Distributed Denial of Service

Géraud Mudry
EPFL, Lausanne

Frédéric Olive
KTH, Stockholm

March 2002

Abstract

Today's Internet is a wide collection of computers interconnected without any laws or security requirements; consequently most systems are vulnerable to attacks perpetuated by some malicious individuals. These attacks can seriously damage systems or, in the case of denial of service attacks, can destroy temporary the availability of a service to legitimate users.

Since the current Internet infrastructure has very few built-in protection mechanisms, it is vulnerable to those attacks and failures. Huge efforts are done in order to increase the security of the Internet but one of the most difficult problem to solve is denial of service attacks because it is simply using the fact that the resources available in the Internet systems are finite and can be easily overwhelmed.

The difficulties in reaching this goal are numerous; some simple tools already exist to limit the effectiveness and scope of these attacks, but they have not been widely implemented. The attack does not have to exploit a security hole at the target to cause a problem (although that would exacerbate the problem, to the attacker's benefit), and there is almost nothing the victim can do to protect it.

1. Introduction

Denial-of-Service (DoS) are among the most dangerous thread to today's Internet security. A DoS attack is meant to overwhelm a victim host or network with illegitimate requests issued by a single source, to the point of unresponsiveness to legitimate user transactions. Since 1999 wide attacks have been launch against several high-visibility e-commerce site like Yahoo, Ebay, Amazon, as well as against some companies and educational institutions network infrastructures, or more vital Internet backbone resources. Besides, many other low-profile sites have also been victims, and most the attacks are not publicized. Reports of attacks to the Computer Emergency Response Team (CERT) show that such attacks have increased by 50 percent per year between 1989 and 1995. Some substantial work has been made to quantify DoS attacks [3], and it appears 4 attacks every 10 min are launched over the Internet!

Even though some partial and still imperfect solutions or containment methods could be applied to limit DoS attacks, recent evolutions show coordinated many-to-one attacks, called Distributed Denial of Service (DDoS) attacks. Rather than using a single source, attackers now take advantage of thousands of compromised host to damage their victim. DDoS require well-designed tools and strategies to compromise hosts throughout the Internet and coordinate attack flows. Section 2 provides background information about DDoS scenario and techniques, and Section 3 gives an overview of the most popular tools used by attackers to initiate and launch such distributed attacks.

A great deal of work and research is done to elaborate strategies and implementations of reliable defenses against DDoS. Different approaches are being investigated depending on where to implant the defense tool: on the victim network, on the attacker's one or on the Internet core infrastructure. Section 4 details motivation, results or expectations of these various approaches, while Section 5 to 8 describe major tools currently expected to defeat denial of service attacks.

2. DDoS Overview

The principle of DDoS is to combine bandwidth of multiple machines into a single victim machine or network. Much higher level of flooding traffic can be generated by this approach, than the traditional DoS style attacks, where a single machine attacks another.

2.1. Model

Distributed attacks are launched from multiple slave hosts. Basically, one or more *attackers* break into a set of hosts over the Internet, exploiting security weaknesses of their system, and install some attack daemons on them. There are two types of compromised hosts and daemons: the *agents*, which run a daemon generating stream of packets directed toward the victim, and the *handlers*, which are capable of controlling multiple agents and coordinate an attack. The attackers remotely control their few handlers. The hierarchical structure of attackers network is described in Figure 1.

The communication between the attacker and its handlers, and between handlers and agents are referred as the *control traffic*. The communication between the agents and their

victim is called the *flooding traffic*, and exploits weaknesses in the TCP/IP suite protocol, described in section 2.3, to inflict denial of service.

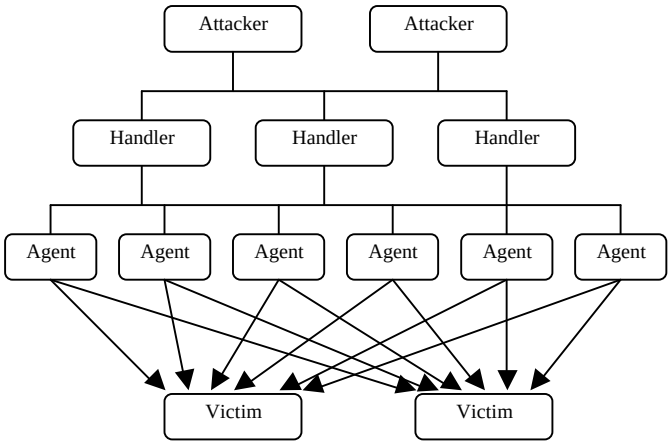

Figure 1 : A typical DDoS Network

Various protocols are used for the control traffic. Usually, an attacker access handlers via encrypted TCP connections, but a simple telnet can also be used. Handlers and agents communications are generally based on a different protocol than the one used between attacker and handlers, in order to avoid correlations. Since this communication does not need to be bi-directional, UDP is mostly used.

2.2. Attack scenario

The classical DDoS attack follows the following steps: recruitment, compromise, control and attack.

