A Comparative Study on Capability v/s. Filtering based Defense Mechanisms

Shubha Mishra
PG Scholar
Maulana Azad National Institute of Technology
Bhopal, India

R. K. Pateriya, PhD
Associate Professor
Maulana Azad National Institute of Technology,
Bhopal, India

ABSTRACT
Denial-of-Service and Distributed Denial-of-Service attacks have been the attack forms with maximum impact on their victims since their origin. The intensity of DDoS attacks is high as the attacker's identity and attack source is safeguarded well behind the bots. Numerous defense mechanisms have been employed to provide robustness against them. In this work, we aim to perform an in-depth study of a few filtering and capability based mechanisms. The advantages and limitations of each along with their architecture and operational services have been discussed in detail. A comparative analysis of their performances with their employment feasibility on the two scales (large or small) had been described as well. The goal of this work is to ease the selection of most robust techniques from these two classifications (filtering and capability based).

General Terms
DDoS defense mechanisms, Network and Communication Security.

Keywords
DoS, DDoS, filtering and capability-based mechanisms, attack traffic and legitimate traffic.

1. INTRODUCTION
With the constantly growing need for faster and uninterrupted communication, the technology advancement is rapidly shifting its phases. With this revolution in the technology and electronic field and the drastic increase in demand of speedy communication the idea of Internet came into existence in 1982 and was finally commercialized in 1995 [1]. It not only offered the possibility to faster communication but also led to the advancement of efficient and reliable communication modes. However, the sharing of data also opened backdoor to several latent vulnerabilities therefore hampering the process and misusing the available resources and data.

Denial-of-Service (DoS) attack refers to restricting a legitimate user from connecting to the victim or exploiting the victim's resources till the extent of its complete damage. The primary aim is to block the target from connecting to other hosts or servers or making the target unavailable to its legitimate and trusted customers.

Denial of service took a more dangerous turn with the origin of *Distributed Denial-of-Service attack*. This attack intensifies the attacker's strength and impact of damage as a single attacker (master) misuses multiple innocent hosts (called bots) to attain its objective successfully. A *bot* is a machine remotely under the control of the hacker for a specified time. The invader can simply send malicious program to multiple systems along with backdoor downloads (downloads without user's consent), e-mail attachments or similar means which on installation communicate to their master and follow the commands. These installed programs acting as communication medium between the attacker and bots are called *handlers*. A network comprising of master, handlers and bots is called a *bot-net*. This makes DDoS far more hazardous as it becomes extremely difficult to detect the real source of the attack. Moreover, the resources get exhausted in lesser time as compared to a single intruder attempting to gain the same results. Figure 1 describes distributed denial of service attack architecture.

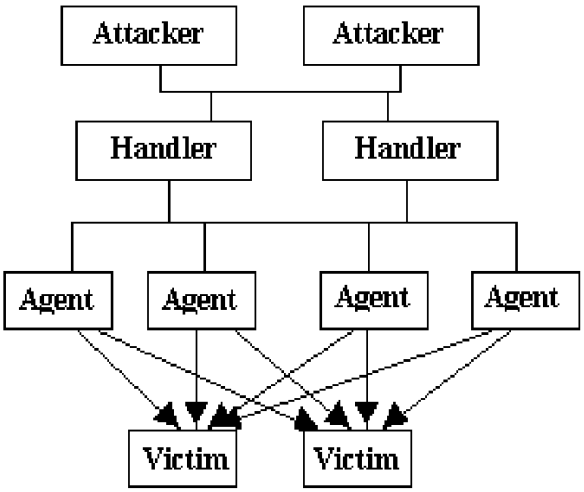

Figure-1. DDoS attack Architecture.

In the summer of 1999, the Computer Incident Advisory Capability (CIAC) reported the first DDoS attack incident [2]. Drastic growth rate was observed from 2000 to 2004 in several types of distributed denial of service attacks such as Flooding attack, Reflection-based and Amplification-based attacks, Smurf and Fraggle attack, R-U-Dead-Yet (RUDY) and loads of other.

The goal of this research is to perform an in-depth study on the performances of various capability and filtering based mechanisms to analyze the feasibility of their implementation under different attack scenarios.

The organization of the remaining article is as follows. In Section 2, a detailed study of various capability-based and filtering-based mechanisms along with the advantages and drawbacks of each has been discussed. Section 3 describes the

comparative analyzes and feasibility of each mechanism as per their range of their coverage area. Finally, the conclusion of this work with the possible future scope is discussed in Section 4.

