END TO END DEFENCE AGAINST DDoS ATTACKS

Anastasia Gemona, Ishbel Duncan, Colin Allison and Alan Miller
University of St Andrews
School of Computer Science

ABSTRACT
Denial of Service (DoS) attacks and Distributed Denial of Service (DDoS) attacks accounted for more losses than Internet financial fraud and viruses combined (CSI/FBI 2003). The Internet has been exposed as being particularly vulnerable to Denial of Service Attacks. This has stimulated research into DDoS and the consequent development of many techniques which aim to control them. This paper aims to contribute to this literature. An holistic approach to combating DDoS is proposed, which places particular stress on the importance of locating functionality in the most appropriate location and that source, intermediate and destination network elements co-operate together. It is argued that attack traffic is best stopped before it leaves its source network, that it is best detected and analysed at the target network and intermediate routers need precise information to allow them to control economically the DDoS traffic that escapes the source network. The design of a passive monitor that is able to use measurements of attack and regular traffic to enable dynamic configuration of network elements is presented along with a detailed discussion of how such a monitor can be deployed to combat the common SYN flood attack. The extension of this approach to combat other forms of DDoS attack is also discussed.

KEYWORDS
Passive Measurement, Denial of Service, Security, Internet, Network Traffic

1. INTRODUCTION
Using back scatter analysis, CAIDA found 12,805 Denial of Service (DoS) attacks on over 5,000 hosts belonging to 2,000 organisations in a three week period (Householder et al. 2001). The fundamental design goals for the Internet stressed survivability in the face of external attack rather than facilitating traffic management (Clark 1988; Cerf, 1978). In the 80s this, combined with the phenomenal growth of the Internet, presented researchers with the challenge of introducing systematic measures for controlling congestion (Jain and Ramakrishnan 1988; Jacobson and Braden 1990). Today the similar challenge of combating malevolent DoS traffic presents itself within the environment of the Internet. This challenge is difficult in part because:

*In the Internet or similar contexts ... we do not know how many potential clients there are, let alone how they will behave. The structure is not merely late bound – it is not bound at all.* (Needham 1994).

The problem of DoS attacks was first identified by Gilgor (Gilgor 1983) within the context of operating systems. DoS attacks attempt to deny legitimate users access to a resource by causing resources to be
expended futilely. In the context of networking the resources that are commonly targeted include: network bandwidth, processing power of a server or a server’s memory resource. Distributed DoS attacks involve multiple hosts co-operating together to consume the resources of some target or targets. There are a number of different tools for generating DoS attacks. They differ in how resources to mount the attack are captured, how communication is maintained between the master computer, handlers and slaves and in the type of traffic generated. There are a number of types of attack traffic. These include: Smurf -- ICMP (Internet Control Message Protocol) ping requests to a directed broadcast address. The forged source address on the request is the target of the attack. The recipients of the directed broadcast ping request respond to the request and flood the target's network. In an ICMP flood large numbers of ICMP packets are sent to the target host. In a UDP flood large numbers of UDP (User Datagram Protocol) packets are sent to the target system. The aim is to saturate the available network bandwidth. In a TCP flood large numbers of TCP packets are sent to the target system with the consequent saturation of network resources. In a SYN flood attack a client sends a request (SYN) to a server announcing its intention to start a connection. The server replies by sending an acknowledgement (SYN ACK), accepting the establishment of a connection to the client and simultaneously reserving an entry for the pending connection in its connection queue. If sufficient bogus connections are established there will be no available entries in the pending connection queue and genuine requests will be refused. An important feature of this attack is that the client does not need to reply to the server’s SYN ACK and does not need to see the server’s response. As a consequence the client can use IP spoofing to hide its identity and make it difficult to differentiate DOS and regular traffic. The second aspect of the attack is that it exploits features of the TCP protocol to magnify its effect, it is therefore not necessary to consume all bandwidth or processor resources as filling the connection queue will suffice.

