

Demystifying Denial-Of-Service attacks, part one

Abhishek Singh, CISSP 2005-12-14

Demystifying Denial-Of-Service attacks, part one By Abhishek Singh, CISSP This paper provides an introduction to Denial of Service (DoS) attacks, their methodologies, common prevention techniques, and how they differ from Distributed Denial of Service (DDoS) Attacks. This article is intended to be a broad overview for the beginner or intermediate-level administrator on the different types of DoS attacks.

1. Definitions

We begin by defining Denial of Service and Distributed Denial of Service.

1.1 What is a DoS?

As the name implies, DoS is a Denial of Service to a victim trying to access a resource. In many cases it can be safely said that the attack requires a protocol flaw as well as some kind of network amplification.

Denial of Services is also an attack on a computer system or network that causes a loss of service to users, typically the loss of network connectivity and services through the the consumption of bandwidth of the victim network, or the overloading the computational resources of the victim system. (see the [Wikipedia definition](#))

The motivation for DoS attacks is not to break into a system. Instead, it is to deny the legitimate use of the system or network to others who need its services. One can say that this will typically happen through one of the following means:

1. Crashing the system.
2. Deny communication between systems.
3. Bring the network or the system down or have it operate at a reduced speed which affects productivity.
4. Hang the system, which is more dangerous than crashing since there is no automatic reboot. Productivity can be disrupted indefinitely.

DoS attacks can also be major components of other type of attacks.

1.2 What is a Distributed DoS?

A Distributed DoS (DDoS) is the combined effort of several machines to bring down victim. In many cases there is a master machine that launches the attack to zombie machines that are part of a bot network, as shown below in Figure 1. Some bot networks contain many thousands of machines used to launch an attack.

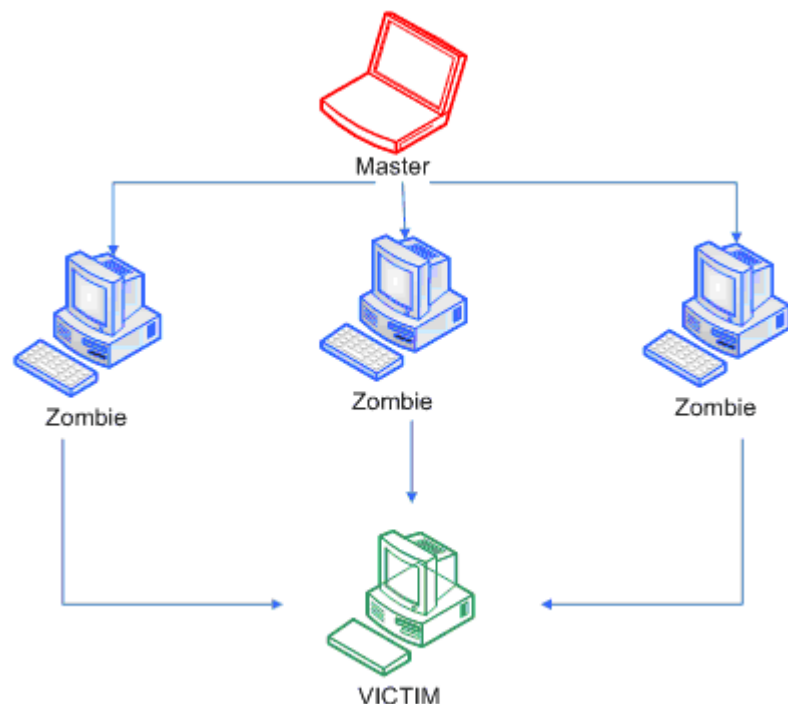


Figure 1. A DDoS attack in operation.

With DoS and DDoS defined, we'll now take a look at attacks that affect the consumption of resources, such as Smurf attacks, and then address attacks like SYN Flood that affect network connectivity.

Note that the consumption of resources is most evident when it involves the exploitation of bandwidth, CPU usage, memory, disk space, or access to other computers and resources.

