Victim-Assisted Mitigation Technique for TCP-Based Reflector DDoS Attacks

Basheer Al-Duwairi and G. Manimaran

Department of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011, USA
{dbasheer,gmani}@iastate.edu

Abstract. This paper develops the concept of victim-assistance for denial of service (DoS) mitigation. The proposed concept is utilized within a simple, yet effective scheme designed for mitigating TCP-based reflector DoS attacks. The proposed scheme, called SYN number based filtering (SNF), takes into account the TCP's connection establishment behavior and the inherent features of the attack itself. The main idea of the SNF scheme is to restrict the choice of the initial sequence numbers of SYN packets to certain pattern, such that corresponding SYN-ACK packets can be validated at the ISP's perimeter. We evaluate the proposed scheme through analytical studies for classical and advanced attacks using two performance metrics, namely, the false positive and false negative rates. Our analysis shows that the proposed scheme offers low false positive and false negative rates. In addition, we identify several research problems based on the proposed concept.

1 Introduction

Recently, reflector-based denial of service (RDoS) attacks are being used by attackers frequently to affect the availability of high profile servers [8, 13]. In these attacks, a large number of compromised computers under attackers control (the slaves) are instructed to continuously send request packets to a set of Internet reflectors (an Internet reflector is an IP host that will reply to any request packet). The source address of each of these request packets is spoofed to be the same as the address of the targeted site. As a result, the reflectors send their replies to the given address causing packet flooding at that site. Using Internet reflectors complicates the problem of DoS attacks. Researchers are more concerned about these attacks because attack packets (reply packets originated from the reflectors themselves) carry legitimate IP source addresses making it difficult to trace original attack sources. Also, because these attacks are usually characterized by an amplification factor that increases their intensity.

The analysis presented in [13] shows that RDoS attacks are feasible in variety of request/reply based protocols, including TCP, UDP, ICMP, and DNS. For example, in Smurf attacks [3], the attacker sends ICMP echo requests (pings) to the broadcast address of a network, so the victim is hit by many more packets. The Fraggle (UDP packet magnification) attack uses UDP echo packets in the same fashion as the ICMP echo packets. In TCP-based RDoS attacks [8], attackers may take advantage of the availability and connectivity of large number of Internet reflectors to coordinate a highly distributed DoS attack. This can be done by abusing the TCP protocol in the following way: An attacker, A, selects a set of Internet reflectors R=$\{R_1$,R_2, ...,$R_n\}$. It then sends low rate faked SYN

packets to each of these reflectors with a spoofed source address equals to that of the final target, V. For each received SYN packet, the reflectors reply with a SYN-ACK packet to the given address, V. Therefore overwhelming the victim site, V, by high aggregate rate of SYN-ACK packets.

Mitigating RDoS attacks is extremely important issue. In this paper, we develop the concept of victim-assistance and use it in the context of a novel scheme to mitigate TCP-based RDoS attacks. This type of attacks has been encountered repeatedly in recent years. For example, attacks against GRC.com were reported and analyzed in [8]. In addition, many TCP-based reflector attacks were captured in the traffic traces taken at Los Nettos ISP network [10]. Its special importance is due to the following reasons: (1) different than other types of RDoS attacks, *TCP-based RDoS cannot be mitigated by blind filtering of attack packets, because such solution would prevent the victim itself from establishing any TCP connection*[1], (2) attack amplification is achieved through multiple retransmission of SYN-ACK packets by the reflectors themselves after each time out, (3) attackers are attracted by the fact that TCP carries 95% of today's Internet traffic and 80% of the total number of flows in the Internet [12], and (4) the fact that any general purpose TCP connection-accepting Internet server could be used as a packet reflection server, which provides attackers with large pool of servers to be used as reflectors.

The proposed scheme, called SYN Number-based Filtering (SNF), is based on restricting the choice of the initial sequence number (ISN) of TCP connection establishment requests initiated by the victim, such that legitimate reply packets can be validated at the ISP (Internet Service Provider) perimeter. The ISN restriction is achieved by requiring it to contain a secret pattern, C_s, known only to the victim and to the ISP's edge routers. We evaluate the proposed scheme by first assuming a classical attack in which attackers do not react to the defense scheme. Then, we take into account the ability of the attacker to perform advanced attacks in which attack packets are generated with ISNs that contain the C_s used by the victim. A simple stochastic analysis are conducted to gain an insight on the performance of the proposed scheme. The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 develops the concept of victim-assistance. Section 4 describes the proposed scheme. Finally, conclusions and future research problems are outlined in section 5.

