International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 6, Nov-Dec-2019, ISSN (Online): 2395-566X

Implementing Dynamic Shrew Attack Protection (Sap) to Defend Against Shrew Attack on Edges

Asst. Prof. Agath Martin
Department of Information Technology,
CUSAT/CUCEK, India

Asst. Prof. Nidhin Sani
Department of Information Technology,
CUSAT/CUCEK,
India

Asst. Prof. Abin John Joseph
Department of Electronics & Communication Engineering,
CUSAT/CUCEK,
India

Asst. Prof. Nishanth R.
Department of Electronics & Communication Engineering,
CUSAT/CUCEK,
India

Abstract **- Today's world is ruled by Internet which is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide. It consists of different type of routers and components to connect between various sources. We have both inner router which connects similar topology and outer router which connects different topology. One of the error types in communication is packet loss by any means which may leads to attacks in TCP connection. Shrew Attack leads to packet drops which uses a low-rate burst carefully designed to exploit TCP's retransmission timeout mechanism, can throttle the bandwidth of a TCP flow in a stealthy manner. We have so many reasons for packet dropping in edges; one of such reason is due to service level agreement (SLA) enforcement where it is a negotiated agreement between two parties (e.g., customer and service provider). In this paper we propose SAP (Shrew Attack Protection) which is an architectural model that used to defend against the shrew attack rather than attempting to detect and isolate. Basic concept which used here is differential tagging & drop rate collector to identify the victims Till now research have been shown mitigating the attack in inner router where the loss is very less. But we implement SAP in edges to reduce the packet-loss (where theoretically more drops in edges) and further stop the TCP connection from closing. SAP is a destination-port-based mechanism and requires only a small number of counters to find potential victims, which makes SAP readily implement able on top of existing router mechanisms.**

Keywords**- Shrew attack, differential tagging, drop rate.**

I. INTRODUCTION

Network security [1] involves all activities that organizations, enterprises, and institutions undertake to protect the value and ongoing usability of assets and the integrity and continuity of operations. An effective network security strategy requires identifying threats and then choosing the most effective set of tools to combat them. Threats to network security include: Viruses, Trojan horse programs, Attacks etc. In order to prevent this we use many security tools include Antivirus software packages, Secure network infrastructure, Identity services, Detection & Tolerance, Encryption, Security management etc.

Security issues arise when an attack is injected into the network. We have so many attacks which plays the major role in today's information transformation. Spoofing [2] attack causes a host or application to mimic the actions of another. Typically the attacker pretends to be an innocent host by following IP addresses in network packets. Eavesdropping [2] is the simplest type of attack. A host is configured to "listen" to and capture data not belonging to it. Carefully written eavesdropping programs can take usernames and passwords from user login network connections. Spamming [2] is the use of electronic messaging systems to send unsolicited bulk messages indiscriminately. While the most widely recognized form of spamming is e-mail spam.

Denial-of service causes the service or program to cease functioning or prevent others from making use of the service or program. These may be performed at the network layer by sending carefully crafted and malicious data-grams that cause network connections to fail. Preventing suspicious network traffic from reaching your hosts and preventing suspicious program commands and requests are the best ways of minimizing the risk of a denial of service attack. There is various Dos attack which is branched now in the network like ROQ attack, Low rated TCP attack, Byzantine attack and Shrew attack etc.

© 2019 IJSRET

International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 6, Nov-Dec-2019, ISSN (Online): 2395-566X

II. SHREW ATTACK- BACKGROUND

Shrew attack [3] is a low-rate DoS attack that attempts to deny bandwidth to TCP flows while sending at sufficiently low average rate to elude detection by counter-DoS mechanisms. Fig. 1 illustrates a single source Shrew attack with a packet stream of a square waveform that has an attack period T, a burst length L, and a peak rate R. Kuzmanovic and Knightly[4] showed that such an attack can reduce the throughput of TCP flows to near zero throughput or cause session resets if the attack has the following characteristics:

- R is large enough to induce victim's packet loss (i.e., R aggregated with existing traffic volume exceeds the link capacity);
- L is long enough to induce timeout (e.g., typically no less than the round-trip time), but sufficiently short to elude detection.
- T is chosen such that when flows attempt to exit timeout, they will face continuous drop (i.e., T is scaled in accordance to the minRTO).