During the recruitment phase an attacker scans hosts on the Internet seeking vulnerabilities to install his daemon and turn these hosts into agents or handlers. They are commonly external to both the victim and the attacker network, to avoid early and efficient response from the victim and detection of the attacker. Attackers often look for powerful machines able to generate large stream of packets. This selection turns to be more and more automatic, as the attack tools are more sophisticated.

When hosts have been selected, the attacker compromises them by installing his attack code and protecting it from discovery and deactivation. This step is also automated. Once hosts are compromised, they can also participate in the recruitment phase, looking for new hosts to propagate the infection, since recent tools integrate some self-propagating features.

As agents are compromised, some control procedures start between agents, handlers and the attacker. In recent tools, Internet Relay Channel (IRC) are used for communications. This mean is quite safe because the discovery of a compromised machine lead to the discovery of the channel but the identity of the other participant is still protected.

Finally, the attacker launch is attack via the handlers, which forward orders via IRC. Target address, duration, port number can be customized and packet types, length, TTL, etc., are often randomized to avoid detection by correlation.

2.3. Attack techniques

There are three major well-known and popular techniques to complete a denial of service by flooding the victim or its network with spurious packets. They use TCP SYN requests, UDP packets streams, and ICMP ECHO messages.

TCP SYN flooding attacks use the standard three-way handshake procedure to cause resource consumption and bandwidth consumption. A stream of TCP packets with various flags set, among them the SYN flag, are sent to the victim, which exhaust its resources trying to acknowledge them, and congests its network keeping this unachieved connections alive.

UDP floods simply send a large number of packets, “UDP Storm”, to victim, whose processing resources cannot handle. This form of attacks also exploits the priority of UDP packets over TCP flow, so that when UDP packets saturate the network, no bandwidth is left to legitimate TCP transactions.

ICMP floods use ECHO_REQUEST (a ping) message stream sent to the victim, which deplete its available resources for replies. A variation of ICMP ECHO attacks is “Smurf attacks”, which misbehave broadcasting on local networks. A ping is sent to a local broadcast address with the broadcast address as the source address. This ping is forwarded to all hosts on the networks, which reply to the broadcast address, rapidly congesting the local network with ICMP traffic.

Most attacks exploit these major weaknesses of TCP/IP protocol suite, and often combine them.

2.4. IP spoofing

In order to conceal their DDoS network infrastructure, attackers typically forge or “spoof” the IP source address of each packet sent from the agents to the victims. Consequently, the packets appear to the victim to be arriving from some legitimate third parties.

It has been long understood that spoofing source address is a major weakness in the IP protocol [2], and even if several attempt have been made to reduce anonymity afforded by IP spoofing, it still remain the main obstacle to agents and attacker identification. Lots of research is done in this domain to trace back spoofed packet to their original source, in order to detect compromised hosts so as to assure that they are cleansed to prevent future misuse.

3. DDoS Tools and Trends

DDoS attacks are so in use now that some techniques and tools appear as standards for security experts. This section

will discuss the different trends in the propagation of the attack and then the main tools used by the attackers.

3.1 DDos Trends

The more technology is increasing, the more automation is introduced in DDoS attacks such as mechanisms of propagation, that can be done following three widely used schemes.

First, the central source propagation scheme uses a mechanism that executes an instruction to transfer a copy of the attack tool to a compromised system, then a script controls the automation of the attack and the propagation to new systems. This process is described by Figure 2.

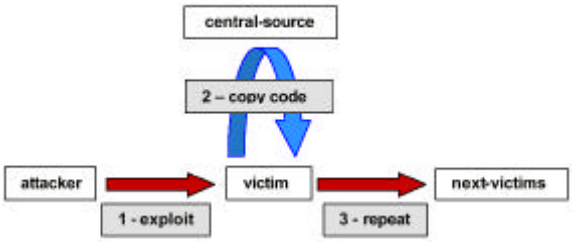

Figure 2 : Central source propagation [1]

Back-chaining propagation is a process where a mechanism executes instructions to transfer a copy of the attack tool from the attacker, and then the process is identical to central source propagation. This is described in Figure 3.

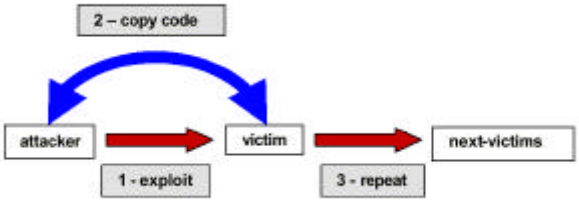

Figure 3 : Back-chaining propagation [1]

Finally, autonomous propagation is a process in which a mechanism injects directly attack instructions into the processing of the victim host, then again the process is identical (Figure 4).

Figure 4 : Autonomous propagation [1]

Another way of propagation attacks or initializing them is to use malicious e-mail attachment files to compromise a system and initiate the attack process.

