International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

MAGIC TRAIN: DESIGN OF MEASUREMENT METHODS AGAINST BANDWIDTH INFLATION ATTACKS

Aravinda Thejas Chandra [1], Ramyashreee D M [2], Sahana Gaonkar [3], Samyuktha B [4], Sushmitha [5]

[1]*Professor, Department of Information Science and Engineering, SJCIT, Chikkaballapur, Karnataka, India*
[2,3,4,5]*Student, Department of Information Science and Engineering, SJCIT, Chikkaballapur, Karnataka, India*

Abstract - *An Bandwidth measurement is important for many network applications and services, such as peer-to-peer networks, anonymity services. To win a bandwidth-based competition for some malicious purpose, adversarial Internet hosts may falsely announce a larger network bandwidth. Some preliminary solutions have been proposed to this problem. They can either evade the bandwidth inflation by a consensus view or detect bandwidth frauds via forgeable tricks. However, smart adversaries can easily remove the forgeable tricks and report an equally larger bandwidth to avoid the consensus analyses. To defend against the smart bandwidth inflation frauds, we design magic train, a new measurement method which combines an unpredictable packet train with estimated round-trip time (RTT) detection. Inflation behaviors can be detected through highly contradictory bandwidth results calculated using different magic trains, or large deviation between the estimated RTT and the RTT reported by the train's first packet.*

Key Words*: *Bid Data, Hdfs, Mapping, Reducing, Clustering.

1. INTRODUCTION

Many measurement methods have been proposed to monitor and measure the rich network properties for the Internet, such as bandwidth (or capacity), latency, packet loss and reordering rate. These properties can well reflect the quality and dynamics of the Internet, and are therefore important for network management and optimization. In this paper, we focus on the bandwidth measurement which is important for many network applications and services. Design of bandwidth measurement methods against inflation attacks, especially the attacks which can falsely enlarge the reported bandwidth beyond network paths physical constraints (i.e., the capacity), is particularly challenging. On the one hand, most of existing techniques employ a train of two (also known as packet pair) or more back-to-back packets for bandwidth measurement. Adversarial hosts can simply delay the leading packet or rush the trailing one (provided that the measurement packets' a priori information, such as the number of measurement packets and each packet's sequence number, can be guessed in advance) to inflate the reported bandwidth to an arbitrary value. Although some secure measurement methods have been proposed for the bandwidth attacking problems, they do not consider more sophisticated bandwidth attack method. For example, an opportunistic bandwidth measurement method has been designed to secure P2P bandwidth evaluation systems. This method considers that a target peer cannot cheat all the measurement traffic with an equally larger bandwidth report. This method therefore employs multiple peers to measure the same target and exclude false bandwidth reports through a consensus analysis. In this project, we propose a new measurement method to detect smart bandwidth inflation frauds. The basic idea is to design an unpredictable, yet long enough, packet train (we call it magic train) for detection. Since the train is designed with unpredictable elements, adversarial hosts cannot have a priori knowledge about the measurement packets and therefore cannot regularly delay or correctly rush trains' packets on the fly. Irregular delays can make the bandwidth results computed based on different trains or a train's different successive packets highly contradictive, and rushing an incorrect packet can immediately reveal a dishonest behavior. By this design, even the smart attackers can avoid consensus analyses by inflating the bandwidth to an equally larger value and evade forgeable trick detections by mimicking normal network conditions, they cannot escape from our magic train detection due to the lack of a priori measurement knowledge. This design will cause the round-trip time (RTT) reported by the train's first packet to deviate significantly from the actual RTT if adversarial hosts receive all the train's packets before echoing them. To further prevent smart adversaries from faking a larger RTT by simply delaying RTT measurement packets, we propose a suite of RTT estimation algorithms to approximate the actual RTTs from nearby routers, subnet neighbors and the hosts from the same country/ISP. We also propose a new magic delay algorithm to measure the bottleneck bandwidth (i.e., capacity) by mitigating the cross-traffic's influence.