The rationale behind this form of attack is as follows. When the initial attack burst of a Shrew attack causes packet drops for a TCP flow, the TCP sender will wait for the retransmission timer to expire before it starts to retransmit.

As such a retransmission timeout value is typically an integer multiple of the minRTO, subsequent retransmissions encounter another attack burst and are dropped repeatedly because the attack interval is synchronized with the retransmission timeout value. As a result, the TCP flow fails to exit the timeout phase and experiences near-zero end-to-end throughput or a session close. Moreover, most TCP implementations use among a small set of fixed minRTO values1, which makes a single Shrew attack effective for a large set of TCP flows. More specifically,

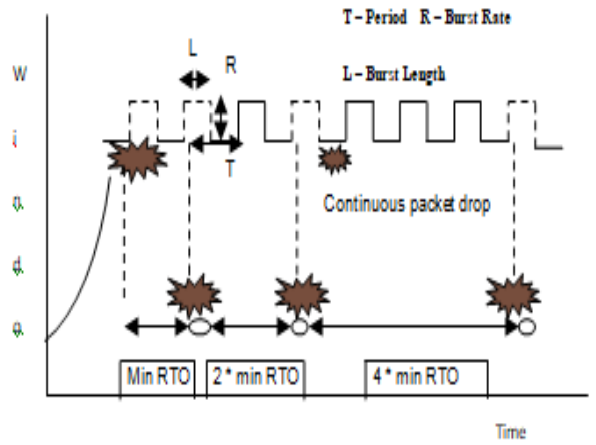

Fig.1. Shrew Attack uses an on-off Wave.

Kuzmanovic and Knightly [4] show that the normalized TCP throughput under a Shrew attack is:

$$\mu_{norm}(T) = \begin{cases} \frac{2T - minRTO}{T}, & T < minRTO \\ \frac{T - minRTO}{T}, & T \geq minRTO \end{cases}$$

where T is the attack period and L the peak length. This shows that a Shrew attack with sufficient peak rate and T = minRTO can cause the TCP throughput to become zero.

1. Drop Rate Collector- The role of the drop rate collector component is to monitor drop rates. We maintain two sets of counters for each application port: one set for arrivals, and the other set for drops. The values of these counters are in terms of cumulative bytes, and we denote their respective values at time interval t by a(t) and d(t). In SAP, we use a time-sliding window (TSW) to provide a smooth estimate of the drop rate for each port, as depicted in Fig. 3. Each time interval is a fixed duration of ts seconds. In particular, at the beginning of each time interval t, we initialize the arrival and drop counters for each port with a(t) = a(t − 1) and d(t) = d(t − 1). Then, during the time interval t, new byte arrivals or byte drops would increment a(t) or d(t) by the appropriate amount, respectively. To calculate the average drop rate [14] for each port over a sliding window of the last wn time intervals, we can compute it as follows:

$$p(t) = \frac{\Delta d(t)}{\Delta a(t)} = \frac{d(t) - d(t - \mathrm{Wn})}{a(t) - a(t - Wn)}$$

(2)

2. Fair Drop Rate Controller- Given the historical drop rates of all ports collected, the role of the fair drop rate controller is to determine the fair drop rate threshold pfair that it wants to limit. This fair drop rate threshold calculation actually depends on the average total drop rate pavg. This average drop rate will be updated every ts intervals using the same TSW algorithm:

$$p_{avg} = \frac{\sum_{i=1}^{i=N} \Delta d_i(t)}{\sum_{i=1}^{i=N} \Delta a_i(t)} \quad (3)$$

where Δdi(t) and Δai(t) are the cumulative arrival and drop counts, respectively, for application port i over the last wn intervals, and N is the number of application ports. In addition to continuously updating the average total drop rate pavg, the fair drop rate controller takes one more parameter, pmin. This parameter specifies a minimum drop rate threshold, under which SAP does not intervene. Specifically, if pavg ≥ pmin, then SAP uses the current average drop rate to serve as the fair drop rate threshold by setting pfair = pavg. If pavg < pmin, then

© 2019 IJSRET

SAP sets pfair = pmin. Since SAP is triggered to protect application ports with high drop rate, SAP effectively ignores any small fluctuation of error rates below pmin. Hence, pmin should be set large enough to tide over small fluctuations and low enough to trigger SAP quickly to protect victims against Shrew attacks.