Once attack daemon installed on several agents, the attacker as to communicate with them, in order to target a victim for instance. While former attacks targeted Unix-based operating systems, the trends are now to attack Windows-based systems since they are more vulnerable because most windows users are less aware of security and less likely to be prepared to attacks or to responds to them.

However; attacks based on selective targeting does not use uniquely the operating system criteria. Another criteria for target selection is a potentially exploitable vulnerability that can be found with the help of scanning in order to test for network connectivity, regular levels of network traffic and available bandwidth.

One of the most recent and disturbing trends in DDoS attacks is selective targeting routers, which for example exploits vendor supplied default password in order to gain access and control to the router. The attacker use then the router as a scanning platform and launch point for attacks.

Finally, the Time-To-Exploit parameter is largely decreasing because of the large existing choice of code-based attack tools, rival groups of attacks will develop new competitive tools to take advantage on other groups. Thus if the tools are exposed to outside group the lifetime of the tool is really short, consequently when public is aware of the existence of a new attack tool it is already in widespread use.

3.2 DDos Tools

Each tool corresponds to different strategy exploiting the numerous vulnerabilities of internet protocols which can in most cases be fixed by applying patches. But since tools are always developing faster than the application that they break into, the attackers always find new vulnerabilities to exploit.

Below is an overview of well-known DDoS attack tools that illustrate the variety of mechanisms employed.

Trinoo is a UDP flood attack using constant size UDP packets to target random port of the victim. The protocol used by the handler to communicate with agents is TCP or UDP but this channel can be encrypted and password protected. For now *Trinoo* does not use spoof source addresses, that could be an easy and incoming development of Trinoo.

Tribe Flood Network (TFN) is a tool that can generate TCP SYN, UDP, and ICMP echo requests floods as well as ICMP directed broadcast (“Smurf”, as explained in 2.3), it has also the ability to spoof source addresses. Handlers and agents communicate through ICMP ECHO_REPLY packets.

Stacheldraht (German for barbwire) is a combination of *Trinoo* and *Tribe Flood Network* and the feature added in this tool is an encrypted channel between handler and agents. New versions have also some means to use different signatures in order not to be detected.

TFN2K is a variant of *TFN* that has features to make traffic hard to identify and filter, since the attack type (TCP SYN, UDP, ICMP) can be varied during the attack. Even the type of messages in the communication channel between handler and agents can be random between TCP, UDP and

ICMP. *TFN2K* has also a feature to defeat ingress filtering by forging packets that appear to come from neighboring machines.

The *mstream* tool uses spoofed TCP packets with the ACK flag set, the communication channel is not protected but it informs all connected users of access to the handler by competing parties.

Shaft is another tool that uses TCP, UDP or ICMP floods but it can use the three simultaneously, it has also random source addresses and source ports. Another interesting feature of this tool is that it can change the IP address and source port of the handler during the attack.

Finally *Code Red* is a self-propagating malicious worm that uses vulnerability in Microsoft IIS servers to propagate. It is a synchronized attack with a preprogrammed scheduled time, abort time, attack method and victim addresses, i.e. there is no handler-agent scheme involved.

We can conclude that these tools use vulnerabilities of the most widely used protocols on the internet and there is a broad variety in the strategies involved, that is why it is really difficult to find solutions to DDos attacks.

4. Defense Strategy Overview

Prevention of DDoS is almost impossible to achieve, since it basically means increasing security of every Internet host to the point where they cannot be compromised by any attackers tools. As a matter of fact, applications and operating systems are not bug free, providing security holes, and it is not possible to check if every users properly used security patches to secure theses holes. Prevention approach will always be vulnerable to new attacks on new application holes, and is not likely to stop the DDoS threat. Hence, the only way to remove DDoS is to design and implement detection defense strategies to protect victims and identify attackers.

A common DDoS attack streams packets which traverse many Internet hosts, from a *source network*, through an *intermediate network*, composed by machines that forward the packets, to the *victim network*. Consequently, we can identify three major categories of detection and defense strategies, considering the part of the Internet where they intend to be deployed.

4.1. Victim Network Strategies.

The victim has the greatest need of protection and DDoS defense system. Most systems developed on the victim networks are based on intrusion detection, attack recognition, and some filtering features. Early systems aimed at monitoring traffic properties, raising alerts when an attack pattern or signature is detected at the victim network gateway, and eventually filtering certain type of traffic supposed to be part of the attack. For a reasonable volume of packet, these systems can efficiently protect the victim but in any case they do not stop the attack, and can be overwhelmed by large scale DDoS attacks. Their only advantage can be to increase the victim ability to recognize early that it is the target of an attack.

Several system and strategies are now being developed to secure the victim network and to stop DDoS attacks. Various trace back algorithms are among the most investigated solutions and they try to identify attacking flows and their to go upstream until they reach the source but those strategies does not address completely the issue of the identification of the attacker since lots of DDoS attacks use spoof source addresses or attacking routers that the attacker has under its control.

Other strategies are also implemented in routers and they analyze incoming flows, categorize them and drop them as soon as they identify, following various criteria's, the flow as an attacking flow.