2 Related Work

In general, RDoS attacks can be defeated either by filtering the reflected attack packets (which hold valid IP source addresses), or by solving the origins of the problem (i.e., filtering IP packets with spoofed source addresses). In this section, we discuss the main research efforts that addressed the RDoS explicitly, and we review some of the research efforts that targeted the filtering of spoofed IP packets.

2.1 Detection and Prevention of RDoS Attacks

In [13], filtering of different types of reply attack packets that share certain attributes, such as the destination port number and IP destination address,

[1] Examples of servers that could be targets of this type of attacks include servers that initiate TCP connections frequently, such as FTP server, proxy server, and Socks server.

was considered. However, the issue of collateral damage (i.e., filtering legitimate replies as well) was not addressed. In [15], a distributed approach for detecting reflector attacks was proposed. The approach is based on sharing beliefs among potential reflectors if any abnormal traffic is observed, such that the reflectors become aware that they are being used in a RDoS attack, and consequently start ignoring incoming request packets that have source address equals to the victim's address. Clearly, this approach cannot be deployed in practice because there is no way by which certain reflector knows the group of reflectors participating in ongoing attack, such that it can share its belief with them. Even if attack detection is possible among set of reflectors, there is no mechanism by which reflectors can distinguish attack packets from legitimate packets. Moreover, it is possible for the attackers to abuse the scheme by sharing their own beliefs with many other innocent reflectors in order to drop legitimate requests received by them.

2.2 Filtering of Spoofed IP Packets

Generally, filtering of spoofed IP packets can be done at the source network, intermediate routers, or at the destination network. For example, in ingress filtering [7], routers are configured to block packets that arrive at the edge router of the source network with illegitimate source addresses. Obviously, the major problem with this approach is that it may violate some existing setups and protocols such as Mobile IP and multi-homing. It is also difficult to convince ISP administrators to support ingress filtering because the benefit is not felt directly by the deploying ISP. The proposed scheme is different in this aspect because an ISP network will directly benefit from its deployment. Another example is the SAVE protocol [11], which is designed to provide routers with the information needed for source address validation. The main problem of this protocol is that legitimate packets may be filtered even in the absence of an attack. This is due to routing instability which leads to errors in the source address validation tables maintained at the SAVE enabled routers. In general, such schemes require large scale deployment to prevent IP source address spoofing efficiently.

Hop count filtering [9] is a simple approach to drop spoofed packets at the destination network. It is based on observing that the distance travelled by a spoofed packet is usually different than that travelled by a packet originated from the actual spoofed source. Therefore, attack packets can be distinguished and dropped directly. The main drawback of this approach is the need to keep up to date database of source addresses and their distances. This might be difficult due to route changes. Also smart attackers may spoof IP addresses that never communicated with the given reflector such that it cannot judge about the validity of these packets.

3 The Concept of Victim-Assistance

It is well known that a DoS attack against an end-system has a direct impact not only on that system, but also on the network in which the targeted system is located. This is due to attack traffic aggregation near the victim which leads to network bandwidth exhaustion. Therefore, it is of primary importance to filter attack traffic as far away from the victim as possible. In this context, the placement of DoS detection and mitigation modules involves several tradeoffs

that need to be considered. (Modules refer to devices together with the required software).

The tradeoff introduced by the placement of DoS detection and mitigation modules. The placement of the modules is of primary importance as it involves a tradeoff between efficiency and practicality. This is due to the fact that DoS attacks consume both network and end-system resources. At one extreme, these modules can be placed at the victim itself. At another extreme, these modules can be placed at the ISP's perimeter. Between the two extremes, the placement of DoS detection and mitigation modules can be along an optimal boundary defined by the victim. The following discussion explains the tradeoffs introduced in each case.

