SCS5607 ETHICAL HACKING AND DIGITAL FORENSICS

UNIT-II-
TCP / IP AND FIREWALLS

TCP/IP-CHECKSUM

The Transmission Control Protocol is designed to provide reliable data transfer between a pair of devices on an IP internetwork. Much of the effort required to ensure reliable delivery of data segments is of necessity focused on the problem of ensuring that data is not lost in transit. But there's another important critical impediment to the safe transmission of data: the risk of *errors* being introduced into a TCP segment during its travel across the internetwork.

Detecting Transmission Errors Using Checksums

If the data gets where it needs to go but is corrupted and we do not detect the corruption, this is in some ways worse than it never showing up at all. To provide basic protection against errors in transmission, TCP includes a 16-bit *Checksum* field in its header. The idea behind a checksum is very straight-forward: take a string of data bytes and add them all together. Then send this sum with the data stream and have the receiver check the sum. In TCP, a special algorithm is used to calculate this checksum by the device sending the segment; the same algorithm is then employed by the recipient to check the data it received and ensure that there were no errors.

The checksum calculation used by TCP is a bit different than a regular checksum algorithm. A conventional checksum is performed over all the bytes that the checksum is intended to protect, and can detect most bit errors in any of those fields. The designers of TCP wanted this bit error protection, but also desired to protect against other type of problems.

TCP Checksum Calculation and the TCP "Pseudo Header"

Advantages of the Pseudo Header Method

So, why bother with this “pseudo header”? The source and destination devices both compute the checksum using the fields in this pseudo header. This means that if, for any reason, the two devices don't use the same values for the pseudo header, the checksum will fail. Now, when we consider what's in the header, we find that this means the checksum now protects against not just errors in the TCP segment fields but also against:

- **Incorrect Segment Delivery:** If there is a mismatch in the *Destination Address* between what the source specified and what the destination that got the segment used, the checksum will fail. The same will happen if the *Source Address* does not match.

- **Incorrect Protocol:** If a datagram is routed to TCP that actually belongs to a different protocol for whatever reason, this can be immediately detected.

- **Incorrect Segment Length:** If part of the TCP segment has been omitted by accident, the lengths the source and destination used won't match and the checksum will fail.

What's clever about the pseudo header is that by using it for the checksum calculation, we can provide this protection without actually needing to send the fields in the pseudo header itself. This eliminates duplicating the IP fields used in the pseudo header within the TCP header, which would be redundant and wasteful of bandwidth. The drawback of the pseudo header method is that it makes checksum calculation take more time and effort (though this is not much of an issue today.)

In the context of today's modern, high-speed, highly-reliable networks, the use of the pseudo header sometimes seems “archaic”. How likely is it that a datagram will be delivered to the wrong address? Not very. At the time TCP was created, however, there was significant concern that there might not be proper “end-to-end” checking of the delivery of datagrams at the IP level. Including IP information in the TCP checksum was seen as a useful additional level of protection.

Of course, there is one interesting implication of the TCP pseudo header: it violates the architectural layering principles that the designers of TCP sought to respect in splitting TCP and IP up. For the checksum, TCP must know IP information that technically it “shouldn't”. TCP checksum calculation requires, for example, that the protocol number from the IP header be given to the TCP layer on the receiving device from the IP datagram that carried the segment. The TCP pseudo header is a good example of a case where strict layering was eschewed in favor of practicality.

Finally, TCP also supports an optional method of having two devices agree on an alternative checksum algorithm. This must be negotiated during connection establishment.

IP SPOOFING

What is IP Spoofing?

A technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host

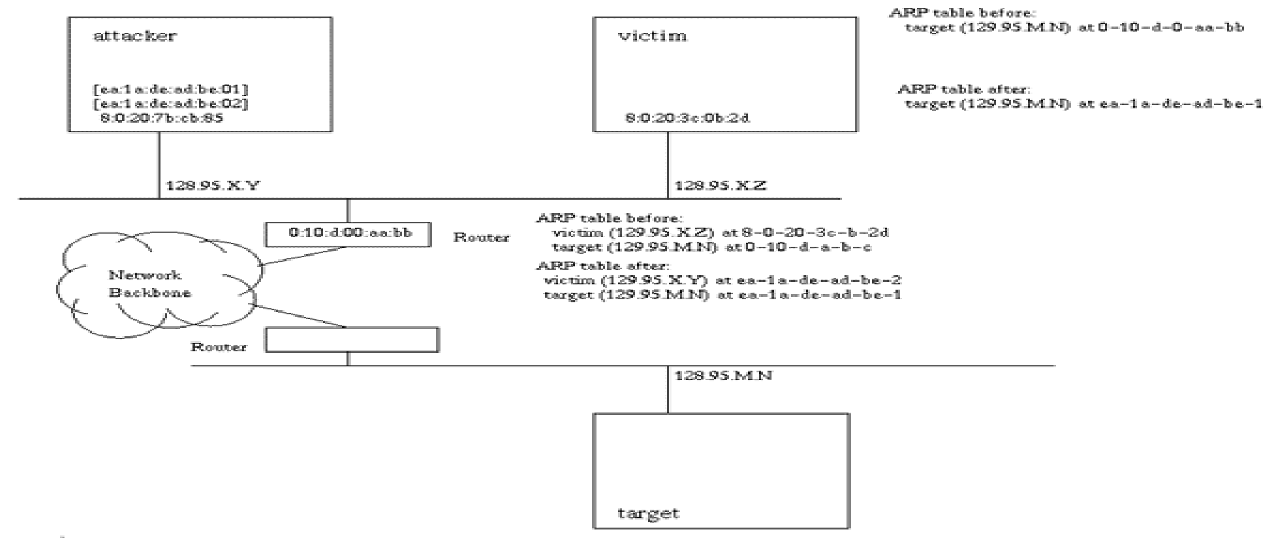
Base for IP spoofing

The concept of IP spoofing was discovered as a security weakness in the IP protocol which carries the Source IP address and the TCP protocol which contains port and sequencing information.

1. Non-Blind Spoofing

Takes place when the attacker is on the same subnet as the

victim. This allows the attacker to sniff packets making the next sequence number available to him.


attacker
[ea:1a:de:ad:be:01]
[ea:1a:de:ad:be:02]
8:0:20:7b:cb:85
victim
8:0:20:3c:0b:2d
ARP table before:
target (129.95.M.N) at 0-10-d-0-aa-bb
ARP table after:
target (129.95.M.N) at ea-1a-de-ad-be-1
128.95.X.Y
128.95.X.Z
0:10:d:00:aa:bb
Router
ARP table before:
victim (129.95.X.Z) at 8-0-20-3c-b-2d
target (129.95.M.N) at 0-10-d-a-b-c
ARP table after:
victim (129.95.X.Y) at ea-1a-de-ad-be-2
target (129.95.M.N) at ea-1a-de-ad-be-1
Network
Backbone
Router
128.95.M.N
target

The first stage of this attack is to prevent Victim from sending RST packets to host Target once the attack begins. This can be done by flooding the Victim with SYN messages.

Attacker initiates handshake message with the Target using the spoofed IP address. Target responds to the Victim with a SYN + ACK message which is sniffed by the Attacker to find out which sequence number is expected next for the ACK messages and sends it.

2. Blind Spoofing

Usually the attacker does not have access to the reply.

e.g.

Host C sends an IP datagram with the address of some other host (Host A) as the source address to Host B. Attacked host (B) replies to the legitimate host (A) .

The sequence and acknowledgement numbers from the victim are unreachable. In order to circumvent this, several packets are sent to the victim machine in order to sample sequence numbers.

Attacker connects to a TCP port on the victim prior to starting an attack to completes the three-way handshake, making sure that the initial sequence number (ISN) is recorded. This is repeated several times to determine the Round Trip Time (RTT) and the final ISN retained. The RTT is necessary to predict the next ISN.

