1 Introduction

In this paper the authors describe a service for secure communication on the times existing IP infrastructure. The aim of the paper is to describe a service secured from DoS attacks. The communication is between predetermined users and the service, where the users could be located anywhere in the wide-area network and must have authorization to communicate with that service. The service is proactive, meaning that the system aggressively filters the packets and blocks all the packets whose source address is not "approved". The attackers have a finite set of resources to perform the attacks and know the IP addresses of the nodes that participate in the overlay and of the target that has to be protected. The service however does not solve the general problem of DoS attacks.

2 SOS architecture Description

SOS is a network overlay, composed of nodes that communicate with one another atop the underlying network substrate. Nodes route messages to each other inside the network. Attackers want to prevent legitimate users from communicating with the service. They can launch a type of distributed DoS attack from a variety of points inside the wide area.

The basic goal of SOS is to distinguish the 'good'(legitimate) packets from the 'bad' packets. In order to achieve that a firewall is placed deep inside the SOS infrastructure. This helps avoiding network congestion (firewalls are by nature susceptible to DoS attacks that's why it is placed deep inside SOS). For this reason also the firewall is distributed.

2.1 Protecting the Target: Filtering

In order to avoid DoS packets to reach the main target, a filter is constructed at a set of high-powered routers. These high-powered routers are around the target and only allow traffic from a set of IP Sources.
2.2 Reaching a well-filtered Target

The target selects a subset of the total nodes (Ns) that participate in the SoS infrastructure, to act as forwarding proxies. The filter now allows traffic whose source address matches an IP Address of a node in the subset Ns. This subset of nodes are aimed to do more complex security techniques in order to verify that a packet comes from a legitimate user. This method prevents attackers who knows the legitimate users IP Address to attack the target. However if the attacker knows the IP Address of the proxy, he can either attack the proxy or even worse the target. For this reason the identity of the proxies must remain hidden. Hidden proxies are also referred as secret servlets.

2.3 Reaching a Secret Servlet

The target chooses and informs some of the overlay nodes that they will act as secret servlets. Now if a packet arrives at the secret servlet, and is verified that it comes from a verified user it passes through the filter to the target. Now we need some nodes that will route packet to the secret servlet. The network infrastructure until reaching the secret servlet is made like a graph. A pathh exists between every pair of nodes.

2.4 Connecting to the Overlay

Some other nodes will act as secure overlay access points(SOAPs). SOAPS receive packets that have not been marked as legitimate and verifies them. Using many SOAPS, requires more resources from the attacker in order to prevent legitimate traffic to flow through the overlay.

2.5 Routing through the Overlay

Choosing at random the next hop can become very insufficient as the number of hops grows. The authors of the paper propose an alternative routing algorithm that has $O(\log N)$ complexity which is called Chord. Each overlay node has a list that contains $O(\log N)$ elements of the active nodes in the overlay. By a technique which is called consisted hashing each overlay node maps an arbitrary identifier to a unique destination node that is active on the overlay. Given the destination identifier, each overlay node chooses a node from the element list such that from an arbitrary chosen node, the destination node is reached in $O(\log n)$ overlay hops. Chord is used to direct packets from any overlay node to the node that the identifier is mapped to, by applying the hash function to the targets IP Address. The reaching node is called beacon. When a packet is approved form the SOAP, a hash is applied to the IP Address of the target and it is used as a key for the rest of the time. However beacons must know the identity of the secret servlet. This way the packet can be delivered from any SOAP to the overlay nodes, then to the beacons, then to the secret servlets and finally to the target through the filters.
2.6 SOS Architecture summary

Putting it all together, SOS firstly creates a filter that only allows packet whose source IP Address is accepted. Then a set of nodes are chosen to act as secret servlets, these servlets will forward traffic to target through the filters. These nodes also compute a key k for each of the hashed functions based on the targets network address. Each of the keys will identify a set of nodes that act as beacons. Beacons are then notified with a secure way with all the necessary information to forward packets to the proper secret servlet. A source that wants to communicate with the target contacts first a SOAP. SOAP then securely forward the packet to the beacons by applying hash function to the source IP Address and identify the next hop. Beacons then route the packet to a secret servlet and then to the target through the filtering.

3 Conclusion

In this paper the authors propose a service that is secured from DoS attacks on the existing IP infrastructure. They designed the service with proactive mechanism that consists of aggressive packet filtering deep inside the architecture that protects a target, an overlay network that can self heal in case of DoS attack, and a scalable access control mechanism that allows legitimate packets to reach the target.