2. Bandwidth exhaustion attacks

A bandwidth exhaustion attack is where an attacker tries to consume the available bandwidth of a network by sending a flood of packets. This is most often accomplished with the help of several other machines. There is soon a flood of malicious nonsense packets on the network in large quantity, whereby the chances of survival of any good, legitimate packets becomes remote. Eventually the network becomes choked with these packets, and the network is effectively cutoff from the Internet and services are denied.

An ideal example of a bandwidth exhaustion attack would be Smurf attacks. Consider a scenario with an ISP and three clients, as shown below in Figure 2. In this scenario, the ISP receives extensive traffic for client 2 on its backbone. Since the connection to client 2 is of

limited capacity and smaller than the ISP's backbone, it can't push all the data received for client 2 through the link to client 2. Therefore it will start to drop packets, and the TCP connections will lead to retransmissions of the lost/dropped packets. There will be a time when a legitimate host wants to connect to Client 2's network, but this will timeout and hence a DoS will occur.

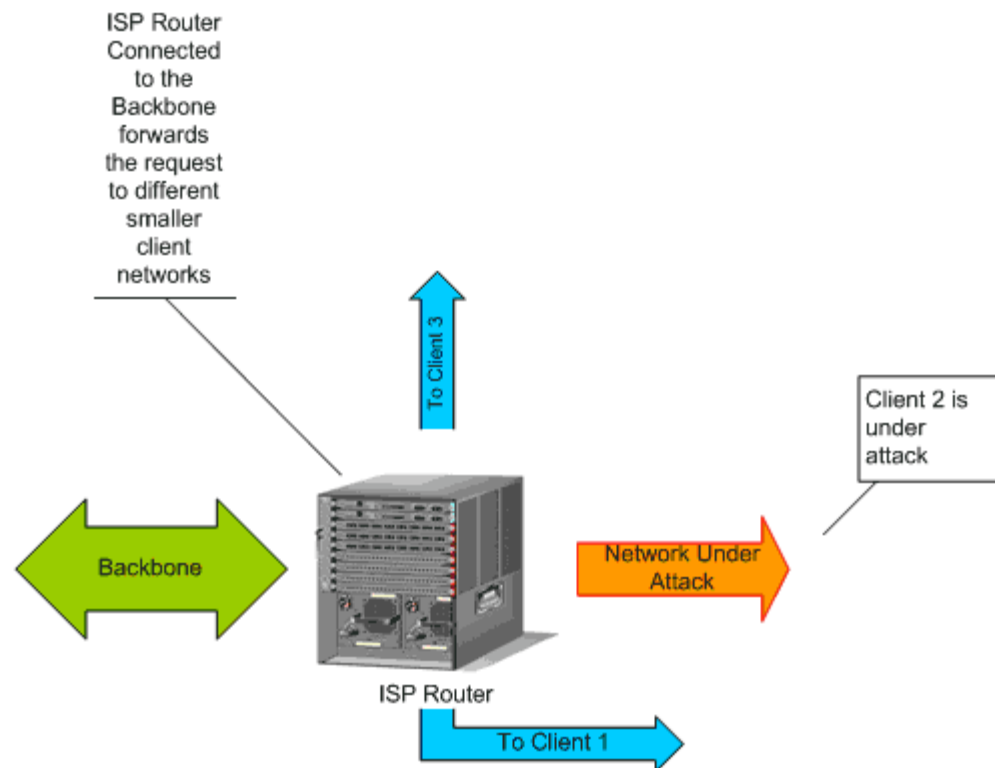


Figure 2. Scenario for a bandwidth exhaustion attack.

2.1 Example: Smurf attack

Named after a popular program which generates this attack, an ICMP echo request is sent to a broadcast network address (acting as an amplifying agent) with the source address of the victim spoofed. This results in a storm of replies from that network which, if large enough, has the power to take the victim's network down. It is to be noted that there is not much a victim can do about this attack since the link is simply overloaded with packets.

There are always three parts of a Smurf Attack:

1. Attacker

2. Amplifier - a router
3. Victim

This attack succeeds because the amplifier is misconfigugred to forward the directed broadcasts.