- *At the victim:* With respect to DoS detection, the victim is the best location for fast and accurate attack detection in most attack scenarios. With respect to DoS mitigation, the victim is also the best location from deployment point of view, because the DoS mitigation module would be installed at a single system (i.e., at victim) which has complete knowledge about the attack, and, in most cases, it can easily classify incoming traffic as attack or legitimate. Unfortunately, this ability does not alleviate the effect of attack packets as they consume significant resources by the time they are identified.
- *At the ISP's perimeter:* With respect to DoS detection, edge routers along the ISP's perimeter cannot detect the existence of an attack because the amount of attack traffic seen by each edge router individually is very small as compared to that seen by the victim. However, with respect to DoS mitigation, installing the DoS mitigation modules far away from the victim provides protection for the network as well as for the targeted system. It is preferred to install the DoS mitigation modules at the ISP's edge routers to form a line of defense at the ISP's perimeter, because (1) any traffic destined to the victim has to pass through one of the ISP's edge routers (2) edge routers are usually computationally capable and can perform the task of packet filtering. However, the major problem with this approach is that edge routers cannot perform accurate traffic classification by themselves due to the lack of knowledge about the ongoing attack and how to make a distinction between attack packets and legitimate packets.
- *At an optimal boundary:* Optimal boundary is a conceptual term that refers to the periphery around the victim where it is possible to protect against end system exhaustion as well as bandwidth exhaustion. Such periphery is located somewhere between the ISP's perimeter and the victim itself. The DoS detection/mitigation modules installed along that periphery can (1) detect attacks that are difficult to detect at the ISP's perimeter (2) perform more efficient DoS mitigation as compared to mitigation at the ISP's perimeter. Although such placement policy achieves the best of both worlds (i.e., protection from bandwidth exhaustion with accurate detection and efficient mitigation), it has several practical limitations that prevent its deployment. For example, the conceptual optimal boundary is victim-oriented. Moreover, it requires modification at core routers. Therefore, raising deployment concerns.

Fig. 1 depicts the tradeoff introduced by the placement of the DoS detection/mitigation modules. Obviously, *performing DoS mitigation at the ISP's perimeter provides protection from both bandwidth and end system exhaustion.* In fact, the perimeter-based DoS mitigation architecture is not new as it has been

used in previous work (e.g., [2, 5]). However, different than earlier research, we introduce the concept of *online* perimeter-based DoS mitigation wherein edge routers of the ISP network take advantage of the inherent features of the ongoing attack to perform per-packet classification and filtering. In this context, ISP's edge routers should be supplemented by special mechanisms in order to perform online attack mitigation. The concept of victim-assistance is proposed to achieve this objective. Victim-assistance refers to the direct role of the victim in identifying attack traffic before reaching its target. The mitigation scheme at ISP's edge routers requires the victim to cooperate with them in a manner that leads to accurate classification of incoming traffic. In this context, several research problems can be identified. For example, what information should be provided by the victim? How this information is to be communicated to edge routers? When to activate/terminate the mitigation scheme and which edge routers? In fact, the answers to these questions depend on the attack type which can be determined by the victim itself. In the next section, we illustrate the use of this concept in mitigating TCP-based RDoS attacks.

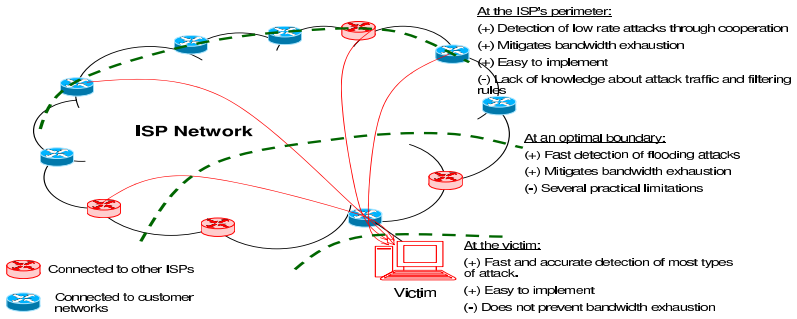

Fig. 1. The tradeoff introduced in the placement of DoS detection and mitigation modules within an ISP network

4 SYN-Number Based Filtering (SNF)

TCP connection establishment procedure is characterized by its deterministic nature regarding the type and content of messages exchanged between the communicating parties. This forms the basis of our scheme for defense against TCP-based RDoS attacks. Theoretically, an incoming SYN-ACK packet (destined to the victim) can be validated by inspecting its sequence number to see if it matches the ISN of the corresponding SYN packet plus one. Although such validation could be done very effectively at the victim itself, it would not be useful to stop bandwidth exhaustion. At the same time, it is more challenging to perform validation at the edge routers of the ISP network that contains the victim, because it would be necessary to maintain state information about each SYN packet sent by the victim.

