Bleeding Edge DDoS Mitigation Techniques for ISPs

Vivek Ramachandran[1], and Sukumar Nandi[2]

[1]Cisco Systems, Inc. Bangalore, India
vivramac@cisco.com
[2]Indian Institute of Technology, Guwahati, India
sukumar@iitg.ernet.in

Abstract.

Distributed Denial of Service (DDoS) attacks are increasing plaguing the Internet since their first big appearance against Yahoo in the year 2000. Using thousands of compromised slave/zombie machines, DDoS attacks are capable of attacking and tearing down the Internet's backbone thus forcing all communication across it to a grinding halt. The early DDoS attacks started as "Script Kiddie pranks", has now evolved to an organized digital crime, targeting networks of government and business establishments, with motives ranging from defamation to extortion. This paper presents various cutting edge practical countermeasures, which an Internet Service Provider (ISP) should adopt to minimize damages inflicted by DDoS attacks. It also provides a detailed study of the latest bleeding edge solution called Traffic Scrubbers. In course of this paper we discuss advantages and drawbacks of these mitigation techniques and outline a set of industry best practices, which should be followed in order to be able to mitigate DDoS attacks and minimize the casualties caused.

Keywords: Security, DDoS, ISP, Blackholing, ACL, Traffic Scrubber, Netflow, Best Practises.

1. Introduction

A Denial of Service (DoS) attack by definition is any attack, which denies a particular service or resource to legitimate users. A Distributed Denial of Service (DDoS) attack is a very large-scale coordinated attack aimed at disrupting the use of a resource or service by its legitimate users. According to the Computer Incident Advisory Capability (CIAC) first DDoS attacks were seen in the year 1999 [1]. The first big commercial impact of DDoS was felt when the popular Internet site Yahoo.com was made offline for almost two hours, leading to huge revenues losses for Yahoo [2]. DDoS attacks have now become a very regular affair on the Internet with dozens of sites being brought down every week. The motivation behind these attacks has drastically transitioned from “casual play” by script kiddies to “extortion demands” by organized professional group of hackers.

In the most general form a DDoS network consists of the Attacker, the Master and the Slave machines (also called Zombies). The Attacker controls the Masters, which in turn control the Slaves. The hierarchy for control and command execution is pyramidal in architecture with the Attacker at the apex, Masters in the middle tier and Zombies at the base. The Attackers mostly use well known publicly released software exploits to compromise Master and Slave machines. Once compromised, DDoS tools are installed on these captured machines. After compromising a few thousand of such hosts, the Attacker gets ready to launch a DDoS attack. During an attack, the Attacker commands the Master machines, which in turn command the Slaves to start attacking

the Victim. DDoS tools installed on these machines are capable of orchestrating a variety of attacks. We now describe some of the most widely deployed DDoS tools [15] such as Trin00, TFN, TFN2K etc.

1.1. Trin00

Trin00 daemons are deployed on Solaris and Linux systems. They use the Master-Slave architecture as discussed above. These compromised systems also have a root kit installed to hide the presence of Trin00. Master-Slave communication is over hard coded TCP (1524, 17665) and UDP (27444, 31335) ports. The Trin00 tool does not use IP spoofing and only sends UDP traffic during an attack. Attacks are initiated to random UDP ports on the victim.

1.2. Tribal Flood Network (TFN)

TFN is a powerful improvement over Trin00 and incorporates four types of attacks [14] viz. UDP flood, TCP SYN flood, ICMP flood and Smurf attack. It is to be noted that all these attacks support IP address spoofing. The Master maintains an IP list of all slaves reporting to it. This IP list is encrypted using the Blowfish algorithm. The Attacker controls the Masters using various communication methods such as SSH terminal sessions, telnet session and remote shell bound to a TCP port. Masters in turn communicate with Slaves using ICMP Echo Reply packets with attack commands set in the 16-bit IP Identifier field (ID).

1.3. Tribal Flood Network 2000 (TFN2K)

TFN2K is a huge leap in sophistication over its predecessors. The source code can be easily ported to Linux, Solaris and Windows operating systems thus make most computers vulnerable to abuse as Zombies. Apart from supporting attacks used in TFN, Targa3 and Mix attacks were introduced in this tool. Targa3 attacks use random values in the header fields to produce malformed IP packets which might cause IP stacks to crash due to improper protocol handling and Mix attacks consist of sending UDP, TCP SYN and ICMP packets in a 1:1:1 relation. Communication between the Attacker and Masters uses a randomly chosen protocol (UDP, TCP or ICMP) and the attack vector data uses its own protocol called “Tribe Protocol”. The Tribe protocol is CAST-256 [16] encrypted and base64 [17] encoded. Also all passwords are generated on demand at compile time. The only weakness is that because of base64 encoding regardless of the protocol and encryption algorithm there is a sequence of 0x41(the character ‘A’). Actual count of the character ‘A’ might vary but there will always be at least one. This is used as a fingerprinting technique to stop the tool’s communication channel.