Suppose the address range 172.30.164.0 to 172.30.164.255 is assigned to a company which has an amplifier, and an attacker sends packets with destination 172.20.164.255. All the routers and systems from attacker to the amplifier will not see the difference between this IP and 172.30.164.10 (an IP from the range). The packet reaches the amplifier and the amplifier notices that this is the broadcast address, so it forwards the request to all the systems on the network/subnet. This is known as directed broadcast.

The two crucial components of this attack were:

1. A misconfigured router forwarding the broadcast request to the subnet.
2. Machines that will respond to this ICMP broadcast request.

Going deeper we can see that the amplifier also makes itself and its network a victim of this attack.

Victims are typically chosen by attackers from IRC where bots (automated programs) are kept to look for the address of victims. Hackers often exchange the information about amplifiers with each other so when a mass attack takes palce it usually appears to come from all over the globe.

Powertech provides [realtime statistics of the top amplifiers](#) currently on the Internet.

Below is a typical depiction of the dumps at the Victim. These are ICMP Echo replies received at the Victim's end. Then Figure 3 provides an overview of a Smurf attack.

```
10:10:17.100000 172.30.164.1 > victim: icmp: echo reply
10:10:18.300000 172.30.164.76 > victim: icmp: echo reply
10:10:18.310000 172.30.164.10 > victim: icmp: echo reply
10:10:19.110000 172.30.164.223 > victim: icmp: echo reply
10:11:09.190000 172.30.164.51 > victim: icmp: echo reply
10:11:09.240000 172.30.164.18 > victim: icmp: echo reply
10:11:10.110000 172.30.164.98 > victim: icmp: echo reply
10:11:10.600000 172.30.164.18 > victim: icmp: echo reply
10:11:10.790000 172.30.164.240 > victim: icmp: echo reply
```

