrypted. Here we have an example now assume Alice sends messages

to Bob but an attacker in the middle are able to detect this communication pattern involved in it. Now attacker is able to conclude that Alice and Bob know each other in the communication. Thus Traffic analysis is a form of Social Engineering where number of messages, their pattern, precise timing but also their absence can be important information for attackers. For example the timings gap between the messages can be used in SSH timing Attacks and absence of messages can be used burglars to determine that you are on holidays and break into your house.

As we already described, Traffic analysis attacks are further categorized into two types Passive and Active attacks. The big difference between active and passive attacks is that in the active attacks the attacker interrupts the connection and modifies the information whereas in a Passive attack the attacker stops transit information with intent to read and analyze information, not to change it. Some kind of attacks and techniques in traffic analysis and confirmation, are described as below:

i. Path Selection Attacks

To retain and security of the anonymity in Tor users, this kind of selection for path is very crucial in anonymity systems. Initiator of path should select first and last relay in communication in such that way no nodes in path should collude. When we are about to select random nodes for circuit creation, chances of selection of non adversary nodes will increases with selection of long path in this way. However, this argument ignores those possibilities that malicious Tor router can choose to facilitate connections with other adverse controlled relays and discard all other connections. Thus, the initiator node either creates a completely malicious circuit by selecting malicious node, or fails in that circuit and then tries. Basically this kind of attack shows that long circuits do not guarantee strong anonymity [4].

A version of this attack called "packet spinning", [5] attempts to force users to practice genuine router to choose malicious router from time to time. Here the attacker makes circular paths in the Tor network and keeps busy in transmitting large amounts of data through those paths to keep it valid.

ii. Wei Dai's attack on Traffic Shaping[6]

In [7], it is describes as a general attack against the system that allocates bandwidth to users because the connections and the traffic shaping are established between the nodes. Here, the attacker makes himself an anonymous route through a pair of nodes which he suspects is related to Alice's route. The attacker then increases the traffic through this route as long as the total traffic between the pair of nodes reaches the fixed bandwidth of the traffic shaping. At these point nodes no longer send padding packets to each other and real traffic throughput can be reduced by reducing the traffic sent by the attacker to the bandwidth limit.

iii. Latency attack

The possibility of using latency data in traffic analysis has been mentioned several times in previous works apparently originating in an article by Back et al. In this paper Latency attack is defined as it is perhaps the most difficult to protect against. It is completely based on the assumption that the latency on different routers will differ and these latencies can be computed easily by attackers that resides on those routers.

iv. Passive attacks

Observing user traffic patterns: Viewing a user's connection will not disclose its destination or data, but it will disclose traffic patterns (both sent and received). Further processing is required for profile through user connection patterns because multiple application streams are simultaneously running together on a circuit or in series in a single circuit.

Observing user content: While the user is finally encrypting the content at the user end, the connection cannot be made to the respondents (in fact, the feedback website itself may be hostile). While filtering content is not the primary goal of onion routing, Tor can use Privoxy and related filtering services to anonymize privacy data streams directly. (Privoxy is a filtering proxy for the HTTP protocol, which is often used in combination with Tor. Privoxy is a web proxy with advanced filtering capabilities for privacy, filtering web page content, managing cookies, controlling access, and deleting ads, banners, pop-ups, etc. It supports both stand-alone systems and multi-user networks.)

Options distinguish ability: This will allow customers to choose the configuration options. For example, customers concerned about request linkability should often walk around the circuit compared to people concerned about traceability. The option of choice can attract users with different needs but the customers living in the minority can be more anonymous by showing them differently by optimizing their behavior [8].

End-to-end timing correlation: Tor only hides at least this kind of correlation. The attacker, who saw patterns of traffic on the beginner and the respondent, will be able to confirm the high probability correspondence. Against this type of confirmation the biggest security available at present is to hide the connections between the onion proxy and the first tor node by running OP on the tor node or firewall. For this approach, an observer needs to isolate traffic from the onion router with traffic passing through it: A global observer can do this but it can be beyond the capabilities of the limited observer.

End-to-end size correlation: Simple packet counting will also be effective in verifying the end points of the stream. However, even without padding, we may have some limited protection: Leaking pipe topology means that different

numbers of packets can enter one end of the circuit instead of exiting the other.