A spoofed ACK message is sent from the attacker to the server:

- If the NSN is less than what is expected by the actual server, it considers it as a resent message and ignores it.
- If the NSN is correctly guessed, the target server responds back.
- If the NSN is greater than the expected NSN but it is within the window of packets expected by the server, the server waits until all the packets prior to that are received.
- If the NSN is greater than the expected NSN and is beyond the window of expected packets, the server just discards the packet.
- **3. ICMP redirect**
- The attacker sends a spoofed ICMP redirect message that appears to come from the host's default gateway.
- e.g. Host 192.168.1.4 sends a forged ICMP packet to host 192.168.1.3, saying the route through 192.168.1.4 is a better way to internet. The source IP address of this forged ICMP packet is the gateway's IP address 192.168.1.1. Then all the traffic from 192.168.1.3 to internet will go through 192.168.1.4.

Services Vulnerable to IP Spoofing

1. RPC (Remote Procedure Call services)

RPC multiplexes many services on top of one framework. Port mapper directs clients to the service that they want. Some of these services include NIS, NFS, and Exchange mail. Port mapper is usually secure, but the services below it often are not.

2. Any service that uses IP address authentication

3. X Window system

You can run programs on other people's displays, snoop their keystrokes and mouse movements, lock their screens etc.

4. R services suite (rlogin, rsh, etc.)

To prevent these sorts of attacks, users should have uncrackable passwords, and all shell access should be strongly authenticated and encrypted.

Port Scanning

- The process of examining a range of IP addresses to determine what services are running on a network.
- Finds open ports on a computer and the services running on it. For example
  - HTTP uses port 80 to connect to a Web service. IIS / Apache
- Port-scanning tools can be complex, must learn their strengths and weaknesses and understanding how and when you should use these tools.
- Find known vulnerabilities by using:
  - Common Vulnerabilities and Exposures (www.cve.mitre.org)
  - US-CERT (www.us-cert.gov) Web sites.
- There are also port-scanning tools that identify vulnerabilities, commercial tool.
  - AW Security Port Scanner (www.atelierweb.com)

Types of Port Scan

- SYN scan —In a normal TCP session, a packet is sent to another computer with the SYN flag set. The receiving computer sends back a packet with the

SYN/ACK flag set, indicating an acknowledgment. The sending computer then sends a packet with the ACK flag set.

- If the port the SYN packet is sent to is closed, the computer responds with an RST/ACK (reset/acknowledgment) packet.
- If an attacker's computer receives a SYN/ACK packet, it responds quickly with an RST/ACK packet, closing the session.
- This is done so that a full TCP connection is never made and logged as a transaction. In this sense, it's "stealthy." After all, attackers don't want a transaction logged showing their connection to the attacked computer and listing their IP addresses.

Port Scanning Tools

- Hundreds of port-scanning tools are available for both hackers and security testers.
- Not all are accurate, so using more than one port-scanning tool is recommended.
- One of the most popular port scanners and adds new features constantly, such as OS detection and fast multiple-probe ping scanning.
- Nmap also has a GUI front end called Zenmap that makes working with complex options easier.
- Open source
- Very Fast, use multiple threads
- Unicornscan can handle TCP, ICMP, and IP port scanning, it optimizes UDP scanning
  - www.unicornscan.org.
- Nessus and OpenVAS – other commercial and open source

- With the Fping tool (www.fping.com), you can ping multiple IP addresses simultaneously.
  - accepts a range of IP addresses entered at a command prompt,
  - Or create a file containing multiple IP addresses and use it
- For example, the fping -f ip_address.txt command uses ip_address.txt, which contains a list of IP addresses, as its input file.
- fping -g Beginning IPaddress Ending IPaddress. The -g parameter is used when no input file is available. For example, the fping -g 193.145.85.201 193.145.85.220 .

DNS SPOOFING

What is DNS Spoofing ?