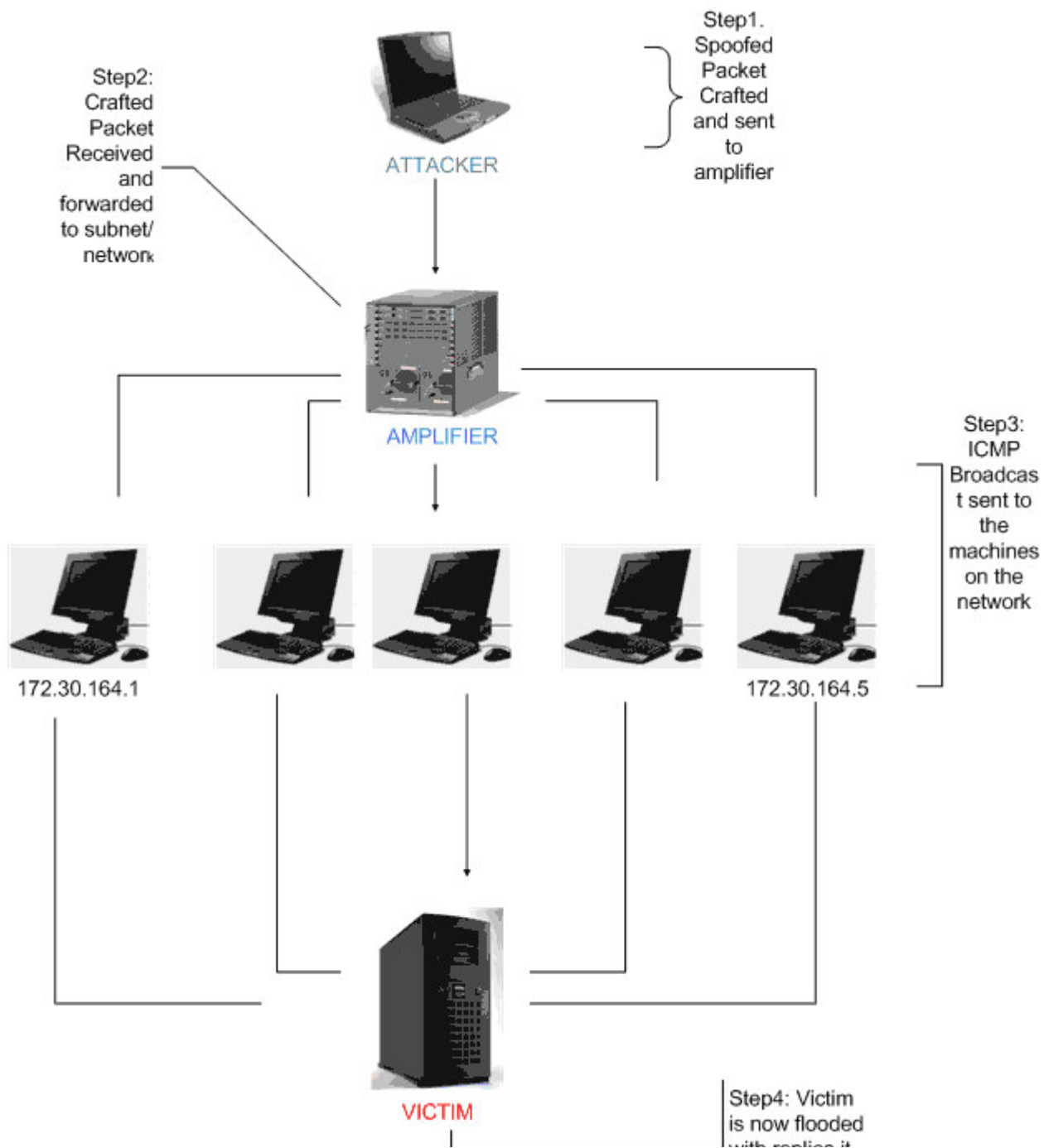


Figure 3. Typical directed broadcast Smurf attack.

2.2 How to protect against Smurf attacks

Step 1. Amplifier Configuration. The router should be configured so that it does not forward directed broadcasts onto networks. It is important to note that the broadcast has to be disabled on all the routers and not merely just the external ones. Command "no ip directed-broadcast" on Cisco routers should do the job in most cases. This will also ensure that employees on the internal network won't be able to launch Smurf attacks. However it is also advisable that one has a filtering device (such as a firewall) on the perimeter, thereby providing an extra layer of security.

Step 2. Configure the server operating systems. Servers should be configured so that they will not respond to a directed broadcast request. FreeBSD is one such system which by default does not respond to this request. Other systems can be similarly configured, and this will be discussed in the next section.

Step 3. Victim issues. As mentioned earlier, not much can be done at the victim's end and damage will be done unless victim's ISP takes some actions to block these ICMP Echo Reply floods. Even if the victim's parameter router denies the ICMP Echo Reply, the link from the ISP to the victim's site will suffer.

2.3 ICMP Ping Flood attacks

Ping Floods are where an attacker floods the victim's network with large number of ICMP Echo Requests - such as by flooding the network as fast as possible. In this scenario, filtering the incoming packets might help, however, if the victim is on a modem instead of a high-speed connection, nothing can be done. However the catch in this attack is that if not done properly the attacker can also be counter-attacked, so he needs to be on a faster network than the victim. In most cases, mitigating this attack involves isolating spoofed IPs. This attack is easy to perform since there are many tools on Internet and little knowledge is required to execute a ping flood.

2.4 Fraggle attacks

A Fraggle attack is a Smurf variant that uses UDP instead of ICMP. In this case, the ports echo, chargen, daytime, qotd are used to trigger responses. These ports are also susceptible to a [pingpong attack](#), and therefore these services should be turned off or blocked.

3. Network connectivity attacks

These attacks overload the victim so that its TCP/IP stack is not able to handle any further connections, and processing queues are completely full with nonsense malicious packets. As a consequence of this attack, legitimate connections are denied. One classic example of a

network connectivity attack is a SYN Flood.

3.1 Example: SYN Flood attacks

A SYN Flood is where an attacker sends packets with a spoofed source IP Address and a TCP SYN Flag set to the server (victim). Let's first assume that the attacker knows which ports are open on the server. Since the source IP is spoofed, the response sent to the SYN packet by the server will never receive a reply back. The server will keep waiting until it times out. If this happens for a very large number of connections the result will be a DoS, since the server won't be available for any legitimate connections and its resources will be choked.

