ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

AUGMENTING THE SECURITY IN TOR

S.Kokilavani[1], G.Priyadharshini[2]

HOD /ECE, Sri Ranganathar Institute of Polytechnic College, Athipalayam, Coimbatore-641110

Lecturer/CSE, Sri Ranganathar Institute of Polytechnic College, Athipalayam, Coimbatore-641110

ABSTRACT: Tor is an anonymous communication network. If more users are becoming interested in their privacy, the need for such anonymous services might increase. The second generation Onion Router designs Tor and its previous designs seem to have been under research and there have been rather recent papers on Tor's vulnerabilities. Various low-latency anonymous communication systems such as Tor and Anonymizer have been designed to provide anonymity service for users. In order to hide the communication of users, most of the anonymity systems pack the application data into equal-sized cells (e.g., 512 B for Tor, a known real-world, circuit- based, low-latency anonymous communication network).This project defends cell-counting-based attack against Tor, which allows the attacker to confirm anonymous communication relationship among users very quickly by adopting reputation based routing allocation among the onion routers. Cell counting attacks by varying the number of cells in the target traffic at the malicious exit onion router, the attacker can embed a secret signal into the variation of cell counter of the target traffic. The embedded signal will be carried along with the target traffic and arrive at the malicious entry onion router. Then, an accomplice of the attacker at the malicious entry onion router will detect the embedded signal based on the received cells and confirm the communication relationship among users. Reputation for each Tor onion router are assigned and measured on real-time. One TOR router is competing for the available routing resources, the routing is allocated when the router completely satisfy the router's demands, unless router's reputation is below a certain threshold, set by the system to mark misbehaviour. In this case, competing router does not receive any routing resource, as a punishment for his zero contributions or for accessing the embedded signal through cell counting attack.

KEYWORDS: Tor, Cell counting, Reputation Algorithm

I. INTRODUCTION

Tor (originally short for The Onion Router) is a system intended to enable online anonymity. Tor client software directs internet traffic through a worldwide volunteer network of servers to conceal a user's location or usage from anyone conducting network surveillance or traffic analysis. Using Tor makes it more difficult to trace Internet activity, including "visits to Web sites, online posts, instant messages and other communication forms", back to the user and is intended to protect users' personal privacy, freedom, and ability to conduct confidential business by keeping their internet activities from being monitored.

"Onion Routing" refers to the layered nature of the encryption service: The original data are encrypted and re-encrypted
multiple times, then sent through successive Tor relays, each one of which decrypts a "layer" of encryption before passing the data on to the next relay and, ultimately, its destination. This reduces the possibility of the original data being unscrambled or understood in
transit.

The latest Onion Routing system is freely available and runs on most common operating systems. There is a Tor network of several hundred nodes, processing traffic from hundreds of thousands of unknown users. (The protection afforded by the system makes it difficult to determine the number of users or application connections.) Exact current and historical number of Tor nodes and global traffic volume processed are graphically depicted here. The code and documentation is available under a free license. Check out the Tor site for more details and instructions for running Tor.

The protection of Onion Routing is independent of whether the identity of the initiator of a connection (the sender) is hidden from the responder of the connection, or vice versa. The sender and receiver may wish to identify and even authenticate to each other, but do not wish others to know that they are communicating. The sender may wish to be hidden from the responder. There are many ways that a web server can deduce the identity of a client who visits it; severaltest sites can be used to demonstrate this. A filtering proxy can be used to reduce the threat of identifying information from a client reaching a server.

Onion Routing currently makes use of the Privoxy filter for this purpose. Concerns about privacy and security havereceived greater attention with the rapid growth and public acceptance of the Internet, which has been used to create our global E-economy. Anonymity has become a necessary and legitimate aim in many applications, including anonymous Web browsing, location-based services (LBSs), and E-voting. In these applications, encryption alone cannot maintain the anonymity required by participants [1]–[3]. In the past, researchers have developed numerous anonymous communication systems. Generally speaking, mix techniques can be used for either message-based (high-latency) or flow-based (low-latency) anonymity applications. E-mail is a typical message-based anonymity application, which has been thoroughly investigated [4]. Research on flow-based anonymity applications has recently received great attention in order to preserve anonymity in low-latency applications, including Web browsing and peer-to-peer file sharing.

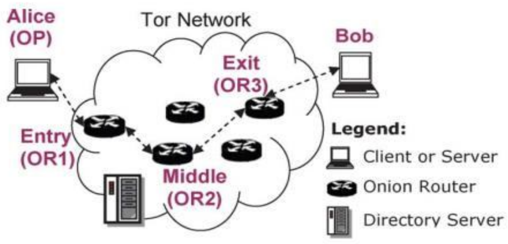
To degrade the anonymity service provided by anonymous communication systems, traffic analysis attacks have been studied. Existing traffic analysis attacks can be categorized into two groups:

passive traffic analysis and active watermarking techniques. Passivetraffic analysis technique will record the traffic passively and identify the similarity between the sender's outbound traffic and the receiver's inbound traffic based on statistical measures.