1.4. Stacheldraht

Stacheldraht (German for “Barbed Wire”) is an enhancement over Trin00 and TFN. This tool uses encrypted communication between the Attacker and Masters. The most notable improvement seems to be an automated update for agents, which allows the Attacker to periodically fix bugs and add enhancements to this already powerful tool. Automatic update feature uses the “RCP” command i.e. Remote Copy on port 514/TCP to update Slaves. Masters and Slaves also communicate via ICMP Echo Reply packets making it very difficult to filter them, without posing a risk of breaking Internet applications, which use ICMP Echo packets. When the Slaves startup, they contact their Master by locating its IP address in a configuration file or use the default Master’s IP hard coded in their binary, if the configuration file is not found. Slaves then send ICMP Echo Reply packets

to their respective Masters with the ID field set to 666 and the data field containing string "skillz". Masters in turn respond with string "ficken" and the ID set to 667. Fortunately as all these strings are sent in plain text and with no authentication it becomes very easy to locate Slaves and to hijack them.

Apart from the popular Master-Slave architecture described above a Botnet Architecture is often used as well. In the Botnet Architecture, the Attacker uses an Internet Relay Chat channel to communicate with his Slave machines. To communicate, both Attacker and Zombies connect to the same IRC Channel. This is almost always a secret channel with encryption so that ordinary users cannot snoop into their conversation. The Attacker now sends attack parameters e.g. Victim IP etc out to the Slaves through IRC and the Slaves comply with the Attacker's request. As IRC is a very popular mode of communication used by legitimate users as well, it is almost impractical to totally firewall this traffic out. Thus IRC provides a very "secure" channel for DDoS control traffic as it blends in with the IRC traffic generated by legitimate users in the network. GTbot [3] is an example of a Bot software used to command such Botnets.

We can broadly classify all DDoS attacks into either Bandwidth based DDoS attacks or Resource Exhaustion DDoS attacks. In Bandwidth based DDoS, the Attacker's goal is depleting the entire Internet bandwidth available to the Victim. This is accomplished by sending a huge flood of packets to victim thereby saturating its entire bandwidth to the Internet. Examples of Bandwidth based DDoS are ICMP and UDP floods[14]. The Resource Exhaustion attacks are targeted at exploiting vulnerabilities in the network stack of the Victim. Examples of this attack are sending TCP SYN floods or malformed packets. These packets cause all resources on victim to be locked up and exhausted thus making them unavailable to legitimate users.

It is to be noted that DDoS attack techniques are constantly evolving along with mitigation techniques. This has lead to a rat race between Attackers and Security community to keep them one step above the other. We will now discuss current mitigation techniques available to ISP's around the world to combat DDoS attacks.

2. Cutting and Bleeding Edge DDoS Mitigation Techniques

DDoS is one of the primary concern of ISPs as they have to strictly adhere to a minimum quality of service that they guarantee to their customers. A loss of Internet connectivity because of a DDoS attack for a customer on a regular basis might cause a loss of customers and hence revenue. The first step an ISP needs to take in mitigating a DDoS attack is to identify "attack traffic" and points (network devices) through which it is entering the network. ISPs today use a variety of techniques to detect, classify and mitigate attacks. We will now discuss these techniques in detail.