III. EVALUATION

In this section, we describe experiments setup for evaluating SAP. Since we cannot easily implement our scheme into a real router due to time constrain, we primarily report simulation results using ns- 2 simulations. However, we can perform a set of controlled experiments on a real Juniper router testbed to verify that our simulation setting is realistic.
The network topology used in our simulations is shown in Fig.3.

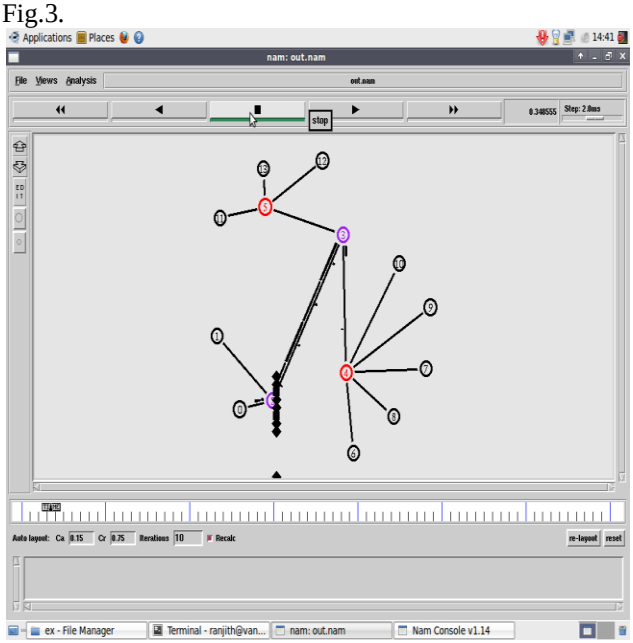

Fig.3. Simulation result of Shrew Attack.

A Shrew attack is launched from node 3 sending attack traffic destined to node 2 where the heavy packet loss is shown in simulation. The attack is created and shown in simulation result. Below table specify the peak rate and attack flow to know about the drop rate of inner and edges in simulation result.

Table I : Attack Flow Drop Rate

	Drop Rate during Attack Flow	
Peak Rate	Inner Router	Outer Router(Edges)
15 Mbps	33.1 %	37.8%
18 Mbps	44.8 %	49.1%
20 Mbps	49.3%	54.1%

The graph shows the difference between inner and edges to know about how the packet losses affect the edges in data transformation. Implementing SAP mechanism [10] is the on going process of my project. First step is to build about the differential tagging of the packet whether high or low priority. For this we use drop rate collector by using TSW for smooth estimation.

The remaining works are under progress for the project. After tagging the packets preferential dropping method is used to control the drop rate of the packet. In this method we use WRED [5] algorithm for fixing minimum threshold drop rate.

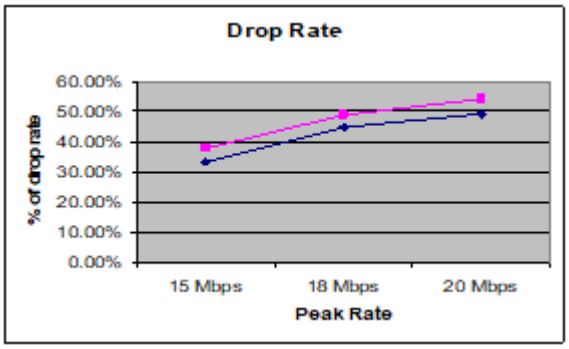

Fig.4. Drop Rate of Attack Flow in both inner and Edges.

IV. RELATED WORK

There have been a number of mechanisms proposed to protect against Shrew attack. Kuzmanovic and Knightly [4] first investigated the use of an active queue management (AQM) scheme to mitigate Shrew attacks. Although they experimented with a rather sophisticated scheme called REDPD [8], which takes the drop history of each flow into account, they found that it cannot satisfactorily mitigate the attack, which also agrees with our own results.

Another method that they explored was to randomize the fixed minRTO value in the TCP implementation in an attempt to mitigate the synchronized attack, which is not sufficient to fully mitigate the attack. Sun et al [7]. proposed a router-based detection approach against Shrew attack, where multiple routers along the path collectively detect attach based on the autocorrelation analysis among attack packets.