Website fingerprinting: The above are all effective passive attack traffic confirmation attacks. There is also a passive traffic analysis attack that is potentially effective. Instead of searching for exhaust connections for time and volume correlations, the opponent can create a database of "fingerprint" which includes file sizes and access patterns for targeted websites. He can later confirm the connection of the user of a given site by consulting the database. This attack has proved effective against SafeWeb [9]. This can be less effective against tor, because the streams are multiplexed in the same circuit, and fingerprinting will be limited to granularity of cells (currently 512 bytes). Additional defenses could include larger cell sizes, large set to set up group can include padding plans and link padding or long range dummies.

v. Active attacks

Compromise key: An attacker who learns TLS session key can see control cells and encrypted relay cells on each circuit on that connection; After learning a circuit session key that allow it to open a layer of encryption. An attacker who learns TLS private key of OR can impersonate TLS key for lifetime but he should also learn the onion key to decrypt make cells (And because of complete further privacy, he cannot hijack the established circuit, without having his session key agreement). Periodic key rotation restricts the window of opportunity of these attacks. On the other hand, an attacker who learns the identity of the node can change the node indefinitely which can change indefinitely by sending new forged descriptors to the directory servers.

Iterated compromise: A wandering adversary who can compromise the ORs can compromise the circuit until the circuit reaches the end. Unless the adversary can meet this attack during the lifetime of this circuit, however, the ORs will abandon the necessary information before the attack is complete. (Thanks to the right forward privacy of the session key, the attacker cannot force the nodes to decrypt the recorded traffic after the attacker is closed.) In addition, construction of circuits crossing the authority can make legal robustness difficult - this event is usually called Jurisprudence Mediation.

Run a recipient. An adversary running a web server who learns the time patterns of users connecting it and can present an arbitrary pattern in its replies. End-to-end attacks are easy if the adversary can inspire users to connect to their web server (perhaps by those advertising content targeted to those users) now he keeps one end of his connection. There is also a danger that application protocols and related programs can be motivated to reveal information about the initiator. Tor depends on Privoxy and similar protocol cleaners to solve this later problem.

Run an onion proxy: It is expected that the end users will almost always run their local onion proxies. However, in some settings proxies may be needed to run remotely usually those institutions that want to monitor the activity of those who connect to the proxy. Compromising an onion proxy compromises all future connections through it.

DoS non-observed nodes: An observer, who can only see some Tor network can stop the value of this traffic to attack the non observed nodes it reduce their reliability or persuade users to stop them from being trusted. Here robustness is the best defense.

Run a hostile OR: In addition to being a local observer, a separated hostile node can create a circuit through itself or may change the traffic patterns to affect traffic on other nodes. Nevertheless, a hostile node should immediately be near both intervals to compromise the circuit's anonymity. If an adversary can run multiple ORs and the directory server can persuade that they are reliable and independent, sometimes some users choose one of those ORs to start and the second is in the form of the end of the circuit. If an adversary controls $m > 1$ of N nodes, he can correlate at most $([m/N])^2$ of the traffic-although an adversary could still attract a disproportionately large amount of traffic by running an OR with a permissive exit policy or by degrading the reliability of other routers.

Introduce timing into messages: It is a strong version of the passive timing attacks which is already discussed.

Tagging attacks: A hostile node could tag the cell by changing it. If the stream for instance, had an unencrypted request on a web site will confirm the bad content association coming out at the right time. However integrity check on cells prevents this attack.

Replace contents of unauthenticated protocols: When relaying an unauthorized protocol such as HTTP, a hostile exit node can pretend to be the target server. Preference should be given to the protocols which provide end -to-end validation by authentication to their users.

Replay attacks. Some anonymous protocols are vulnerable to replay attack. Tor is not; Replaying again on one side of the handshake will be a separate conversation session key, and therefore the remaining record sessions cannot be used.

Smear attacks: An attacker can use the Tor network for socially disallowed acts to defame the network and turn off its operators. Exit policies reduce the chances of abuse, but ultimately the network requires volunteers who can afford some political heat.

Distribute hostile code: An attacker could trick users into running the above Tor software which is in fact did not anonymize their connection or worse could run the OS in running weakened software which used to reduce users to a less anonymity. We address this problem (but do not completely solve it) by signing all the torrent releases with an official public key, and with an entry in the directory, which lists what versions are currently safe.

IV. GUARD NODE DISCOVERY OF ONION SERVICE

All the anonymity exploits introduced in this paper rely on the dropping behavior of the Tor routing protocol. The aim of this attack is to find the entry node used on any onion service by any onion service without any relay in the network. This method does not reveal the location of the hidden service directly, but gives the relay on which the hidden service is always connected which significantly reduces the work required in the direction of complete denomination.