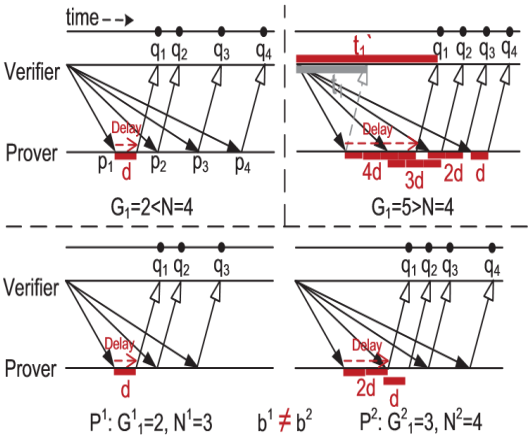
1.1 PROBLEM DEFINITION

The magic train is designed as a TCP packet train with an unpredictable length (i.e., the length is a random value and cannot be known by the prover in advance). By this design, the only way an adversarial over can obtain the length is

blind guess. Those unsupervised delaying and rushing behaviors can be detected due to either highly contradictory bandwidth results reported by different magic trains or nonexistent response packets. Given a magic train P = {(pi, qi) |i = 1, 2, …. N}, the verifier can randomly generate the train's length $N^{\check{}} \leq N \leq N^{\hat{}}$, where $N^{\check{}}$ is the smallest candidate value of the length and probability that the value N is selected as the magic train's length mainly introduced to address the computational challenge. $N^{\hat{}}$ is the large stone. Let PR[N] be the be the probability that the value N is selected as the magic train's length.

1.2 Detection of a priori delay attack

Examples of a priori delay attacks based on G1 < N (top left in the figure) and G1 > N (top right), as well as our MMTD algorithm (bottom). As can be seen, a priori delay attacks



based on G1 < N cause different magic trains to report different bandwidth results, whereas a longer train will induce a larger t1 for detecting the attacks based on G1 >N.

Fig -1: Priori delay attack

1.3 Detection of a priori rush attack

Examples of a priori attacks based on G1 < N (left) and G1 > N (right). As can be seen, if G1 < N, a priori rush attacks cannot inflate the bandwidth result (the dispersion between q1 and q3 remains the same). While G1 > N can immediately expose the attacks due to nonexistent response packet q4.

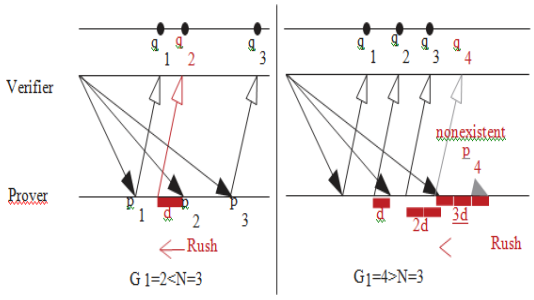


Fig -2: Priori rush attack

2. OBJECTIVE

To improving the accuracy in the Data Packets. To detect the errors in the Data Packets and Correct using Alternative Correcting Frequency. To prevent from the human errors. Detecting and the Bandwidths.

To improving the accuracy in the Data Packets

Improving accuracy of network intrusion detection systems under bibliometric data packets that are less likely to affects its detection accuracy.

To detect the errors in the Data Packets and correct using Alternative Frequency.

Detecting errors like dropped packets or retransmissions on the network level are relatively easy. During Transmission certain frequency is set to the data packets. Whatever the frequency is set we find errors in that. To correct the errors, we use the Alternative frequency.

To prevent from the Human errors.

There are many different human error classifications, which are being generated to assist analysis of human error and behavior.

3. METHODOLOGY

This section presents the design of magic train for detecting and preventing bandwidth inflation attacks. This train is designed unpredictable by the prover, long enough to prevent posterior adversaries and round-trip linkable by the verifier. After those sections, we further extend the train using a magic delay algorithm and thus make the train robust and capable of securing capacity measurement. In the given fig:1 provider sends Input which generates signal, using that signal data packets are created using TCP. Those data packet can detect bandwidth using priori delay attack and prior rush attack. After detecting the bandwidth verifier checks the error in data packet if error occurs correct data packet using magic train algorithm, RTTED,

MMTD, RTTPD otherwise it goes to verifier. In general, the magic train is designed as a TCP packet train with an unpredictable length. The only way an adversarial prover can obtain the length is blind guess. The adversary cannot accurately control the bandwidth report by delaying or rushing response packets on fly. Unsupervised delaying and rushing behaviors can be detected due.

