Communications and Network, 2015, 7, 106-116
Published Online May 2015 in SciRes. http://www.scirp.org/journal/cn
http://dx.doi.org/10.4236/cn.2015.72010

A Defense Framework against DDoS in a Multipath Network Environment

Ahmed Redha Mahlous

Department of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia
Email: armahlous@psu.edu.sa

Received 20 February 2015; accepted 3 April 2015; published 9 April 2015

Abstract

The Internet is facing a major threat, consisting of a disruption to services caused by distributed denial-of-service (DDoS) attacks. This kind of attacks continues to evolve over the past two decades and they are well known to significantly affect companies and businesses. DDoS is a popular choice among attackers community. Such attack can easily exhaust the computing and communication resources of its victim within a short period of time. Many approaches to countering DDoS attacks have been proposed, but few have addressed the use of multipath. In this paper, we analyze, how multipath routing based solutions could be used to address the DDoS problem. The proposed framework traces back the attack to its source and blocks it. It also calculates multiple paths to the attacker (if they exist) and alerts all gateways near the attacker to block possible traffic originating from this source in case another path(s) is (are) later used to attack the victim again. We demonstrate that our scheme performs better that other single path schemes.

Keywords

DDoS, Multipath, Filtering, Traceback

1. Introduction

A distributed denial-of-service (DDoS) attack is a serious threat to the security of the Internet. It consists of exhausting the bandwidth, processing capacity or memory of a targeted machine or network. It is a distributed, cooperative and large-scale attack. It has been widespread on wired [1] and wireless networks [2].

Attackers usually hide their identity and DDoS is an example of anonymous attacks where currently there is no obvious way to prevent or trace them. Practically it is impossible to prevent all DDoS attack(s) that originate from Internet; however we can at least find a mechanism to identify the source(s) of the attack in situation where prevention fails. In this context security community proposes IP traceback techniques.

How to cite this paper: Mahlous, A.R. (2015) A Defense Framework against DDoS in a Multipath Network Environment. *Communications and Network*, **7**, 106-116. http://dx.doi.org/10.4236/cn.2015.72010

In this paper we present a filtering framework mechanism for a DDoS attack, which is similar to that one used in [3]. However, our new framework computes multiple paths, if they exist, to the attack source(s), then it generates an alert to the nearest gateway(s) of those paths, to block all traffic originating from the attack source(s), thing that has not been investigated in [3]. In this way our proposed framework prevents the attacker from using other available paths to attack the same destination.

Our source address spoofing solution is a hardware-friendly variant of the IP route record (RR) technique. It uses recording packet route information to block attack traffic from different paths. The marking techniques used are different from traditional ones [4] [5] (which do not provide an explicit recorded route in each packet). The aim of our multipath traffic filtering (MTF) system is to use the recorded route on each incoming packet to block that traffic at the last point of trust on each attack path, which means blocking untrusted traffic at the nearest possible point from the attack source(s).

The remainder of the paper is organized as follows: Section 2 presents related work, while Section 3 describes the MTF components. Section 4 describes the MTF protocol in detail. MTF simulation and performance are evaluated in Section 5. And Section 6 concludes the paper.

2. Related Works

Many techniques against DDOS have been studied in recent years, which lead to the proposal of a defense mechanism to be implemented on routers [6]-[10], servers [11] [12], or both servers and clients [13]-[16].

Authors in [17] proposed a targeted filtering mechanism that blocks stateful DDoS attack called targeted filtering. Filters are established at a firewall and automatically converges the filters to the flooding sources [18]; authors studied the impact of application layer Denial-of-Service (DoS) attacks that targets web services. They proposed an adaptive system that detect against application layer attacks such as XML and HTTP. Furthermore, authors state that the system is also capable of detecting malicious request, spoofing and regular flooding attacks. It is intended to be deployed in a cloud environment where it can transparently protect the cloud broker and cloud providers.

A Bandwidth DDoS (BW-DDoS) attack was the study of [19]. Authors pointed that, this type of attack can disrupt network infrastructure operation by causing congestion, this is due to the increase of total amount traffic inducing connectivity degradation or more than that, loss of connection between the Internet and victim networks or even whole autonomous systems (ASs). Authors presented some type of attack agent, attack mechanism, protocol manipulation, and attack target and response mechanism.

A New approach to IP traceback named Deterministic Flow Marking (DFM) was presented in [20]. DFM allows the victim to traceback the origin of spoofed source addresses up to the attacker node even if this later is hidden behind a NAT or a proxy server. It has the capability to trace simultaneous distributed attacks in near real time while providing authentication according to authors.

Authors in [21] argued that the existing application layer methodologies couldn't detect DDoS attack. They proposed a set of algorithms that are capable of detecting and blocking DDoS attacks whilst allowing through legitimate user traffic, including flash traffic. Discriminating the attack flows from the flash crowds was also the study of [22] and [23].