The client runs a local software called onion proxy (OP) to anonymize the client data into Tor. In this we separate two types of clients: normal clients use the Tor core network and bridge clients use the bridge to access Tor core network. The Tor core network consists with various elements in it these are Onion Proxy, Onion Router, Directory Server, and Bridges which are responsible for their own functionalities.

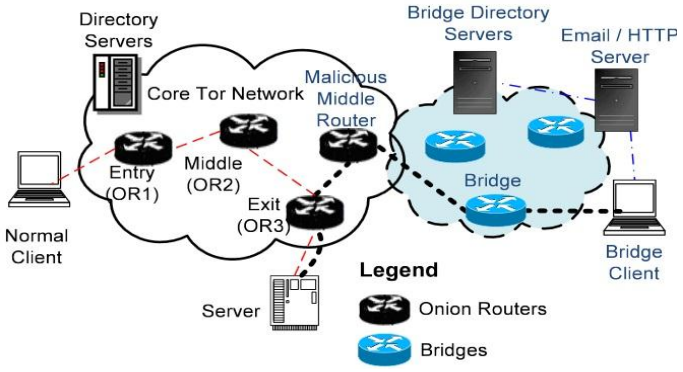

Figure 4: Core infrastructure for Bridges and Onion Routers in searching of Guard node.

Onion routers (or Tor routers) relay the application data between clients and servers. Directory server keeps information related to onion router such as IP addresses. Copy of onion router list locally is available to its entire user. This is the reason that it is easy to block Tor core network. Bridges are special Tor routers which publish their information on the bridge directory server. This server has information about all bridges. The client bridge can retrieve bridge information by accessing the https / email server or receiving it privately from the friend-to-friend network. The main purpose of Onion Proxy, Onion Router, Directory Server and Bridges are integrated into TOR Software Package. A user can configure the configuration file to configure for different combinations of those purposes.

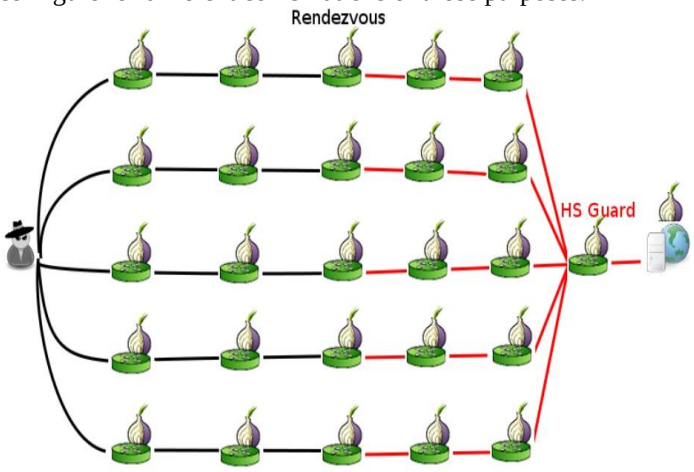

Figure 5: Guard discovery technique: relay drop attack against an onion service. Black links are built by the adversary. Red links are built by the onion service. [10]

Algorithm for Tor Path Selection Algorithm at Client : In this, following steps are carried out:

1: Create a new circuit and initialize global circuit ID etc.
2: Add the circuit into the global circuit list.
3: Decide a suitable length (3 by default) for the circuit.
4: if Option 'ExitNodes' is defined in the configuration file, Then
5: Use one of 'ExitNodes' as the exit router
6: else
7: Exclude nodes by Node Exclusion for Selecting Exit Nodes
8: Select an exit node by Bandwidth Weighted Node Selection algorithm
9: end if
10: Exclude the chosen exit node, and select an entry node by the of Selection of an Entry/Middle Node
11: Exclude the chosen exit node and entry node, and select a middle node by the Selection of an Entry/Middle Node [11].