2. DDoS DEFENSE MECHANISMS

2.1 Filtering-based mechanisms:

Filter-based techniques allow validation and handling of packets by filters installed at various levels in a network. These packets may be validated on the basis of several factors such as their source address, destination address, path-based or their purpose. This allows pre-validation of traffic restricting the entrance of suspected traffic into the network. Some of the widely deployed filtering techniques are discussed below in detail:

2.1.1 Ingress Filtering [20]

This technique is one of the most initial methods deployed under filtering mechanisms. It is helpful prominently to provide defense against source address spoofing attack (where an attacker may send data with some other host's IP address) and enhance the traceability of the real attacker.

Each interface is connected to a block containing a set of IP addresses. A filter is installed at the interface between ISP and the user, edges of ISPs or routers between networks to monitor the traffic before it is allowed to enter the network. A packet is dropped if its source does not belong to the set of valid IP addresses to which a particular interface is connected. This filter can be installed on any ordinary system and not specifically on a router. *RFC 2827 [28]* and *3704* [29] are defined by IETF (Internet Engineering Task Force) to reduce the impact of DDoS attacks.

Using ingress as close to the source as possible improves the traceability of the actual source of packet. Ingress filtering can be implemented in several ways using ingress access list and reverse path forwarding.

2.1.1.1 Ingress Access List

It maintains a list of network interfaces mapped to the set of acceptable prefixes. This list is referred by the filter for each data packet and ones with no match are dropped being considered as spoofed source packets. It enhances the security of a network or host but is not a very practical approach as manual update of list is needed and delay or inconsistency in updates may lead to dropping of legitimate traffic and ample chaos. Implementing this on multiple levels and higher granularity level is the key requirement for effectiveness of this technique.

2.1.1.2 Reverse Path Forwarding (RPF)

It lies on the concept of curbing the data based on the current status of Routing Information Base (RIB) or Forwarding Information Base (FIB). RPF uses unicast routing tables (for implicit construction of source based spanning trees) to make the decision of packet forwarding. A packet is forwarded only if it arrives from the best possible route from source to the router, else it is discarded. It is also dropped if its source address does not exist in the valid domain of IPs reachable by a particular interface. It is a more restrictive approach than access list as it considers dynamic routing information and does not require updating manually. Drawback of it lies where asymmetric routing is implemented. For symmetric routing (packets trace same route in forward as well as reverse direction) this approach is very effective.

Ingress filtering is not one of the very successful methods developed due to multiple reasons. *Firstly,* it is not applicable in case of multi-homed hosts (host with multiple IP addresses but single upstream link or multiple interfaces with single IP per interface) and transit network traffic. It is not a practical idea to overload the ingress with the mappings of interfaces and their respective IPs of other networks along with the details of the network on which it is installed. *Secondly,* it may deny traffic from legitimate users in case of mobile IPs when a user may dynamically change its address. *Thirdly,* it is not effective when address spoofing is done by a host within the same network. In such case, filtering of packet is not done as ingress only monitors the packets coming from outside the network.

2.1.2 Egress Filtering [20] [24]

This technique is similar to ingress filtering, except that it monitors the outgoing traffic from a network and restricts internal hosts from participating in attacks. A firewall is installed for this purpose on the network egress and tables are maintained for storage of each connection's information and other relevant data. The filter validates each packet based on certain policies before it is allowed to leave the network. Policies may be either**: *i.) Allow-all (or default-allow)* or, *ii.) Deny-all (or default-deny) policy***. It is always recommended to use default-deny policy as it offers stronger constraint on the outgoing traffic avoiding any mistake of flow of spoofed or suspected data out of the network. For every data packet to depart the network requires explicit administrative permission. Accompanied with this benefit is drawback of the amplification of overhead to a significant range. Allow-all policy demands lesser overhead but, it limits the constraint for traffic which renders it lesser efficient in security enforcement. The key role in egress filtering is played by firewall. All the policies decided by network authorities are enforced onto it and it is only responsible for their effective usage. This firewall is preferably a *NAT (Network Address Translation)* device which veils the internal address of the network from the Internet. A router may be also be used for egress filtering in a network. This may be implemented in two ways:

2.1.2.1 Access List

This scheme is similar to ingress filtering where access control list is maintained. The mapping of each interface to the permissible domain of IPs it desires to communicate is retained and decision is made on this basis.