Some other proposed solution on monitoring buffer queues and when congestion is detected, packet belonging to a certain type of problematic traffic are dropped, this approach could be seen as an adaptation of the RED congestion control mechanism to DDoS attacks.

Lots of strategies increase the ability of a victim to detect quickly an attack and thus to obtain time to respond but that does not address the problem of avoiding the attack.

The other strategies that try to rate limit some specific traffic are more effective to relieve the attack affect but there is problem when the offending traffic share common characteristics with the legitimate traffic and when the amount of offending traffic is so huge that the rate limit mechanism cannot handle it.

Consequently victim network strategies are not the most effective strategies to relieve the effect of the attack since usually huge amount of flows are involved, because of convergence of the flows, so this could be efficient if the other upstream strategies are able to limit the attacking flow or even to block it.

4.2. Intermediate Network Strategies

Intermediate network strategies have to be done by routers in order to preserve the victim even if that is not enough to stop the attack that can help the victim.

The main problem in intermediate network strategies is to identify attacks, since the attack is generally spread, i.e. the attacking flow is following different path in the network. Then the routers have to implement some algorithm in order to limit the attack.

Two aspects of intermediate network strategies are detection of the attack in order to rate limit it or trace the attack back to the source and some algorithm have been designed in order to perform one or both of these tasks.

Algorithms have also been designed in order to detect misbehaving routers that launch denial of service attacks but it does not help to identify misbehaving hosts and it needs explicit communication between routers.

Another mechanism used to identify the source of attack is traceback but it does not help to stop the attack and traceback mechanisms are ineffective when the volume of the attack flow is small. Traceback is also hard to achieve when the attack uses spoof source addresses, that is why some mechanisms such as ingress filtering have been designed. These mechanisms simply help to prevent spoofed source address but since spoofed addresses are not necessary for a DDoS attack, it does not prevent against all attacks.

It is also possible to identify, at the routers, aggregates that participate in an attack and then trying to rate limit these flows but it implies collaboration between routers because each router on the path of the flow has to monitor traffic and rate limit the aggregate responsible for the attack. This approach is the one used by pushback but one disadvantage of this method is that some normal flows will suffer from the rate limitation since the real identification of an attack flow is a hard and imprecise task.

A response from the intermediate network is more effective than one initialized by the victim network sine larger volume of attack can be handled and the attacks can be traced back to the source but there are problems that prevent the development of these approaches.

- the intermediate network performance is a problem since it handle large amount of traffic and allocating resources to traffic profiling or rate limiting could degrade the network's performance.
- the attack detection is also problematic since it is difficult for the intermediate network to detect the attack and identify the victim since traffic is spread. A solution could be to ask the victim to signal the attack to intermediate network but this requires some identification process since this signal could be used by that attacker as well.
- lack of inter-domain cooperation is also a problem since lots of solutions proposed rely on cooperation between routers but there is currently very little cooperation between different administrative domains.
- general deployment is important since all the methods rely on a cooperation from all the system of the network, so if one of those system does not participate, the performance of the whole system is degraded.

4.3. Source Network Strategies

One obvious idea could be that taking a good network based intrusion detection system on the source side could achieve a good DDoS defense system, but the major problem is that the attack does not look like an attack at the source since the flows originate from different sources while at the victim end all the flows converge and significantly affect the system so that it is easy to detect the attack.

That is why there is not so many research efforts on source network strategies but some research effort try to address this issue. The main strategies, such as the source router defense, are based on the idea that each network has to monitor traffic characteristics and store statistical data. Then an attack is detected by abnormal values of packet ratios and then abnormal traffic is rate limited.

Of course placing DDoS defense systems close to the sources of the attack have many advantages because the flows can be stopped before entering the Internet and before converging to the victim, moreover being close to the source allows easier traceback and attack investigation. Another advantage is that a source network is more able to sacrifice some of its resources to defense systems. However, due to the low degree of flow aggregation, more complex detection strategies need to be employed, since the misclassification of flow can cause problems by restricting legitimate traffic.

5. IP Traceback

The main problem and weakness in IP protocol is that there are no guarantee that what is filled in the source address field of a packet is the actual IP address of the sender. Nothing in the protocol implementation nor in the current operating systems can prevent IP spoofing, which is widely exploited by attackers to disguise the real origin of the attacks. Traceback techniques aim at determining the original source of spoofed IP packets, in order to identify agents participating in an attack. This technique then relies on attack detection and path reconstruction to the approximate source.

5.1. IP Traceback Techniques Overview

Most existing traceback techniques start from the router closest to the victim and iteratively checks upstream links to determinate which carried the attacker's packet. This iterative process can be done during the attack or after its completion, and is based on querying information to the routers to reconstruct the attack path (Figure 5).