2.1 Attack Detection and Classification using Netflow

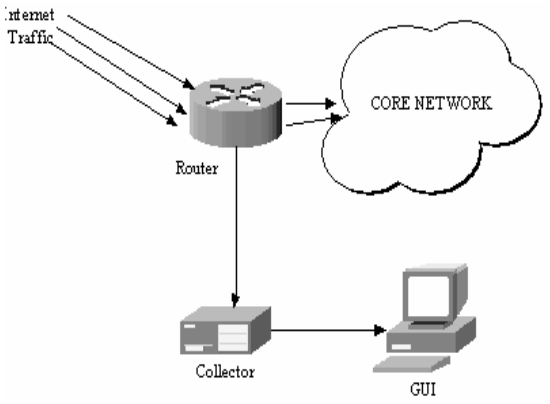

Figure 1: A Netflow Network Architecture

Netflow is a form of network telemetry and accounting technology available on routers. Netflow is used to provide application layer visibility and detailed traffic information. It allows administrators to track IP flows across their network. An IP flow in Netflow's terms is defined by seven unique keys namely Source IP, Destination IP, Source port, Destination port, Layer 3 protocol Type, Type of Service Byte and Input logical interface on the device for packet. Netflow version 5 onwards, other relevant information such as packet counts, byte count, output interface, next hop address, source and destination autonomous system number etc are also available. Figure 1 shows a typical setup where individual routers export the Netflow data they collect to a central Collector [4]. The Administrator can analyze the data from a Graphical User Interface (GUI) [5]. These exported Netflow packets are around 1500 bytes in size and contain around 20-50 flow records per packet. Netflow provides an insight into IP flows across a network, thus allowing for traffic correlation across devices.

As Netflow offers an easy way of visualizing traffic in the form of flows, one can detect DDoS attacks by constant monitoring of exported data. The most widely used technique for detecting attacks using Netflow is to monitor the number of flows on every device. If at any point in time current number of flows exceeds the normal average number of flows expected to be seen on that device, then it can be concluded that an attack is underway. The above algorithm is shown in Figure 2. The values of *X, Y, Z* and *N* are empirically chosen and depend on the normal traffic seen on the concerned device. Netflow is widely used to detect such sudden changes in flow statistics or what is called "Flow Anomaly".

Once the Administrator is aware that his network is under attack, he can also use Netflow to categorize attack traffic. Doing this using Netflow is very simple as the problem is almost always reduced to classifying flows to a single destination IP address i.e. Victim of DDoS attack. Once we have detected the characteristics of the attack traffic the Administrator can employ other techniques such as ACLs, Blackholing etc to mitigate the attack. We will discuss these techniques later in the paper.

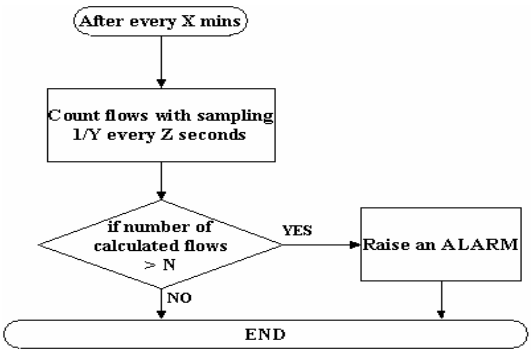

Figure 2: Detecting DDoS using Netflow

2.2 Classification and Mitigation of Attacks using ACLs

Access Control Lists (ACL) are rules, which can be applied on a router or switch to filter unwanted traffic. To fight a DDoS attack we first classify and fingerprint attack traffic using ACLs and then add a rule to explicitly drop all traffic bearing the same characteristics. To fingerprint attack traffic we add various ACLs corresponding to different types of traffic (ICMP, TCP, UDP etc) on interfaces. ACLs provide a counter which shows the number of packets which match the rules for individual ACLs on an interface. During a DDoS attack one can easily classify attack traffic by periodically checking the counters of the different ACLs. While detecting and classifying the attack we set ACL action to permit. Once we have detected and fingerprinted the malicious traffic we explicitly set the matching ACL action to deny. This will drop all the malicious traffic. It is to be noted here that

ACLs are a CPU consuming action and under heavy packet floods might degrade performance of the router. Also categorizing complex DDoS attacks with ACLs, where the attack traffic might vary with time would be a bad idea, as ACLs require human intervention.

2.3 SNMP and Remote Monitoring (RMON)

Simple Network Management Protocol (SNMP) is a widely used protocol to query real time statistics from network devices. SNMP tools such as Net-SNMP [6..8] provide a good platform to collect and analyze device statistics almost in real time. SNMP Management Information Base (MIB) support polling of information on a device ranging from chassis temperature, CPU utilization, bandwidth consumed per interface, packet speeds etc. A lot of this information can be used to detect DDoS attacks by identifying any anomalous behavior as a possible indication. A high CPU utilization or a large number of packets coming into an interface on an edge router could be a possible sign of an attack. Thus SNMP is a handy tool to detect DDoS.

RMON is a standard, which defines how a set of network remote probes or agents relay networks traffic information; they individually collect to a central location for further analysis. RMON is not as popular as SNMP or NETFLOW and normally uses raw traffic via SPAN/RSPAN and generates statistics for further analysis. These statistics can be analyzed periodically for signs of an attack.