However, their detection scheme requires multiple data manipulation steps (e.g., noise filtering, feature extraction, etc.), which is often prohibitive for an on-line mechanism due to a huge number of packets going through high-speed network links. Shevtekar et al [6]. proposed an approach to detect these attack flows at edge routers by monitoring any periodic traffic pattern that is synchronized with minRTO and RTTs observed for other connections. Kwok et al [9]. proposed a scheme called

© 2019 IJSRET

HAWK, where they record attack flows into a small table and drop packets from those flows to halt the attack. In practice, it is difficult to identify and isolate the attack flows. In contrast, SAP does not attempt to identify the attack flows; it simply controls the drop rates of victim flows. In recent years, researchers have also explored the application of signal analysis techniques to Shrew attack detection [11]. Chen et al [12].

used frequency domain spectrum analysis to identify attacker flows. Luo et al. proposed a Wavelet-based approach to study the characteristics of low-rate TCP-targeted DoS attacks. In general, these detection algorithms are based on complicated signal analysis, which can be prohibitively expensive to realize at wire speeds for high-speed networks. We are unaware of any such solutions that can effectively mitigate Shrew attacks in real network environments. In contrast, our proposed SAP requires a small number of counters, and we expect the hardware implementation can handle the huge number of packets in today's high speed networks.

V. CONCLUSION

From the above result, we proposed a protection mechanism for Shrew attack called SAP which is an architectural model that used to defend against the shrew attack rather than attempting to detect and isolate. Basic concept which used here is differential tagging & drop rate collector to identify the victims. Till now research have been shown mitigating the attack in inner router where the loss is very less. It provides network operators with a broad first line of proactive defense against Shrew attacks, significantly neutralizing their impact. More broadly, our results show that SAP is also effective in allowing TCP flows in general to recover their throughput under a Shrew attack.

REFERENCES

[1]. http://www.cc.boun.edu.tr/network_security.html
[2]. http://www.wordiq.com/definition/Spamming, http://k12linux.mesd.k12.or.us/nag2/x-087-2-firewall.html,
[3]. http://www.cs.northwestern.edu/~akuzma/rice/shrew/
[4]. A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial ofservice attacks (The shrew vs. the mice and elephants)," in Proc. ACMSIGCOMM, 2003.
[5]. Cisco Systems, "Distributed Weighted Random Early Detection."[Online].Available:http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.pdf.
[6]. A. Shevtekar, K. Anantharam, and N. Ansari, "Low rate TCP denial of-service attack detection at edge routers." IEEE Commun. Lett., Apr.2005.
[7]. H. Sun, J. C. Lui, and D. K. Yau, "Defending against low-rate TCPattacks: dynamic detection and protection," in Proc. International Conf. Netw. Protocols, 2004.
[8]. R. Mahajan, S. Floyd, and D. Wetherall, "Controlling high-bandwidthflows at the congested router," in Proc. International Conf. Netw.Protocols, 2001.
[9]. Y. K. Kwok, R. Tripathi, Y. Chen, and K. Hwang, "HAWK: haltinganomalies with weighted choking to rescue well-behaved TCP sessionsfrom shrew DoS attacks," in International Conf. Computer Netw. MobileComputing, 2005.
[10]. Chia-Wei Chang, Seungjoon Lee, Bill Lin and Jia Wang, "The Taming of The Shrew:Mitigating Low-Rate TCP-Targeted Attack," in IEEE Transaction on Network Service Management, Vol. 7, No. 1, March 2010
[11]. X. Luo and R. K. C. Chang, "On a new class of pulsing denial-of-serviceattacks and the defense," in Proc. Netw. Distributed Syst. Security Symp.,2005.
[12]. Y. Chen, Y.-K. Kwok, and K. Hwang, "Filtering shrew DDoS attacksusing a new frequency-domain approach," in Proc. IEEE LCN Workshop Netw. Security, 2005.
[13]. M. Allman and V. Paxson, "On estimating end-to-end network path
[14]. properties," in Proc. ACM SIGCOMM, 1999.
[15]. M. Guirguis, A. Bestavros, I. Matta, and Y. Zhang, "Reduction of quality(RoQ) attacks on Internet end systems," in Proc. IEEE INFOCOM, 2005.
[16]. Y. Zhang, Z. M. Mao, and J. Wang, "Low-rate TCP-targeted DoS attacks disrupts Internet routing," in Proc. 14th Annual Netw. Distributed Syst. Security Symp. (NDSS), 2007.

© 2019 IJSRET