DNS Spoofing is the art of making a DNS entry to point to an another IP than it would be supposed to point to. To understand better, let's see an example. You're on your web browser and wish to see the news on www.cnn.com, without to think of it, you just enter this URL in your address bar and press enter.

DNS Spoofing Tools

- Dsniff
- dnsspoof
- Example
  1. abc.com IP address is 10.0.0.1
  2. Make it spoof to respond 100.0.1.1
  3. In the text file dnssniff.txt write
  4. 100.0.1.1 abc.com
  5. [gateway]# dnsspoof -i eth0 -f /etc/dnssniff.txt
  6. [bash]# host abc.com abc.com has address of 100.0.1.1

- DNS Replies are verified for

  Coming from same IP address

  1. Coming to the same port from which request was sent
  2. Reply is for the same record as was asked in the previous question

     Transaction ID match

Now let's see how someone could poison the cache of our DNS Server. An attacker his running is own domain (attacker.net) with his own hacked DNS Server (ns.attacker.net) Note that I said hacked DNS Server because the attacker customized the records in his own DNS server, for instance one record could be www.cnn.com=81.81.81.81

1) The attacker sends a request to your DNS Server asking it to resolve www.attacker.net

2) Your DNS Server is not aware of this machine IP address, it doesn't belongs to his domain, so it needs to asks to the responsible name server.

3) The hacked DNS Server is replying to your DNS server, and at the same time, giving all his records (including his record concerning www.cnn.com) Note : this process is called a zone transfer.

4) The DNS server is not "poisoned". The attacker got his IP, but who cares, his goal was not to get the IP address of his web server but to force a zone transfer and make your DNS server poisoned as long as the cache will not be cleared or updated. 3

5) Now if you ask your DNS server, about www.cnn.com IP address it will give you 172.50.50.50, where the attacker run his own web server. Or even simple, the attacker could just run a bouncer forwarding all packets to the real web site and vice versa, so you would see the real web site, but all your traffic would be passing through the attacker's web site.

DNS ID Spoofing

We saw that when a machine X wants to communicate with a machine Y, the former always needs the latter IP address. However in most of cases, X only has the name of Y, in that case, the DNS protocol is used to resolve the name of Y into its IP address. Therefore, a DNS request is sent to a DNS Server declared at X, asking for the IP address of the machine Y.

Meanwhile, the machine X assigned a pseudo random identification number to its request which should be present in the answer from the DNS server. Then when the answer from the DNS server will be received by X, it will just have to compare both numbers if they're the same, in this case, the answer is taken as valid, otherwise it will be simply ignored by X. Does this concept is safe ? Not completely. Anyone could lead an attack getting this ID number.

If you're for example on LAN, someone who runs a sniffer could intercept DNS requests on the fly, see the request ID number and send you a fake reply with the correct ID number... but with the IP address of his choice. Then, without to realize it, the machine X will be talking to the IP of attacker's choice thinking it's Y.

By the way, the DNS protocol relies on UDP for requests (TCP is used only for zone transferts), which means that it is easy to send a packet coming from a fake IP since there are no SYN/ACK numbers (Unlike TCP, UDP doesn't provide a minimum of protection against IP spoofing).

DOS ATTACK

A distributed **denial-of-service** (**DDoS**) **attack** occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. Such an **attack** is often the result of multiple compromised systems (for example a botnet) flooding the targeted system with traffic.

Ping of Death

POD is an old denial of service attack that was quite effective back in the day, but is not really much of a threat anymore. Ping of Death has also been called Teardrop, and a few other names.