As will be shown, this attack exploits a vulnerability of the TCP protocol, by the way in which the TCP three-way connection is established. This is shown below in Figure 4.

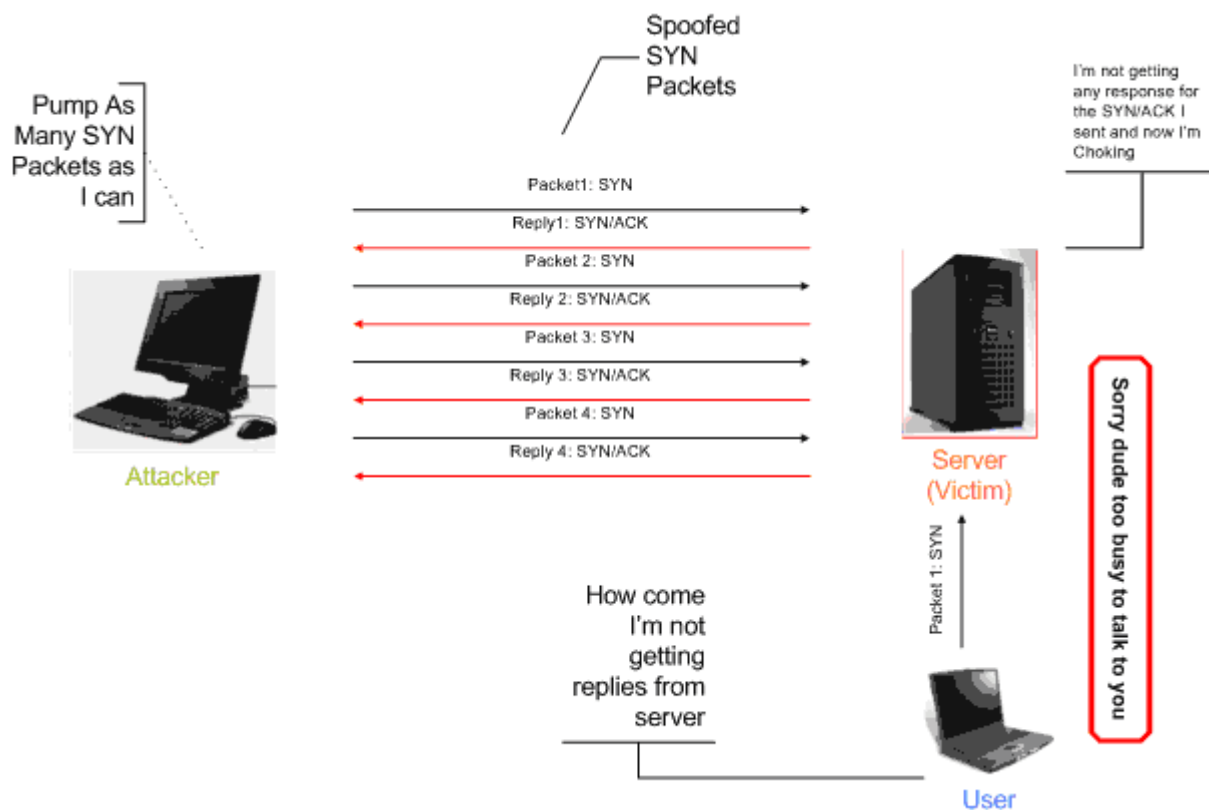


Figure 4. SYN Flood attack.

3.2 Protecting against SYN Flood attacks

There are several things that can be done to protect against SYN Flood attacks.

1. Decrease the TCP Connection Timeout on the victim server.
2. Use a filtering device, like a firewall, at the perimeter which works as an intermediary in forwarding the connections to the server.
3. Use of a server farm: this can also help in fighting the SYN Flood since you will have number of the servers to answer the request, but this also has limitations and overhead considerations.

A [detailed article on SYN Flood protection](#) by Mariusz Burdach was previously published on SecurityFocus. Therefore, only a short overview of SYN Flood protection will be discussed in this section.