Whilst the challenge of DoS attacks has provoked widespread research and the development of numerous techniques for combating them, this has often been on a somewhat piecemeal basis. This paper offers the outline of a taxonomy of defence mechanisms from which a functional decomposition of where best to locate defence functionality is developed. This is followed by a detailed discussion of detecting and defending against a common form of DoS attack the SYN Flood. It is then shown that the approach is built around a flexible framework which facilitates its extension to support defence against other types of known attacks.

2. DEFENCE TAXONOMY

In the last few years there has been much research into how to defend effectively against Distributed DoS attacks. A significant number of techniques have been developed. Whilst there has been work in developing taxonomies of DoS attacks (Hussain 2002; Mirkovic 2002; Manzano 2003; Houle 2001), there has been little written that would provide an overview and taxonomy of defense techniques (Cisco). There has also been little written on how these techniques could combine together to provide an effective and holistic approach to preventing and mitigating the effects of DDoS (Householder et al., 2001).

We offer here the outline of a taxonomy, which focuses on the location where the defence is mounted and the functionality that is deployed. The starting point is to think about the attack traffic as an End-to-End process and consider the network elements it passes through before reaching the target. For a typical attack this might include: subverted clients, source network firewall, source network boundary router, intermediate routers, destination network boundary router, destination network boundary firewall and target server. Next the various types of defense that are advocated in the literature and the functionality that they deploy are considered.

**Source:** A number of measures focus on preventing a client becoming subverted in the first place. These include Intrusion Detection Systems (IDS) (Paxson 1999; Roesch 1999) and defensive programming (Xiaohu Qie 2001). Fooling an attacker into thinking that they have subverted a host so that the attack methodology can be studied has also been proposed in the Honeypot (Weiler 2001) scheme.

The second point at which a defense is possible is the Egress Firewall/Router, IP spoofing can be combated by checking source addresses to ensure that all packets leaving the network have genuine source IP addresses. DWARD proposes detecting DDoS traffic at the source from the statistical relationship between data and acknowledgement packets and then rate limiting the attack traffic (Mirkovic 2002). Wang et al (Wang et al. 2002) use the monitoring of SYN and FIN pairs at the source network to detect a SYN flood attack.
Stopping DDoS traffic at the source has the advantage that none of the targets’ resources or Wide Area Network resources will have been wasted. It also makes sense to deploy intelligence at the edge of the network and thereby avoid adding complexity to the core where high data rates predominate. However, it may be difficult to detect DDoS traffic as the sources may be distributed across many networks. Universal or close to universal deployment is required otherwise a potential attacker can simply source from networks that are not defended. Detection strategies based upon statistical observations may be undermined where there is a small amount of attack traffic from any one network. From the point of view of a service provider they do not have control over the source networks and can only hope that effective measures are put in place. It follows that whilst defences at the source are important and can be effective they are not adequate by themselves.

**Core:** A number of defense mechanisms have been proposed that require implementation in intermediate routers. In Denying DoS attacks a router based solution is advocated (Shu Zhang 2000) where hardened routers detect DoS attacks and drop traffic near the source. This technique depends upon the embedding of cryptography in the core of the network. Floyd builds on congestion control work to suggest the monitoring and regulation of traffic aggregates (Manajan et al. 2002) in the core of the network to defend against Flash Crowds and DoS Attacks. A mechanism for communicating with upstream routers is included so that control can be pushed back to the source. IPTraceback suggests placing router IDs in the headers of packets so that the target can reconstruct the path that attack traffic is taking and thereby discover the source (Minho Sung 2002). MULTOPS is a data structure which facilitates the efficient monitoring of traffic to detect a DoS attack. A network device using MULTOPS detects attacks by looking for anomalies in the ratio of incoming and outgoing packet rates (Gil and Poletto 2001).

Employing defence mechanisms in the core of the network helps protect the network against misbehaving aggregates of flows and may lessen the impact of an attack upon target networks. However the question of economic resource usage is important here as in general schemes that rely upon per connection state have found it difficult to gain a foothold (Braden et al. 1997). Using router resources for untargeted monitoring may reduce their ability to carry out their traffic forwarding duties. Consequently, statistical approaches to dropping and rate limiting approaches to control are probably the best that can be achieved. The idea of pushing back information can however be extended, if routers could be provided with an accurate characterization of attack traffic or simple rules for discriminating between attack or regular traffic they are in a good position to take ameliorating action.