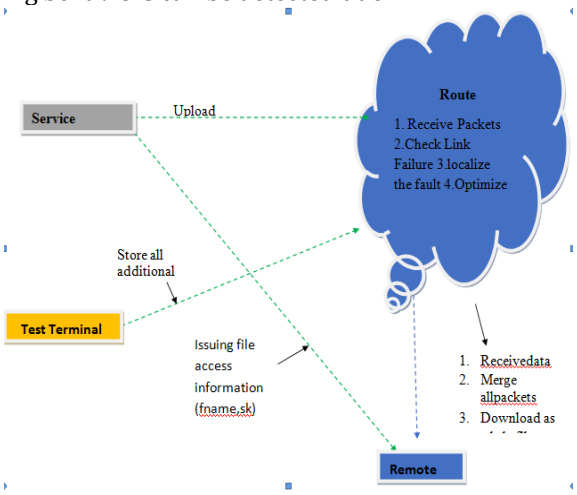


Fig -3: Architectural Diagram

4. IMPLEMENTATION

a. Service Provider

In implementation, we generate the TCP packets with a variable payload to balance the measurement intrusive characteristic and accuracy. Implementation of magic train for commodity computers and run it in the user space of Linux operating system, the time resolution we can accurately capture cannot be smaller than one millisecond. Service providers are the central place to configure all data packets.

b. Main Router

The main Router manages a multiple sub router (router1, router 2, router 3, and router 4) to provide data storage service. In a main router we can do some operations such as assign energy for nodes, view details of nodes and change status of nodes. In router n-number of nodes (A, B, C, D, E…) are present, and in a main router which has more energy & status is ON, it will communicate first. In a router service provider can assign energy for nodes, view details of nodes and change status of nodes. Router will accept the file from the service provider, the file wi split into five packets, then the highest energy router will select first, in a sub router highest energy sensor node are select and file will send to particular receiver.

c. Test Terminal Router

The test terminal router will control the all router, when the main router will receive the file from the service provider, then test terminal router will be activated and check the all sub router status and energy, that details will send to main router. In a test terminal router, we can do some operations such as assign energy for router, view details of router, change status of router and view log of router. In a test terminal router, we can view all sub router energy and status. The sub router which has more energy and status is ON, that router will select and then file will send to particular receiver.

d. Receiver

In this, there are n-number of receivers are present (A, B, C and D). the receiver can receive the data file from the service provider via router. The receivers receive the file by without changing the File Contents. Users may receive particular data files within the network only.

e. Attacker

Attacker is one who is injecting the fake message to the corresponding sensor nodes. The attacker adds malicious data to the particular sensor node and packets in a sub router (router1, router 2, router 3, and router 4). After attacking the nodes, that node information will change in main router.

5. CONCLUSION

In this paper, we advanced the state-of-the-art secure bandwidth measurement. We designed, analyzed, implemented and evaluated a new bandwidth measurement algorithm, we call it magic train, for securing uncooperative bandwidth measurement in adversarial networking environment. Our magic train is carefully designed to use an unpredictable measurement train to defeat even the smart adversaries.

REFERENCES

[1] Linux programmer's manual - raw (7) [online]. http://man7.org/linux/ man-pages/man7/raw.7.html, 2012.

[2] The libpcap project [online]. http://sourceforge.net/projects/libpcap/, 2013.

[3] Cloc: Count lines of code [online]. http://cloc.sourceforge.net/, 2013.

[4] John W Lockwood, Nick McKeown, Greg Watson, Glen Gibb, Paul Hartke, Jad Naous, Ramanan Raghuraman, and Jianying Luo. NetFPGA–an open platform for Gigabit-rate network switching and routing. In Microelectronic Systems Education, 2007. MSE'07. IEEE International Conference on, pages 160–161, 2007.

[5] The dummynet project [online]. http://info.iet.unipi.it/~luigi/ dummynet/, 2013.

[6] D-itg, distributed Internet traffic generator [online]. http://traffic.comics. unina.it/software/ITG/, 2013.

[7] Ghassan Karame, Boris Danev, Cyrill Bannwart, and Srdjan Capkun. On the security of end-to-end measurements based on packet-pair dispersions. IEEE Transactions on Information Forensics and Security, 2013.

[8] Joseph B. Kowalski and Kasimir Gabert. Tor network status [online]. http://torstatus.blutmagie.de/, 2014. [10] Alexa - actionable analytics for the web [online]. http://www.alexa.com/, 2014. [11] Bittorrent - delivering the world's content [online]. http://www.bittorrent. com/, 2014.