3.2.1 Protecting Microsoft Windows from a SYN Flood attack

Microsoft Windows has a mechanism to detect and start SYN Flood protection. The SYN flooding attack protection feature detects symptoms of SYN flooding and responds by reducing the time the server spends on connection requests that it cannot acknowledge.

Specifically, TCP shortens the required interval between SYN-ACK (connection request acknowledgements) retransmissions. TCP retransmits SYN-ACKS when they are not answered. As a result, the allotted number of retransmissions is consumed more quickly and the unacknowledgeable connection request is discarded faster.

When enabled, the system monitors the connections maintained by TCP and starts the SYN attack flooding protection when the any of the following conditions, symptomatic of SYN flooding, are found:

- The total number of connections in the half-open (SYN-RCVD) state exceeds the value of **TcpMaxHalfOpen**
- The number of connections that remain in the half-open (SYN-RCVD) state even after a connection request has been retransmitted exceeds the value of **TcpMaxHalfOpenRetried**
- The number of connection requests the system refuses exceeds the value of **TcpMaxPortsExhausted**. The system must refuse all connection requests when its reserve of open connection ports runs out.

Microsoft suggests the following registry settings:

```
hkey_local_machine \system \currentcontrolset \services
\tcpip \parameters \synattackprotect=1 REG_DWORD

hkey_local_machine \system \currentcontrolset \services \tcpip
```

```

\parameters \tcpmaxconnectresponseretransmissions=2 REG_DWORD
hkey_local_machine \system \currentcontrolset \services \tcpip
\parameters \tcpmaxdataretransmissions=3 REG_DWORD

```

3.2.2 Check Point protections against a SYN Flood attack

In the first scenario, we look at Check Point as a simple proxy to the victim server. This is shown below in Figure 5.

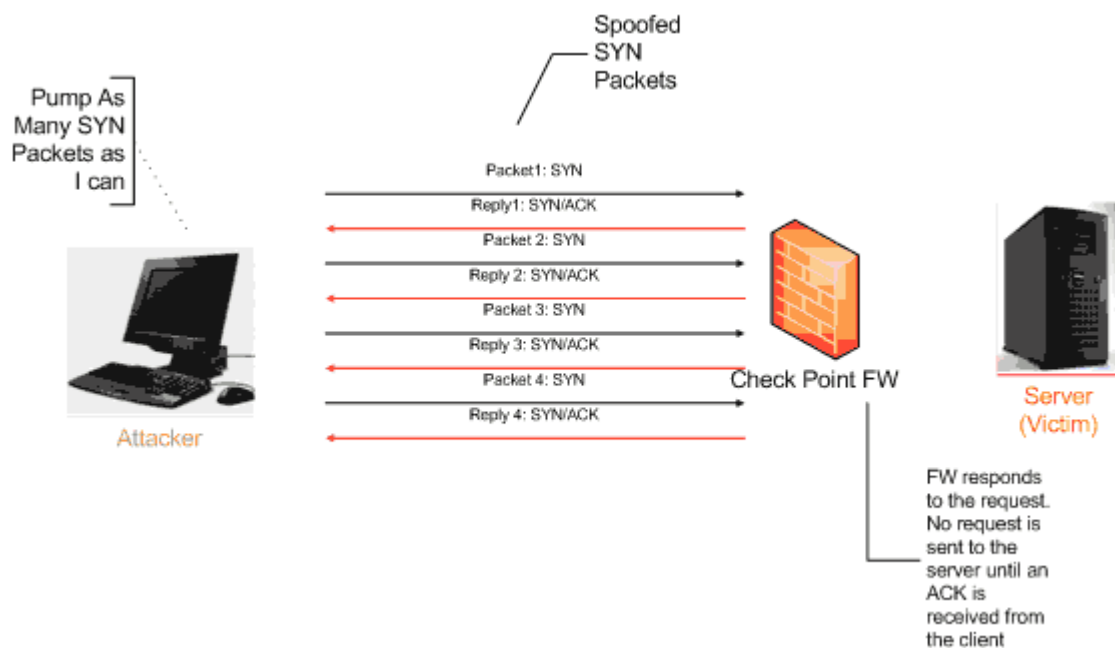


Figure 5. Check Point firewall acting as a proxy.