**Destination:** At the Ingress router or Firewall a number of actions can be taken. The CISCO IOS Firewall is an example of a commercially available IDS, which monitors traffic for suspicious activity, looking for matches to signature attack types. Upon detection three options are possible; issue a warning, drop the packets or reset the TCP connection (Cisco). The reliance upon signatures without reference to measurements of regular traffic often leads to a high proportion of false positives. CISCO IOS Intercept replies to TCP SYN packets and sets up two connections one between itself and the client and one between itself and the server. This allows it to prevent DOS traffic from reaching the server, but for valid connection requests the two connections have to be merged into a single session which the system manages (McNealis 1997). This has a significant performance impact even when no attack is in progress consequently many system administrators do not utilize the facility. Comparing IP addresses to that of regular traffic has been proposed as a way of differentiating between flash floods and DOS attacks (Jaeyeon Jung 2000) and could be used more generally to probabilistically identify DOS packets. Examining TTL values in IP packets is another suggested way of identifying attack traffic.

A number of actions are possible for a defence to be mounted at the target host (Spatscheck 1999). SYN cookies have been implemented in Linux to reduce the server resources that are tied up in a SYN attack. A SYN Cache is used to a similar end in FreeBSD. One of the motivations for the establishment of Peer to Peer file distribution systems was to overcome the problems of Flash Floods; this distributed property also makes them resilient in the face of DoS attacks. Another approach is to widely distribute resources content using for example Xenoservers. In general peer to peer distribution may be more resilient against both DoS attacks and Flash Floods.

It is the destination of the attack which is in the best position to identify an attack and characterise its nature. In addition monitoring of traffic is easier at the edge than the core as bandwidths are lower. It is also the place where there is the most information and therefore the best possibility of finding a solution. There is further motivation to mount a defence here as this is the place where damage will be done. It would be beneficial to mount a defence automatically as waiting for human intervention may take too long.

The destination network can carry out a number of important actions to mitigate the effect of a DoS
It can:

1) Protect the internal bandwidth of the network by dropping attack traffic at the edge
2) Protect processing resources by dropping traffic
3) Protect the rest of the network from packets generated in response to attack traffic (e.g. Reset packets, ACK packets SYN_ACK packets etc.)
4) Signal to upstream routers so that they can have sufficient information to selectively drop or rate limit traffic.

3. DEFENCE ARCHITECTURE

In developing congestion control on the Internet significant thought was given to a functional decomposition of roles. Irland (Irland et al. 1979) observed that as a router has a global view of traffic it was best placed to detect the onset of congestion. However, once traffic has reached the router valuable network resources have been expended. Consequently it is desirable for the host to control the flow of data into the network. This split in responsibility introduces the need for communication between the router and host. Jain builds on Irland’s functional decomposition (Jain and Ramakrishnan 1988). A similar approach is arrived at for the Internet with the combination of Random Early Detection Gateways and Explicit Congestion Notification (Floyd and Fall, 1999).

Figure 1 shows a proposed architecture for detecting and defending against DoS attacks. Online passive monitoring techniques are used to observe traffic that is coming into the network. If an attack is detected then the network is placed in a state of alert. Special measures are put in place, which ensure that service is maintained. It would be undesirable to be in a constant state of alert as some of the measures used to defend against a DoS attack may themselves lead to a degradation of service. Consequently, it is also important that the monitor can detect when an attack is over and is able to switch between active defence and passive monitoring modes.