Because this type of attack relies on correlating the timings of messages moving through the anonymous system and does not change the traffic characteristics, it is also a passive timing attack.

For example, Serjantov et al. [7] proposed a passive packet-counting scheme to observe the number of packets of a connection that arrives at a mix node and leaves a node. However, they did not elaborate how packet counting could be done. To improve the accuracy ofattacks, the active watermarking technique has recently received much attention. The idea of this technique is to actively introduce special signals (or marks) into the sender's outbound traffic with the intention of recognizing the embedded signal at the receiver's inboundtraffic.

Tor is a popular overlay network for providing anonymous communication over the Internet. It is an open-source project and provides anonymity service for TCP applications. As shown in Fig. 1, there are four basic components in Tor.



Tor network.

1) Alice (i.e., Client): The client runs a local software called onion proxy (OP) to anonymize the client data into Tor.

2) Bob (i.e., Server): It runs TCP applications such as a Web service.

3) Onion routers (ORs): Onion routers are special proxies that relay the application data between Alice and Bob. In Tor, transport-layer security (TLS) connections are used for the overlay link encryption between two onion routers. The application data is packed into equal-sized cells (512 B as shown in Fig. 2) carried through TLS connections.

4) Directory servers: They hold onion router information such as public keys for onion routers. Directory authorities hold authoritative information on onion routers, and directory caches download directory information of onion.

II. RELATED WORKS

Perfect forward secrecy: In the original Onion Routing design, a single hostile node could record traffic and Later compromise successive nodes in the circuit and force them to decrypt it. Rather than using a single multiply encrypted data structure (an onion) to lay each circuit, Tor now uses an incremental or telescoping path-building design, where the initiator negotiates session keys with each successive hop in the circuit. Once these keys are deleted, subsequently compromised nodes cannot decrypt old traffic. As a side benefit, onion replay detection is no longer necessary, and the process of building circuits is more reliable, since the initiator knows when a hop fails and can then try extending to a new node. Separation of–protocol cleaning□ from anonymity: Onion Routing originally required a separate –application proxy□ for each supported application protocol—most of which were never written, so many applications were never supported. Tor uses the standard and near-biquitous SOCKS [32] proxy interface, allowing us to support most TCP-based programs without modification. Tor now relies on the filtering features of privacy-enhancing application-level proxies such as Privoxy [39], without trying to duplicate those features itself.

Onion Routing originally called for batching and reordering cells as they arrived, assumed padding between ORs, and in later designs added padding between onion proxies (users) and ORs. Tradeoffs between padding protection and cost were discussed, and traffic shaping algorithms were theorized to provide good security without expensive padding, but no concrete padding scheme was suggested. Recent research [1] and deployment experience [4] suggest that this level of resource use is not practical or economical; and even full link padding is still vulnerable [13]. Thus, until we have a proven and convenient design for traffic shaping or low-latency mixing that improves anonymity against a realistic adversary, we leave these strategies out. Many TCP streams can share one circuit: Onion Routing originally built a separate circuit for each application level request, but this required multiple public key operations for every request.

III. PROPOSED METHODOLOGY

Tor is a communication network that provides Anonymity service for users. Tor packs the application data into equal-sized cells (e.g., 512 B for Tor). Via extensive experiments on Tor, It is found that the size of IP packets in the Tor network can be very dynamic because a cell is an application concept and the IP layer may repack cells. Based on this information a new attack has been created by inducing DSSS signals in to the tor network. The embedded signal will be carried along with the target traffic and arrive at the malicious entry onion router. Then, an accomplice of the attacker at the malicious entry onion router will detect the embedded signal based on the received cells and confirm the communication relationship among users. To counter this attack we propose a new rank based algorithm for each newly added onion router in the network. Each newly added nodes (Routers) are monitored for a specific period of time. If the Particular onion router at question Exhibits any suspicious behaviour, then the router is systematically removed from service. This essentially protects the anonymity of the Tor.

IV. EXPERIMENTAL STUDY

We have implemented the cell-counting-based attack presented in against onion router. we use real-world experiments to demonstrate the feasibility and effectiveness of this attack. All the experiments were conducted in a controlled manner, and we experimented on TCP flows generated by ourselves in order to avoid legal issues.

