TRAFFIC REDIRECTION ATTACK PROTECTION SYSTEM (TRAPS)

Vrizlynn L. L. Thing[1,2], Henry C. J. Lee[2] and Morris Sloman[1]
[1]Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK, [2]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613

Abstract: Distributed Denial of Service (DDoS) attackers typically use spoofed IP addresses to prevent exposing their identities and easy filtering of attack traffic. This paper introduces a novel mitigation scheme, TRAPS, whereby the victim verifies source address authenticity by performing reconfiguration for traffic redirection and informing high ongoing-traffic correspondents. The spoofed sources are not informed and will continue to use the old configuration to send packets, which can then be easily filtered off. Adaptive rate-limiting can be used on the remaining traffic, which may be attack packets with randomly-generated spoofed IP addresses. We compare our various approaches for achieving TRAPS functionality. The end-host approach is based on standard Mobile IP protocol and does not require any new protocols, changes to Internet routers, nor prior traffic flow characterizations. It supports adaptive, real-time and automatic responses to DDoS attacks. Experiments are conducted to provide proof of concept.

Key words: Distributed Denial of Service; Attack Response System; Adaptive Security.

1. INTRODUCTION

In Denial of Service (DoS) [1] or Distributed DoS (DDoS) attacks, a large number of malicious packets are sent from single or multiple machines respectively, with the aim of exhausting the target's computational and networking resources. The DDoS attacks that shut down some high-profile Web sites (e.g. Yahoo, Amazon) in February 2000 [2], demonstrated their severe consequences and the importance of efficient defense mechanisms. Measurements collected in [3] shows the prevalence of DoS attacks in the

Internet, whereby more than 12,000 attacks against over 5,000 distinct targets were observed in a 3-week data collection period.

In DDoS attacks, the attack packets are often sent with spoofed IP addresses to hide the attackers' identity. Traceback mechanisms [4-12] have been proposed to trace the true source of the attackers to institute accountability. In [13,14], authenticity of IP packet addresses are verified to eliminate spoofing. Rate limiting [15] can be used to decrease malicious traffic as a response technique when the probability of false positive is high. When the data stream is reliably detected as malicious, filtering mechanisms [16,17] can be used to drop the attack traffic. Reconfiguration mechanisms [18,19] change the topology of the victim's network to isolate the attacks or add more network resources. Detailed discussions continue in Section 6.

This paper proposes a novel comprehensive adaptive DDoS mitigation scheme named "Traffic Redirection Attack Protection System" (TRAPS). It consists of traffic congestion and overloading detection, DDoS alleviation by performing good traffic redirection, bad traffic filtering and suspicious traffic adaptive rate-limiting. This scheme does not require prior traffic flow characterizations compared to most existing DDoS defense systems, and allows for a quick real-time response even when attacks constitute flooding of the victim with legitimate service requests. We examined the various approaches of achieving the TRAPS reconfiguration for redirection functionality, and concluded that the end-host approach is comparatively more efficient and requires the least deployment effort. We used Mobile IP (MIP) [20,21] protocol to implement the end-host approach, to avoid the need for new Internet protocol. Although MIP is used, TRAPS is applicable regardless of whether the victim is a wired or wireless node, at home or in a foreign network, and operating in static or mobile mode.

Section 2 of the paper specifies the design objectives and key assumptions. Section 3 describes TRAPS. The experimentation to prove the concepts is presented in Section 4. Section 5 considers the security issues of the protocol and possible attack scenarios, followed by comparisons with existing techniques in Section 6. Conclusions follow in Section 7.

2. DESIGN OBJECTIVES AND KEY ASSUMPTIONS

In this section, we present the design objectives and discuss the key assumptions on which the TRAPS' design is based.

Design Objectives:

i) Should not require any changes to the Internet infrastructure as it would raise conformance issues
ii) Minimal processing and overhead requirements so as not to overload the host or network under attack

iii) Simple and fast algorithms as time and processing power are critical factor and resource during DDoS attacks
iv) Should achieve zero false positive to filter off packets to prevent self-inflicted DoS
v) Should guarantee QoS for high-bandwidth legitimate users
vi) Should guarantee communication of signals required for mitigation purpose to ensure that victim's "call for help" is not overwhelmed.