2.4 RFC 2827/BCP 38 Ingress Packet Filtering

DDoS attacks almost essentially use IP address spoofing to obfuscate the real location of Attacker and Zombie machines. RFC 2827/BCP 38 [11] was drafted as a counter measure to prevent IP address spoofing. The goal of this filtering is to make sure that all packets leaving a network should be sourced from a valid allocated address space, which is consistent with topology and space allocation. To counter IP spoofing it is highly desired that devices filter packets as close to the concerned edge as possible and filter based on both source and destination addresses. We will now discuss the various measures, which can be implemented, on a network to be able to concur with the draft.

2.4.1 Unicast Reverse Path Forwarding (uRPF)

Unicast Reverse Path Forwarding [12] is a technique used to drop packets with forged IP addresses on routers. uRPF uses the principle of symmetric routing based on which a packet is expected to arrive on an interface which is on the best return path for that source address, on that router. Depending on the network topology one may configure either Strict or Loose uRPF checks. When a Strict uRPF is configured on a router, only packets arriving on a router interface, which is the best return path interface for the source address of the packet are forwarded and all other packets are dropped. In Loose uRPF, a packet is forwarded if there exists at least one interface on the router, which is the best return path for that source address. If such an interface is not found on the router matching the above criterion then the packet is dropped. uRPF is most applicable to single homed environments where there is a single upstream channel i.e. symmetric routing. uRPF produces best results if it is applied as much downstream as possible. As IP address spoofing is a major part of any DDoS attack, an Internet wide deployment of uRPF would be very

effective in thwarting this. It is however cautioned that uRPF should not be applied to networks which have asymmetric routing topologies i.e. networks having multiple routes to same source, as this would result in wrongly dropping packet instead of forwarding it.

2.4.2 IP Source Guard

The IP source guard [13] feature prevents IP spoofing at the access edge itself. This feature is available on a variety of switches. This works along with DHCP Snooping [13] to detect IP spoofing. When a switch port comes up all traffic on it is blocked except DHCP traffic. Once the host connected to the switch port receives a valid IP address through a DHCP server a Port Access Control List (PACL) is set on that port. This PACL will only allow packets sourced from the valid IP address assigned previously by DHCP, to communicate through this port. The Switch keeps track of the valid IP address binding for that port through a process called DHCP Snooping. In DHCP Snooping the switch peers into DHCP traffic passing through that port and keeps track of the DHCP server assigned IP address in a special table. If traffic is seen from any other address other then the learnt one in its table, an alarm is raised and the appropriate configured violation action is taken. This technique succeeds in mitigating IP spoofing totally at the access edge. As a worst case violation measure the Administrator can configure that the switch port be shut down and the offending host be denied all network access.

2.4.3 Access Control Lists (ACLs)

ACLs can also be used to drop spoofed packets. Using ACLs an Administrator can enable ingress traffic filtering on a network. In the simplest form of implementation the Administrator needs to configure a static ACL which permits all traffic from source addresses belonging to the allocated block for a given network and deny all other source address traffic.

In some cases it is also desirable to allow traffic to only certain ports on some hosts from the outside world e.g. Port 80 on a Web server.

An Internet wide deployment of IP spoofing mitigation techniques such as the above will help curb this problem totally. This in turn will help in locating the Attacker and his Slaves faster and thus help mitigate the attack sooner.

2.5 Device Security

Router security is one of the key issues in securing a network. The easiest DDoS attack would be to take over all Routers on the network and configure them to drop all traffic. There are many things one must remember to do in order to “harden” the router’s configuration. Foremost thing one must remember is that Routers ship with factory defaults (default configurations, passwords etc) making them inherently insecure. It is advisable to “harden” the router configuration before placing it on a live network. In the process of hardening one must ensure that all unnecessary services should be shut down e.g. finger, http etc. Also one should be cautious when allowing the use of protocols such as Cisco Discovery Protocol (CDP) between devices on the network, as they convey valuable device and network information to the requestor. On the networks where SNMP is being used for device configuration, it would be advisable to shift to SNMP version 3 as it offers authentication and encryption. SNMP v1 and SNMP v2 are known to have widely publicized vulnerabilities. The Administrator should also be careful to make access to routers available only from

trusted sources by judiciously enabling password protections at every possible authorization level. The preferred communication medium to connect to the router should be an encrypted one with the use of protocols such as SSH instead of Telnet. The Administrator should also configure local user level passwords on the router and also authentication through a AAA [20] server such as RADIUS, TACACS+ etc is highly recommended. Also one must be alert enough to patch Routers as soon as any new vulnerability disclosure is made by the Router's vendor. Router security is an important building block in securing a network against DDoS.