A. Experiment Setup

In our experiment setting, we deployed two malicious onion routers as the Tor entry onion router and exit onion router. The entry onion router and client (Alice) are deployed. The server (Bob) is located in LAB, and the exit onion router is at an off-campus location in SRIPC LAB as well. All computers are on different IP address segments and connected to different Wireless Routers.

B. Experimental Results

To obtain the empirical property of IP packet size for the traffic within the Tor network, we downloaded a file with the size of 20 M using the Tor network. Fig. 15 shows the empirical cumulative probability function (CDF) of the IP packet size in the traffic. As shown in Fig. 5, we know that the packets with non-MTU size are around 50%. This validates that the size of packets transmitted over the Tor is dynamic. Consequently, it also indicates that our embedded signal will be hidden in the normal traffic and hard to be detected by victims.

To validate the accuracy of the cell-counting-based attack, we let the client download 30 files in our experiments. The size of each file is around 10 MB. At the exit onion router, we generate a random signal with 100 b. When the target traffic from server Bob arrives at the exit onion router, we vary the number of cells in the circuit and embed the signal into the variation of the cell count during a short period in the target traffic. At the entry onion router, the cells in the circuit queue are recorded in the log, and the recovery mechanisms will be applied to recognize the embedded signal. When we evaluate the reputation rate, the client downloads 30 files via Tor again. However, no signal is embedded into the traffic at the exit onion router. Denote the traffic with no signal as clean traffic. We generate a 100-b random signal. By checking how many bits of this signal show up in the clean traffic, we can calculate the reputation rate. Based on the reputation rate the exit and entry path for the onion routers are selected for the users. If cell counting attack found the reputation rate goes to negative for the onion router placed at position X. The node is avoided at future network transmission.

To validate the accuracy of the cell-counting-based attack, we let the client download 30 files in our experiments. The size of each file is around 10 MB. At the exit onion router, we generate a random signal with 100 b. When the target traffic from server Bob arrives at the exit onion router, we vary the number of cells in the circuit and embed the signal into the variation of the cell count during a short period in the target traffic. At the entry onion router, the cells in the circuit queue are recorded in the log, and the recovery mechanisms will be applied to recognize the embedded signal. When we evaluate the reputation rate, the client downloads 30 files via Tor again. However, no signal is embedded into the traffic at the exit onion router. Denote the traffic with no signal as clean traffic. We generate a 100-b random signal. By checking how many bits of this signal show up in the clean traffic, we can calculate the reputation rate. Based on the reputation rate the exit and entry path for the onion routers are selected for the users. If cell counting attack found the reputation rate goes to negative for the onion router placed at position X. The node is avoided at future network transmission.

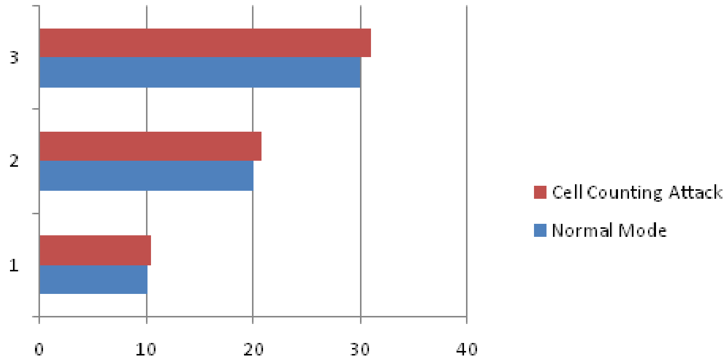


Fig 2. Shows the cell counting attack packet size variations

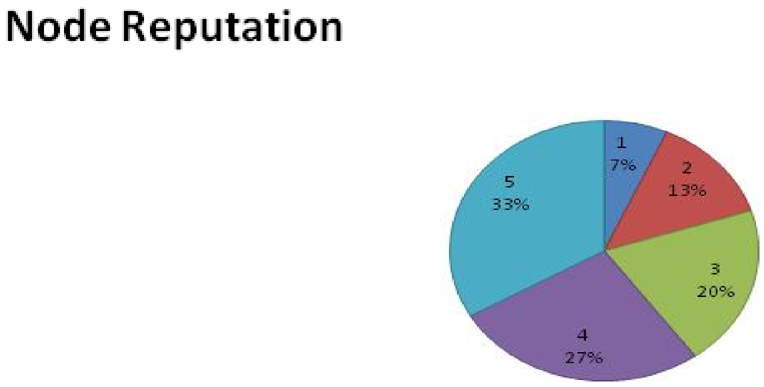


Fig. 3 shows the reputation percentage of the onion routers available during experiment.