The proposed scheme, called SYN Number based Filtering (SNF), enables edge routers of the ISP network to validate incoming SYN-ACK packets destined to the victim without maintaining state information about *individual* SYN packets sent originally by the victim. The main idea of the SNF scheme is to restrict

the choice of the Initial Sequence Numbers (ISNs) of SYN packets generated by the victim to certain pattern, such that corresponding SYN-ACK packets can be validated at the ISP perimeter. In the following subsections, we develop two variants of the SNF scheme to counter both *classical* and *advanced* TCP-based RDoS attacks. Classical attacks are those in which the attacker does not react to defenses employed by the victim during attack period. Advanced attacks are those in which the attacker monitors victim's reaction and acts accordingly. In the rest of this section, we develop a simple analytical models to evaluate the performance of the proposed scheme under both classical and advanced attacks focusing on the following performance metrics:

- False positive rate (FPR): The percentage of attack SYN-ACK packets that are falsely allowed to pass the ISP perimeter toward the victim.
- False negative rate (FNR): The percentage of legitimate SYN-ACK packets that are falsely filtered.

Clearly, we need both metrics to be minimized.

4.1 Countering Classical TCP-Based RDoS Attacks

In the classical TCP-based RDoS attacks, SYN packets generated by attack nodes (recall that TCP-based reflector attacks start by generating spoofed SYN packets that hit the reflectors, forcing them to flood the victim with the corresponding SYN-ACK packets) will continue to hold ISNs according to the rules specified originally by the attack tool itself. To counter such attacks, we propose the Basic-SNF scheme in which the victim chooses a *secret pattern* for its ISNs. The secret pattern, C_s, is set by fixing a randomly chosen k-bits out of the 32 sequence number bits to form a specific combination. The victim then initiates a filtering request by multicasting a message to all edge routers of the ISP network. The message includes the specific combination, C_s, that should be used to validate incoming SYN-ACK packets. Edge routers inspect incoming SYN-ACK packets destined to the victim according to the algorithm shown in figure 2. In this algorithm, if the (ACK number - 1) of the SYN-ACK packet contains the secret pattern, C_s, then the packet is passed. Otherwise, it is filtered.

Basic-SNF (SYN-ACK packet P)
a. X=P.ACK_number-1
b. if (X contains C_s) pass P
c. else drop P

Fig. 2. Basic-SNF algorithm. This algorithm is performed at each edge router while attack is going on. It is applied only to SYN-ACK packets destined to the victim

Assuming that the ISNs of attack packets are generated randomly, the probability that a given attack packet will contain a bit combination that matches the secret pattern currently in use, C_s, is equal to $\frac{1}{2^k}$, where k is the length of C_s. This implies that the Basic-SNF scheme is very effective against classical attacks. However, a major weakness of the this scheme is that the attacker does not have to discover the actual secret pattern used by the victim. In fact, it is sufficient to intercept a single legitimate SYN or SYN-ACK packet and using its corresponding ISN as an input to subsequent attack packets. In order to address this issue,

we propose to change the secret pattern, C_s, in a way that significantly reduces attacker's chances of launching replay attacks (i.e., using previously used secret pattern). To achieve this, the secret pattern can be changed either periodically; where the C_s is changed regularly every T time units, or reactively; where C_s is changed whenever a replay attack is detected by the victim.

4.2 Countering Advanced TCP-Based RDoS Attacks

In this model, we assume that an attacker experiences some delay D_a before being able to reconfigure its attack tool to generate SYN packets that carry valid SYN-numbers to the reflectors (i.e., packets that hold the secret pattern currently in use by the victim). This delay consists of the time required to intercept SYN packets generated by the victim itself, the communication time to the zombies under attacker's control, and the attack tool reconfiguration time. To deal with this type of attacks, we modify the Basic-SNF scheme by changing the secret pattern periodically.