Key Assumptions:

i) If the packets' contents match an attack signature, it could be easily detected and filtered off by an Intrusion Detection Systems (IDS). Co-operation of IDS with TRAPS would allow faster detection of attacks with known signatures and reduce false positive. In this paper, we only focus on attack traffic with seemingly legitimate packet contents and proceed to differentiate them into the good, bad and suspicious types. Prior knowledge of attack signatures and characterization based on packet contents are thus not required. We define 4 classes of DDoS attacks as follows:

Class A: High-bandwidth traffic with legitimate source addresses
Class B: High-bandwidth traffic with randomly generated spoofed source addresses
Class C: Low-bandwidth traffic with legitimate source addresses
Class D: Low-bandwidth traffic with randomly generated spoofed source addresses

Class A attacks are similar to legitimate user traffic flows and cannot be classified as attacks as they would have the same rights as legitimate users. They are using their own source addresses and transferring legitimate traffic and so will be informed of any host/network reconfiguration by TRAPS. TRAPS should not attempt and would not be able to prevent such attacks. TRAPS assumes these are high-bandwidth users (even if they are zombies), who have negotiated a QoS agreement, and so aims to preserve their QoS. Therefore, this should be handled by mechanisms such as resource allocation at the protected network and we do not consider this form of "attack" here. However, Class A attackers might try to obtain a protected host/network's latest configuration information to support attacks in the other three classes. We discuss this further in Section 5 and propose a solution.

Class B attackers will be sent TRAPS notification of the latest reconfiguration information, but they will not receive them as the addresses are spoofed. Thus, they are not able to send subsequent packets based on the latest protected host/network's configuration, and so the subsequent traffic can be easily identified and filtered off.

Class C and Class D attack traffic are not notified as they constitute a vast distributed set of distinct addresses, so sending individual notifications is not practical. This attack traffic would be treated as suspicious traffic along with any new incoming legitimate requests, and be subjected to lenient treatment (i.e. rate limiting).

ii) We assume that legitimate correspondents are willing to co-operate upon receiving notifications generated by TRAPS. As they would like to have access to the services provided by the protected host/network, they would be motivated to co-operate so they would not block off notifications or refuse to act upon receiving notifications. As such, authentication of notifications becomes an important consideration and we would discuss this further in Section 5.

iii) We assume that the protected network is one under an administrative domain (e.g. enterprise network) and there exists the ability to reconfigure gateways (e.g. for rate limiting) or routers within the network to support TRAPS.

3. DESIGN OF TRAPS

When severe traffic congestion or overloading is detected at the victim, all the gateways and the victim's access router (AR) are informed to drop packets for it, to maintain resource utilization at a "safe" level. The gateways, with a specified probability, discard packets destined for the victim from external sources. The AR ensures that aggregate traffic destined for the victim does not exceed the "safe" level and performs additional rate limiting if required (to take care of possibility of internal attackers, whereby implications will be discussed later). The above-mentioned step will ease the congestion to prevent the victim and network from being overwhelmed by the flood. This is very important during an attack to allow nodes within the protected network to be able to achieve communication for activating TRAPS mitigation support – *satisfies Objective (vi).* At the same time, reconfiguration of the victim/network will be performed to support traffic redirection and the victim will determine recent correspondents with high on-going traffic. TRAPS will inform these correspondents to send future traffic based on the new configuration information. Some notifications may fail due to spoofed addresses. When the acknowledgements are received from the correspondents (after allowing time for retries), the victim informs the gateways and its AR to drop all subsequent packets which do not contain the latest configuration information. Legitimate on-going high bandwidth traffic will have received the redirection information and so will be passed through to reach the victim – *satisfies Objective (v).* Bad on-going high bandwidth traffic using spoofed source addresses, will be filtered off. This traffic detected as attacks, are without doubt from illegitimate users and thus zero false positive is achieved – *satisfies Objective (iv)*.

In DDoS attacks, multiple small-volume bad traffic flows are directed at the victim with randomly spoofed source addresses, and traffic redirection is not feasible. This remaining (Class D) traffic is instead rate-limited (i.e. a more lenient approach) as it might include newly initiated connection requests or small streams of traffic from legitimate sources. In a DDoS attack, a high percentage of the remaining traffic belongs in the category of attacks as compared to the small volume of legitimate traffic and therefore, rate-limiting improves the probability of letting the legitimate requests get through. Next, we propose the various approaches of achieving the TRAPS reconfiguration for traffic redirection.