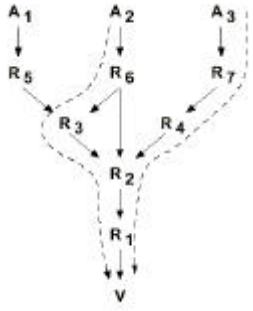

Figure 5 : The attack paths are $\{A_2,R_6,R_3,R_2,R_1\}$ and $\{A_3,R_7,R_4,R_2\}$

There are two distinct approaches to the IP traceback problem. The first is based on logging packets at key points on the network (for instance routers or end hosts) and perform auditing after an attack and try to recompose the attack path. The second approach is to reconstruct the path

during an attack using schemes that marks packet on the fly, without needing to keep track of records of any kind. These approaches are described in subsequent sections 5.2 and 5.3 respectively.

5.2. Logging and auditing

This approach suggests that packets or sample of packets can be logged in routers throughout the network and that data mining extraction techniques can be used to determine the path that the attack packets traversed. This scheme is especially useful to trace attack already completed, but can also be used during attacks since routers could be provided with sophisticated real-time auditing features. However, basic logging has obvious drawbacks: logging all packets passing routers requires enormous storage capacity.

A Hash-based IP traceback technique [7] has been developed and tested in order to reduce storage requirement by several order of magnitude. Traffic logging is achieved by computing and storing 32-bit packet digests rather than packet themselves. In addition to that, encrypted packet digest preserve traffic confidentiality.

The hash function takes some part of the IP packet content: enough to uniquely identify the packet, but not too much for good performances. IP packet fields included in the digesting function are invariant fields, e.g. TTL, ToS, Option, and checksum are taken into account, in addition to the first 8 bytes of the payload. [7] shows that computing the digest on the first 28 invariant bits of a packet is enough to avoid packet collision (collision rate is then approximately 0.00092%).

To reduce even more dramatically storage requirement some space efficient data structure can be used, as suggested in [7], resulting in a finally feasible storage of most of packet digests on various routers on the Internet.

When an attack occurs or had occurred, some intrusion detection system queries the routers for path information on offensive packets. Only routers having digests records reply. The attack graph is rebuilt by a reverse-path flooding (RPF) algorithms. Figure 6 shows the reverse path flooding proceeding from the victim backwards to the attackers.

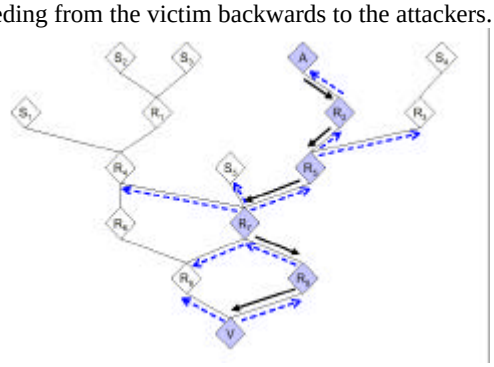

Figure 6 : Reverse path flooding illustration : black arrows show the attack path, dashed arrows the queries [7].

Anyway, even by significantly reducing storage requirements, hash-based traceback techniques do not solve the other main problem that logging techniques have to cope with: large scale inter-provider database integration is still required to perform data mining based auditing.

5.3. Marking Schemes

The basic idea of IP marking schemes is that routers probabilistically write some partial path information into the packet during forwarding. This technique usually uses the edge sampling algorithm to write edge information consisting in start and end IP address along the path, and a distance field. Each router marks the incoming packets with a probability *p*. When a router decides to mark a packet it writes its own IP into the start field and write zero in the distance field. If the packet was already marked (e.g. its distance field is zero or more), it can choose to write its own IP in the end field, and increment the distance field by 1. Finally, every router must increment the distance field of marked packets, so that the distance field is always greater than or equal to the attack path.

The router path information is written in the 16-bit IP Identification header used for fragmentation. Since recent measurements have shown that less than 0.25% of packets are fragmented it does not disturb significantly the traffic. However 16-bit is generally not enough to store two 32-bit IP addresses and the distance field: some encoding schemes are needed to reduce storage requirement in each packet.

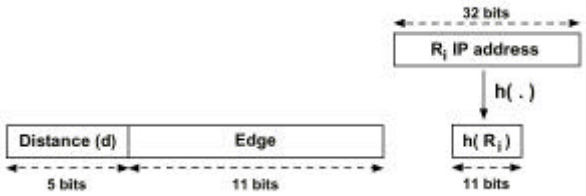

Figure 7 : Encoding the IP Identification header [6]

The most common approach is to use a 5-bit distance field and an 11-bit edge field (Figure 7). A 5-bit distance field can represent 32 hops, which usually sufficient for most of Internet paths. Hash functions reduce a 32-bit IP address into an 11-bit sequence. Two functions are actually used h_1 and h_2. When a router R marks for the first time a packet it writes h_1(R) in the edge field, and if the packet was already marked it XORs h_2(R) and the edge value and writes the result in the field. The reason to use two hash functions is to distinguish the routers order in the XOR result.

The reconstruction process starts with the victim extracting the distance value and decoding the edge information. The victim has to know about the network topology and often stores some kind of upstream router map representation. At a distance *d*, the victim enumerates all router one hop away and check if their hash valued IP match the edge field value.

