Dos Attacks exploit the bandwidth unbalance that can exist between a single victim’s network and a pack of compromised machines/servers that are used maliciously to launch an attack. By using this bandwidth to flood the victim’s servers, an attacker can effectively make it difficult for legitimate traffic to go through, and, in the extreme cases, make the victim’s servers crash. It is important to note that DoS attacks do not exploit any network protocol weakness, other than the ones that let the attacker compromise his zombies in order to launch the attack.

A denial-of-service attack (DoS), which purports to deny a victim (host, router, or entire network) providing or receiving normal services in the Internet, Flooding-based distributed DoS attack, or simply DDoS attack, is another form of DoS attack, and is the focus of this article. do not rely on particular network protocols or system weaknesses. Instead, they simply exploit the huge resource asymmetry between the Internet and the victim in that a sufficient number of compromised hosts is amassed to send useless packets toward a victim around the same time.

The widely publicized DDoS attacks against Yahoo!, eBay, Amazon.com, and several other popular Web sites in February 2000 revealed the vulnerability of even very well equipped networks. In fact, DDoS attacks are prevalent events in the Internet today, but most of them go unreported.

launching a DDoS attack is made very easy by the availability of a number of user-friendly attack tools. On the other, there is still a lack of effective solutions to defend

two types of flooding attacks: direct attacks and reflector attacks,

direct attack, an attacker arranges to send out a large number of attack packets directly toward a victim TCP SYN packets is sent to a victim’s server Port, victim would respond by sending back SYN-ACK packets to spoofed random source address, then repeat SYN-ACK until giving up UDP packets may also be used to achieve the same result. In these cases, the victim usually responds with the corresponding ICMP reply and error messages, and UDP packets

Before launching a direct attack, an attacker first sets up a DDoS attack network, consisting of one or more attacking hosts, a number of masters or handlers, and a large number of agents (also referred to as daemons or zombies

Each attacking host controls one or more masters, and each master in turn is connected to a group of agents.

A reflector attack is an indirect attack in that intermediary nodes (routers and various servers), better known as reflectors, are innocently used as attack launchers. An attacker sends packets that require responses to the reflectors with the packets’ inscribed source addresses set to a victim’s address.

any protocol that supports this type of “automatic message generation” can be exploited to launch reflector attacks,

When SYN-ACK packets are involved, the reflector in fact behaves like a victim of SYN flooding attacks, because it also maintains a number of half-open connections.

attack packets inscribed with inactive destination ports would trigger hosts to send ICMP port unreachable messages. Attack packets with very small time to live (TTL) values would trigger routers to send ICMP time exceeded messages.

The attack sources (the reflectors) could also be more dispersed on the global Internet, because the attacker does not need to implant agents to launch such attacks. Moreover, the reflected packets are in fact normal packets with legitimate source addresses and packet types, which therefore cannot be filtered based on address spoofing or other route-based mechanisms.

three lines of defense against the attack: attack prevention and preemption (before the attack), attack detection and filtering (during the attack), and attack source traceback and identification (during and after the attack).

In the past, most work in this area has focused on tolerating attacks by mitigating their effects on the victim

determining the source generating attack traffic is surprisingly difficult due to the stateless nature of Internet routing.

ATTACK PREVENTION AND PREEMPTION

hosts may be securely protected from master and agent implants or monitor network traffic for known attack messages sent between attackers and masters active side, cyber-informants and cyber-spies can be employed to intercept attack plans. –gay

ATTACK SOURCE

TRACEBACK AND IDENTIFICATION

identifying the actual source of any packet sent across the Internet without relying on the source information in the packet. There are generally two approaches to the IP traceback problem. One is for routers to record information about packets they have seen for later traceback requests [8]. Another is for routers to send additional information about the packets they have seen to the packets’ destinations via either the packets [9] or another channel, such as ICMP messages.

IP traceback is ineffective in reflector attacks in which the attack packets come from legitimate sources. => might block an AS

ATTACK DETECTION AND FILTERING

The detection part is responsible for identifying DDoS attacks or attack packets. After identifying attack packet flows or attack packets, the filtering part is responsible for classifying those packets and then dropping themThe false positive ratio (FPR) and false negative ratio (FNR) can quantitatively measure the effectiveness of the attack detection. FPR = #False P / # Real P

The effectiveness of packet filtering, on the other hand, refers to the level of normal service that can be maintained by the victim during a DDoS attack by filtering the attack packets.

effectiveness of packet filtering can be measured by normal packet survival ratio (NPSR), which gives the percentage of normal packets that can make their way to the victim in the midst of a DDoS attack.

At Source Networks

ISP networks that are directly connected to source networks can effectively ingress-filter spoofed packets [10]. Thus, a ubiquitous deployment of the ingress packet filters can drop all attack packets in direct attacks, and all attack packets sent from agents to reflectors, if all attack packets use spoofed addresses.Of course, ensuring all ISP networks in the Internet to install ingress packet filtering is an impossible task in itself.