3.1 En-route Routers Nomination

We propose the network based approach as follows. The victim or a central node nominates routers (e.g. randomly) within the network. These nominated routers are assigned as en-route routers in newly constructed path/s (different set of routers could be nominated for different (set of) correspondents). These new alternative path/s are assigned to the high-bandwidth traffic correspondents, through TRAPS notifications, to allow them to reach the victim. The gateways are then informed of the {correspondent/s' address, victim's address, designated path/s} matching data sets. They will check the incoming packets and if they do not contain any valid designated path information in the packets when checked with the matching data sets, these packets will be dropped. Another set of routers, the Guard Routers, are also randomly chosen within the network and they too, are informed of the matching data sets; Though gateways will be responsible for dropping off attack packets from external attackers, attack packets from internal attackers will bypass gateways and therefore, there exists a need for these Guard Routers to check packets in transit based on the matching data sets and make decisions whether to forward or drop the packets. Guard Routers would filter packets based on both the information in the packets and whether the packets are supposed to visit it. The AR is also informed of the matching data sets to provide a final line of defense. It will perform final checks before forwarding packets to the victim

The disadvantages of this approach are that mechanisms such as source routing have to be used to ensure that the packets follow the designated paths and a new signaling protocol is required to notify the correspondents of the path/s they are assigned. The default route is cut off and the alternate path/s might not be the optimal ones, and high overhead will be incurred as the packets need to encapsulate the en-route routers' addresses. The advantages are that the packets must follow the designated paths or be filtered off and as it would be difficult for the attackers to guess what are the nominated en-route routers (security strength is dependent on the no. of routers selected and no. of address bits (minus away no. of bits for network prefix)), and to derive the exact routers sequence in the alternate path/s. Another advantage would be that the approach could be applied to reduce the load on the gateways by having them perform only random checks and leaving the mandatory verifications to the Guard Routers. Therefore, work distribution across the protected network could be achieved.

3.2 Passcode Approach

Instead of assigning alternate path/s to the correspondents, passcodes could be generated for assignments instead. Packets with matching source address, destination address and valid passcode are allowed to be forwarded. The advantages of this approach are that mechanisms such as source routing is not required (e.g. passcode could be placed in an optional header in the packet), lesser overhead is incurred as passcode is shorter than the entire path information and the default route, which is normally the optimal path, is not “cut off”. A disadvantage is that a new signaling protocol for TRAPS notification is still required as in the En-route Router Nomination approach. Attackers having knowledge of this scheme will have a success rate α $1/2^{n+32}$ or $1/2^{n+128}$ of breaking it, for IPv4 or IPv6 networks respectively (i.e. guessing matching correspondent’s address and passcode of *n* bits for each victim it’s targetting).

3.3 Virtual Relocation Approach

The following describes the Virtual Relocation Approach, which is end-host based. The victim performs a virtual relocation by requesting a new IP address (different addresses could be used for different correspondents or set of correspondents), while still maintaining its old one for use with correspondents not chosen for notifications. It informs high-bandwidth traffic correspondents of the new IP address, and all gateways and a selected set of Guard Routers in the protected network of the {correspondent/s’ address, victim’s new address} matching data sets. The AR is also informed of the matching sets to provide a final line of defense. The required forwarding or dropping of packets are performed by the gateways, Guard Routers and AR, based on the matching data sets.

This approach has the least overhead as it does not require additional data in the packets (i.e. just replace the destination field). The gateways, Guard Routers and victim’s AR will drop packets with source address = notified correspondents and destination address = victim’s old address. The default route need not be “cut off” and multiple paths could still exist between correspondents and victim. The attackers having knowledge of the scheme could guess the new address (having network prefix of *m* bits) and matching correspondent's address with a success rate α $1/2^{64-m}$ for IPv4 and $1/2^{256-m}$ for IPv6 networks.

Although the possibility of success of attackers breaking the scheme is not very high (e.g. $1.39e^{-17}$ and $2.43e^{-63}$ for Virtual Relocation Approach in IPv4 (assuming 8-bit network prefix) or IPv6 networks (having known 48-bit public topology IDs respectively), security strength could be further increased, by performing dynamic reconfigurations more frequently. However, this increases the signaling overhead.