Analysis of Periodic-SNF In this scheme, we propose changing the secret pattern of the SYN number periodically (i.e., every T time units). This is done by dividing the time since attack is detected into fixed intervals, each of length T. A secret pattern, C_i, is then assigned for the i-th interval, for $i = 1, 2, 3, ...$ It is assumed that interval length and secret pattern assignment are known to edge routers via authentic multicasting. Any SYN packet generated by the victim during the i-th interval must hold C_i as part of its ISN. For SYN-ACK packet validation, each edge router acts independently according to the filtering algorithm shown in Fig. 3.

The edge router specifies the interval in which the packet to be inspected has arrived (step a). It is expected that SYN-ACK packets that correspond to SYN packets which are sent at the end of a given interval (e.g., the $(i-1)$-th interval) may arrive at the edge router during the subsequent interval (i.e., the i-th interval). Such packets will be dropped if the validation is done based on the current secret pattern in use (i.e., C_i) alone. To reduce the impact of this problem, a packet is considered to be valid if its secret pattern matches either of C_i or C_{i-1} given that its arrival time falls between $(i-1)T$ and $(i-1+\alpha)T$, where $0 \leq \alpha \leq 1$ (step c). Otherwise the validation is performed based on C_i alone (step d).

Fig. 4 shows two scenarios for connection establishment by the victim assuming α to be zero. In scenario 1, the request (i.e., a SYN packet) is sent by the victim at time t_1 during the $(i-1)$-th interval, the reply (i.e., the corresponding SYN-ACK packet) takes x_1 time units to arrive at the ISP perimeter. Since $(t_1 + x_1 < T)$, the reply is passed. In scenario 2, the reply that corresponds to the request made at t_2 arrives the ISP perimeter at $t_2 + x_2 > T$ (i.e., in the i-th interval. The reply is filtered in this case because it holds a secret pattern (C_{i-1}) that is no longer in use.

Generally, because of the attacker's ability to generate valid attack packets after D_a seconds each time a new secret pattern is applied (recall that D_a represents the amount of delay experienced by the attacker before being able to reconfigure its tool), it can be seen that the attacker can pass its packets during the window of time specified by $(1 + \alpha)T - D_a$. It is clear that the choice of T and α introduces a tradeoff that shapes the false positive and false negative rates. The following analysis focuses on evaluating both metrics.

Periodic-SNF (SYN-ACK packet P)

a. $i = \lceil \frac{P.at}{T} \rceil$
b. X=P.ACK-1
c. if ($(i-1)T \leq P.at \leq (i-1+\alpha)T$)
   - if ((X contains C_i) OR (X contains C_{i-1})) pass P
d. else if (X contains C_i) pass P
e. else drop P

Fig. 3. Periodic-SNF algorithm. This algorithm is performed at each edge router while attack is going on. It is applied only to SYN-ACK packets destined to the victim. $P.at$ stands for the packet arrival time. P. C_i stands for the secret pattern assigned for interv

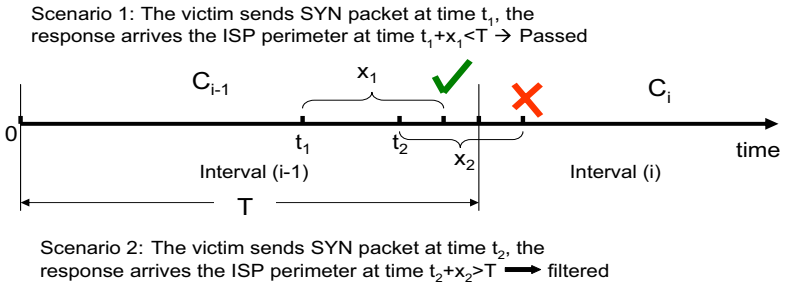

Fig. 4. The impact of changing the secret pattern of the ISN for connections generated by the victim. The secret pattern is set to C_{i-1} during the $(i-1)$-th interval, and to C_i during the i-th interval

FPR for Periodic-SNF ($FPR_{periodic}$): $FPR_{periodic}$ can be expressed as the expected value of the ratio of time in which the attack traffic penetrates the ISP perimeter. Let $A_{periodic}$ represent this ratio. It is clear that:

$$A_{periodic} = \frac{(1+\alpha)T - D_a}{(1+\alpha)T} \tag{1}$$

Assuming that D_a has an exponential distribution[2] with rate μ_a, the expected value of $A_{periodic}$ is given by Equation (2).