2.1.2.2 Black-Hole Routing [26] [27]

It refers to a location within the network designed to dump all the discarded traffic without rendering any information to the source. This process is not valuable when attacker aims at congesting the bandwidth of link shared by the victim (or the victim's network). If routing is not performed close to the source, impacts such as low bandwidth utilization and congestion of links will be visible. Major challenges of this scheme are described as follows:

i.) In order to avoid the filtering of data, attacker may use encrypted data. This may result in bypass of traffic unchecked. Solution to this problem may involve use of *SSL Bridge* for passage of entire encrypted traffic through it. This permits validation and processing of data as per the requirement.

ii.) Similar to ingress, egress filtering also requires continuous modification of table which increases overhead.

2.1.3 Aggregate based Congestion Control (ACC) [11] and Pushback [10] [11] [12] [13]

Pushback and ACC are filtering based techniques which rate limits the traffic on a specific path passing through specific routers. An *aggregate* is a subset of entire network traffic which could be distinguished based on certain characteristics. In this approach, a destination can request its nearest access router to rate-limit traffic coming through a specific router for a specific aggregate. This router would then forward the request towards the uplink routers and the process continues. It becomes mandatory for routers on the links upstream to rate-limit the flow or packets will ultimately be dropped by the routers on the links downstream.

Each path router then verifies the packets to check if it is part of congestion or attack traffic and dropped if true. There may be some traffic which is not intended for the victim but share the path through the same routers as by the victim. Verification is performed to identify such packets and endeavor is to permit the flow of this traffic without much hindrance. A rate limit value is defined at the output queue of routers which determine the dropping or forwarding of packets. John Ioannidis and Bellovin [12] implemented the pushback daemon, *pushbackd* which provides updates for the rate limiter values to upstream daemons on the basis of the packets dropped. Routers implementing Pushback exchanges three types of messages: *request, response* and *status.*

Header fields	RLS-ID	Maximum depth
Depth of Requesting Node		Bandwidth Limit
Expiration Time		Congestion Signature

Figure 2: Pushback request packet format.

The extent to which the request has to be forwarded is defined by the originating router using *depth* field in the request packet. Another message called *cancel* message is generated when originating router intends to stop the rate limiting process. Pushback requires an efficient aggregate detection algorithm to detect congestion and decide on methods to respond effectively.

Pushback and ACC offer several features which motivates their deployment on a large scale.

i.) These mechanisms are deployed close to the destination and gradually shift towards the source which offers the maximum defense in case of bandwidth flooding attacks.

ii.) Due to identification of aggregates, blocking is not restricted on the basis of source IP address. The destination is capable of selecting the direction of the route from which it intends to curb the traffic.

The limitations of these mechanisms are as discussed below:

i.) Their deployment requires involvement of all the possible path routers which could be a large count for a large network. This increases the complexity and overhead considerably.

ii.) Another setback is in the scenario of link congestion attack, pushback requests may not be forwarded timely and in an attempt to do so, it may result in further flooding of the link.

iii.) To be fully effective Pushback protocol demands secure transfer of its request, response and status messages. Modification in any of these may lead to severe chaos in the network. Worst case scenario arises when the closest access router for a destination is itself malicious.

iv.) Pushback and ACC are very less effective against uniform attacks from multiple sources transmitting data at a constant (unsuspected) rate.

2.1.4 StopIt [4]

This is another filtering-based mechanism that allows user or possible victim to permit the blocking of traffic from a suspicious source for some definite period of time when attack situation is suspected. Each autonomous system (AS) have a separate StopIt server which handles the filtering request of that AS. StopIt servers of connected AS's are aware of each other's IP addresses. We consider the following scenario for implementation of StopIt architecture in brief.

Consider a destination D who wishes to block a suspected source S for a specific period of time T. It first sends a StopIt request to its access router R_d which then forwards a request to StopIt server of destination's autonomous system S_d. The request includes source, destination and time period for which to block the traffic *(S, D, T).* This request which is inter-domain in nature is then sent to StopIt server S_s of source's autonomous system. S_s then generates a request and sends it to R_s, access router of the attacking source which installs a filter on receiving the request after verifying it and sends a request to source S to stop transmitting data to D. An amenable host will immediately halt the transmission minimum till time 'T' expires. If in case it does not respect the request, the router will handle such host as per the protocol but anyhow, the blocking process will be performed.