Comparing the methods proposed, we could see that Virtual Relocation has the least overhead (no additional fields in data packets and minimal signaling within protected network). The processing overhead is low due to its simplicity as it does not require a hashing algorithm or network support in alternate path/s construction – *satisfies Objectives (ii) and (iii)*. The most important factor here is that it requires the least deployment effort – *satisfies Objective (i) (although all the methods do not require modifications to the Internet infrastructure)*. The network based approaches require a signaling protocol for communications with the correspondents and customized TRAPS activation software at all potentially legitimate correspondents. However, with the Virtual Relocation Approach, we could make use of MIP, and thus no special software is needed at the correspondents. As long as the correspondents comply to the MIP standards, they have the necessary mechanisms to support communications and react to relocation of their correspondents. This approach could be used even if their correspondents are not actually mobile.

The following sub-sections describe the details on the components of TRAPS, namely high-bandwidth traffic selection, traffic congestion and overloading detection, rate-limiting, and flooding subsidence.

3.4 High-bandwidth Traffic Selection

A Correspondent Database (CD) is maintained by the victim to record information about the traffic it receives and contains the following fields.

- Source address (S_k - unique key field)
- Amount of traffic (e.g. in bytes), M_k, received from this source

where k (from 0 to K-1) is the sequence number of the entries in the CD, and K is the total number of entries in CD. CD is refreshed every T_u secs to keep the data set updated for monitoring the latest on-going traffic of the last T_p secs ($T_p > T_u$).

When congestion or overloading is detected, the victim looks up its CD to select those correspondents with high-bandwidth ongoing traffics, to be notified about the reconfiguration. This also applies to Class B attacks.

In the event of Class C and D attacks, most of the source addresses in the CD will be widely distributed and short-lived (in the case of Class D). With the record interval, T_p, there will be very little recorded traffic for each unique spoofed source address. However, setting an absolute threshold of traffic received for TRAPS activation would require monitoring normal traffic flow and attack traffic to derive how much traffic are considered heavy good traffic or low unique bad traffic with widely distributed range of source addresses. Therefore, we propose (1) as the first condition for choosing the correspondents to perform notifications. In this case, only entries in the CD with traffic equal or greater than the average traffic received will be chosen. The second condition is that the selected traffic must also be high enough (> threshold, M_T) to justify selection for notifications. This is to prevent massive activation in the event that there are many sources of low bandwidth attack traffic while there is no ongoing high-bandwidth legitimate traffic – this is likely for DDoS.

$$M_k \geq \frac{\sum_{k=0}^{K-1} M_k}{K} \tag{1}$$

3.5 Traffic Congestion and Overloading Detection

The traffic and resource monitoring system on the victim detects flooding and severe resource consumption. A simple method is to observe the resource utilization (i.e. bandwidth and computing resources) at the victim and activate TRAPS when a threshold is reached. Another way would be through monitoring gradual depletion of resources at the victim. For example, in traffic monitoring, the aggregate incoming traffic will be observed for checking bandwidth utilization. Traffic growth rate is then computed, so as to detect seemingly abnormal traffic behavior. As for the computing resource monitoring, parameters such as CPU load or memory consumption would be observed and consumption growth rate could then be computed to detect any signs of attack directed at the victim. The following describes the detection method in details.

1) Let x_n (bandwidth or other resources' utilization in percentage) be the alerting points whereby resource consumption growth rate monitoring has to be started, with $n > 0$ and $x_n > x_{n-1} > \ldots > x_2 > x_1$.
2) Let g_n (consumption growth in percentage) correspond to each x_n whereby an alarm has to be triggered and traffic redirection activated. Detection sensitivity has to be increased as the resource utilization gets larger. Therefore, allowable consumption growth rate should be set smaller for increasing monitoring stages.
3) Let t_n be the sampling rate of each stage (in seconds, $n = 0$ for sampling rate before first alerting point and $n > 0$ for sampling rate during alerting

stages). Similar to the consumption growth, the detection sensitivity should be increased as the alerting point is advanced. This could be set through the sampling rate by allowing more frequent sampling at later/crucial monitoring stages.

4) Let *y* be the final alert point or the alarm point, whereby an alarm is immediately generated as soon as the resource utilization reaches or exceeds this point.