To the best of our knowledge all traceback techniques that were studied confines to only one attack path. In this paper we propose a novel multipath traceback technique to defend against DDoS attack.

3. MTF Components

3.1. Path Recording

MTF appends the node's address to the end of the packet as it travels through the network from the attacker to victim. Consequently, every packet received by the victim arrives with a complete ordered list of the routers that it traverses providing a built-in attack path. So a path fingerprint is embedded in each packet, enabling a victim to identify packets traversing the same paths through the Internet on a per packet basis, regardless of source IP address spoofing. In this way the victim needs only classify a single packet as malicious to be able to trigger a filtering action that will be applied to subsequent packets with the same marking (all packets traversing the same path carry the same marking). This approach is very lightweight, both on the routers for marking, and on the victims for decoding and filtering.

We assume that each router that participates in our scheme has a capability to add its IP address to the packets that traverse it and is MTF capable. We also assume that only the border router participates in this scheme and not the core router connected to the Internet backbones. As a result, each packet carries the identities of a sub-list of the border routers that forwarded it.

To avoid the performance overhead of a traditional IP route record, the path header (PH) is introduced at the beginning of the IP payload and it works as a "shim" protocol between the IP and transport layers.

The PH used follows the recommendations stated in [24] and is composed of three fields: the path is a list of (initially empty) slots; the size is the total number of slots in the path; the pointer points to the first empty slot in the path. The size of the number of slots represents the maximum distance that a packet can reach.

As the length of route cannot be predicted and each hop increases the packet header size (that grows linearly with hop count) which can lead to fragmentation due to the limit of the maximum transmission unit (MTU), we set the maximum hop to 15 as used by the well-known RIP network protocol [25]. If a packet reaches a router beyond this number it is dropped (the router does not have an empty slot to write its IP address).

When a packet traverses an MTF-enabled router it does the following: (1) inserts a PH header in the packet; (2) writes its own IP address and random value in the first slot; and (3) sets the pointer to point to the second slot. Each subsequent MTF-enabled border router that ingresses the packet into a new AS: (1) reads the pointer; (2) writes its own IP address and random value in the indicated slot; and (3) increments the pointer to point to the next slot. If no room is left in the PH header (after 16 hops), the router drops the packet.

3.2. Multi-Path Calculation

The victim's gateway will calculate multiple shortest paths to the source of attack (if they exist) and locate the nearest border gateway(s) to the attacker. To calculate the multiple shortest paths (if they exist), we use a modified variant of Dijkstra's algorithm [26] [27], as it is the most widely used algorithm in present day routers. By choosing the shortest paths we aim to send a warning message (blocking request) as fast as possible.

3.3. Flow Classification

When a packet crosses an MTF-enabled router, its recorded route includes an authentic (non-spoofed) suffix. Specifically, the last "n" components of the recorded route are authentic, when the last n border routers crossed by the packet are MTF-enabled, and there is no malicious node on the path that interconnects them using malicious node detection scheme [28].

The recorded path of a packet is defined as a sequence of IP addresses that corresponds to: the packet's source, the list of border routers included in the PH header, and the packet's destination. Packets are marked as belonging to the same flow if they share a common recorded path suffix. This procedure enables a receiver to identify distinct incoming traffic flows in the face of source address spoofing.

A flow F with recorded path P is denoted as FP. A DDoS node victim uses a filtering policy to classify incoming traffic in different flows, and then decides which one to apply a filtering procedure to and either reject it or accept it. The operation of the policy module depends on the specific service run by the victim and is outside the scope of this paper. The blocking of any undesired flow that may come from different paths and enforcing such a policy is achieved by deploying our proposed mechanism (MTF).

It is up to the policy module to classify incoming traffic in multiple flow levels. For example, consider that in **Figure 1** an attack source, PC 1, is sending high-rate traffic to the victim, SERVER. If network Net1 prevents source address spoofing, SERVER can easily identify F1 [PC1, R1 R3, SERVER] as a high-rate flow, and thus undesired. If PC1 is able to spoof multiple source IP addresses, SERVER can only identify F2 [*, R1, R3, SERVER] as the undesired flow.

Once the policy module identifies an undesired flow, it sends a filtering request to the local MTF process.

4. Basic MTF Protocol

4.1. Terminology

- The recorded path P of an undesired flow (**Figure 2**) has the form [PC1, R1, …R3, SERVER], where PC1 is the attack source, *i.e.* the node thought to be generating the undesired traffic; if PC1 = *, all traffic through R1 is undesired.

- R1 is the attack gateway, *i.e.* the border router thought to be closest to PC1.
- R3 is the victim's gateway, *i.e.* the border router closest to the victim.
- SERVER is the victim.