More sophisticated marking algorithms are fully studied in [6]. This paper also shows that these techniques have very low routers overhead and can be easily deployed. Marking schemes are considered as the more efficient and handier traceback techniques.

6. Pushback

Lots of denial of service attacks are based on aggregates, that is different flows that share something in common such as destination or source addresses, certain application type or types of packets such as TCP/SYN packets or ICMP ECHO packets. A hard task in preventing DoS attacks is to identify these aggregates and determine a policy in order to prevent them to flood the victim, this is not the purpose of pushback but pushback will use this identification services and will require collaboration between adjacent upstream routers. The purpose of pushback is to allow adjacent upstream routers to rate-limit traffic corresponding to identified aggregates and it prevent upstream bandwidth from being wasted on packets that will be dropped later in the network. This is not the only purpose of pushback since pushback tries also to protect "normal" traffic from these aggregates.

The hard task of differentiating good flows from bad will be addressed by developing heuristics that try to identify most of the bad packets sent by the attackers. Bad traffic will be characterized by an attack signature and a congestion signature.

Poor flows will be those that satisfy the characteristics of bad flows without being attack flows and thus suffer from these common characteristics.

The principle to identify aggregates is that as soon as congestion is detected we try to identify characteristics between packets or flows and classify them. Another fundament in DDoS identification is the packet drop history that gives information on the attack in progress and let the pushback agent detect in an easier way the aggregate that cause the attack.

As soon as an aggregate is identified and classified as a bad flow the pushback mechanism will be started, i.e. the root server that identified the aggregate will begin to rate-limit packets corresponding to this bad flow and if this is not enough it will send pushback messages to adjacent routers in order to tell them to rate limit this flow as well.

Pushback for an aggregate can be visualized as a tree, as shown in Figure 8, where an attack is detected by router R_0 that then informs the adjacent upstream routers of the ongoing attack.

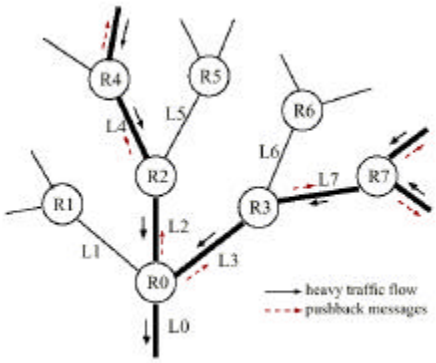

Fig 8 : Illustration of pushback [9]

Congestion Signature
Bandwidth Limit
Expiration Time
RLS-ID
Depth of Requesting Node
Pushback Type

Fig 9 : Content of a pushback request [9]

Figure 9 describe the content of a pushback request that contains :

- the congestion signature identifying the aggregate
- the requested upper bound for the amount of the traffic sent belonging to this aggregate (i.e. at what rate have to rate limit the traffic corresponding to this aggregate)
- the expiration time of the request since routers have to determine when the pushback process should be stopped
- RLS-ID is used in the feedback messages sent by upstream routers to the router that initiated the process in order to inform of the arrival rate of the aggregate at each node
- depth of the requesting node is used to set timers for sending feedback
- the pushback type will influence the decision taken by the upstream router whether it has to propagate the pushback request upstream or not.

This process of pushback will continue until the router is not any more significantly congested or if the aggregate being limited is not any more one of the main responsible aggregate for congestion, thus all routers implemented pushback has to monitor congestion before their rate limiter.

Another characteristics of pushback is that it can allow a trace back upstream to the source: it is possible to identify the sources, but since sources are usually spoofed, this property usually require some special implementation.

One of the main problems of pushback is that it will rate limit poor flows that are not part in an attack but share characteristics with some bad flows. Consequently this traffic will suffer from the rate limiting process and the solution to solve this problem is to improve the detection of

bad flows. This is a hard task and a lot of work has still to be done in order to improve the process.

7. Ingress filtering

Most of denial of service attacks employ forged source addresses that make trace back difficult. A simple and effective method can be to prohibit forged source address DoS attacks to be propagated from “behind” an ISP’s aggregation point by ingress traffic filtering. Ingress traffic filtering does not protect against flooding that originates from valid source IP addresses, but it prevent from the use of IP addresses that does not conform to ingress filtering rules. Moreover, using ingress traffic filtering enables the attacker to be traced back to its true source (at least to its true ISP) since the attacker has to use a valid source address for its attack.

Considering a classical TCP SYN attack, the attacker usually uses randomly changing source addresses which are not present in the global internet routing tables, consequently these addresses are unreachable and the attacker floods the victims without allowing the victim to trace the packets back to the attacker.

Another common method is to use valid source addresses from within another legitimate network which appears as the source of the attack, this results in a double denial of service attack, since the apparent source address is flooded by TCP/ACK packets incoming from the victim.