2.6 Blackholing and Remote Trigger Blackholing (RTBH)

Blackhole filtering or Blackhole Routing is a popular technique used by ISP's to drop undesired packets on their routers. The Administrator can configure static routes on routers to drop all packets destined to a particular IP address. To achieve this the Administrator can setup a static route for the IP address, packets destined to which are desired to be dropped and configure it to send it to Null0. The Null0 is actually a special interface on routers. A packet sent to Null0 implies that the packet be dropped. These packets are dropped based on their destination address in router hardware, thus having almost no performance impact on routers. This technique is also called "Route to Null0". During a DDoS attack the Administrator can configure a "Route to Null0" or "Blackhole" for the destination IP address (IP of Victim under DDoS attack) on edge routers. This will ensure that all traffic destined to the Victim is dropped at the network edges thus lowering impact of DDoS attack on the whole network. The disadvantage of this technique is that even legitimate good traffic is dropped along with bad traffic thus making the Victim inaccessible through the Internet. This technique is a drastic measure, which has to be taken often to save the rest of the network from DDoS attack.

Remote Trigger Blackholing uses the Border Gateway Protocol (BGP) to trigger a network wide "Blackholing" response to an attack. This technique works by exploiting the fact that BGP allows arbitrary next hops to be defined administratively. This technique requires that a static route to Null0 be configured for a "Reserved IP" [10] address on all edge routers (Blackholing). We then designate one of the routers on the network as the "Trigger Router". This router will be used to trigger a network wide BGP update. During a DDoS attack targeting a victim with IP address X on the network, we will configure a static route for IP address X on the Trigger Router and administratively force the next hop to be configured to the "Reserved IP address" we had earlier set network wide, to be routed to Null0. This change in configuration on the Trigger Router will cause a network wide iBGP (Interior BGP) update to be sent. All the routers in the network will update their routing tables accordingly. This will in turn cause all packets destined to IP address X to be dropped on all routers, thus mitigating DDoS attack at the network edges. It is to be noted that this technique does not impose the usage of BGP as routing protocol for the network. It just requires deployment of an iBGP mesh, internal to the network and need only contain the Blackholed addresses in its table. The routing on the network can use any protocol it wants e.g. Route Information Protocol (RIP) and coexist with these Blackholed routes used by BGP. RTBH is actually an improvement over Blackholing, making it possible to trigger a fast network wide response to an attack.

2.7 Sinkholes

Sinkholes are a topological feature use to divert attack traffic to a dedicated network, which can withstand the attack. The traffic directed to the Sinkhole can be used for further analysis and classification using ACLs, Sniffers and Network based traffic anomaly detectors. In a typical installation the sinkhole will advertise routes for the IP address range under attack to all routers on the network. After receiving the routing update routers redirect all traffic destined for the Victim IP address range to the sinkhole. Routing updates are done using a CIDR[18] advertisement for the block of IP addresses under attack. Once traffic enters the Sinkhole it is thoroughly scrutinized for bogus scans i.e. packets from unallocated IP addresses, allocated and announced but unused IP address spaces and for RFC 1918 addresses. Results of detailed analysis from the Sinkhole can be used to fingerprint and classify attack traffic. Also one can try back tracing the source of the attacks by using the Backscatter data (this technique is described in the next section) attracted to the Sinkhole.

2.8 Back Scatter

The Backscatter technique uses the RTBH and Sinkhole architectures to provide a better view of the ongoing attack. When a network with RTBH deployed, comes under a DDoS attack, the RTBH technique sets next hop for the Victim IP address under attack, to be sent to Null0. This causes packets to be dropped but at the same time the router sends an ICMP Unreachable error message to source address of the dropped packets. To be able to collect some of these ICMP messages we use the Sinkholes to advertise for a large block of unallocated space (after referring to [10]).

If we assume that the source IP addresses in the spoofed packets are randomly generated then the probability that a given host on the Internet will receive an unsolicited response from victim of DDoS attacks is:

$$E(X) = \frac{1}{2^{32}}$$

If the Sinkhole advertises N distinct IP addresses and the network receives M attack packets then the probability of receiving a packet in the Sinkhole is

$$E(X) = \frac{MN}{2^{32}}$$

One should remember that these prefix advertisements made by the Sinkhole should be confined within the local network and should not be exported to other ISPs. This can be easily done by using BGP's no export option and by the use of BGP egress routing filters. Once the Sinkhole receives all these backscattered ICMP packets it can construct the entry points for the spoofed IP addresses into the local network. Analysis of source addresses in these ICMP packets helps us identify if the attack is coming from internal or external sources to the ISP's network.