We assume that the only node affected by the attack is SERVER; e.g. if this is a flooding attack, the only part of the network that is congested is the tail-circuit from R3 to SERVER. If R3 were also affected, it itself would be the victim, and its closest upstream border router would be the victim's gateway. As the majority of DDoS attacks are performed using TCP as summarized in **Table 1** [29], we assume that only TCP packets are used in our model.

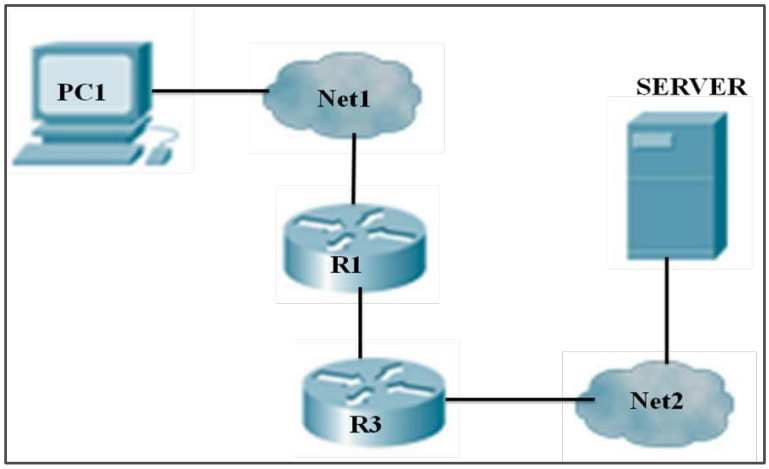

Figure 1. MTF network.

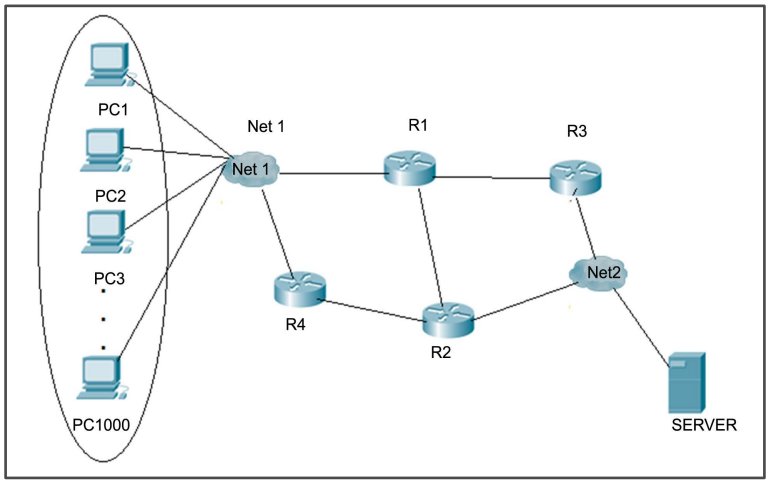

Figure 2. DDoS network attack scenario.

Table 1. Distribution of DDoS attack types.

Protocols	Ratios
TCP	90% - 94%
UDP	2.4% - 5%
ICMP	2.1% - 2.6%
Other	2.06% - 2.93%

4.2. Blocking Close to the Attack Source

MTF involves 4 units:

1) When an undesired flow(s) arrives at the victim SERVER, it sends a filtering request to R3, specifying an undesired flow F.

2) The victim's gateway R3:

(a) Installs a temporary filter to block F for Ttmp seconds.

(b) Initiates a 3-way handshake with R1 (**Figure 3**).

(c) Removes its temporary filter, upon completion of the handshake.

3) The attack gateway R1:

(a) Responds to the 3-way handshake.

(b) Installs a temporary filter to block F for Ttmp seconds, upon completion of the handshake.

(c) Sends a filtering request to the attackers' source (PC1…PC1000), to stop F for Tlong >> Ttmp minutes.

(d) Removes its temporary filters if the attackers comply within Ttmp seconds; otherwise, it disconnects the attackers.

4) the attack source stops F for Tlong minutes or risks disconnection.

When R1 blocks traffic it sends a positive message that the traffic has been blocked. During this time, R3 calculates whether there are other paths to the attacker. If yes, it sends a message to the gateways(s) closest to the attacker (R4 in **Figure 2**) warning it of a possible attack from a source attack. The gateway(s) starts monitoring traffic and blocking any packets coming from the warned sources. R4 will install a filter to block any future traffic from the attack sources for time T_{block}. T_{block} will have the same value as T_{temp}.

If the victim's warning message arrives after the attacker has used another path, some unwanted traffic might arrive at the victim, but progressively they are blocked after the victim carries out the same process described previously to block the attack.