This approach uses a mechanism called PASSPORT [30] to avoid address spoofing and cryptographic authentication to avoid modification of StopIt requests in their path. The cryptographic operations used are however, very simple and require only limited processing overhead and resources. StopIt filtering technique has better resistance to bandwidth flooding attacks and strategic filter exhaustion attacks. But, it is ineffective when link is already under congestion.

2.1.5 AITF (Active Internet Traffic Filtering) [6]

AITF is a network-layer defense system based on the deployment of filters at proper locations. A receiver can block any suspicious traffic flow '*F*' from a specific sender for a definite period of time '*T*'. All traffic for a particular domain passes through its border routers. Numerous path-identification mechanisms as suggested in [8] and [9], exist which allows a destination to identify the path traced by a packet. The destination can then decide the set of hosts it intends to block. By default, AITF supports accept-all policy.

A victim can send an AITF request enclosing the flow details; it intends to halt to its respective border router which instantly installs a temporary filter to block the flow. This filter is uninstalled the moment its request is satisfied by border router at sender end. This includes source address, destination address, path-information (path followed by packet) and time-period *(S, D, P_i, T).* A three-way handshaking protocol is established to verify the legitimacy of the request. To block *'F'*, using a local key and hash function, the nonce is calculated as:

$$\boldsymbol{nonce_i = HASH_{key}\ (F)} \quad (1.)$$

An inter-domain request is then generated and sent to border router of source to appeal the discontinuation of flow. A filter is then installed immediately at attacker's end by it and is removed as the attacker halts the flow. *Filtering Gain* is a metrics to analyze the tradeoff between cost and performace in this approach. It is defined as the ratio of number of flows blocked to the required number of filters given as:

$$\mathbf{FG= N_{BFlows}/ N_{Filters}} \quad (2.)$$

The benefits offered by this mechanism are discussed below:
i.) Receiver gains complete authority to deny traffic from unwanted sources, not necessarily attackers.
ii.) It provides instant blocking rather than forcing the victim to wait till the attacker stops transmitting packets.
This mechanism, however, does not entirely provide effective defense against DDoS attacks:
i.) AITF offers best defense in case of destination-flooding and bandwidth attacks but, it is still highly vulnerable to two-way link flooding attack. Also, in a scenario where link between two domains is already under attack, the AITF request could not reach the source timely.
ii.) This mechanism assumes that the best possible path-identification technique underlies which is not always true. There may be multiple interfaces and transit networks which need to be considered as well.
iii.) Another major setback of this approach is that it relies on the border routers and the sender side to a large extent. A malicious gateway on either side can result in the worst consequences.

2.2. Capability-based mechanisms

These mechanisms are based on the approach that each source can decide for itself from whom it agrees to accept the data by granting them certain privileges in form of *capabilities*. The capabilities are then attached to each packet transmitted by the sender during subsequent communication and each packet is verified at every path router before forwarding. However, these techniques have led to another vulnerability called Denial of Capability (DoC) attack where legitimate users are denied of attaining capabilities.

2.2.1 Portcullis [5]

A newer version of denial-of-service attack was originated with the development of initial capability-based mechanisms. This attack was named as *Denial-of-Capability (DoC)* attack as it results from the flooding of request channel with the request traffic. Only a few capability based techniques like TVA and Portcullis are robust against DoC attack.
Portcullis is a technique based on the concept of granting privileges to selected destinations by the receiver. The source needs to solve a set of problems called *puzzles*. It is based on per-computation fairness which implies that even when under attack, no attacker could obstruct a genuine host from connection setup request. A brief overview of its operational deployment is described next.
The *Seed Generator* releases *seeds* which are made available to the users by a S*eed Distribution Service.* The latest seed can then be used for generating puzzles by *Puzzle Generation Algorithm*. Clients can obtain the puzzles in different ways one of which is the DNS (used in [5]). Client can then use the seed to perform puzzle computation and append that seed and its solution into the capability setup packet. The seed generator randomly picks a number h_0. Beginning from h_0 to h_n, a hash chain is then computed using a public hash function H as in equation 3:

$$\boldsymbol{h_{k+1} = H(h_k \|_k)} \quad (3.)$$

Using this latest seed h_i, a randomly chosen 64-bit nonce *r*, a difficulty level *l* (for the puzzle) and a 64-bit number *x* such that last l bits of p are zero the puzzle is generated as follows:

$$\boldsymbol{p = H(x\|r\|h_i\|dest\ IP\|\ell)} \quad (4.)$$

These values *x, r,* h_i and *l* are then appended into the request packet while p is regenerated at each router during puzzle verification. The last seed of the hash-chain, h_n is called the *anchor* and it is released only about once per year. The router verifies the puzzle by computing the value $\boldsymbol{H(h_i\|i)}$ and comparing it with the last seed generated $\boldsymbol{(h_{i+1})}$**.**On the basis of the difficulty level of puzzles, the priority of a specific flow is decided. Hosts with more challenging puzzle levels are preferred for connection setup over simpler ones.
Portcullis is more robust approach to DoC attacks and provides maximum possible delay in attacker's attempt to obstruct legitimate users. Portcullis offers some advantages which defines its uniqueness amongst other existing capability techniques:
i.) Portcullis requires only border routers to be active for puzzle verification and provides 2^{64} possible values of randomly chosen 64-bit value *r*. This reduces the probability of generation of same puzzle for the same combinational values and being valid for same time period; to almost zero.
ii.) Another advantage is that its performance does not degrade much with the rise in the number of attackers and so, it is capable of defending against denial-of-capability attacks successfully.
Limitations of Portcullis are as discussed below:
i.) It assumes equal availability of resources to all hosts which is not always valid. In a situation where a fair user lacks ample resources, it may undergo infinite starvation.
ii.) With Portcullis, in case of a victim already under attack, high-level puzzles may create a situation of link congestion or dropping of legitimate traffic intended for some other destinations.

2.2.2 SIFF (Stateless Internet Flow Filter) [7]

SIFF permits a destination to decide the set of hosts it agrees to communicate and the ones it desires to block. This protocol is based on the assumption that a host is capable of distinguishing between legitimate and attack traffic flows. The entire network traffic is classified into two types:
Privileged traffic and *Unprivileged* traffic. Always *privileged* traffic is given higher priority over the other one as it is the traffic that only authenticated host possessing capability access can transmit. *Unprivileged* packets are transmitted when a source initially requests a destination for connection establishment. Due to space constraint, only a brief discussion on this protocol had been done below. However, the complete working of this mechanism have been described in [7] with certain assumptions made including the modification in the IP packet fields by Yaar, Perrig and Song.
The capabilities are valid only for a definite time period *'T'* and destination must continuously update them even for the same source. The router originally appends path-specific information into the request packet *'P_r'* (considered unprivileged as per this protocol). With this information capabilities are granted to selected hosts by the destination. Each sender will then compulsorily attach the retrieved capabilities to each communication packet *'P_c'* for data transmission thereafter. Each router then compares the capabilities with information they would have inserted had it been a P_r packet type. If the two values match packet is forwarded. If any P_c fails the verification, it is dropped by that router at that point. It is due to this validity of capabilities for a fixed time that a receiver could halt the flow from a specific sender by simply stopping the updating process for it. This approach offers several benefits:

i.) Always privileged traffic is prioritized over request traffic. Therefore, even when under attack, already established connections remain unaltered with the ascending number of attackers.
ii.) Routers need not maintain per-flow information (or state), rather, only a small amount of data update is required with respect to each router interface. This restricts the space and resource requirement.
Still we do not consider this as an entirely robust approach for defense against DDoS attacks as:
i.) The situation where colluders are sharing the common bottleneck link with the victim, privileged traffic can also be hindered along with unprivileged packets. Also, this mechanism is vulnerable to denial-of-capability (DoC) attack
ii.) The foremost setback lies in the assumption that a destination is capable of differentiating between legitimate and attack traffic, which in itself is a serious challenging issue. The successful deployment of this protocol is possible only when a strong differentiating algorithm is implemented beforehand, which is not always necessary.
A better approach could be as suggested in [7], combination of SIFF with a puzzle auction mechanism to obtain the capabilities. This would eliminate the above mentioned assumption as sender with more challenging level of difficulties will be prioritized over simpler ones. But, this approach also does not offer a solution for DDoS attacks in its entirety.

2.2.3. TVA (Traffic Validation Architecture) [3] [25]

This technique eliminates the limitations of the capability-based techniques like SIFF to a large extent. Each source provides tokens called capabilities to specific destinations whose data it wants to receive. Since the capabilities need to be unforgeable and fine-grained, they are generated with the help of strong hashing algorithms like SHA and AES. Also, to restrict hosts from readily offering capabilities, scheduling techniques such as fair-queuing (on per-source or per-destination basis) are implemented over the available bandwidth. Each router on the path from source to destination verifies every packet forwarded by it after appending its own pre-capability as shown in Figure 2.