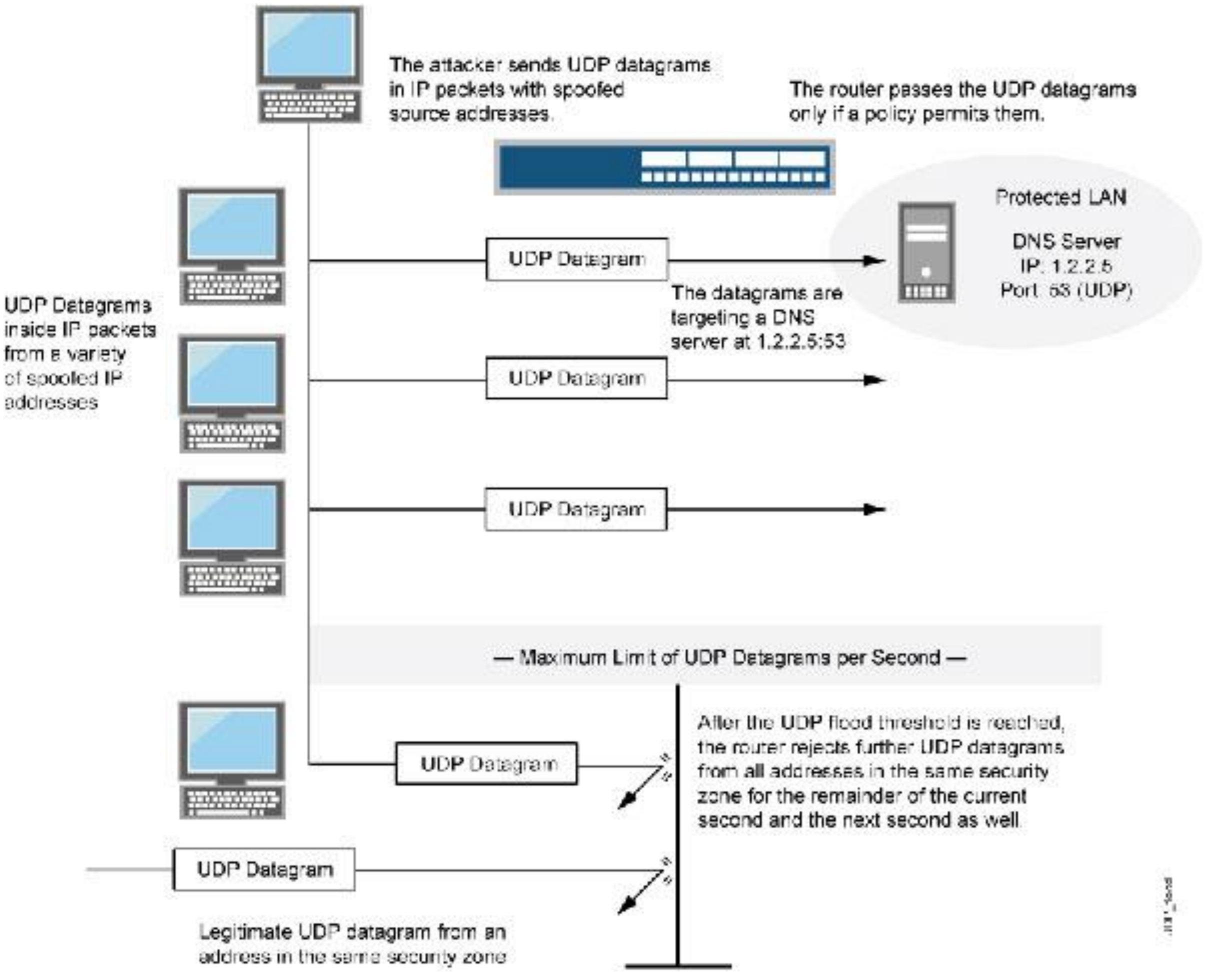
Within the IP protocol there are maximum byte allowances for packets (information) sent between two machines. The max allowance under IPv4 is 65,535 bytes. When a large packet is sent it is separated across multiple IP packets, and when reassembled creates a packet so big it will cause the receiving server to crash.

SYN Flood

This type of attack is a classic DDoS that sends rapid amounts of packets at a machine in an attempt to keep connections from being closed. The sending machine does not close the connection, and eventually that connection times out. If the attack is strong enough it will consume all resources on the server and send the website offline.