V. METHODOLOGY

This section discusses a methodology used for new traffic analysis attack named as drop mark attack. The steps and description of attack involved in this paper are as follows:

Using Forward Compatibility to generate drop mark Attack

We can also try to create end-to-end correlation attacks based on the flexibility of the Tor Protocol. In this paper we consider a model where the relays control by the adversary node and wish to efficiently mark traffic going through one of his guards is coming out on one of his exits. Few research

papers tried this attack, which results in tagging attack [12, 13]. This attack changes the data flow at the entry node in the direction of outbound to generate an integrity error and this is recognizable by the relay, during the decryption which is by the other compromised relay, but also induced the tear of the circuit on non-collude it exit. This section is close to focus on this paper to designed; Biryukov et al performed the circuit construction finger printing to deanonymize onion services. The fact which uses this attack focused that the Tor protocol is gentle with most unrecognized cells [14]. The base idea of our paper is similar to proposed by [14], known as drop mark attack for this paper. There are two types of cells in onion routing: Link-level cells and circuit cells [12]. Structure of Link-level cells and circuit cells are given in figure 6 below. Cell format in onion routing has different fields. The fields CircID and Cmd are not encrypted while the remaining bytes might be encrypted several times, depending on the number of relays in the circuit. All circuit-level cells have the same Cmd value (set to "Relay") but not the same Rel cmd value (the second Cmd field in Figure 6), defining the subtype of circuit-level cell. The default behavior emphasis on an architecture design promoting forward compatibility in TOR: every cell with a Cmd or an unrecognized Rel cmd is silently dropped. In this section, we use this dropping behavior (forward compatibility) to make traffic confirmation in Tor.

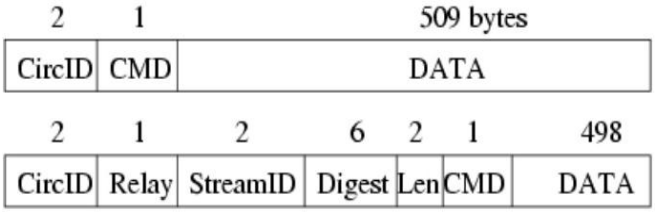

Figure 6: Link-level cells and circuit cells structure, image from [12].

i. Silently Dropping Packets (Forward Compatibility)

We took Tor protocol to elaborate the functionality of drop mark attack of this paper. We can send any number of cells from an edge node in the communicating network of Tor which would be silently dropped by the other edge. We target timing intervals where flows are used to be idle in order to avoid latency of our victim flow and easily extraction of the mark that we added drop mark in our case.

Now consider, what happens when a service is requested from the Tor and in order to connect with website a *begin cell* is sent to the circuit from the Tor client. This cell triggers a DNS lookup at the edge of the circuit. As soon as DNS resolver succeeds and the connection of desired service is set up, the edge relay in circuit sends a connected cell towards the client in inbound direction. The upcoming cells that would be responsible to the GET request that the client has issued.

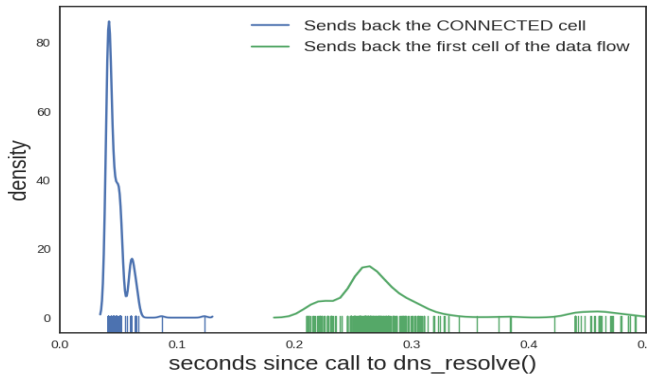

Figure 7: Density plot of the connected cell and the first data cell.

Figure 7 shows the density plot of the connected cell and the first data cell sent towards the client. This plot results from 24 hundreds web connection that an exit relay performed. These connections arrive at many different web servers locations simulated to inspire a similar response compared to web servers on the Internet.

The idle time when no cells go towards the client correspond to the Round Trip Time (RTT) of the both the Tor circuit and the edge connection. Before the client receives the connected cell this will be idle till the request sent optimistically and then idle time corresponds to RTT of the edge connection. This feature is also described in the documentations of Tor's code as a method to speed up the HTTP protocol. This entry point window is still large enough to encode the drop mark. Encoded drop mark with relay drop cells or any other which are silently drop by other edges in the circuit.

ii. About attack:

This paper represents attack, a method to carry along the circuit on bit of information without adding latency. This attack needs two characteristics within anonymizer of low latency, first the protocol will drop unrecognized package silently over the circuit and second the circuit will remain idle when no cover traffic (at some moment).