Time-stamp (8 bits)	**hash (src IP, dest IP, T, secret) (56 bits)**

Pre-capability appended by routers

Time-stamp (8 bits)	**hash (pre-capability, N, T) (56 bits)**

Capability appended by hosts

Figure3: Format of capabilities in TVA [3]

The packets which lack the valid capabilities are considered as low priority data. A packet may be a request packet or a regular packet (or priority packet) depending upon whether a sender is requesting for fresh set of capabilities or it had already received them. Request traffic can be granted a small fraction of total available bandwidth to enhance congestion avoidance. Legacy traffic is the lowest priority traffic which may be allotted only about 1% fraction of bandwidth. Each packet is piggybacked along with normal packets to reduce overhead. The sender sends a request packet to obtain capabilities along with normal packet such as TCP SYN initially. It then receives a set of capabilities valid for a specific time period *(T)* and number of bytes *(N)* the sender can transmit using current capabilities with TCP SYN/ACK packet.
TVA works well but some issues this technique needs to handle include: *First*, with the involvement of say, x malicious hosts each requesting for capabilities from the same destination, $(1/x)^{th}$ of the bandwidth allotted for request traffic is utilized when per-source fair queuing is enforced, thereby hindering legitimate users considerably for greater values of x. This issue could be resolved by enforcing fair-queuing for request traffic along with regular traffic.
Second issue, is the deployment of malicious routers at any point in the network. This may enhance the attack by deliberately forwarding capabilities to false sender or unverified traffic to false receiver, therefore resulting in congestion. This issue does not cause a dangerous impact as even due to presence of one or two such routers the ones next perform fair verification and will drop such traffic immediately on either side.
In order to ensure secure transfer of capabilities, a new technique called TVA+ [4] have been developed. This technique is robust against source-address spoofing on its request channel. It uses per-AS along with per-source (hierarchical queuing) unlike TVA for request bandwidth fair sharing. Hence, it offers equal opportunity to legitimate users to request connection setup even under attack conditions.

3. COMPARATIVE ANALYSIS

In this article, a depth-wise survey of some of the advanced capability and filtering based techniques is performed. Observation was that the filtering based techniques are a better approach when large scale deployment is the need. This is because their architectures are solely dependent on the positions where the filters are installed and the criteria or the factors on which the filtering is based. Ingress and Egress filters are unable to offer security from the intruders within the network. Also, they are highly vulnerable to the source and destination address spoofing and are ineffective for large networks due to increased overhead and reduced processing speed.
ACC and Pushback are healthier than ingress filtering as far as security is concerned but these rely on network and its elements (routers and ISPs) entirely. This makes them less feasible and overhead is still high. StopIt and AITF protocols reduce the overhead as only few network elements are involved. Defense against attack is much more enhanced as receiver has full authority to take blocking decisions. AITF offers more robustness then even StopIt for most attacks as temporary filters are installed instantly to satisfy the request within negligible time. Both of these methods still demand network support which raises the possibility of attack with indulgence of malicious network elements. In overall, AITF is found to be the most effective filtering technique as well as StopIt is also a considerable option along with some enhancements.
The detailed study of capability mechanisms showed that they offer more safeguarded environment to legitimate users due to

accessibility to privileges (*capabilities*). However, their deployment is confined to small scale as capability distribution becomes tedious with oversized networks. Capability-based systems still suffer from a critical weakness: they cannot protect the initial capability request, because that request is sent unprotected as non-prioritized traffic. With Portcullis, genuine hosts may be declined access due to their lack of resources. It is based on the assumption that all the hosts hold identical resources which, being a rare possibility, fails its practicality. SIFF architecture does not hinder the legitimate user from gaining access to a destination, unlike Portcullis. SIFF is not a very feasible solution as it relies on the strength of the receiver to efficiently differentiate between legitimate and attack traffic. It's major setback is that it provides negligible defense against denial-of-capability (DoC) attacks.

TVA and TVA+ are the most effective capability-based approaches and outperform almost all filtering as well as capability based techniques. Their deployment on a large scale however, is still a challenging issue which is why they could not be titled as entirely robust against DDoS attacks.