A TCP connection is uniquely identified by the tuple of source and destination IP addresses and source and destination ports this allows the monitor to uniquely identify each connection. The monitor contains a connection layer (Figure 2), which contains an entry for each open connection. Associated with each connection in the monitor is the state necessary to track its evolution. Upon receipt of a packet the monitor identifies the connection that it belongs to and updates that connection’s state. Thus if a connection gets stuck at a certain point this will be known to the monitor. Above the connection layer is a host layer, where a table of all the hosts in the local network is maintained. The host layer allows the monitor to identify hosts that are under attack and decide on the appropriate remedial action to be undertaken. When events occur at the
connection layer the appropriate host object is notified. For each host in the home network the monitor maintains a count of the number of connections in each state. State is also maintained for the local network as a whole so that a global view can be maintained. The control module is informed by the host layer of events which suggest the start of an attack. It is able to query both the connection and host layers for more information and is responsible for formulating a defence strategy. This may include configuring the firewall, sending messages to upstream routers and warning servers.

A TCP connection’s start up and closure procedure are governed by its state machine or Macro State. Data transfer is governed by what is referred to as Micro State. The start up and close down of a TCP connection is controlled by the TCP state machine (Postel 1981), this can be referred to as TCP’s Macro State. During the lifetime of a TCP connection it will normally move through a number of state transitions. A client will actively open a connection by sending a SYN packet, it will then move into the SYN_SENT state. The server, upon receipt of a SYN, will normally generate a SYN_ACK, allocate memory resources and move from the LISTEN to the SYN_RECEIVED state. Upon receipt of the SYN_ACK the client will send an acknowledgement and move into the ESTABLISHED state. When the server receives the ACK packet it will move into the ESTABLISHED state.

During the ESTABLISHED state, data transfer will take place. This transfer is controlled by TCP’s micro state which includes the variables that control TCP’s flow and congestion control windows.

The state transitions that a third party passively monitoring the network traffic would observe differ from the view of an end point in three important respects:

1. The monitor is unable to observe interactions between either of the TCP stacks and their respective applications. The transition of a server from CLOSED to LISTEN can not be observed, the open command which triggers an active open can not be observed and neither can the close commands which triggers the sending of a FIN.

2. A number of transitions that are atomic for the client or the server are not for the monitor. For example when the server receives a SYN it replies with a SYN_ACK and moves into the SYN_RECEIVED state. For the monitor to track this transition it needs to introduce a new state called here SRV_SYN_ACK, which indicates that the monitor has seen the SYN and is waiting for the SYN_ACK reply before it can be certain that the Server’s transition to SYN_RECEIVED has been made. Similar issues pertain with the server transitions from ESTABLISHED to CLOSE_WAIT.
3. The monitor is positioned on the LAN/WAN interface. Hosts that are located in the LAN or collection of WANs we shall refer to as Home hosts, hosts connected on the other side of the WAN we shall refer to as Foreign hosts. In tracking the development of the TCP connection the monitor associates itself with the connection which is in the home network. It would be possible to track the evolution of both sides of the connection but this would add complexity and yield little new information.

Figure 3 shows the state transition diagram for a third party monitor tracking traffic to and from a home network. The CLOSED and LISTEN states have been merged as there is no way of distinguishing between them. A number of new temporary states have been introduced which indicate where the monitor having observed an input that would stimulate a state change is waiting for the appropriate output to be observed before concluding that the transition has been successful. These are necessary as the input might be dropped by the network or become corrupted between being observed by the monitor and arriving at the appropriate home machine.

4. DEFENDING AGAINST SYN FLOOD ATTACKS

Having presented an architecture for passively monitoring the evolution of TCP connections in this section we describe how this can be utilised for detecting a DoS attack. For the sake of brevity we will focus on one sort of attack the SYN Flood. As described above a SYN flood works by tying up the resources of the server in lots of uncompleted TCP connections. The approach outlined here focuses on protecting the local network in the here and now and has the advantage of requiring no changes to the protocols of attached hosts.

With the monitor in place and tracking the evolution of TCP connections detecting a SYN flood attack is straightforward. A connection which is participating in a SYN flood will first show a SYN packet arriving. The server will reply to the SYN with a SYN ACK. The original SYN was probably a forged packet, so the SYN ACK will never arrive at the source of the original SYN and the server will be stuck in the SYN_RECEIVED state. However, all server side TCP connections should pass through the SYN_RECEIVED state. In addition as the Internet provides unreliable transport it will be normal for some of the SYN_ACK packets to be lost, meaning that it will be normal for some connections to stay in the SYN_RECEIVED state for a considerable period of time. At the level of a single connection it will therefore be difficult to distinguish between normal connections that have been delayed and those connections that are part of a SYN Flood attack. Consequently the monitoring framework needs to consider aggregates of connections in order to detect an attack.

