Project 4: "(E)DoS Attacks"

Secure Systems and Applications

2009

Ben Smeets
(C) Dept. of Electrical and Information Technology,
Lund University, Sweden

Introduction

A particular troublesome type of attack on networked (computer) systems is the so-called Denial-Of-Service (DOS) attack. The purpose of a DOS attack is to attack a system in such a way that the provided service is not more available or has become so poor that practical use of the service is no longer possible. DOS attacks have shown to be difficult to avoid but much has been done since the first DOS attacks were reported. Most modern servers have been designed to have some robustness against these types of attacks. Yet full protection has not been achieved and probably never will be achieved. In this project you will do some experiments and perform DOS attacks against a laboratory target machine. The purpose of these experiments is to let you understand the different types of DOS attacks and to give you insight in the difficulty to realize a complete protection against DOS attacks. You have to perform three attacks. The targets of these attacks are a Window95 box, a Linux box, and a Windows XP. DOS attacks are usually classified as

- System resource consuming attacks
- Bandwidth consuming attacks

In a system resource consuming attack critical resources on the attacked computer are driven by the attack into a state where either the resource stops working or becomes so overloaded that system throughput is almost zero. In a bandwidth consuming attack the service provided by attacked machine (often a server) is no longer available because the bandwidth available to the attacked machine is filled with bogus traffic. In the next section we describe the SYNC attack. It is today not a real threat anymore in most system but it illustrates the nature of a resource consuming attack. Before you start to work with this laboratory we advise you to read the Master Thesis report on Distributed Denial of Service attacks; a link to this report is available via the project4 web page (Thanks to the authors!).

It is important that you read and understand the rules of conduct for this project (see Assignments section). We give you a unique possibility to do experiments. Use this possibility under responsibility!

The TCP/SYN Flood Attack

(free from http://grc.com/files/drdos.pdf)

This is a now classic attack which will not affect most modern commercial computer systems. To understand this attack we recall some facts on TCP. TCP packets contain so-called "flag bits" which specify the contents and purpose of each packet. In particular the “SYN”, “ACK”, and “FIN” flags are important. A packet carrying the "SYN" (synchronize) flag bit is initiating a connection from the sender to the recipient. A packet with the "ACK" (acknowledge) flag bit set is acknowledging the receipt of information from the sender. A packet with the "FIN" (finish) bit set is terminating the connection from the sender to the recipient.

The establishment of a TCP connection typically requires the exchange of three Internet packets between two machines in an interchange known as the TCP Three-Way Handshake. Figure 1 illustrates the process:

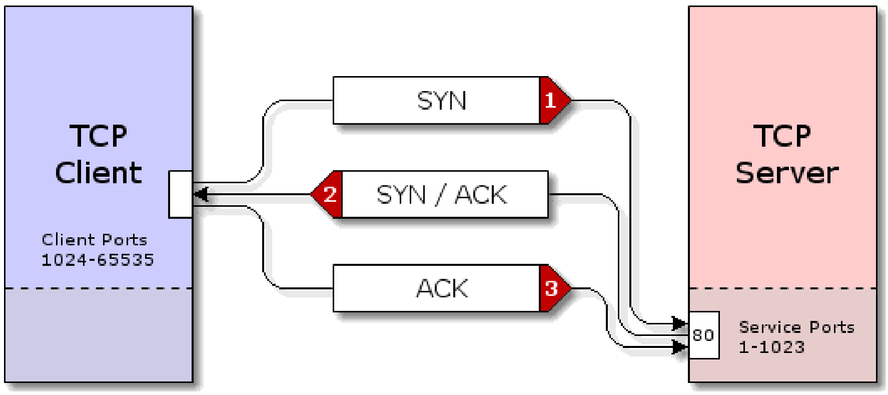


Figure 1: Three-way handshake.

SYN: A TCP client (such as a web browser, ftp client, etc.) initiates a connection with a TCP server by sending a "SYN" packet to the server.

As shown in Figure 1, this SYN packet is usually sent from the client's port, numbered between 1024 and 65535, to the server's port, numbered between 1 and 1023. Client programs running on the client machine ask the operating system to "assign them a port" for use in connecting to a remote server. This upper range of ports is referred to as the "client" or "ephemeral" port range. Similarly, server programs running on the server machine ask the operating system for the privilege of "listening" for incoming traffic on specific port numbers. This lower port range is known as "service ports." For example, a web server program typically listens for incoming packets on port 80 of its machine, and web browsing clients generally send their web packets to port 80 of remote servers.