We can guess that this attack impacts on any low latency network which has above cited characteristics. This paper is basically focused on the effectiveness of drop mark attack on the Tor Network. The edge node of a circuit is responsible for the encoding part of this attack. This edge node could be an exit relay or an onion service or an intro point. For smooth functionality of this we considered only exit relay in this paper to elaborate our experiment.

As shown in figure 7, we took advantages of the idle entry point that exists on any Tor Circuit. Each time a relay *begin cell* is sent towards the *exit relay* that carries the desired domain name or IP address to connect to the network. The communication process is start as the reception of relay *begins cell* towards the client. As *dns_resolve()* function is

called, we log the IP address and can easily send 3 relays drop cells towards the client. Figure 7 gives us ability to understand the decoding function that would further decide a drop mark is embedded in the flow or not. The decoding function on the side of guard relay in circuit considers the first few cells of the circuit in inbound direction and completely responsible to verify this observation. Even we can say if three cells have the same arrival time (with the difference of more or less a few milliseconds) within the first four cells, we mark the flow as having a drop mark. This work basically based on forward compatibility feature of Tor.

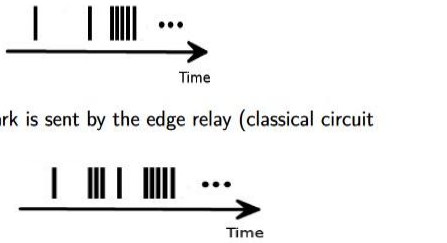

Figure 8: Simplified view of cell timings flowing towards the client from the perspective of a guard. [10]

iii. Implementation and Explanation:

The complete implementation of this attack is simulated in shadow simulator at Linux (ubuntu) platform. To simulate the behavior, web pages are downloaded from Alexa dataset [15]. Every time each web client should use a fresh circuit. By doing so, it will be responsible to capture the behavior of Tor browser. Tor browser uses a new circuit every time for each web address. During this complete process of simulation we imagine that traffic flowing of any type of application through Tor is correspond to this drop mark attack. Figure 7 shows the advantages taken by drop mark attack used in Transport layer (TCP). During simulation we considered two situations to test the functionality, first when no exit relays send the drop mark but all other relays in the circuit apart from it applies decoding function on guards and second all exit nodes send drop mark and guard relay apply the decoding function.

The first situation focuses on accessing the number of false positives that will raise by decoding function and the second one aims at evaluation of number of false negatives in the circuit after decoding function. These tests are performed under a network loaded similar to real Tor network which evaluates the impact of efficiency by congestion on the circuit in shadow.

In experiment of first situation light loaded network is simulated in which no drop marks are sent and approximate 30 thousands circuits have been tested among the guard. The fraction of HTTP transfer errors over successful transfer was near about or can say approximate 0.3%. In the experiment of second situation the overall congestion is increased by increasing number of web clients. In this situation light loaded network where every exit relay sends drop mark for all connection and no false negatives were detected. By increasing the size of network by adding some web clients we obtained some false negatives result over 30 thousand tested circuits (approx) with a fraction of HTTP transfer error over successful transfer which is approximate 4.6%. Most of those false negatives were due to failure in the circuit. In final observation this method shows good results (as we were expecting during experimentation). It loaded network with 99.86% true positive rate and 0.03% false negative rate. Moreover this method is not based on the top layer of network and does not perturb it. This leads to successful correlation even if a few bytes are exchanged between the source and the destination.

VI. CONCLUSION

In the Tor protocol, forward compatibility refers to dropping of unrecognized packets by cell structure of the circuit. This paper explained how dropping of unrecognized packets in network can exploit anonymity of anonymous communication. In network, protocols forward compatibility is one of the desirable features. Even more when the network is scattered since many different versions can compose the overall network. If anonymity can be broken in some situations, paper does not claim the design of Tor is broken or it has any kind of loop hole. However it sheds light over the fact that ensuring anonymity at server side is complex. This research also explained about various attacks of traffic confirmation or possible attack methods for analyzing the traffic in any anonymous communication network.

ACKNOWLEDGMENT

We take this opportunity to thank Er Jagtar Singh, Head of Department of Computer Engineering for making essential facilities available for us.

REFERENCES

[1] "Traffic Analysis." Wikipedia, Wikimedia Foundation, Aug. 2018, en.wikipedia.org/wiki/ Traffic_analysis.

[2] Leuven, K.U. Introducing Traffic Analysis Attacks, Defences and Public Policy Issues. Dec. 2005, www0.cs.ucl.ac.uk/staff/G.Danezis/talks/TAIntro-prez.pdf.