3.6 Rate Limiting at Gateways and Victim's AR

After TRAPS is activated, resource consumption at the victim is constantly monitored to adjust the rate-limiting parameters at the gateways and victim's AR in the protected network. An allowable stable resource consumption level, R_c, is configured at the victim. We define the probability of rate-limiting, p, as the probability of dropping the incoming traffic. The initial value of p, p_0, is derived from R_c when alarm is triggered for TRAPS activation. For example, if R_c is 85% of bandwidth and aggregate incoming traffic at the victim is utilizing 95% of it's bandwidth, p_0 will be (95-85)/95, which is approximately 0.105. This value will be sent to the gateways to perform rate limiting for this particular victim (i.e. destination of packets = victim). Resource consumption, which is constantly monitored at the sampling rate, t_n, as in Section 3.4, will be used for adjusting the probability setting. To provide a last line of defense (e.g. in case of internal attackers), victim's AR will be asked to perform further rate-limiting to maintain victim's resource consumption within a "safe" level (e.g. limit victim's aggregate incoming traffic bandwidth at 100kbps).

3.7 Flooding subsidence

To prevent frequent toggling between activation and deactivation of TRAPS resulting in high overhead, three parameters would be used to determine if the DDoS attack has subsided. Therefore, TRAPS will only be deactivated if possible resource consumption without TRAPS is maintained within an acceptable level ($R_a < x_1$, where x_1 is defined in Section 3.4), for at least T_a seconds with a low probability (P_a) of rate limiting at the gateways. Possible resource consumption without TRAPS is measured by totaling resource consumption at the victim, resource conservation due to filtering and rate limiting at the gateways and victim's AR. The choice of the three parameters (R_a, P_a, and T_a) would affect the frequency of toggling as in the following equation.

$$\text{Frequency of toggling } \alpha \ (R_a \text{ x } P_a)/T_a \qquad (2)$$

4. PROOF OF CONCEPT (INCORPORATION WITH MIP)

We used MIP for performing the signaling as it is well-suited for carrying out the required virtual traffic redirection. It is virtual in the sense that traffic is not really redirected to another route but rather to the victim's new address, and the same default or optimal route might still be used. Another reason is that since MIPv4 and MIPv6 are IETF standards, widespread implementations of the protocols are in place (e.g. versions in Windows, Linux, BSD are available). No change will be required in the rest of the Internet infrastructure and the correspondents. In MIP, Home Agents (HAs) are responsible for proxying and intercepting the packets on behalf of Mobile Nodes (MNs, i.e. the victims here), therefore the tasks of filtering and forwarding of the packets destined to MNs can be performed by HAs instead of the gateways. In this case, the gateways are relieved from having to handle all the hosts, which might be activating TRAPS, in the network. In this way, more effective workload distribution and thus higher scalability is achieved.

We developed the TRAPS prototype by implementing the necessary modifications on the MIPL MIPv6 code [22] and additional supporting modules for deployment in a testbed. The systems were running Linux kernel 2.4.22. The supporting modules implemented on the Gateway are the Rate Limiting daemon, which listens for signals from MN and provides rate limiting based on the received parameters, and the Router Bandwidth Monitoring application, which monitors all incoming traffic and records bandwidth utilization for previous interval. The Filtering daemon on the HA listens for signals from MN and filters packets with old correspondent-victim address pair. MN runs the Host Bandwidth Monitoring and TRAPS activation application, which monitors all incoming traffic, computes the bandwidth utilization, monitors the alert stages, sends TRAPS activation signal to the MIP code to trigger TRAPS, notifies gateway regarding rate limiting activation and parameter updates, and notifies HA of filtering updates, and the Test Server, which listens for data transfer from CN before, during and after TRAPS activation to test that there's no cutting off of messages. The Attack module on the Attacker system is an UDP packet generator with adjustable attack rate and configurable spoofed address. The Test Client on CN sends continuous data to MN before, during and after TRAPS activation to test that there's no cutting off of messages

Experiments were performed by setting 3 stages of resource monitoring (2 alert stages at 50 and 60kbps respectively, and the alarm stage at 80kbps) at MN. Test Server module at MN and Test Client module at CN were started to continuously carry out data transfer. The Attacker's spoofed address was set to be CN's IP address. When the attack traffic was gradually increased through each stage corresponding to those set at MN, the alert events and finally the alarm event were triggered. MN then sent rate limiting signal to the gateway and BU to the CN regarding its new IP address. The gateway started rate limiting traffic destined to MN. When CN received MN's BU, it sent a BAck to MN. After that, MN sent the filtering signal to HA to activate filtering on the CN's address, MN's HoA pair. After which, the attack traffic from the Attacker was intercepted by HA and filtered off. On the other hand, the data transfer between the Test Server and Test Client was able to continue.