UDP Flood

A User Datagram Protocol Flood works by flooding ports on a target machine with packets that make the machine listen for applications on those ports and send back an ICMP packet.


The attacker sends UDP datagrams
in IP packets with spoofed
source addresses.
The router passes the UDP datagrams
only if a policy permits them.
Protected LAN
DNS Server
IP: 1.2.2.5
Port: 53 (UDP)
UDP Datagrams
inside IP packets
from a variety
of spoofed IP
addresses
UDP Datagram
The datagrams are
targeting a DNS
server at 1.2.2.5:53
UDP Datagram
UDP Datagram
— Maximum Limit of UDP Datagrams per Second —
After the UDP flood threshold is reached,
the router rejects further UDP datagrams
from all addresses in the same security
zone for the remainder of the current
second and the next second as well.
UDP Datagram
UDP Datagram
Legitimate UDP datagram from an
address in the same security zone

Reflected Attack

Forged packets are sent out to as many computers as possible. When the packets are received the computers reply, but because the packets are spoofed, instead of responding to the real sender, the machines will all attempt to communicate with the machine at the spoofed address. Eventually, if the attack is strong enough the server will shut down.

Nuke

This is an old distributed denial of service attack that uses corrupted ICMP packets with a modified ping utility to delivers bad packets to the target server. With enough volume the attack can be successful.

Slowloris

Types of DDoS attacks like these are way more complex than some of the other DDoS attacks we've talked about. Slowloris is a DDos toolkit that sends out partial requests to a target server in an effort to keep the connections open as long as possible. At the same time it does this, it sends out HTTP headers at certain intervals, which ramps up the requests, but never makes any connections. It doesn't take long for this type of DDoS attack to take down a website.

Peer-to-Peer Attacks

These types of attacks exploit peer-to-peer networks by maliciously redirecting legitimate visitors to the site or server they want to attack. If the attacker is able to pull it off with enough people, the resulting DDOS shuts down the site.

Unintentional DDoS

Exactly what it sounds like: you get so much traffic you overload your server and it poops out. This isn't necessarily a bad thing. It means your site is growing.

But it also means it's time to upgrade.

Degradation of Service Attacks

There really is only one purpose for this type of attack and that is overloading the server until it is so painstakingly slow it's all but worthless. This type of attack relies on the fact that no one is going to use a slow website for long, so the slower they can make it, the more of your visitors will find their way off your site.

What makes these types of attacks a pain is because it is hard to tell if you are experiencing a DDoS attack, or are just getting a boost of solid traffic — which is what every site owner is looking for. The key here is to analyze what your "visitors" are doing on the site and benchmark that with historical data. From there you should be able to tell if it is an attack or not.

Application Level Attacks

These are what's known as Layer 7 DDoS attacks. An attack like this will target the weakest points on your website. Layer 7 attacks are very difficult to stop without having the infrastructure, software, and knowledge to combat them.

Multi-Vector Attacks

A Multi-vector DDoS attack is quite possibly the most complex form of DDoS. This is where attackers not only blend attack strategies, but they often use a variety of tools as well. When you are faced with this type of DDoS attack you will notice the attacker pinpointing applications on your server, while at the same time flooding your site with bad traffic.

Zero Day DDoS

"Zero Day" attacks are a type of DDoS that is just being used. In other words, it is an attack being used for the first time.

DDoS Attacks Are Constantly Evolving

Distributed denial of service attacks are devastating to businesses. They motivations of attackers are evolving just as fast. From politically motivated to criminal weapon, DDoS attacks are used for a variety of purposes and target many applications: websites, email, and VoIP.