At the Victim’s Network Detect a DDoS attack based on an unusually high volume of incoming traffic (of certain packet types) or degraded server and network performance.

IP hopping or the moving target defense, in which a host frequently changes its IP address or changes its IP address when a DDoS attack is detected. However, it is not difficult to add to the attack tools a DNS tracing function, which could attack the new address after discovering the address change.

INTERNET FIREWALL

Internet firewall attempts to detect DDoS attacks in the Internet core so that it can drop the suspected attack packets well before reaching a victim.

2 implementations

A ROUTE-BASED PACKET FILTERING APPROACH

Extends the ingress packet filtering function to the Internet core

distributed packet filters to examine whether each received packet comes from a correct link according to the inscribed source and destination addresses, and the BGP routing information. A packet is considered an attack packet (or other illegitimate packet) if received from an unexpected link and is therefore dropped. However, note that the dropped packet may still be legitimate due to a recent route change.

drawback of this approach is to require BGP messages to carry source addresses. Inclusion of the source information would significantly increase the BGP message size and the message processing time.

cannot filter attack packets using valid source addresses, such as reflected packets.

Therefore useless: all Dos Attacks are going to be reflected attackes = lots of waisted efforts and resources not fixing the issue at hnad.

A DISTRIBUTED ATTACK DETECTION APPROACH

placed in “strategic” locations in the Internet, and they nonintrusively monitor and analyze the traffic passing through them for possible DDoS attacks. detects DDoS attacks based on network anomalies and misuses observed from a set of distributed detection systems (DSs). Anomaly detection involves both determining a set of normal traffic patterns and detecting traffic patterns that “significantly” deviate from the normal ones. Misuse detection, on the other hand, aims to identify traffic that matches a known attack signature. Example, known port number for a known attack agent Issues: # required, location, design of the individual DSs and communication among themselves

DS Design Considerations — One major challenge in DS design is to process packets at very high speeds, especially when placed in backbone networks. Due to the distributed nature of a typical DDoS attack, each DS can observe only partial traffic anomalies. As a result, the entire attack detection process consists of two levels: local detection and global detection. If a DDoS attack is confirmed, the DS notifies the packet filtering component to install packet filters for the corresponding packet stream, and it may also notify the upstream networks to filter the attack packets there

Another important consideration is to ensure that any DS can reliably flood attack alert messages to other DSs. Therefore, the most important requirement for this network of DSs is that all DSs must always be “connected,” which includes both physical connectivity and usability of the paths among the DSs.

Since the two-level detection induces a certain amount of delay to reach a global detection decision, this DS network is not very useful for DDoS attacks of very short durations.

Flash crowds on the Internet can trigger false alarms in the detection systems.

Based on the previous discussion, there are four detect-and-filter approaches to defend against general flooding-based, DDoS attacks in the Internet: ubiquitous ingress packet filtering (UIPF), local attack detection (LAD), RPF, and DAD. The UIPF approach refers to the perfect ingress filtering scenario where every packet in the Internet is subject to ingress filtering. Although UIPF is quite impossible to realize, it is included here for the sake of completeness in the comparison study. The LAD approach refers to the attack detection at a victim’s network and/or its ISP networks. Other defense mechanisms for specific attacks, such as SYN flooding, are not considered here.

**Practicle support for IP tracebac**

technique for tracing anonymous packet flooding attacks in the Internet back towards their source.

we address the more limited operational goal of simply identifying the machines that directly generate attack traffic and the network path this traffic subsequently follows. We call this the traceback problem and it is motivated by the operational need to control and contain attacks.

efficacy of measures such as packet filtering improve as they are applied further from the victim and closer to the source.

Our solution is to probabilistically mark packets with partial path information as they arrive at routers. This approach exploits the observation that attacks generally comprise large numbers of packets. While each marked packet represents only a “sample” of the path it has traversed, by combining a modest number of such packets a victim can reconstruct the entire path. This allows victims to locate the approximate source of attack traffic without requiring the assistance of outside network operators. Determination can be made even after an attack has completed.

adds little or no overhead to the router's critical forwarding path and may be incrementally deployed to allow traceback within the subset of routers supporting our scheme.

ingress filtering, is to configure routers to block packets that arrive with illegitimate source addresses . This requires a router with sufficient power to examine the source address of every packet and sufficient knowledge to distinguish between legitimate and illegitimate addresses.

on many deployed router architectures the overhead of ingress filter becomes prohibitive on high-speed links.

Doesn’t save anyone against reflected attacks

**Link testing**

Most existing traceback techniques start from the router closest to the victim and interactively test its upstream links until they determine which one is used to carry the attacker's traffic. Ideally, this procedure is repeated recursively on the upstream router until the source is reached. This technique assumes that an attack remains active until the completion of a trace and is therefore inappropriate for attacks that are detected after the fact, attacks that occur intermittently, or attacks that modulate their behavior in response to a traceback.