Note that in addition to source and destination port numbers, each packet also contains the IP address of the machine which originated the packet (the Source IP) and the address of the machine to which the Internet's routers will forward the packet (the Destination IP).

SYN/ACK: When a connection-requesting SYN packet is received at an "open" TCP service port, the server's operating system replies with a connection-accepting "SYN/ACK" packet.

Although TCP connections are bi-directional (full duplex), each direction of the connection is set up and managed independently. For this reason, a TCP server replies to the client's connection-requesting SYN packet by ACK(nowledg)ing the client's packet and sending its own SYN to initiate a connection in the returning direction. These two messages are combined into a single combined "SYN/ACK" response packet.

The SYN/ACK packet is sent to the SYN's sender by exchanging the source and destination IPs from the SYN packet and placing them into the answering SYN/ACK packet. This sets the SYN/ACK packet's destination to the source IP of the SYN, which is exactly what we want.

Note that whereas the client's packet was sent to the server's service port — 80 in the example shown above — the server's replying packet is returned from the same service port. In other words, just as the source and destination IPs are exchanged in the returning packet, so are the source and destination ports.

The client's reception of the server's SYN/ACK packet confirms the server's willingness to accept the client's connection. It also confirms, for the client, that a round-trip path exists between the client and server. If the server had been unable or unwilling to accept the client's TCP connection, it would have replied with a RST/ACK (Reset Acknowledgement) packet, or an ICMP Port Unreachable packet, to inform the client that its connection request had been denied.

ACK: When the client receives the server's acknowledging SYN/ACK packet for the pending connection, it replies with an ACK packet.

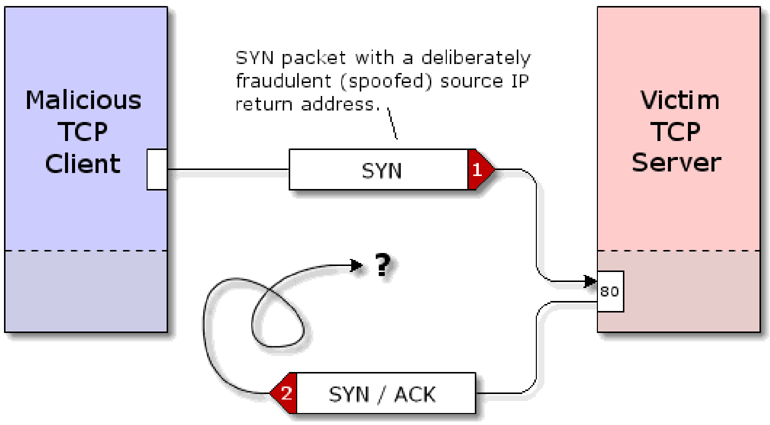
The client ACKnowledges the receipt of the SYN portion of the server's answering SYN/ACK by sending an ACK packet back to the server. At this point, from the client's perspective, a new two-way TCP connection has been established between the client and server, and data may now freely flow in either direction between the two TCP endpoints.

The server's reception of the client's ACK packet confirms to the server that its SYN/ACK packet was able to return to the client across the Internet's packet routing system. At this point, the server considers that a new two-way TCP connection has been established between the client and server and data may now flow freely in either direction between the two TCP endpoints.

Several years ago, a weakness in the TCP connection handling of many operating systems was discovered and exploited by malicious Internet hackers.
As shown in the TCP transaction diagram of Figure 1, the server's receipt of a client's SYN packet causes the server to prepare for a connection. It typically allocates resources in form of memory buffers for sending and receiving the connection's data, and it records the various details of the client's connection including the client's remote IP and connection port number. In this way, the server will be prepared to accept the client's final connection-opening ACK packet. Also, if the client's ACK packet should fail to arrive, the server will be able to resend

its SYN/ACK packet, presuming that it might have been lost or dropped by an intermediate Internet router.

But think about that for a minute. This means that memory and other significant server "connection resources" are allocated as a consequence of the receipt of a single Internet "SYN" packet. Clever but malicious Internet hackers figured that there had to be a limit to the number of "half open" connections a TCP server could handle, and they came up with a simple means for exceeding those limits, see Figure 2.


Malicious TCP Client
SYN packet with a deliberately fraudulent (spoofed) source IP return address.
SYN
1
Victim TCP Server
80
?
2
SYN / ACK
Figure 2: SYN attack.