If the warning message arrives before the attackers use another path(s), the victim, in this manner, has blocked probable attacks. Thus the attackers will have less chance to attack SERVER using other paths after being blocked.

When SERVER receives the attack, it sends a message to R1 with a suggested rate limit value. Based on the requirements in the message, R1's system will decrease the rate limit value exponentially. After the traffic at the victim's end has returned to normal for a period, an update message is sent to the source end asking it to increase the rate limit value linearly.

If the defense system has not found any anomalous changes at the victim's end since the update message, a cancel message to remove the rate limit at the source end is sent.

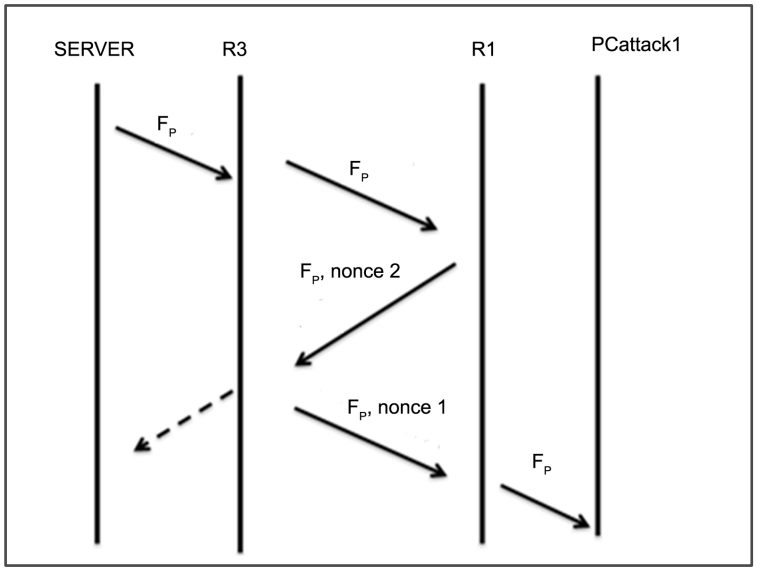

Figure 3. Three-way handshake.

However, if the volume of attack traffic still increases aggressively, an update message will be sent again to block the traffic coming from the attacker.

4.3. Preventing Man-in-the-Middle Attack

In order to secure communication between nodes and trigger a filter in the gateway near the source of the attack we use a 3-way handshake as shown in **Figure 3**.

The 3-way handshake is depicted in **Figure 3**: in which R3 sends a request to R1 to block F; R1 sends SERVER a message that includes F and a nonce; R3 intercepts the message and sends it back to R1.

R3 proves its location on the path to SERVER by intercepting the nonce sent to SERVER. This prevents malicious node H, located off the path from R1 to R3 (**Figure 4**), from causing a filter to be installed at R1 and block traffic to SERVER. By picking a sufficiently large and properly random value for the nonce, it can be made arbitrarily difficult for H to guess it. To defend against forced delay attack, the nonces are tied in with short response time outs.

To avoid a buffering state on incomplete 3-way handshakes, R1computes the nonce as follows:

$$\text{Nonce1} = \text{Hk}(\text{F}) \tag{1}$$

where Hk is a keyed hash function.

To verify the authenticity of a completion message, R1 just hashes the flow included in the message and compares the result to the nonce included in the message.

For example, in **Figure 4**, traffic sent by Net1 to Net2 has recorded path [R1:RND1, R3:RND2, Net2], where RND1 and RND2 are random values inserted by R1 and R3 respectively. To avoid keeping per destination states, each router computes the random value to insert in each packet as follows:

$$\text{R} = \text{Hk}(\text{D}) \tag{2}$$

where D is the packet's destination.

4.4. Filter Timeout Values

The goal of a temporary filter on the victim's gateway is to block an undesired flow until the corresponding handshake is complete. Considering that Internet roundtrip times range from 50 to 200 msec [30], we take the approximate average of T_{tmp} = 1 sec as a reasonable choice since it offers a safe value for a round trip time.

The choice of the long-term filter timeout T_{long} involves the following trade-off: An attack source PC1 is typically an innocent end-host compromised through a worm. A large T_{long} of, say, 30 minutes guarantees that the victim SERVER will not receive any undesired traffic from the corresponding attack PC1 for at least 30