5. DISCUSSIONS

Security Considerations of Protocol

Traffic redirections as used in TRAPS can pose a major security problem in the Internet if the protocol messages are not properly authenticated. Therefore, we will now consider the MIP related security issues, which are of concern to TRAPS.

In MIPv4, it is specified that each MN, FA, and HA must be able to support a mobility security association for mobile entities, indexed by their security parameter index (SPI) and IP address. Registration messages between MN and its HA must also be authenticated with an authorization-enabling extension. This prevents a malicious node from impersonating MN to redirect away its traffic or HA to intercept MNs' packets.

The MIPv4 Route Optimization Authentication extension [23] is used to authenticate the protocol messages with an SPI corresponding to the source IP address of the message and it must be used in any binding update message sent by the HA or MN to the CNs. The calculation of the authentication data is specified to be the same as in the base MIPv4. This is HMAC-MD5 [24]. A security association must be present between CN, which could be any node in the Internet, and MN/HA. It is suggested in [20] that the mobility security association at a CN could be used for all MNs served by a particular HA. The effort of establishing such an association with a relevant HA is more easily justified than the effort of doing so with each MN.

In MIPv6, binding updates are protected by the use of IPSec extension headers [25] or the Binding Authorization Data option, which employs a binding management key established through the return routability procedure [21]. It is specified that MN and HA must use an IPSec security association to protect the integrity and authenticity of the binding management messages.

The protection of binding updates to CNs does not require the configuration of security associations or the existence of an authentication infrastructure between the MN and CNs. The return routability procedure is used to prove the authenticity of the MN by testing whether packets addressed to the two claimed addresses (i.e. HoA and CoA) are routed to the

MN. MN can only pass the test if it is able to supply proof that it received the keygen tokens which CN sends to those addresses. The return routability procedure also protects CN against memory exhaustion DoS attacks as CN does not need to retain any state about individual MNs until an authentic binding update arrives.

If the gateways are not implemented with the HA functionalities to perform filtering, security associations must be set up between the MN and the gateways, which are responsible for rate-limiting. Finally, it is important to note that TRAPS presents no additional security vulnerability to the MIP protocols.

Random Hit

We mentioned that Class B attacks are singled out by TRAPS for notification of the latest reconfiguration information. As they could not be "reached", they could be easily identified as attack traffic flows and would then be filtered off. However, what if there happens to be a random hit (e.g. randomly generated spoofed addresses by attackers within an address range resulting in an address belonging to one of the attackers)? In this case, that particular attacker would be notified of the latest information and continue attack on the victim using randomly spoofed addresses. However, in this second round of attack, the traffic volume will be lower and distributed across the spoofed address range (and will be rate-limited instead), since the other attackers were "stopped" in the first round. A solution to strengthen the scheme and lower the chances of this happening (recommended in Section 3), is by performing regular dynamic reconfigurations and updates.

Spying by Class A attackers

It was mentioned in Section 2 that there's a possibility that Class A attackers might be used as spies to obtain protected host/network's latest configuration information to support attacks in the other 3 classes. However, even with this information, the other forms of attacks would not be successful as prevention from filtering not only acts on knowledge of this information but also matching correspondent's address. In any case, a solution could be in place to catch the spy. The victim could have multiple sets of configuration information (e.g. multiple addresses in the Virtual Relocation Approach) and provide each set of correspondents with different configuration information. If exploitation of a particular set of configuration information is detected, we would know that a spy is within this set of correspondents. We could narrow down to the exact correspondent by performing iterations of this procedure.

6. COMPARISONS WITH RELATED WORK

Traceback mechanisms [4-12] have been proposed to trace the true source of the DDoS attackers, as attack packets are often sent with spoofed IP addresses. In traceback, the attack path or graph is constructed to provide information on the route/s the attack packets have taken to arrive at the victim. It is an attacker identification tool which requires further deployment of a detection and mitigation tool to counter DDoS attacks.

Pushback [15] is a rate limiting mechanism which imposes a rate limit on data streams characterized as "malicious". It involves a local mechanism for detecting and controlling high bandwidth aggregate traffic at a single router by rate limiting the incoming traffic, and a co-operative pushback mechanism in which the router can ask upstream routers to control the aggregate. However, all high bandwidth traffic, whether good or bad, will be subjected to this rate limiting. Filtering mechanisms [16,17] on the other hand, filter out attack stream completely. This is used when the data stream is reliably detected as malicious; else, it may run the risk of accidentally denying service to legitimate traffic.