$$FPR_{periodic} = E[A_{periodic}] = (1+\alpha) - \frac{1}{\mu_a(1+\alpha)T} \tag{2}$$

FNR for Periodic-SNF ($FNR_{periodic}$): In order to gain an insight on the performance of the periodic SNF scheme in terms of the false negative rate $FNR_{periodic}$, we assume that the victim is generating TCP connections at a Poisson rate of λ per unit time, and that the distribution of connection response time X (defined as the time between sending the SYN packet by the victim and receiving the corresponding SYN-ACK at the perimeter) is given by $X = c + Y$, where c is a constant and Y is a variable component. This assumption is based on the definition of the round trip time (RTT), which consists of a *fixed* component given by the sum of link propagation latencies and transmission and processing delays on all nodes in the forward and reverse direction, and additional *variable* components due to queuing and processing delays at overloaded routers and end-hosts. Assuming that Y can be expressed as the summation of n identical

[2] This assumption is made because actual measurements of D_a are not available.

exponentially distributed random variables each with rate β, Y is said to be an Erlang random variable with parameters β and n.

Based on these assumptions, we are interested in finding an expression for $FNR_{periodic}$, which is the same as the probability of filtering a legitimate packet. The following equation represents the probability distribution function of X:

$$f_X(x) = \begin{cases} \frac{\beta^n (x-c)^{n-1} e^{-\beta(x-c)}}{(n-1)!} & \text{if } x > c; \\ 0 & \text{if } x \leq c. \end{cases}$$

We proceed in our analysis for $x > c$. The cumulative distribution function (CDF) of X is given by:

$$F_X(x) = 1 - \sum_{k=0}^{n-1} \frac{e^{-\beta(x-c)} \beta^k (x-c)^k}{k!} \tag{3}$$

It can be seen that Pr(a legitimate packet P is filtered | P is sent at time t) = Pr($X \geq (1+\alpha)T - t$ | P is sent at time t) = 1 - Pr($X \leq (1+\alpha)T - t$ | P is sent at time t). We call this probability the conditional probability of legitimate packet filtering $CPLP_{periodic}$. This probability is given by the following equation:

$$CPLP_{periodic} = 1 - F_X((1+\alpha)T - t) \tag{4}$$

Substituting 3 with $(x = (1+\alpha)T - t)$ in 4, we obtain:

$$CPLP_{periodic} = \sum_{k=0}^{n-1} \frac{e^{-\beta((1+\alpha)T - t - c)} \beta^k ((1+\alpha)T - t - c)^k}{k!} \tag{5}$$

Unconditioning on the time at which each request is generated, we obtain:

$$FNR_{periodic} = \frac{1}{T} \int_0^T CPLP_{periodic} dt \tag{6}$$

In practice, each end-to-end path is characterized by its own minimum possible RTT (i.e., the fixed component c of the RTT along the given path). Also, the variable component represented by the erlangian variable Y has its own parameters β and n for each TCP segment due to the diversity of destinations and the variability of load on network routers. However, for the purpose of obtaining an upper bound on the probability of a legitimate SYN-ACK packet being filtered, due to secret pattern change, we can assume worst case values for c, β, and n.

We performed extensive simulation experiments to evaluate the value of $FNR_{periodic}$ for several values of α and T. It is to be noted that $FNR_{periodic}$ does not depend on λ since all arrivals are independent. However, the value of λ was fixed to 0.004 just for the sake of experimentation. Fig.5 plots the percentage of false positive and false negative rates side by side to gain an insight on the tradeoff introduced by α and T. To obtain a worst case value for the random variable X, we assigned a value of 1 second to the constant c, which was found to be the maximum reported minimum RTT [1]. Also, to eliminate the variability of parameters n and β, we assigned the values 30 and 2 for them, respectively. This corresponds to the maximum reported path length [4], and a relatively large queuing and processing delay at intermediate routers. α was varied along the x axis in the range of 0 to 0.2.