We observed that even though majority of the discussed techniques are hybrid (not source or destination positions restricted) in their deployment, none of them is a standalone defense approach. A hybrid mechanism which combines these two categories is a very practical solution. This may require effective handling of overheads but still would be far stronger than each of these individually.

The graph shown in figure 4 depicts the difference in the effectiveness of Capability and Filtering mechanisms under different attack intensities. Attack power (also called attack intensity in this article), in general, is considered as a measurement parameter based on number of attackers and size of packet. The effectiveness is a secondary term based on the number successful (completed) TCP transfers by legitimate users. Based on these parameters, the above graph proves that initially, when attack intensity is low, both the mechanisms are effective against the attack.

However, filters are slightly more effective then capabilities. With the rise in the attacker's power, the effectiveness of both the approaches declines drastically. It is clearly shown that filters are visibly ineffective due to the delay or inability of their timely installation. The scenario when the attacker is able to gain unhindered long-term access to capabilities, this approach also becomes ineffective for defense. Capability mechanisms show a constant performance for larger range of attack power as compared to its counterpart. However, with the continuous increase in the attack intensity, both the mechanisms fail in defending against the attack. It is hence, clear that a fail-safe mechanism is the need of the hour. An approach which could effectively handle high attack intensities, thereby, offering a more robust environment for secure communication is vitally essential.

Table 1 summarizes the comparative study on Capability and Filtering mechanisms based on several parameters. While for some parameters capabilities are better performers, for others Filtering are superior ones. However, neither of them is a standalone defense mechanism, capable of completely outnumbering the other when all the parameters are considered.

Table 1: Summary of comparison between Capability and Filtering mechanisms.

Parameters for comparison	**Capability mechanisms**	**Filtering mechanisms**
Deployment position	Hybrid (source to destination)	Source or destination/Hybrid
Scalability	Small-Medium	Upto Large scale
Performance	High if capabilities are secure.	High only upto limited attack intensity
Complexity	High	Low - Medium
Dependability	On path routers	On routers, ISPs
Cost effectiveness	Low (much costly)	Medium

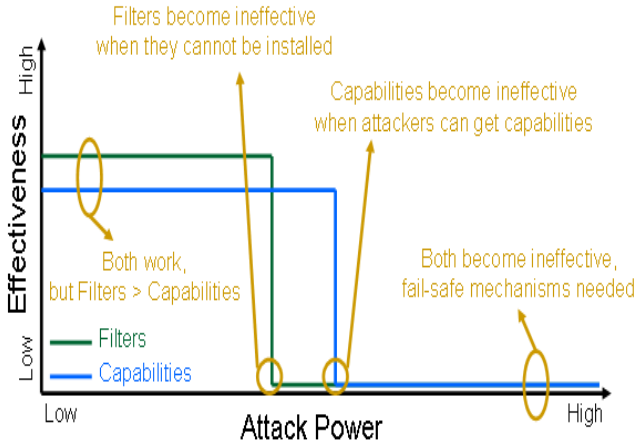

Figure 4: Effectiveness of Capability and Filtering mechanisms under different attack intensities [30].

4. CONCLUSION

The goal of this survey was to analyze the filtering and capability based mechanisms and the focus lied mainly on the advantages offered by each technique over others and their respective operational constraints.

The conclusion drawn from this survey is that neither installation of filters nor granting the capabilities is a complete solution for DoS problem. There are multiple other approaches based on varied concepts besides those discussed in this article. We found that no technique is a standalone defendant against DDoS attacks. A more effective approach which deploys capability based techniques such as TVA, with installation of filters at proper positions which perform filtering on the basis of some criteria, is the current need. A wide scope exists in future for a combination of these two categories.

5. REFERENCES

[1] History of the Internet, [online] http://en.wikipedia.org/wiki/History_of_the_Internet.

[2] P. J. Criscuolo, *Distributed Denial of Service*, Tribe Flood Network 2000, and Stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory Capability (CIAC), UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, February 14, 2000.

[3] X.Yang, D.Wetherall, and T.Anderson, *TVA: a DoS-limiting network architecture*, IEEE/ACM Trans. Netw., vol. 16, no. 6, pp. 1267-1280, 2008.

[4] Xin Liu, Xiaowei Yang and Yanbin Lu, "*To Filter or to Authorize: Network-Layer DoS Defense against Multimillion-node Botnets*", ACM SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.

[5] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu, "*Portcullis:Protecting Connection Setup from Denial-of-Capability Attacks."* In *ACM SIGCOMM*, 2007.