Through the use of "Raw Sockets", the packet's "return address" (source IP) can be overridden and falsified. When a SYN packet with a spoofed source IP arrives at the server, it appears as any other valid connection request. The server will allocate the required memory buffers, record the information about the new connection, and send an answering SYN/ACK packet back to the client.
But since the source IP contained in the SYN packet was deliberately falsified (it is often a random number), the SYN/ACK will be sent to a random IP address on the Internet. If the packet were addressed to a valid IP, the machine at that address might reply with a "RST" (reset) packet to let the server know that it did not request a connection. But with over 4 billion Internet addresses, the chances are that there will be no machine at the address and the packet will be discarded.

The problem is, the server has no way of knowing that the malicious client's connection request was fraudulent, so it needs to treat it like any other valid pending connection. It needs to wait for some time for the client to complete the three-way handshake. If the ACK is not received, the server needs to resend the SYN/ACK in the belief that it might have been lost on its way back to the client.

As you can imagine, all of this connection management consumes valuable and limited resources in the server. Meanwhile, the attacking TCP client continues firing additional fraudulent SYN packets at the server, forcing it to accumulate a continuously growing pool of incomplete connections. At some point, the server will be unable to accommodate any more "half-open" connections and even valid connections will fail, since the server's ability to accept any connections will have been maliciously consumed.

It is possible to built-in protection mechanisms into the systems that will prevent this attack from causing problems. Yet it is good illustration of the nature of a resource consuming DOS attack

Assignments

Report the details of the attacks in the experiments in such a way that somebody else could repeat your experiments.

We do not have the (financial) resources to let all the groups do these experiments at the same time. There are a limited number of machines available (in the assigned rooms) and in some experiments your need 3-4 computers. Hence you have to coordinate with your colleagues.

IMPORTANT
Rules of conduct

Because of the potential harmful use the following rules of conduct applies to every student that logs in and accesses the software for this project.

It is forbidden to take copies of the software used in this laboratory. It is not allowed to conduct attacks on machines other then setup for this course. The department will monitor users and behavior and will react if misconduct is observed. In such a case the student will be exposed to a process to be expelled from this course. Furthermore, a student not complying to these rules of conduct may be made liable for damage inflicted.

Preparation 1: Read the Master Thesis and the laboratory manual

Purpose is to get enough background on DOS attacks and details to operate the software that is needed for this project.

Preparation 2: Installation of software that is needed on 3 or 4 PCs.

One PC will contain the ControlCenter, one run an Apache Server, and the others will run zombies.
Note1: That you need not to store VMware player on all machines (installing VMware player takes a while! Check if the player is already installed). In fact we may have preinstalled it already so installation should be not necessary.

Experiment I.
For this attack you will need only one computer. After all the software is installed, you run Debian OS in VMWare player. After the OS is loaded, it will show up the IP of the virtual computer the actual victim of this attack. You do not need to login.

Then, you should run Control Center (CC), and one Zombie. To initialize the zombie and make it ready for this attack, you should type in CC:

```
attack type pod2 destip <VictimIP> srcip <FakeIP> interval 1000
```

After this you can check the status of the Zombie by status. After that, the attack can be started from CC, and you will see the result;

Experiment II.
In this example you should load Win95 OS into VMWare. This time Win95 will be the victim, and you should _rst check its IP using winipcfg. This attack is similar to the previous one, but the type is winnuke. Also, instead of the fake IP you should give the Zombie's real IP;

Experiment III.
For this attack you will need to run CC on one computer, then 4-5 Zombies on each of the 3-4 other computers. Choose one more computer as a victim running Apache, and one computer for receiving returning packets from the victim. To initialize the attack you can type

```
attack type udpflood srcip <ReturningTrafficReceiverIP> destip <VictimIP> datasize <n>
```

First try with the packages of size 32 bytes, and then of 1470 bytes. When the attack is running, try to access the webserver on the victim computer. You should also observe the bandwidth of the switch through PRTG but it is sufficient if you measure the network traffic on the machine where Apache is running.

Do not forget to configure your Apache with the correct port number when installing it. Otherwise you have to change in httpd.config file in C:\Program Files\Apache Software Foundation\Apache2.2\conf and restart Apache. Port number 8080 and 1337 are open in the firewall

How to document?
In the final report for each attack you should include:

1. A short description of the attack, and the reasons why the attack is possible;
2. Describe the parties and roles;
3. Describe your steps;
4. Describe the result of the attack;
   For attack-III also give the graphs of bandwidth used at the attacked machine before and during the attack. Also give the switch bandwidth or the returning packet traffic, before and during the attack as well;
5. Explain how to prevent the attack.

Before you rush to do all this read the last comments in
http://www.eit.lth.se/fileadmin/eit/courses/eit015/AddComments4.pdf