V. CONCLUSION & FUTURE WORK

This paper presented a complete methodology for identifying the cell counting attack and allows the onion routers to contribute their resources in a loaded network where the capacity- limited access link of each onion router is shared among his download and upload streams. Our proposed allocation scheme is implemented in a distributed manner at each onion router, without the necessity of any global information. This System Considered attacked peers who seek to maximize their utility with the least possible contributions and showed that under the presence of our proposed reputation system they are inclined to cooperation. In the homogeneous scenario where all onion routers have the same capacity and request generation rate, rational routers fast reach a cooperative operating point under which they offer half of their capacity for their download and half for their upload streams. In heterogeneous systems, utilization of all resources is not always possible since some of the higher capacity (powerful) peers' resources may not be exploited because of the inability of other routers to offer and receive Resources at the rate that powerful peers can accept and provide, respectively.

System has much more scope in the future and reputation-based allocation protocol significantly improves the performance of onion routers under the related work in over heterogeneous systems and even in more general systems where parallel downloads and uploads take place and peers with different request generation profiles exist.

It further system significant advantages over Bit Torrent protocol which lacks of a dynamic capacity allocation algorithm for single capacity-limited link peers to help them improve their performance, and fails to deter misbehaviour and motivate seeds to remain and contribute in the system.

VI. ACKNOWLEDGEMENT

The authors would like to thank Director cum secretary, Correspondent and Principal of Sri Ranganathar Institute of Polytechnic College, Coimbatore for their motivation and constant encouragement. The authors would like to thank the Dean, HOD, faculty Members of Department of Computer and Engineering, Department of Electronics and Communication Engineering for the critical review of the manuscript and for their valuable input and fruitful discussions. Also we take privilege in extending gratitude to their family members and friends who rendered their support throughout this research work.

REFERENCES

[1] Q. X. Sun, D. R. Simon, Y. Wang, W. Russell, V. N. Padmanabhan, and L. L. Qiu, –Statistical identification of encrypted Web browsing traffic,□ in Proc. IEEE S&P, May 2002, pp. 19–30.

[2] X. Fu, Y. Zhu, B. Graham, R. Bettati, and W. Zhao, –On flow marking attacks in wireless anonymous communication networks,□ in Proc. IEEE ICDCS, Apr. 2005, pp. 493–503.

[3] L. Øverlier and P. Syverson, –Locating hidden servers,□ in Proc. IEEE S&P, May 2006, pp. 100–114.

[4] G. Danezis, R. Dingledine, and N. Mathewson, –Mixminion: Design of a type III anonymous remailer protocol,□ in Proc. IEEE S&P, May 2003, pp. 2–15.

[5] R. Dingledine, N. Mathewson, and P. Syverson, –Tor: The secondgeneration onion router,□ in Proc. 13th USENIX Security Symp., Aug. 2004, p. 21.

[6] –Anonymizer, Inc.,□ 2009 [Online]. Available: http://www.anonymizer.com/

[7] A. Serjantov and P. Sewell, –Passive attack analysis for connectionbased anonymity systems,□ in Proc. ESORICS, Oct. 2003, pp. 116–131.

[8] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, –Timingattacks in low-latency MIX systems,□ in Proc. FC, Feb. 2004, pp.251–565.

[9] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, –On flowcorrelation attacks and countermeasures in Mix networks,□ in Proc.PET, May 2004, pp. 735–742.

[10] S. J. Murdoch and G. Danezis, –Low-cost traffic analysis ofTor,□ in Proc. IEEE S&P, May 2006, pp. 183–195.

[11] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker,–Lowresource routing attacks against anonymous systems,□ in Proc.ACM WPES, Oct. 2007, pp. 11–20.

[12] X. Wang, S. Chen, and S. Jajodia, –Network flow watermarkingattack on low-latency anonymous communication systems,□ in Proc.IEEE S&P, May 2007, pp. 116–130.

[13] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, –DSSS-basedflow marking technique for invisible traceback,□ in Proc. IEEE S&P,May 2007, pp. 18–32.

[14] N. B. Amir Houmansadr and N. Kiyavash, –RAINBOW: Arobust and invisible non-blind watermark for networkflows,□ inProc.16th NDSS, Feb. 2009, pp. 1–13.

[15] V. Shmatikov and M.-H. Wang, –Timing analysis in low-latencyMIX networks: Attacks and defenses,□ in Proc. ESORICS, 2006, pp.18–3

[16] http://www.planet-lab.org, 2008.

[17] I. Ahmad and Y.-K. Kwok, –A New Approach to SchedulingParallel Programs Using Task Duplication,□ Proc. Int'l Conf. ParallelProcessing (ICPP '94), pp. 47-51, 1994.

[18] H. Attiya, –Two Phase Algorithm for Load Balancing inHeterogeneous Distributed Systems,□ Proc. 12th Euromicro Conf. Parallel, Distributed and Network-Based Processing (PDP '04), p.434,2004.