Mechanisms such as traceback, rate limiting, and filtering need to be triggered by a third-party detection tool. The way the detection tools detect an attack is therefore very important to determine how reliable it is and which of the above-mentioned mechanisms is to be used. Detections are classified in two main categories, which are "Anomaly Detection" and "Misuse Detection" [26]. Anomaly detection techniques assume that a "normal activity profile" could be established for a system. Activities not matching the profile would be considered as intrusions. However, an action which is not intrusive but not recorded formerly in the profile would then be treated as an attack, resulting in false positive. Filtering would then result in DoS by the defense system itself. In situations whereby intrusive activities, which are not anomalous, occur, it would result in attacks not detected and therefore false negatives. Such scenarios are possible if DDoS attacks are launched by flooding the victim with legitimate service requests. In misuse detection schemes, the attacks are represented in the form of a pattern or signature so that even variations of the same attack can be detected. However, they can only detect known attacks. For new attacks whereby the characteristics of the attack packets and pattern are unknown, they would of little use. They are also unable to detect attacks that are launched by flooding of legitimate packets. The advantage of TRAPS over these mechanisms is that it does not require prior traffic characterizations.

A preventive measure to DDoS attacks is to ensure the authenticity of packets by eliminating source address spoofing. Ingress filtering [13] filters packets with spoofed source addresses at the first router encountered on entering the Internet. This router typically has information about valid source addresses that are allowed to pass through it. However, enforcement on supporting ingress filtering on all outbound routers to the Internet is difficult. Source Address Validity Enforcement (SAVE) [14] messages propagate valid source address information from the source to all destinations, for en-route routers to build an incoming table that associates each incoming interface of the router with a set of valid source address blocks. Packets with invalid source addresses are identified as attack packets. Widespread deployment is required for this scheme to be effective.

Reconfiguration mechanisms change the topology of the victim or the intermediate network to add resources or isolate attack machines. The Secure Overlay Services (SOS) [18] architecture is constructed using a combination of secure overlay tunneling, routing via consistent hashing, and filtering. The overlay network's entry points perform authentication verification and allow only legitimate traffic. The route taken by the traffic is computed to be designated beacons and then servlets, both of which are kept secret from the correspondents. Potential targets are protected by filtering which only allow traffic forwarded by the chosen secret servlets. Randomness and anonymity is in this way introduced into the architecture, making it difficult for an attacker to target nodes along the path to a specific SOS-protected destination. The XenoService [19] is a distributed network of web hosts that respond to an attack on a web site by replicating it rapidly and widely. It can then quickly acquire more network connectivity to absorb a packet flood and continue providing services.

TRAPS belongs to the category of reconfiguration mechanisms by changing the routes to the victim under attack. However, unlike SOS, an overlay network and complex algorithms (e.g. Chord routing algorithm, consistent hashing) need not be implemented. In SOS, only certain destinations are chosen for protection. These destinations are protected by filtering to only allow traffic forwarded by selected servlets. However, beacons and servlets could be subjected to attacks instead. It is recommended in [18] to have a large number of beacons and servlets to provide redundancy. Nodes overwhelmed by the attacks would then be "removed" and their jobs will be handled by the remaining active ones. In TRAPS, any node running the MN module would be able to bring itself under protection in the event of attacks. Redundancy by providing additional resources is also not required in TRAPS, unlike XenoService.

7. CONCLUSIONS

This paper proposes TRAPS, an adaptive real-time DDoS mitigation scheme. In TRAPS, the victim under attack verifies the authenticity of the source by performing adaptive reconfigurations, either host or network based, and requesting senders of high-bandwidth traffic streams to send subsequent data based on the victim's latest configuration. If the source is illegitimate, it would not be updated with this information. This traffic can be easily identified as attacks, with absolute confidence and be dropped. Suspicious traffic for the victim will be rate limited as most good traffic will have been redirected, leaving mainly attack packets with randomly generated IP addresses.

The basic mechanisms of TRAPS, and various approaches (i.e. En-route Routers Nomination, Passcode and Virtual Relocation) of achieving the TRAPS reconfiguration for redirection were explained in detail. We discussed and evaluated the various approaches, and concluded that the end-host based approach, Virtual Relocation, is comparatively more efficient (e.g. requires least processing at gateways/routers/victim and overhead), and requires the least deployment effort among the proposed approaches. We suggested incorporating this approach with the MIP protocol to avoid proposing new protocols for Internet-wide deployment. Implementation of TRAPS was carried out and deployed in a testbed environment. It was observed that the operations of each module were functioning correctly and TRAPS was able to successfully mitigate an attack launched with spoofed source IP address. The security considerations with regards to MIP are discussed and we showed that TRAPS does not introduce any additional security vulnerability. Other possible scenarios of random hit and spying were also discussed with possible solutions proposed.