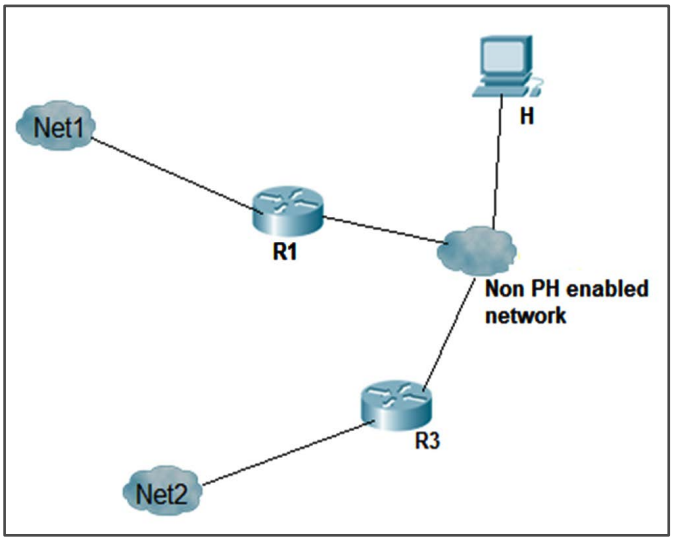

Figure 4. Man in the middle.

minutes; on the other hand it also guarantees that SERVER will not receive any traffic at all from PC1, even if PC1 is appropriately patched before T_{long} expires.

5. Simulation and Performance Results

5.1. Filtering Gain

In this section we use a NS-2 simulator [3] to conduct a set of simulations to evaluate the performance of the proposed MTF framework in the presence of DDoS attacks with different scenarios.

Figure 5 shows the experimental topology. Trusted users (PC 1 to PC 100) generate a TCP based FTP flow with a packet size of 1000 bytes. While DDoS traffic is generated by PCattack (PCattack 1 to PCattack 1000) by sending packets of 50 bytes in size. The victim's gateway uses up to 10,000 filters to protect the victim. For each scenario we plot the bandwidth of the attack traffic that reaches the victim as well as the victim's goodput as a function of time, *i.e.* we show how fast attack traffic is blocked and how much of the victim's goodput is restored. In all scenarios T_{temp} = 1 sec and T_{long} = 2 min. The choice for such value (T_{long} = 2 min) is to reduce the event handling overhead for the simulation run.

5.1.1. Scenario 1: Attackers Use One Path Only

The victim receives a flooding attack from 1000 attack sources. The bandwidth of the attack (before defense) is 1 Gbps. From **Figure 6** we can see that at time t = 0 - 1 sec, before the attack, SERVER is receiving 80 Mbps of goodput, then when the attack starts from path 1 [R1-R3-SERVER] (**Figure 5**) the goodput is reduced to 12%

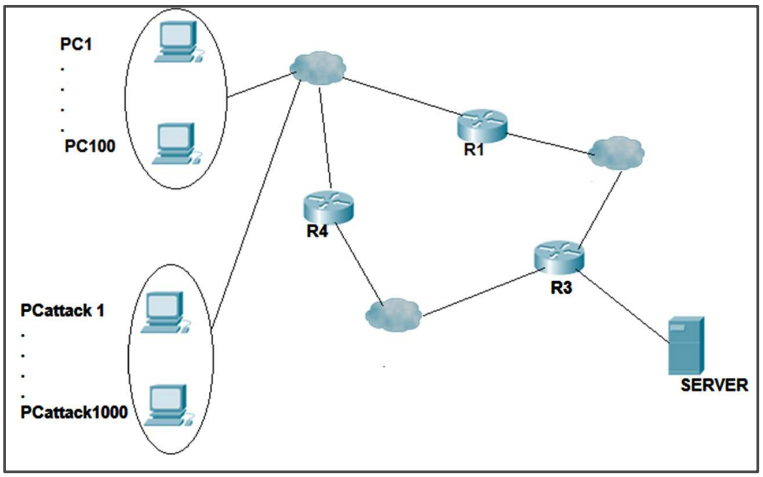

Figure 5. Experimental topology.

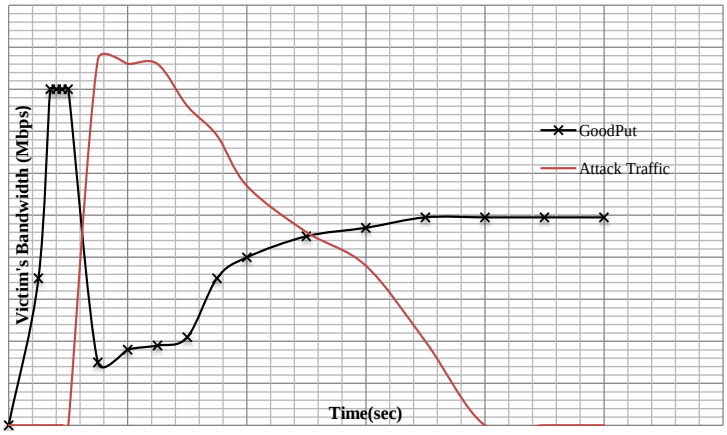

Figure 6. One path attack.

from its original value. At t = 2 sec, SERVER starts sending requests to the gateway to block the attack (it sends 100 filter requests); t = 2 - 8 half of goodput is restored progressively until all the attack traffic is blocked at t = 8.