By recalling that αT represents the amount of extra time in which a secret pattern of a given interval remains valid throughout the subsequent interval, it is easy to explain the decrease of the false negative rate by increasing α. It is also straightforward to explain the increase of false positive rate at the same time. It can be observed that as T increases, the false negative rate decreases. This is expected since secret pattern change will be less frequent, resulting in lower percentage of legitimate packets being dropped. At the same time, the false positive rate increases sharply. For example, the false positive rate exceeds 30% for $T = 350$. This is due to the fact that increasing T provides larger window of time for the attacker to pass his traffic toward the victim. These results lead to an important conclusion: The Periodic-SNF scheme is expected to perform well in terms of false negative rate. However, it is expected to perform well in terms of false positive rate only if the value of T happens to be close to the rate at which the attacker reconfigures its tool with a valid secret pattern. As a future work, we plan to investigate the effectiveness of a reactive version of the SNF scheme in which the secret pattern change is triggered by certain event rather changing it periodically.

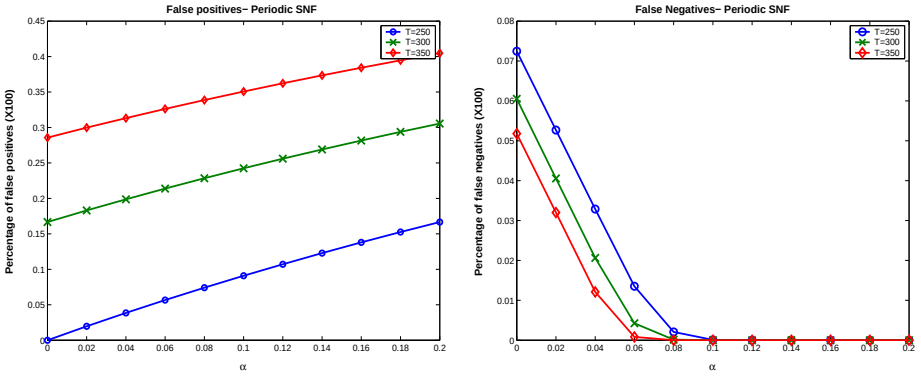

Fig. 5. The effect of α on the (left) percentage of false positive and (right) percentage of false negative for different values of T. μ_a, λ, β, and n were fixed to 0.004, 5, 2, and 30, respectively

4.3 Practical Consideration

In this section, we discuss the practicality of the proposed schemes focusing on the following two aspects.

The impact of restricting the ISN to specific pattern: It is known that when an end system sends its SYN packet to establish a TCP connection, it chooses an (ISN) for that connection. Typically, the ISN should change over time, so that each connection has a different ISN. New instances of a connection may be established if the connection is opened and closed in quick succession, or if the connection breaks with loss of memory and is then reestablished. The TCP should be able to identify duplicate segments from previous instances of the same connection. To achieve this, RFC 793 [16] specifies that the ISN should be viewed as a 32-bit counter that is incremented by one every 4 microseconds. Although this practice is common in most TCP implementations, it is not necessary for proper functionality of the protocol, and it can be violated in favor of mitigating

the effect of an ongoing attack. The main impact of restricting the choice of ISN is the increase the possibility of *duplicate segment problem* (i.e., having duplicate segments from previous instances of the same connection) if multiple instances of the same connection are created frequently. However, this drawback can be tolerated during TCP-based RDoS attack, because it is more important to mitigate the effect of the attack. It is also, important to point out that restricting the ISN to certain pattern does not increase the vulnerability of connection hijacking through TCP sequence number guising. This argument is based on the fact that for a connection to be hijacked, the sequence number of the server, rather the client, needs to be guessed.

The need to perform TCP's header inspection: The proposed scheme requires that all edge routers of the ISP network that contains the victim to inspect each packet closely enough to determine if it is a TCP SYN-ACK packet, so the mechanism can more closely analyze the packet to see if it is a legitimate SYN-ACK. This requires, at least for all TCP packets destined to the victim, examining the TCP header fields. Fortunately, recent advancements in router manufacturing technology has allowed performing such functions at high Internet speeds. For example, Cisco routers currently have the TCP intercept feature [6] which implements a software to protect TCP servers from TCP SYN-flooding attacks. In this context, a router is configured to perform TCP intercept in which the software actively intercepts each incoming connection request (SYN) for the purpose of legitimacy check. We believe that a similar approach can be adopted to support the proposed SNF scheme. The main difference will be in intercepting SYN-ACK packets instead of SYN packets, and accordingly the router should be configured to perform packet filtering based on the specified rules.