The solution to this problem proposed by ingress traffic filtering is that if the attacker’s machine is attached to a router implementing ingress filtering, the input link of the router will restrict traffic to only traffic originating from source addresses that the router is serving on this ingress link. For example if the router is serving addresses 155.0.0.0/8 the ingress filter on the router will check:

> IF packet's source address from within 155.0.0.0/8
> THEN forward as appropriate
> IF packet's source address is anything else
> THEN deny packet

This allows also network administration to save information on the dropped packets in order to monitor malicious activity. The ingress filtering method can also recommend checking that the return path to the source address will flow the same interface as the packet arrived upon.

Ingress filtering can also cause problems to some “special” services such as mobileIP, DHCP or BOOTP. For example, since in mobileIP the traffic from the mobile node is not tunneled and thus originates from an invalid source address for the router to which the mobile node is currently attached, the packets will be dropped but the Mobile IP Working Group is trying to specify reverse tunnels in the Mobile IP specifications, so this problem will be soon resolved. Concerning BOOTP and DHCP, the administrator has to ensure that the ingress filter don’t drop packets originating from 0.0.0.0 and destined to 255.255.255.255 are not dropped in order to be able to reach the relay agents in router when appropriate.

As we have seen in this section ingress traffic filtering will reduce the possibility for attackers to spoof source addresses but it lets the attacker use addresses from within the same network as he is using. Consequently it is still possible to spoof those addresses but an ISP can locate the attack within its network and can disconnect some addresses until the problem is solved. This allows also trace back to a precise network and thus an ISP is able to monitor in order to prevent such further attacks from the same attacker. Another problem is that ingress filtering has to be implemented by all the IPS in order to be efficient and that will take time before the ingress filtering become a standard for the ISP, so some holes in this system will still persist.

8.Source Router Defense

8.1. Principle and Mechanism

A source router defense system is deployed in the source network, precisely on the router that serves as a gateway between the source network and the rest of the Internet. Indeed this source router is the ideal position to protect the victim from individual DoS originating from some agents in source network because of its ability to arbitrate communication between them. This approach assumes that the source router is able to determine on which interface it receives the incoming and outgoing traffic form the source network and that all machines on the source network use it as the exit router to the Internet.

The goal of this system is to detect outgoing packets that are part of an attack. Attacks can therefore be stopped before they reach and damage the victim. However, detection of this packet at the source is particularly hard since the attack flow is not aggregated as it is at the victim network gateway.

The underlying principle of a source router defense system is monitoring the behavior of each destination the source network is communicating with, looking for difficulties in communication. These difficulties, such as the reduction of response packets, long inter-arrival times, or any other severe destination network congestion characteristics, can be the sign of a denial of service attack against this specific destination. The detection of these abnormal difficulties is achieved by comparing the parameter of the two-way communication for each destination and a predefined normal traffic model. If the difference between the model and the actual measure on the traffic turns to be important, it can reveal a possible DDoS attack. In this case, the router responds by imposing a rate limit on all outgoing traffic for the destination. The rate limit is updated with subsequent traffic observation: the rate limit is even more restricted if the DDoS attack is confirmed, whereas refutation leads to slow increase.

8.2. System architecture

A source router system requires an observation and a throttling component to be added to the common router functionalities. These components can be part of the source router itself or can be external units that communicate with the router (Figure 10).

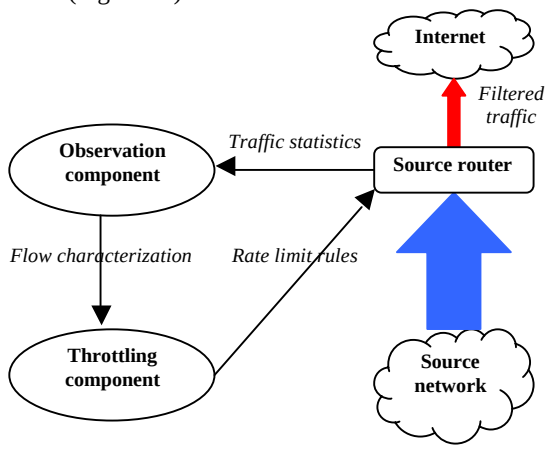

Figure 10 : Source router defense system architecture.
adapted from [4]

The observation component is the actual monitoring tool, which classify each packet as incoming or outgoing, based on source and destination address, and arriving interface. It computes statistics on current flows using packets header information, and periodically compared these statistics with a normal traffic model. The comparison result is a characterization of the flow as being normal, transient or attack flow. Normal flows match the model profile and have not been recently classified as attack flows. Transient flows parameters match the model but were recently detected as attack flows, thus their rate limit must be slowly increase to avoid recurring attacks. Finally attack flows are clearly out of the model boundary and trigger a rate limiting response.

The throttling component is responsible for computing and adjusting the value of the rate limit of a particular flow. Since the detection of an attack based on statistical comparison is likely to give false-positive results, an adaptive filtering is better suited to a complete discard of the flow. Thus some packets are allowed to go through and reach the destination, keeping the connection alive and allowing misclassified flows to achieve the transaction they were intended to perform. Since DDoS attacks impacts are based on the overall volume of packet, it is acceptable to let some packet go through.