5.1.2. Scenario 2: Attackers Use Multiple Path

In this scenario we assume that the victim receives a flooding attack from 1000 attack sources from another path [R4-R3-SERVER] (**Figure 5**). We can see clearly from **Figure 7** that at t = 0 - 1 SERVER receives normal traffic using 80 Mbps of goodput; at t = 1 - 2 the attack drives SERVER goodput to ~50% of its original; at t = 2 sec SERVER starts sending 100 filtering request/sec to the gateway, when this later blocks unwanted traffic, the attacker at t = 2 - 3 uses another path to flood the victim with unwanted traffic. The goodput decreases to 60% of its original, then at t = 3 sec SERVER's goodput is restored to ~78% of the original and the attacker is blocked.

R4 blocks 10 flows in 10 seconds and 1000 flows in 100 seconds. Without MTF, a router needs one thousand filters to block one thousands flows; these experiments demonstrate that, with MTF, R4 needs only one hundred filters to block one thousand flows.

In this experiment we demonstrated that MTF achieves a filtering response time equal to the one-way delay from the victim to its gateway.

Hence, MTF reduces the number of filters required to block a certain number of flows by two orders of magnitude. A critical improvement, since routers typically accommodate tens of thousands of filters, whereas DDoS attacks can easily consist of millions of flows.

5.2. Attack Path Reconstruction and False Positive Ratio

An important performance indicator for IP traceback is the number of packets that need to be collected for reconstructing an attack path. **Figure 9** shows the number of packets required to reconstruct attack paths. To better evaluate MTF's effectiveness; we repeat the same experiments with FMS [31] and AMS [32].

The result in **Figure 8** shows that the number of packets needed by MTF is the smallest among all PPM schemes. This is expected since MTF only needs to collect one marked packet along an attack path, whereas other schemes need multiple marked packets.

We are also concerned with the relationship between the false positive ratio and the number of attackers. **Figure 9** shows the false positive ratios versus the number of attackers.

The false positive ratio is the number of legitimate clients that are mistakenly recognized as attackers by the MTF divided by the number of legitimate clients. In each simulation we fix the total number of clients to 5000 and vary the number of randomly chosen attackers from 1 to 4101 in increments of 10. The results show that MTF incurs relatively high false positive ratios at the beginning of attack before gateways are involved in decreasing their rate traffic. The reason is that as soon as MTF detects an attack it will block all traffic from the attacker paths. This is because attacker and innocent packets reside behind the same gateway, and all packets coming from the latter are considered unwanted as soon as an attack occurs and the victim triggers the filtering process.

This cannot make things any worse, since most good traffic is being dropped anyway due to the flood; in

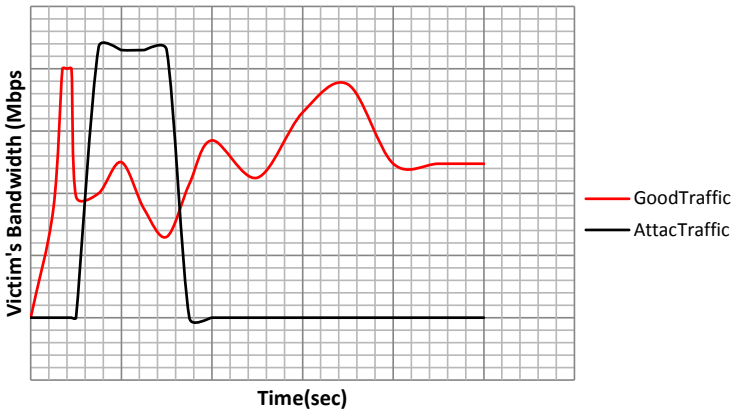

Figure 7. Multiple path attack.

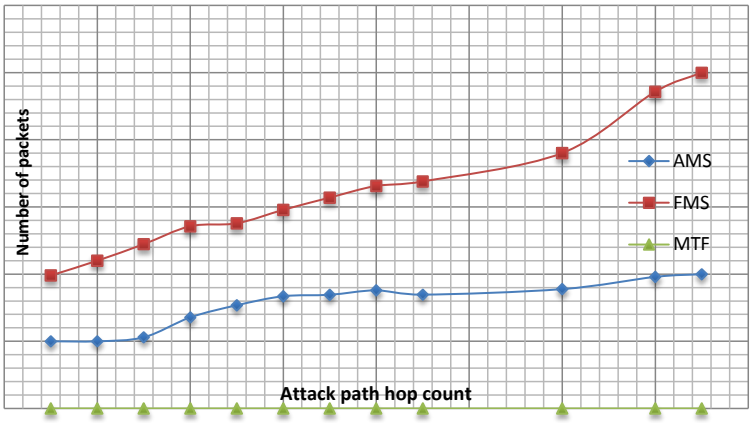

Figure 8. Numbers of packets.