To do this we propose a host layer. At this layer there is an entry for each host on the home network. This list is created dynamically by observing traffic to and from hosts in the network. For each host a list of pointers is maintained for open connections. Every time there is foreign activity on a connection its pointer is moved to the back of the queue. Associated with each pointer is the state that a connection is in. A SYN Flood attack can be detected by observing when the number of connections in the SYN_RECEIVED state moves above some threshold, the threshold should be calculated using two components some absolute number of connections and some proportion of open connections. The exact algorithm can be adjusted as a result of experimentation, which is currently underway.

Upon detection of a DoS attack the framework outlined above and the monitoring of traffic means that the monitor has at its disposal the information required to mount an effective defence. In the absence of an attack only passive monitoring is undertaken. There is no interference with the normal workings of network traffic and consequently no degradation in the service provided by the network. In the presence of an attack the monitor will move from this passive mode into an active mode. The aim is to ensure that a core service is maintained even if it has to be degraded in some ways in order to cope with the attack. In the face of an attack we identify three actions; to clear existing bad connections, to filter new connections and to pushback defence towards the source(s).

Clear: The first action is to clear out all the existing connections that have got stuck in the SYN_RECEIVED state. This can be achieved by spoofing reset packets for the relevant connections and sending them to the home hosts. This will result in the hosts closing the relevant connections and the resources that were associated with them being freed.

Filter: The second action is to try and prevent bogus requests from getting through to servers on the home network, whilst allowing genuine ones to complete. One approach would be to limit the number of
connections that are allowed from a foreign host. The monitor could intercept SYN packets and only forward them if there were no existing SYN_RECEIVED connections or when any existing connections became cleared. The problem with this is that if the SYN packets are being spoofed then whilst they might all be coming from a small number of clients there is no way of telling from where each individual packet comes from. Therefore all SYNs would have to be intercepted with the consequence that the monitor becomes the bottleneck in the system.

It is difficult for a host which forges a packet to engage in secondary interactions with the server. To do so it needs to know what sequence number to acknowledge and this is difficult to guess without observing traffic from the server. Thus it can be surmised that SYN packets purporting to be from foreign hosts that have engaged in sending and receiving data from the server in a TCP legal manner are unlikely to be part of the SYN Flood attack. These packets could be allowed through. What if it turns out that the packets are part of the SYN attack? They will be entered in the queue of connections maintained by the monitor and if no activity occurs will be removed and a reset sent to the server. Thus although it is difficult to identify attack traffic it is possible to use measurements to identify traffic that is not attack traffic and treat it preferentially. Traffic from hosts or networks that have a history of using the service will be allowed through.

There are likely to be genuine hosts wanting to connect to the server in a legal manner who have not connected before. One option would be to simply disallow this on the grounds that it is acceptable to limit the service given whilst a DoS attack is taking place and that its duration is likely to be limited in any case. However, it is possible to allow new connections in a controlled manner.

All SYN packets from unknown sources can be intercepted and held in a queue for processing. When a packet reaches the front of the Queue it will be forwarded to the appropriate server in the normal way. The connection will however still be monitored and treated as suspect. If there is no response to the servers SYN_ACK before a second SYN_ACK is generated the SYN will be deemed to have been part of the attack and a reset generated to close the connection. When one of the connections moves beyond the SYN_RECEIVED state the source host is deemed safe and all future connections from that host are treated as safe requests.

**Pushback:** A further action for the monitor is to deduce from observing traffic at the destination network what the upstream routers should do to control the DoS traffic. At its simplest this may involve rate limiting SYN packets. Intermediate routers do not therefore have to monitor all traffic. When they are told an attack is happening and what they should do about it, they act, but only on the victim’s traffic under the victim’s instructions.