Input debugging

First, the victim must recognize that it is being attacked and develop an attack signature that describes a common feature contained in all the attack packets.

Then ISP installs a corresponding input debugging filter on the victim's upstream egress port.

process is then repeated recursively on the upstream router, until the originating site is reached or the trace leaves the ISP's border

Problem:

considerable management overhead. Communicating and coordinating with network operators at multiple ISPs requires the time, attention and commitment of both the victim and the remote personnel

Incompetence or unwillingness to help = chain is broken.

Controlled flooding

does not require any support from network operators.

tests links by flooding them with large bursts of traffic and observing how this perturbs traffic from the attacker. Using a pre-generated “map” of Internet topology, the victim coerces selected hosts along the upstream route into iteratively flooding each incoming link on the router closest to the victim. Since router buffers are shared, packets traveling across the loaded link – including any sent by the attacker – have an increased probability of being dropped. By observing changes in the rate of packets received from the attacker, the victim can therefore infer which link they arrived from. As with other link testing schemes, the basic procedure is then applied recursively on the next upstream router until the source is reached.

drawbacks and limitations:

Most problematic among these is that controlled flooding is itself a denial-of-service attack. Makes it unsuitable for routine use.

requires the victim to have a good topological map of large sections of the Internet in addition to an associated list of “willing” flooding hosts.

poorly suited for tracing distributed denial-of-service attacks because the link-testing mechanism is inherently noisy

only effective at tracing an on-going attack and cannot be used “post-mortem”.

**Logging**

log packets at key routers and then use data mining techniques to determine the path that the packets traversed. Good: can be used post-attack

bad:

drawbacks, including potentially enormous resource requirements (possibly addressed by sampling) and a large scale inter-provider database integration problem.

**ICMP Traceback**

router to sample, with low probability (e.g., 1/20,000), one of the packets it is forwarding and copy the contents into a special ICMP traceback message including information about the adjacent routers along the path to the destination.

Problems: ICMP traffic is increasingly differentiated and may be filtered or rate limited differently from normal traffic, the ICMP Traceback message relies on an input debugging capability (i.e. the ability to associate a packet with the input port and/or MAC address on which it arrived) that is not available in some router architectures, if only some of the routers participate it seems difficult to positively “connect” traceback messages from participating routers separated by a non-participating router, and finally, it requires a key distribution infrastructure to deal with the problem of attackers sending false ICMP Traceback messages.

**Packet Marking**

does not require interactive cooperation with ISPs and therefore avoids the high management overhead of input debugging. Unlike controlled flooding, it does not require significant additional network traffic and can potentially be used to track multiple attacks. Moreover, like logging, packet marking can be used to trace attacks “post-mortem” – long after the attack has stopped. Finally, we have found that marking algorithms can be implemented without incurring any significant overhead on network routers.

Definitions:

attack path from A is the unique ordered list of routers between A and V.

exact traceback problem is to determine the attack path and the associated attack origin for each attacker.

approximate traceback problem as finding a candidate attack path for each attacker that contains the true attack path as a suffix.

All marking algorithms have two components: a marking procedure executed by routers in the network and a path reconstruction procedure implemented by the victim. A router “marks” one or more packets by augmenting them with additional information about the path they are traveling. The victim attempts to reconstruct the attack path using only the information in these marked packets.

The convergence time of an algorithm is the number of packets that the victim must observe to reconstruct the attack path.

Issues:

When a router receives a packet, it has no way to tell whether that packet has been marked by an upstream router or if the attacker simply has forged this information. In fact, the only invariant that we can depend on is that a packet from the attacker must traverse all of the routers between it and the victim.

BASIC MARKING ALGORITHMS

Node append

The simplest marking algorithm: append each node's address to the end of the packet as it travels through the network from attacker to victim (see Figure 2). Consequently, every packet received by the victim arrives with a complete ordered list of the routers it traversed – a built-in attack path.

Limitations: high router overhead incurred by appending data to packets in flight.

impossible to ensure that there is sufficient unused space in the packet for the complete list. This can lead to unnecessary fragmentation and bad interactions with services such as MTU discovery

4.2 Node sampling

A single static “node” field is reserved in the packet header – large enough to hold a single router address each router chooses to write its address in the node field with some probability p. After enough packets have been sent, the victim will have received at least one sample for every router in the attack path

efficient to implement because it only requires the addition of a write and checksum update to the forwarding path. Current high-speed routers already must perform these operations efficiently to update the time-to-live field on each hop.

Limitations: random variability of path, not robust against multiple attackers.

4.3 Edge sampling

explicitly encode edges in the attack path rather than simply individual nodes. reserve two static address-sized fields, start and end. significant practical limitation of this approach is that it requires additional space in the IP packet header and therefore is not backwards compatible.

They propose an interesting encoding alogorithm but still have these issues:

 backwards compatibility,

 distributed attacks,

 path validation, and

 approaches for determining the attack origin.