Related work on the existing DDoS detection, tracking and mitigation techniques is presented. Comparison of some of their important features with TRAPS is carried out. Advantages of TRAPS over existing DDoS mechanisms are: it does not require prior traffic flow characterizations and allows for a quick real-time response even in the event whereby DDoS attacks constitute brute-force flooding of victim with legitimate service requests; no need for additional resource allocation for providing redundancy; QoS is maintained for good high bandwidth traffic; very suitable for both high-end powerful systems and embedded systems as it is simple to implement and does not require sophisticated algorithms.

8. ACKNOWLEDGEMENTS

We gratefully acknowledge the support from the Institute for Infocomm Research and the EU funded Diadem Distributed Firewall FP6 IST-2002-002154. We would also like to thank Dr. Robert Deng for the valuable suggestions on the paper.

REFERENCES

1. K. J. Houle, G. M. Weaver, "Trends in Denial of Service Attack Technology", CERT Coordination Center, Oct. 2001
2. L. Garber, "Denial-of-Service attacks rip the Internet", IEEE Computer, Vol. 33, No. 4, pp. 12-17, Apr. 2000
3. David Moore, Geoffrey M. Voelker, Stefan Savage, "Inferring Internet Denial-of-Service Activity", Usenix Security Symposium, Aug. 2001
4. Alex C. Snoeren et al, "Hash-Based IP Traceback", ACM Sigcomm 2001, Aug. 2001
5. Stefan Savage et al, "Practical network support for IP traceback", ACM Sigcomm 2000
6. Dawn Song, Adrian Perrig, "Advanced and authenticated marking scheme for IP traceback", IEEE Infocom 2001
7. K. Park, H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack", IEEE Infocom 2001
8. Steve Bellovin et al, "ICMP Traceback Messages", IETF Internet Draft, Version 4, Feb. 2003 (Work in progress)
9. Allison Mankin et al, "On Design and Evaluation of "Intention-Driven" ICMP Traceback", IEEE International Conference on Computer Communication and Networks, Oct. 2001
10. Henry C. J. Lee, Vrizlynn L. L. Thing, Yi Xu, Miao Ma, "ICMP Traceback with Cumulative Path, An Efficient Solution for IP Traceback, International Conference on Information and Communications Security, Oct. 2003
11. Abraham Yaar, Adrian Perrig, Dawn Song, "Pi: A Path Identification Mechanism to Defend against DDoS Attacks", IEEE Symposium on Security and Privacy, May 2003
12. D. Dean, M. Franklin, A. Stubblefield, "An algebraic approach to IP Traceback", Network and Distributed System Security Symposium, Feb. 2001
13. P. Ferguson, D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000
14. Jun Li et al, "SAVE: Source address validity enforcement protocol", IEEE Infocom 2002
15. Ratul Mahajan et al, "Controlling High Bandwidth Aggregates in the Network", ACM Sigcomm 2002
16. T. Darmohray, R. Oliver, "Hot spares for DDoS attacks", http://www.usenix.org/publications/login/2000-7/apropos.html
17. Mazu Enforcer, http://www.mazunetworks.com
18. A. D. Keromytis, V. Misra, D. Rubenstein, "SOS: Secure Overlay Services", ACM Sigcomm 2002
19. J. Yan, S. Early, R. Anderson, "The XenoService - A Distributed Defeat for Distributed Denial of Service", Information Survivability Workshop 2000, Oct. 2000
20. C. Perkins, "IP Mobility Support for IPv4", IETF RFC 3344, Aug. 2002
21. D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", IETF RFC 3775, June 2004
22. MIPL Mobile IPv6 for Linux, http://www.mipl.mediapoli.com
23. C. Perkins, D. B. Johnson, "Route Optimization in Mobile IP", IETF Internet Draft, Version 9, Feb. 2000 (Work in progress)
24. H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, Feb. 1997
25. S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, Nov. 1998
26. Aurobindo Sundaram, "An Introduction to Intrusion Detection", ACM Crossroads, Vol. 2, Issue 4, pp. 3-7, Apr. 1996