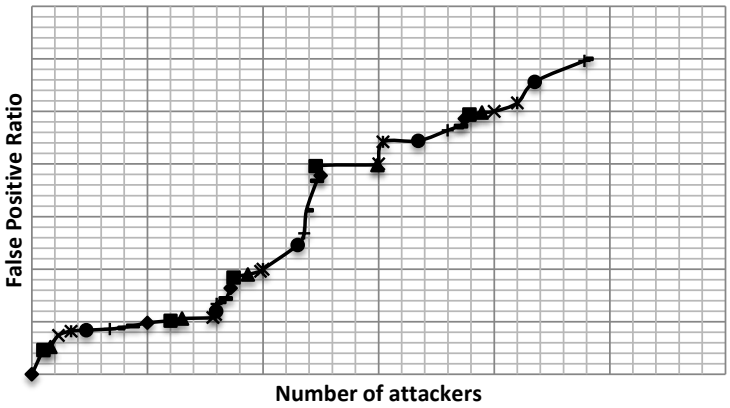

Figure 9. False positive ratio.

contrast it allows good traffic from other gateways to get through. Once the attacks are filtered, we notice that false positive ratio gradually decreases.

6. Conclusions

We presented a Multipath Traffic Filter (MTF), a mechanism for filtering highly distributed denial-of-service attacks. We showed that MTF can block a thousand undesired flows, while only requiring a hundred filters.

It also prevents abuse by malicious nodes seeking to disrupt the communications of other nodes. We showed that MTF has the capability of blocking unwanted flows from the same source from different paths, even before they can occur.

Our plans for future work include the development of a congestion control algorithm that will detect and control high bandwidth flow. Furthermore, we will look at mechanism for reducing computation and storage overhead induced by multipath calculation on control and data plane. Finally, we are interested in deploying such techniques on a larger scale network.

References

[1] Chonka, A., Xiang, Y., Zhou, W.L. and Bonti, A. (2010) Cloud Security Defense to Protect Cloud Computing against HTTP-DoS and XML-DoS Attacks. *Journal of Network and Computer Applications*, **34**, 1097-1107.

[2] Arunmozhi1, S.A. and Venkataramani, Y. (2011) DDos Attack and Defense Scheme in Wireless Ad Hoc Networks. *International Journal of Network Security & Its Applications* (*IJNSA*), **3**, 182-187.

[3] Argyraki, K. and Cheriton, D.R. (2010) Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks. *USENIX Annual Technical Conference*, Berkeley, CA, 10-10.

[4] Yaar, A., Perrig, A. and Song, D. (2003) Pi: A Path Identification Mechanism to Defend against DDoS Attacks. *Proceedings IEEE Symposium on Security and Privacy Symposium*, 11-14 May 2003, 93-107.

[5] Savage, S., Wetherall, D., Karlin, A. and Anderson, T. (2000) Practical Network Support for IP Traceback. *ACM SIGCOMM Computer Communication Review*, **30**, 295-306.

[6] Ferguson, P. and Senie, D. (1998) Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. IETF, Request For Comments 2267, United States.

[7] Wang, H., Zhang, D. and Shin, K.G. (2002) SYN-Dog: Sniffing SYN Flooding Sources. *Proceedings of the* 22*nd International Conference on Distributed Computing Systems*, 421-428.

[8] Park, K. and Lee, H. (2001) On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets. *Proceedings of the* 2001 *SIGCOMM Conference*, **31**, 15-26. http://dx.doi.org/10.1145/383059.383061

[9] Sung, M. and Xu, J. (2003) IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDos Attacks. *IEEE Transactions on Parallel and Distributed Systems*, **14**, 861-872. http://dx.doi.org/10.1109/TPDS.2003.1233709

[10] Mahajan, R., Bellovin, S.M., Floyd, S., Ioannidis, J., Paxson, V. and Shenke, S. (2002) Controlling High Bandwidth Aggregates in the Network. *ACM SIGCOMM Computer Communication Review*, **32**, 62-73. http://dx.doi.org/10.1145/571697.571724

[11] Bernstein, D.J. (1997) SYN Cookies. http://cr.yp.to/syncookies.html

[12] Xu, J. and Lee, W. (2003) Sustaining Availability of Web Services under Distributed Denial of Service Attacks. *IEEE Transactions on Computers*, **52**, 195-208. http://dx.doi.org/10.1109/TC.2003.1176986

[13] Juel, A. and Brainard, J. (1999) Client Puzzles: A Cryptographic Countermeasure against Connection Depletion Attacks. *Proceedings of the Network and Distributed System Security Symposium*, *NDSS*'99, San Diego, February 1999.