In this scenario Check Point acts a proxy to the server and responds to all the requests sent to the server. A request is forwarded to the server only if there is a corresponding ACK. The drawback of this configuration is that normally a perimeter firewall is very heavily loaded and this configuration will induce further load on it. The advantage is that the server will always be free to only take legitimate connections.

In the second scenario, we look at Check Point preventing a SYN Flood attack while residing in a transparent proxy configuration. This is shown below in Figure 6.

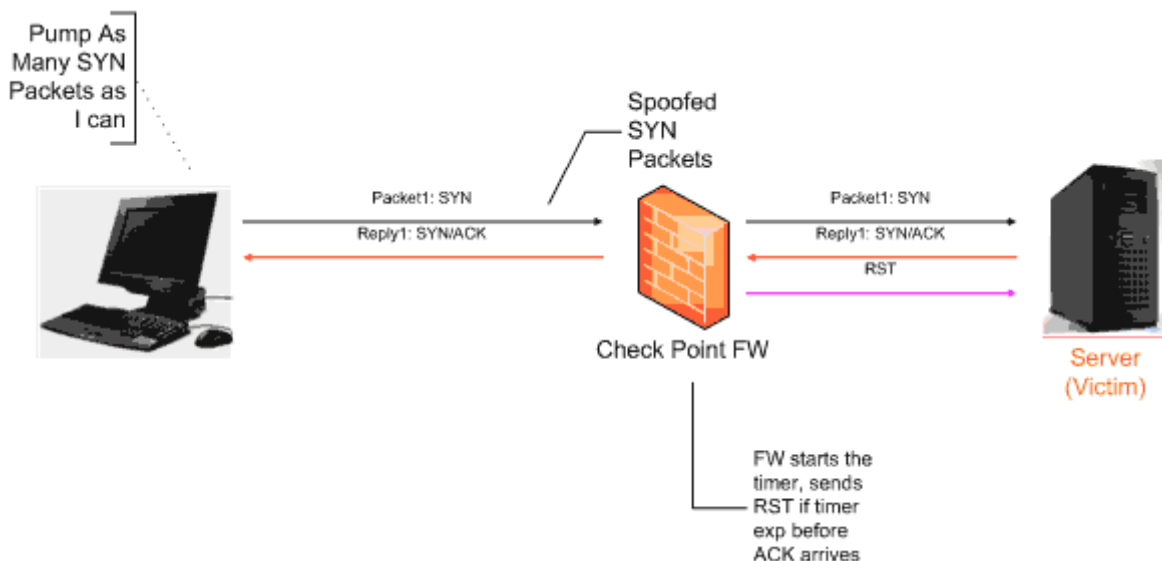


Figure 6. Check Point in transparent mode.

Here Check Point passes all the connections, irrespective of whether they are legitimate or not, to the server but also starts a timer once it sees a ACK/SYN from the server. If there is no corresponding ACK from the Client and the timer expires, the firewall will send a RST to the server thereby preventing its queue from overflowing with illegitimate connections. The advantage of this configuration is that load on firewall is reduced considerably as compared to previous configuration. The drawback, however, is that now the server sees all the connection attempts.

4. Concluding part one

In part one we've defined DoS and DDoS and looked at attacks that affect the consumption of bandwidth: Smurf attacks, ICMP ping floods, and Fraggle attacks. We've also taken a first look at attacks that affect network connectivity, such as SYN Flood attacks and some of the ways to prevent them.

Next time in part two, we'll look at the consumption of other precious resources such as CPU time, disk space, memory utilization, and then examine any vulnerable printers that may be DoS attack vectors on the network. We'll also discuss Teardrop attacks, LAND attacks, Ping-of-death, and finally discuss some common Win32 worms that have been used to build botnets that perform broad DDoS attacks. Finally, we'll discuss mitigation techniques and best practices for preventing DoS attacks. Stay tuned.

[Privacy Statement](#)
Copyright 2006, SecurityFocus