The rate to a particular destination depends on the history of the destination and the current classification of its current flow. Therefore the throttling component has to establish a strategy that will effectively block attack flows, while avoiding complete discard of misclassified flows and allowing them to recover quickly. A good trade-off between high protection and fast recovery has to be implemented in the calculation algorithm of the rate limit [4].

8.3. Statistics Gathering and Traffic Models

The crucial point of this approach is to generate traffic models for every possible destination on the Internet, as accurate as possible in order to minimize false-positive comparison. The information about the normal behavior of foreign hosts are recorded and sorted in the observation component. Since keeping record on each existing IP address is infeasible, the size of the cache is limited and is periodically purged. In this case, only normal flows entries are removed, attack and transient flows information are kept, but their statistics are reset.

Most routers are now optimized to perform fast lookup on IP header fields. The statistics gathering uses information provided by the source router on IP headers. For a particular destination, a common record contains the following information [4]:

- number of packet sent
- number of packet received
- number of bytes sent
- number of bytes dropped due to the rate limit
- inter-arrival time of packet sent
- inter-arrival time of packet received
- smoothed mean of the ratio of number of packet sent and received
- timestamp when fields were reset

These data are used to infer normal traffic behavior, by applying to them some statistical principles. As a matter of fact, most communications between hosts involve a two-way traffic. The TCP protocol for instance uses acknowledgment traffic reverse to the data traffic, thus the ratio of packet sent and received is ideally 1, and any substantial deviation from this ratio is interpreted as a possible TCP attack. However, UDP and ICMP traffic do not require reverse traffic. Models for non-TCP communications are created in advance from statistical observation and analysis of the destination behavior. For some interval of time the traffic to a destination and the packet type are monitored, in order to gather significant traffic statistics and classify the destination as a TCP, UDP or ICMP destination, based on the majority of packet type sent to it. The traffic model and destination classification are regularly updated to reflect new trends of the destination's behavior and the network traffic. Once a model has been establish, the detection of abnormal traffic consists in comparing traffic model with the actual traffic flow, and packet type in the flow with the destination classification. Hence, unusual traffic volume, delays and packet type ratio can be interpreted as an attack and imply a rate limitation response.

Source router defense systems are among the most efficient since it is performed at the source level. Possible attacks can almost perfectly be destroyed from its start, but the system needs to be widely deployed at the exit of all ISP networks.

9.Conclusion

Denial of service attacks exploits both TCP/IP protocol suite vulnerabilities and security holes on hosts connected to the Internet. They inflict growing damage as they tend to be widely distributed, and are still difficult to prevent. IP spoofing is the main tool used by the attacker to stay anonymous and dissimulate his network. Some strategies have appeared concerning hosts defense, and identification of compromised agents that launch attacks, and eventually identification of the actual attacker.

Most of the defense techniques involve attack detection system that trigger an alert and starts a protection system. The protection is mainly based on filtering the attack flow, on the victim network or at the attack source gateway. Traceback techniques can also be used to reconstruct the path to the attacking hosts, which is often use to identify and cleansed agents, to avoid further attacks from it.

Even if some solution could significantly reduce the number and the power of denial of service attacks, none has been deployed yet, mainly because of the lack of inter-domain cooperation in this domain.

References

[1] CERT Coordination Center, “Trends in Denial-of-Service Attack Technology”, October 2001.

[2] R. Morris, “A Weakness in the 4.2BSD Unix TCP/IP Software”, AT&T Bell Laboratories, February 1985.

[3] D. Moore, G. Voelker, S. Savage, “Inferring Internet Denial-of-Service Activity”, In *Proc. USENIX LISA’01*, 2001.

[4] J. Mirkovic, “DDoS Network Attack Recognition and Defense”, PhD Dissertation Prospectus, UCLA, January 2002

[5] S. Savage, D. Wetherall, A. Karlin, T. Anderson, “Practical Network Support for IP Traceback”, Dpt of Computer Science and Engineering, University of Washington, 2001.

[6] D. Song, A. Perrig, “Advanced and Authenticated Marking Schemes for IP Traceback”, Computer Science Dpt, University of California, Berkeley, June 2000.

[7] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, S. Kent, W. Strayer, “Hash-Based IP Traceback”, BBN Technologies, 2001.

[8] D. Dean, M. Franklin, A. Stubblefield, “An Algebraic Approach to IP Traceback”,

[9] H. Burch, B. Cheswick, “Tracing Anonymous Packets to their Approximate Source”, In *Proc. USENIX LISA’00*, December 2000.

[10] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, S. Shenker, “Controlling High Bandwidth Aggregates in the Network”, AT&T Center for Internet Research and AT&T Labs Research, July 2001 (Draft).

[11] J. Ioannidis, S. Bellovin, “Implementing Pushback: Router-Based Defense Against DDoS Attacks”, AT&T Labs Research,