Effectiveness of the Backscatter technique depends on the fact that attacker will spoof IP addresses uniformly across the entire address space and that all unsolicited packets received by the Sinkhole are backscatter traffic. We will now describe the bleeding edge solution of Traffic Scrubbers, which is a huge improvement over all the above-mentioned techniques.

2.9 Traffic Scrubbers

All mitigation techniques discussed above involve dropping all traffic destined for the Victim of the DDoS. This saves the rest of the network from being impacted from the DDoS but the Victim loses total connectivity to the outside world. Thus the Attacker actually, accomplishes his goal of denying services on the Victim to legitimate clients. To combat this problem industry research

gave way to the innovation of "Traffic Scrubbers". These Scrubbers have capabilities, which allow them to distinguish between good and bad traffic. They mitigate DDoS attacks by forwarding only good traffic and dropping attack traffic.

Traffic scrubbers isolate good traffic from bad traffic by using various protocol specific features to "authenticate" TCP and UDP traffic. Here "authenticate" means to be able to identify clearly the packets coming from legitimate sources (good traffic) and those coming from spoofed sources (bad traffic). In all these techniques the Scrubber authenticates the traffic first and then after successful authentication, allows a normal flow between two end hosts for the entire lifetime of this session. The TCP Intercept [19] is one such technique where the router does in line source address validation of all clients requesting a TCP connection to any host in the inside network.

The TCP Intercept technique works in two modes viz. Intercept and Watch. In the Intercept Mode, the Router intercepts the TCP SYN packet destined for an inside host and sends a TCP SYN/ACK on its behalf. If the connecting host responds back with a TCP ACK packet for TCP SYN/ACK sent by the router, thus completing the TCP three way handshake, then the Router initiates a connection to the inside host and transparently proxies all data between these two hosts. In case the originally sent TCP SYN was spoofed then the three-way handshake will not be successful. In such a case the router will either receive no response or a TCP RST packet. In such a scenario the Router shields the internal host from the attack. This technique ensures that all spoofed TCP traffic will be dropped and will never reach protected hosts in the internal network. This unfortunately still does not stop an attack where the Zombie machines might risk getting detected but use their IP addresses to initiate a legitimate connection. This technique has an additional overhead of the Router having to act as a proxy for these connections. In the Watch mode the TCP Intercept mechanism passively watches TCP half open connections and will actively close connections on the internal hosts after a configurable length of time. The Traffic Scrubbers internally employ techniques like the TCP Intercept to clean the attack traffic.

We can broadly categorize Traffic Scrubbers based on their location on the network with respect to traffic flow into two types viz. "Inline" and "Diversion" model Scrubbers.

In the "Inline Model" the Scrubber is placed in the line of incoming traffic. It cleanses traffic and forwards good traffic to destination or to the next hop. An example of an Inline Scrubber is the DDoS Guard from Green Gate Labs [9].

In the "Diversion Model" all traffic destined for Victim of DDoS attack is redirected to the Scrubber. The Scrubber cleans all bad traffic and forwards good traffic to the destination host. An example of a Diversion Scrubber is the Cisco Guard XT[21] and CloudShield CS-2000 [22].

Among the products in the Traffic Scrubber product line, Cisco Guard XT currently leads the industry over others. We will now discuss the Cisco Guard in more detail to be able to provide an insight into architecture of Traffic Scrubbers.

The Cisco Guard XT and Anomaly Detector XT work hand in hand to detect and mitigate DDoS attacks. The Cisco Anomaly detector is placed on the network we desire to protect so that it can analyze traffic sent to hosts on the network. The Cisco Guard is placed upstream near the edge of the network, near border routers. On detecting a possible attack the Detector activates the Cisco Guard by sending the IP address and other information of Victim under attack to it over a secure channel. The

Cisco Guard now sends a BGP update message to the peer router to divert all traffic meant for the victim to it. The peer router now sends all traffic destined for the Victim to the Guard.