[14] Aura, T., Nikander, P. and Leiwo, J. (2000) Dos-Resistant Authentication with Client Puzzles. Cambridge Security Protocols Workshop 2000. Lecture Notes in Computer Science. Springer, Berlin, 200.

[15] Dean, D. and Stubblefield, A. (2001) Using Client Puzzles to Protect TLS. *Proceedings of the Tenth USENIX Security Symposium*, Washington DC.

[16] Wang, X. and Reiter, M.K. (2003) Defending against Denial-of-Service Attacks with Puzzle Auctions. *IEEE Symposium on Security and Privacy*, Los Alamitos, 11-14 May 2003, 78-92.

[17] Chen, S.G., Tang, Y. and Du, W.L. (2007) Stateful DDos Attacks and Targeted Filtering. *Journal of Network and Computer Applications*, **30**, 823-840. http://dx.doi.org/10.1016/j.jnca.2005.07.007

[18] Vissers, T., Somasundaram, T.S., Pieters, L., Govindarajan, K. and Hellinckx, P. (2014) DDos Defense System for Web Services in a Cloud Environment. *Future Generation Computer Systems*, **37**, 37-45. http://dx.doi.org/10.1016/j.future.2014.03.003

[19] Geva, M., Herzberg, A. and Gev, Y. (2013) Bandwidth Distributed Denial of Service: Attacks and Defenses. *IEEE Security & Privacy*, **12**, 54-61. http://dx.doi.org/10.1109/MSP.2013.55

[20] Foroushani, V.A. and Zincir-Heywood, A.N. (2013) Deterministic and Authenticated Flow Marking for IP Traceback. *IEEE* 27*th International Conference on Advanced Information Networking and Applications* (*AINA*), Barcelona, 25-28 March 2013, 397-404.

[21] Sivabalan, S. and Radcliffe, P.J. (2013) A Novel Framework to Detect and Block DDos Attack at the Application Layer. *IEEE TENCON Spring Conference*, Sydney, 17-19 April 2013, 578-582. http://dx.doi.org/10.1109/TENCONSpring.2013.6584511

[22] Sowkarthiga, P. and Suguna, N. (2013) Finding the DDoS Attacks in the Network Using Distance Based Routing. *International Conference on Current Trends in Engineering and Technology* (*ICCTET*), *ICCTET*'13, Coimbatore, 3 July 2013, 410-412.

[23] Tao, Y. and Yu, S. (2013) DDoS Attack Detection at Local Area Networks Using Information Theoretical Metrics. *TRUSTCOM*'13, *Proceedings of the* 2013 12*th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Melbourne, 16-18 July 2013, 233-240. http://dx.doi.org/10.1109/TrustCom.2013.32

[24] Anderson, T., Roscoe, T. and Wetherall, D. (2003) Preventing Internet Denial-of-Service with Capabilities. *Newsletter ACM SIGCOMM Computer Communication Review*, **34**, 39-44

[25] Ioannidis, J. and Bellovin, S.M. (2002) Implementing Pushback: Router-Based Defense against DDos Attacks. *Proceedings of the Network and Distributed System Security Symposium*, San Diego, ISOC, Reston, VA.

[26] Mahlous, R., Chaourar, B. and Fretwell, R.J. (2008) A Comparative Study Between Max Flow Multipath, Multi Shortest Paths And Single Shortest Path. In: *Proceedings of PGNet* 2008, PGNet, Liverpool.

[27] Walfish, M., Vutukuru, H., Karger, D. and Shenker, S. (2010) DDos Defense by Offense. *Journal of ACM Transactions on Computer Systems* (*TOCS*), **28**.

[28] Lokanath, S. and Thayur, A. (2013) Implementation of AODV Protocol and Detection of Malicious Nodes in Manets. *International Journal of Science and Research* (*IJSR*), **2**.

[29] Yaar, A., Perrig, A. and Song, D. (2004) SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. *In Proceedings IEEE Symposium on Security and Privacy Symposium*, 9-12 May 2004, 130-143.

[30] Osipov, E., Kassler, A., Bohnert, T.M. and Masip-Bruin, X. (2010) Wired/Wireless Internet Communications. 8*th International Conference on Wired/Wireless Internet Communications* (*WWIC*), Luleå, 1-10 June 2010.

[31] Lee, W. and Xu, J. (2003) Sustaining Availability of Web Services under Distributed Denial of Service Attacks. *IEEE Transactions on Computers*, **52**, 195-208.

[32] Beitollahi, H. and Deconinck, G. (2011) A Cooperative Mechanism to Defense against Distributed Denial of Service Attacks. *IEEE* 10*th International Conference on Trust, Security and Privacy in Computing and Communications* (*TrustCom*), Changsha, 16-18 November 2011, 11-20.