[3] Murdoch, Steven J., and George Danezis. "Low-cost traffic analysis of Tor." Security and Privacy, 2005 IEEE Symposium on. IEEE, 2005.

[4] Borisov, N., D Anezis, G., Mittal, P.,And Tabriz, P." Denial of service or denial of security? How attacks on reliability can compromise anonymity. In CCS '07: Proceedings of the 14th ACM onference on Computer and communications security (New York, NY, USA, October 2007), ACM, pp. 92–102

[5] Pappas, V., A Thanasopoulos , E., Ioannidis, S.,And Markatos, E. P. Compromising anonymity sing packet spinning. In Proceedings of the 11th Information Security Conference (ISC 2008) (2008), T.-C. Wu, C.-L. Lei, V. Rijmen, and D.-T. Lee, Eds., vol. 5222 of Lecture Notes in computer Science, Springer,pp. 161–174.
[6] A. Back, U. Moller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," Lecture Notes in Computer Science, vol. 2137, no. 245-257, p. 76, 2001.
[7] Evans, Nathan S., Roger Dingledine, and Christian Grothoff. "A Practical Congestion Attack on Tor Using Long Paths." USENIX Security Symposium. 2009.
[8] A. Acquisti, R. Dingledine, and P. Syverson. On the economics of anonymity. In R. N. Wright, editor, *Financial Cryptography*. Springer-Verlag, LNCS 2742, 2003.
[9] A. Hintz. Fingerprinting websites using traffic analysis. In R. Dingledine and P. Syverson, editors, *Privacy Enhancing Technologies (PET 2002)*, pages 171-178. Springer-Verlag, LNCS 2482, 2002.
[10] Rochet, Florentin, and Olivier Pereira. "Dropping on the Edge: Flexibility and Traffic Confirmation in Onion Routing Protocols." Proceedings on Privacy Enhancing Technologies 2018.2 (2018): 27-46.
[11] Ling, Zhen, et al. "Extensive analysis and large-scale empirical evaluation of Tor bridge discovery." INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.
[12] Syverson, Paul, R. Dingledine, and N. Mathewson. "Tor: The second generation onion router." Usenix Security. 2004.
[13] X. Fu, Z. Ling, J. Luo, W. Yu, W. Jia, and W. Zhao. "One cell is enough to break tor's anonymity", In Proceedings of Black Hat Technical Security Conference, pages 578–589.Citeseer, 2009.
[14] A. Biryukov, I. Pustogarov, and R.-P. Weinmann. Trawling for Tor Hidden services: Detection, measurement, deanonymization In Proceedings of the 2013 IEEE Symposium on Security and Privacy, May 2013.
[15] Top-1000 alexa data set. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip, 2017. Accessed: 2017-05-20.
[16] Ling, Zhen, et al. "Extensive analysis and large-scale empirical evaluation of Tor bridge discovery." INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.
[17] Diaz, C., Preneel, B.: Taxonomy of mixes and dummy traffic. In: Deswarte, Y., Cuppens, F., Jajodia, S., Wang, L. (eds.) SEC 2004. IIFIP, vol. 148, pp. 217–232. Springer, Boston, MA (2004).
[18] Kohls, Katharina Siobhan, and Christina Pöpper. "POSTER: Traffic Analysis Attacks in Anonymity Networks." Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017.

Authors Profile

Narinder Sharma pursed Bachelor and Master of Technology in Computer Science & Engineering from N. C. College of Engineering affiliated to Kurukshetra University, Kurukshetra, India in 2005 and 2010 respectively. He is currently working as Assistant Professor in Department of Computer Science & Engineering in N. C. College of Engineering affiliated to Kurukshetra University, Kurukshetra, India since 2007. He has published more than 10 research papers in reputed international journals and conferences. His main research work focuses on Computer Networks, LINUX in Technical Applications, Computer Hardware, Network Security, Open Source Technology, Design Of Enterprise Networks, Installation And Configuration Of Servers, Client Server Architecture. He has 12 years of teaching experience and 3 years of Research Experience.

Parul, is pursuing her Master of Technology in Computer Science & Engineering from N. C. College of Engineering affiliated to Kurukshetra University, Kurukshetra, India. She has completed her Bachelor of Technology in Computer Science & Engineering from Geeta Engineering College affiliated to Kurukshetra University, Kurukshetra, India in 2015. Her Area of interest for research work focuses on Cryptography Algorithms, Network Security and Privacy.