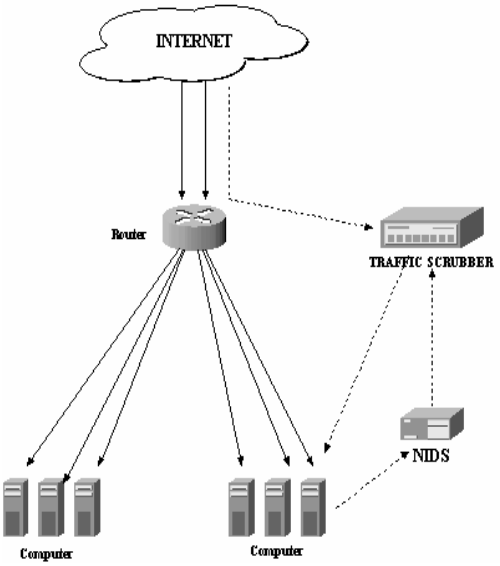

Figure 3: Diversion Traffic Scrubber

It is to be noted that rest of the traffic for the internal network still flows through the normal route i.e. through peer router to individual hosts. The Cisco Guard "cleans" traffic redirected to it using a patented Multi Verification Process, which uses various Anti-Spoofing techniques to differentiate legitimate user traffic from attacker generated traffic. From what has been made public, the Guard authenticates TCP traffic with a method similar to the TCP Intercept but may/may not act as a proxy depending upon the configuration options. The Guard also provides techniques, which can be used to authenticate HTTP and DNS traffic. The Guard drops all malicious traffic and the legitimate traffic is forwarded to destination. This ensures that the destination host only receives clean traffic and not the attack traffic thus allowing it to function normally. The Cisco Guard can be triggered using the Cisco Anomaly Detector or using an Arbor product called Arbor Networks Peakflow SP[23]. Arbor Peakflow SP uses Netflow data from across the network and analyses it in real time to detect attacks. If it detects an anomaly in the traffic to any of hosts it is protecting then it sends a trigger to the Cisco Guard. The Guard in turn cleans traffic as mentioned previously and allows normal functioning of host.

Traffic Scrubbers are definitely a huge leap for DDoS Mitigation technologies but are not fool proof. These Scrubbers are very effective against TCP based attacks but still a lot of progress needs to be made on the UDP front. The problem with UDP is the fact that it is a datagram service, so the only way most UDP protocols can be "authenticated" and labeled as "legitimate and non malicious" is to peer into application layer while a session is in progress and check for malicious activity. Even though Traffic Scrubbers provide a way to authenticate TCP traffic, an Attacker can launch a TCP DDoS attack even in the presence of a Scrubber by having thousands of his Slaves attack Victim by opening a legitimate connection simultaneously. The TCP authentication mechanism of the Scrubbers would allow all these packets to go through because the Zombies are completing TCP three-way handshake. Though this will expose the real IP address of the Zombie but the Victim host will succumb to the overwhelming number of connections made to it. To combat such an attack we use rate limiters like the Committed Access Rate (CAR) feature which we discuss in the next section.

2.10 Committed Access Rate (CAR)

CAR is a traffic rate-limiting feature working on Layer 3. This feature rate limits input traffic before forwarding it towards the destination. The advantage of this technique is it allows the Administrator to flexibly configure rate limits for various kinds of

traffic differently e.g. ICMP and TCP may have different rate limits configured. Using CAR rate limits can also be set on a per IP address basis. In a typical deployment CAR is used with BGP and this combination is referred to as Remote Triggered CAR. As DDoS traffic characteristics change dynamically with time, it is necessary to have a solution, which can allow for fast updates on routers to be able to successfully mitigate attacks. Using QoS Policy Propagation on BGP (QPPB) we can dynamically update our CAR rate limiter specifications on edge routers through BGP. CAR has proved to be one of the most important mitigation techniques to keep the amount of traffic coming into the network in check.

3. Industry Best Practices

The discussion in Section 2 has definitely brought out the point that there is no "one solution" for a DDoS attack but a "set of solutions". Mitigating a DDoS attack requires a careful design of the network much in advance of the attack. The choice of the mitigation technique depends upon the size of the network as well as the amount of investment one wants to make on security i.e. An RTBH will just leverage the existing Routers but a Cisco Guard XT will require an investment of a couple of thousand of dollars.

The best practices are guidelines, which if followed mitigate and greatly minimize damages caused by a DDoS attack attempt. For preventing becoming an unwitting aid to a DDoS attacker, Administrators should have a patch management system in place. As soon as a new security exploit is made public the Administrator should download patches from the vendor's site and patch all vulnerable systems. This will ensure that an Attacker cannot use publicly available exploits to hijack the network's computers and use them as Zombies. To be able to detect Zero day exploits, Administrators should deploy anomaly based network intrusion detection systems across the network. It is also a good practice to employ external network penetration testers from time to time to audit the network's security and patch problems, which they uncover (which the Administrator might have previously overlooked). To curb IP address-spoofing ISPs worldwide should deploy uRPF, ACL filtering, IP Source Guard etc counter measures as described in this paper.