[6] K. Argyraki and D. R. Cheriton, *Scalable network-layer defense against internet bandwidth-flooding attacks*, IEEE/ACM Transaction Netw., 17(4), pp. 1284-1297, August 2009.

[7] A. Yaar, A. Perrig, and D. Song, *SIFF: a Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks*, in Proc. 2004 IEEE Symposium on Security and Privacy, pp. 130-143, May 2004.

[8] X. Yang, *"NIRA: A new internet routing architecture*", Proc. ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA), Karlsruhe, Germany, Aug. 2003.

[9] K. Argyraki and D. R. Cheriton, *"Loose source routing as a mechanism for traffic policies",* Proc. ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA), Portland, OR, Aug. 2004.

[10] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, *Controlling high bandwidth aggregates in the network*, presented at Computer Communication Review, pp.62-73, 2002.

[11] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker, *"Controlling high bandwidth aggregates in the network"*, Submitted to ACM SIGCOMM 2001.

[12] John Ioannidis and Steven M. Bellovin, *"Implementing Pushback: Router-Based Defense Against DDoS Attacks",* in Proc. of Network and Distributed System Security Symposium, 2002.

[13] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker, *Controlling high bandwidth aggregates in the network – extended version.* [online] *http://www.aciri.org/pushback/.*

[14] Criscuolo, P. J. (2000) *Distributed Denial of Service Trinoo, Tribe Flood Network, Tribe Flood Network 2000, and Stacheld-raht*, CIAC-2319, Department of Energy Computer Incident Advisory (CIAC). Rev. 1 UCRL-ID-136939.

[15] Dietrich, S., Long, N., and Dittrich, D. (2000), *Analyzing distributed denial of service tools: The shaft case*. Proceedings of the 14th USENIX conference on System administration, New Orleans, Louisiana, USA, 3-8 December, pp. 329–340. USENIX Association.

[16] Hancock, B. (2000), *Trinity v3: A DDoS tool hits the streets*. Computers & Security, 19, 574.

[17] Batishchev, A. M. (2004), " *LOIC(Low Orbit Ion Cannon)*", [online] *http://sourceforge.net/projects/loic/.*

[18] T. Peng, C. Leckie, and K. Ramamohanarao, *Survey of network-based defense mechanisms countering the DoS and DDoS problems*, ACM Comput. Survey 39, 1, Article 3, April 2007.

[19] C. Douligeris, and A. Mitrokotsa, *DDoS attacks and defense mechanisms: classification and state-of-the-art*, Computer Networks, Vol. 44, No. 5, pp. 643-666, April 2004.

[20] P. Ferguson, and D. Senie, Network Ingress Filtering: Defeating Denial of Service Attacks that employ IP source address spoofing, Internet RFC 2827, 2000.

[21] ha.ckers.org, *Slowloris HTTP DoS*, Retrieved Oct. 19, 2012, [online] *http://ha.ckers.org/slowloris/*

[22] K. J. Higgins, Researchers To Demonstrate New Attack That Exploits HTTP, Nov. 01, 2010, [online] *http://www.darkreading.com/vulnerabilitymanagement/167901026/security/attacks-breaches/228000532/index.html*

[23] Egress Filtering, [online] *http: // en.wikipedia.org/wiki/Egress_Filtering.*

[24] X. Yang, D. Wetherall, and T. Anderson, *A DoS-limiting Architecture*, ACM SIGCOMM, Philadelphia, PA, USA, August 2005.

[25] CISCO, "*Remotely triggered black-hole filtering-destination based and source based*", [online] *http://www.cisco.com/c/dam/en/us/products/collateral/security/ios-network-foundation-protection nfp/prod_white_paper0900aecd80313fac.pdf*

[26] Black-hole filtering, [online] *http://en.wikipedia.org/wiki/Black_hole (networking)*

[27] IETF, *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, May 2000, [online] *https://tools.ietf.org/html/rfc2827.*

[28] IETF, Ingress Filtering for Multihomed Networks, March 2004, [online] https://tools.ietf.org/html/rfc3704.

[29] Liu, X., Li, A., Yang, X., and Wetherall, D. 2008, " *Passport: secure and adoptable source Authentication*", In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation. NSDI'08. USENIX Association, Berkeley, CA, USA, 365-378.

[30] X. Yang, A DoS Limiting Network Architecture, [online] *http://www.cs.duke.edu/nds/ddos/*