In this way the service can be maintained for all genuine hosts in the face of a SYN Flood attack. It is guaranteed that the server will not run out of slots for connections. Unknown hosts may experience delay in setting up their first connection, but subsequent connections should be unaffected. Known hosts should receive their normal service. Some SYN flood packets will get through, but will not seriously deteriorate the service provided and mechanisms are in place to quickly clear up the connections. In the absence of a SYN Flood attack the service provided by the servers will not be interfered with in anyway.

### 5. EXTENDING THE APPROACH

The use of passive monitoring and active defence can be extended to other sorts of DoS attacks. The sending of TCP packets that are not part of a valid connection can be detected as the monitor is already tracking all connections in and out of the network. Once an attack of this nature is detected such packets can easily be blocked and discarded.

During the lifetime of a connection it is possible for a third party monitor to track TCP’s micro state which is used to control the flow of data in both directions. This has been used to make measurements of the Quality of Service being received by streams. The Round Trip Time, RTT, can be measured by observing the time between a packet being sent and its acknowledgement being returned. Subtleties such as delayed acknowledgements are accounted for. Keeping count of sequence numbers allows repeat packets to be identified and therefore for packet loss to be detected. TCP’s effective window size can also be ascertained by observing the amount of data sent per RTT. In addition the evolution of TCP’s congestion window can be tracked as the behaviour of TCP’s congestion control algorithms are well understood.

This tracking of TCP’s micro state would be useful for identifying DoS attacks that work by ignoring
TCP’s congestion control algorithms and sending large amounts of data through a connection. Such connections could be identified and terminated. DoS attacks which took TCP into the ESTABLISHED state or left a connection dangling in the half closed position could also be identified.

The detection of and defence against Ping floods would be relatively easy for a monitor to achieve. The volume of Ping traffic would indicate an attack. Upstream routers can be signaled to temporarily drop or rate limit such packets.

The defence against UDP floods would be more difficult to achieve as the transport headers do little more than identify the port to which the packet is headed. It would be possible to define ports and hosts that expect to receive UDP packets. If there are then sufficient UDP packets on undefined services to have a detrimental effect on other traffic they could be shut out. This approach has the advantage of not imposing a ban on undefined post usage as is common with many firewalls. Instead the monitor is able to react intelligently to network usage patterns. An alternative or complementary approach would be to assign a bandwidth share to UDP traffic and define a policy for blocking traffic if the aggregate exceeds this limit.

6. CONCLUSIONS

It has been argued that a passive monitor observing traffic entering and leaving the target network is best placed to build profiles of the regular and attack traffic and to recognise the start of an attack. This monitoring is best done in the destination network as here the most complete view of both DOS and regular traffic. The monitor is in a position to deduce a set of rules that will allow attack and regular traffic to be distinguished. These rules can then be used to configure the Firewall or ingress router to drop attack traffic for the duration of the attack. However, once DoS traffic has reached the target network it is already doing damage. Ideally attack traffic would be dropped at the source network before it gets a chance to consume resources, but the dispersion of administrative authority and likely dispersion of attack sources means a proportion of traffic is likely to escape the source networks. There is therefore a need for intermediate routers to throttle attack traffic. Consequently, rules need to be securely communicated to routers and should facilitate the economic dropping of a high proportion of DoS traffic whilst minimising the workload for the router and disruption to regular traffic.

This paper has described how this approach can be applied to the role of detecting denial of the common SYN flood attacks. It has been shown that tracking the state of a TCP connection and keeping count of the number of connections in each state allows the monitor to detect when a SYN flood is taking place. By keeping a record of hosts that have legally communicated with the network it is possible to distinguish between likely attackers and friendly clients. This information can then be used by the monitor to allow friendly connections through and filter possible attackers. In this way service can be maintained and the attack defeated. It has been argued that similar techniques can be used to defend against other types of DoS attacks.

REFERENCES


Cisco IOS Firewall intrusion Detection System.

Cisco Strategies to protect against distributed denial of service attacks (DDoS): 7.


CSI/FBI (2003). COMPUTER CRIME AND SECURITY SURVEY CSI/FBI.


END TO END DEFENCE AGAINST DDOS ATTACKS