As stressed throughout this paper the mitigation capabilities of a network against DDoS attack is determined by how well the network has been designed with a possibility of DDoS attack in mind. It is advisable that Netflow or SNMP be used for periodically polling routers for statistics so that a DDoS attack can easily be detected and fingerprinted at the earliest. Sinkholes and Backscatter techniques allow one to determine very quickly the nature and direction of the attack i.e. routers through which attack traffic is entering. RTBH should be deployed on networks which do not have a Traffic Scrubber in place or as a backup in case the Scrubber fails to do its job properly. If one can afford then a Traffic Scrubber should definitely be used to protect the network. A router with CAR enabled should be placed before the main core network to make sure that even if the Traffic Scrubbing lets a huge flood of seemingly legitimate packets through, one can still rate limit traffic before it reaches its final destination. It is very clear that the best practice is to deploy and have in place both detection and mitigation techniques while designing the network, well in advance of an actual DDoS attack attempt.

4. Conclusion

DDoS attacks are a growing menace on the Internet and are here to stay. Though the industry has come up with some innovative techniques to combat this menace, but unfortunately none of these solutions are fool proof. The best solutions still seem to be the ones where we drop or rate limit all packets destined to victim on the edge routers. Doing this ensures that other hosts on the network can still remain functional in the event of a DDoS attack but the victim of the DDoS is inaccessible to the outside world, which meets Attackers original objective. The bleeding edge solution provided by Traffic Scrubbers is definitely a step in the right direction but the technology is still far from mature and needs to address various issues like a generic UDP traffic authentication. Our concluding remark would be that more research is required to formulate a full proof solution to defeat DDoS attack but till that day comes, following the best practices will definitely help mitigate or at least minimize the casualties of a DDoS attack.

5. Acknowledgements

The first author wishes to thank Prof. Sukumar Nandi for constantly encouraging him and guiding him at every step. He would also like to thank Ms. Seema Nagabhushana and Ms. Tulasi from Cisco Systems, Inc. for proof reading the paper at such a short notice.

References

1. Paul J. Criscuolo. "Distributed Denial of Service Trin00, Tribe Flood Network, 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.
2. "Yahoo on Trail of Site Hackers", Wired.com, February 8, 2000. http://www.wired.com/news/business/0,1367,34221,00.html (15 May 2003).
3. GTbot , A DDoS tool http://golcor.tripod.com/gtbot.htm
4. The OSU flow tools, Mark Fullmer, http://www.splintered.net/sw/flow-tools
5. Flowscan Tool for Netflow data visualization, Dave Plonka, http://net.doit.wisc.edu/~plonka/FlowScan
6. Net-SNMP Tool, http://www.net-snmp.org
7. Multi router traffic grabber , Tobi Oetiker http://people.ee.ethz.ch/~oetiker/webtools/mrtg/
8. Round Robin Database Tool RRDTool, http://people.ee.ethz.ch/~oetiker/webtools/rrdtool
9. DDoS-Guard, Green Gates Guard, http://www.greengatelabs.com/
10. IP version 4 Address Space Assignment http://www.iana.org/assignments/ipv4-address-space
11. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2827, www.faqs.org/rfcs/rfc2827.htm
12. Unicast Reverse Path Forwarding www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/uni_rpf.pdf
13. DHCP Snooping and IP Source Guard http://www.cisco.com/en/US/products/hw/switches/ps708/products_configuration_guide_chapter09186a00801eb955.html
14. DDoS Resources, http://www.anml.iu.edu/ddos/types.html
15. DDoS Tools, Dave Dittrich, http://staff.washington.edu/dittrich/misc/ddos/
16. CAST-256 Encryption, RFC 2612, http://www.faqs.org/rfcs/rfc2612.html

17. Base 64 Encoding Scheme http://www.freesoft.org/CIE/RFC/2065/56.htm
18. Classless Interdomain Routing, http://public.pacbell.net/dedicated/cidr.html
19. The TCP Intercept Feature, http://www.sans.org/resources/idfaq/syn_flood.php
20. AAA Server Configuration Guide, http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7a7.html
21. The Cisco Guard XT and Anomaly Detector XT, http://www.cisco.com/en/US/products/ps5894/
22. The CloudShield CS-2000, http://www.cloudshield.com/what_we_do/cs2000.html
23. The Arbor Peakflow SP, http://www.arbornetworks.com/products_sp.php