5 Conclusions

TCP-based reflector distributed denial of service attacks represent a challenging problem. Using SYN-ACK packets to flood certain system complicates the process of distinguishing these packets from legitimate packets destined to the same system. In this paper, we developed the concept of victim-assistance to mitigate such attacks. Then, we proposed a scheme, called SYN number-based filtering (SNF), which is based on the idea of restricting victim's choice of the initial sequence numbers of its generated connection establishment requests during an attack, such that legitimate incoming SYN-ACK packets can be verified at the ISP perimeter by checking specific bit pattern.

Through analytical studies, we showed that the two variants of the SNF scheme (i.e., the Basic-SNF and the Periodic-SNF) offer performs very well for classical attacks and advanced attacks, respectively. Our analysis shows that the Basic-SNF scheme is very effective against classical attacks as it ensures an extremely low false positive rate and exactly zero false negative rate. For advanced attacks, the periodic-SNF performs very well in terms of false positive and false negative rates especially if the period of changing the secret pattern is close to attacker's response time.

Obtaining victim's assistance represents the basis in defending against TCP -based RDoS attacks in the proposed scheme. The importance of this approach is reflected in the fact that new source of information (i.e., the victim) is now available to make distinction between attack and legitimate packets. This approach

opens new directions of research in designing efficient defenses for DoS attacks, which includes (1) designing and analyzing a reactive version of the SNF scheme to counter advanced DoS attacks more efficiently, and (2) investigating the viability of victim's assistance in defending against other types of DoS attacks. In this context, several issues needs to be addressed. For example, what information should be provided by the victim? How this information is to be communicated to edge routers? When to activate/terminate the mitigation scheme and on which edge routers? How to extend the concept of victim-assistance to inter-domain setting?

6 Acknowledgments

The authors would like to thank Prof. Ahmed E. Kamal for his constructive suggestions and help to improve the quality of the work presented in this paper.

References

1. J. Aikat, J. Kaur, F. D. Smith, and K. Jeffay, "Variability in TCP Round-trip Times," in *Proc. of the ACM SIGCOMM Internet Measurement Conference (IMC'03)*, Miami, FL, October 2003.
2. S. Chen and Q. Song, "Perimeter-Based Defense against High Bandwidth DDoS Attacks," to appear in *Proc. IEEE Transactions on Parallel and Distributed Systems (TPDS 2005)*.
3. CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks. Available at: http://www.cert.org/advisories/CA-1998-01.html.
4. B. Cheswick, H. Burch, and S. Branigan, "Mapping and Visualizing the Internet", in *Proc. USENIX Annual Technical Conference 2000*.
5. Cisco systems, "Defeating DDoS Attacks," white paper.
6. Cisco systems, "Configuring TCP Intercept (Prevent Denial-of-Service Attacks)," available at: http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt3/scdenial.htm
7. P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial-of-service attacks which employ IP source address spoofing," RFC 2827, 2000.
8. S. Gibson,"Distributed Reflection Denial of Service," February 22_{nd}, 2002. Available at http://grc.com/dos/drdos.htm.
9. C. Jin, H. Wang, and Kang G. Shin," Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic," in *Proc. ACM Conference on Computer and Communications Security (CCS)'2003*, Washington, DC, October 2003.
10. A. Hussain, J. Heidemann, and C. Papadopoulos,"A Framework for Classifying Denial of Service Attacks," in *Proceedings of ACM SIGCOMM 2003*.
11. J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang., "SAVE: Source address validity enforcement protocol," In *Proc. of IEEE INFOCOMM 2002*, April, 2002.
12. M. Mellia, I. Stoica, and H. Zhang, "TCP Model for Short Lived Flows," in *Proc. IEEE Communication Letters*, Feb. 2002.
13. V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," in *Proc. Computer Communication Review 31(3)*, July 2001
14. T. Peng, C. Leckie and K. Ramamohanarao, "Protection from Distributed Denial of Service Attack Using History-based IP Filtering," in *Proc. of IEEE International Conference on Communications (ICC2003),* May 2003.
15. T. Peng, C. Leckie and R. Kotagiri,"Detecting reflector attacks by sharing beliefs," in *Proc. IEEE Global Communications Conference*, October, 2003
16. J. Postel, "Internet Protocol - DARPA Internet Program Protocol Specification," RFC 791.