SYN Flood

- Attacker sends continuous stream of SYN packets to target
- Target allocates memory on its connection queue to keep track of half-open connections
- Attacker does not complete 3-way handshake, filling up all slots on connection queue of target machine

- If target machine has a very large connection queue, attacker can alternatively send sufficient amount of SYN packets to consume target machine's entire network bandwidth

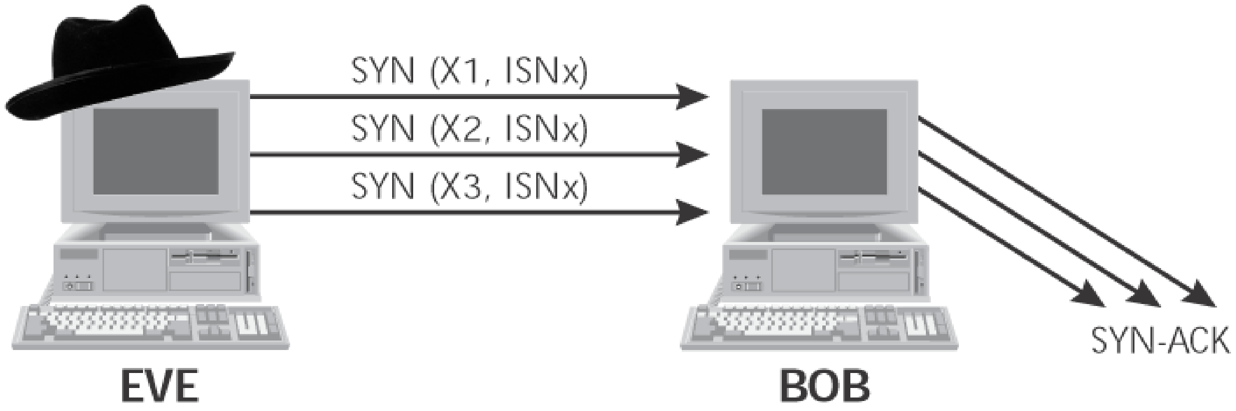


Figure shows -A SYN flood using spoofed source IP addresses that are not live

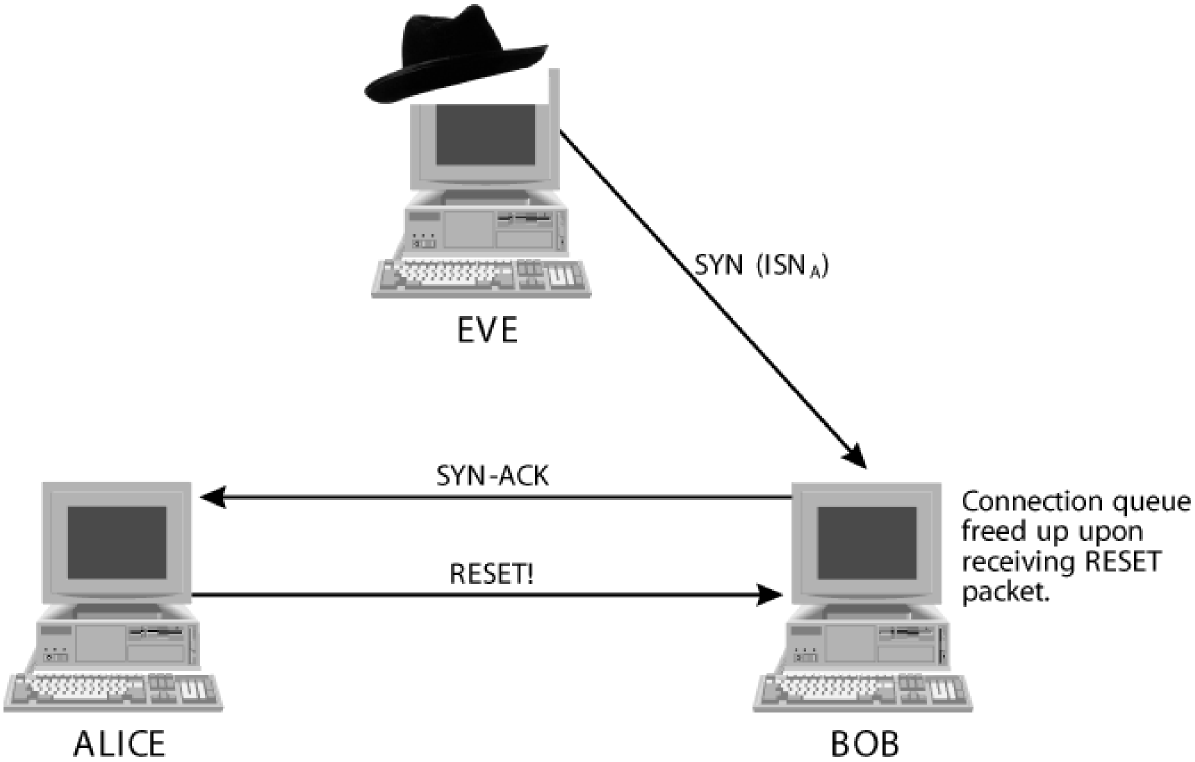
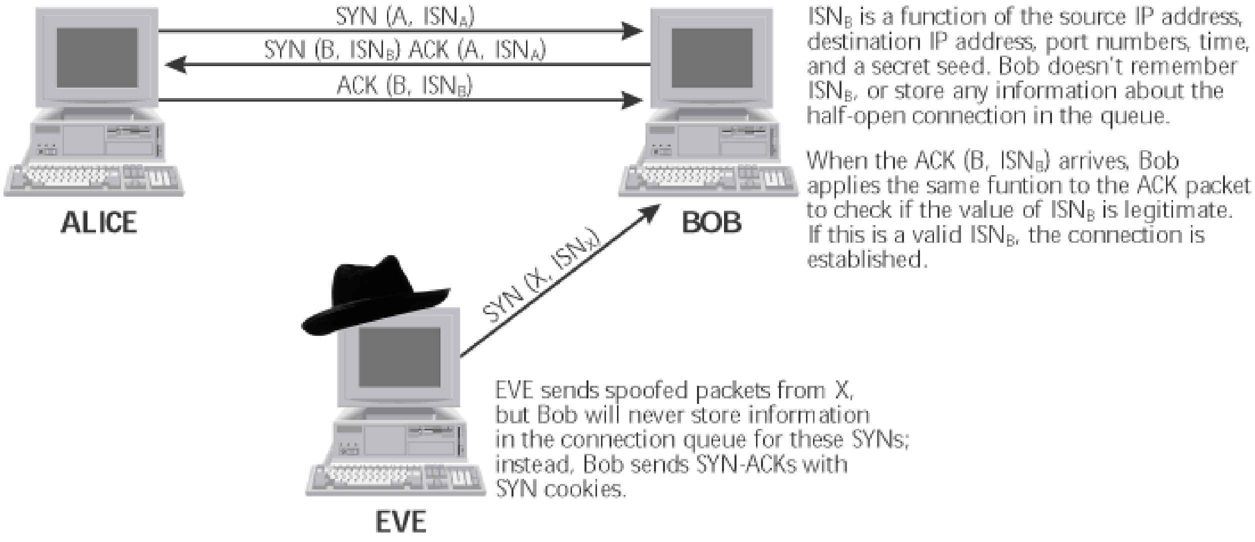


Figure shows-Attackers often spoof using unresponsive addresses to prevent RESET from freeing up the target's connection queue resources.

- Critical servers should have adequate network bandwidth and redundant paths
- Use two different ISPs for Internet connectivity
- Install traffic shaper to limit number of SYN packets
- Increase the size of connection queue or lower the timeout value to complete a half-open connection
    - http://www.nationwide.net/~aleph1/FAQ
- Use SYN cookies on Linux systems
    - A calculated value based on the source and destination IP address, port numbers, time, and a secret number
    - Calculated SYN cookie is loaded into the ISN of SYN-ACK response
    - no need to remember half-open connections on the connection queue

Activated via “echo 1 > /proc/sys/net/ipv4/tcp_syn cookies”



ISN$_B$ is a function of the source IP address, destination IP address, port numbers, time, and a secret seed. Bob doesn't remember ISN$_B$, or store any information about the half-open connection in the queue.

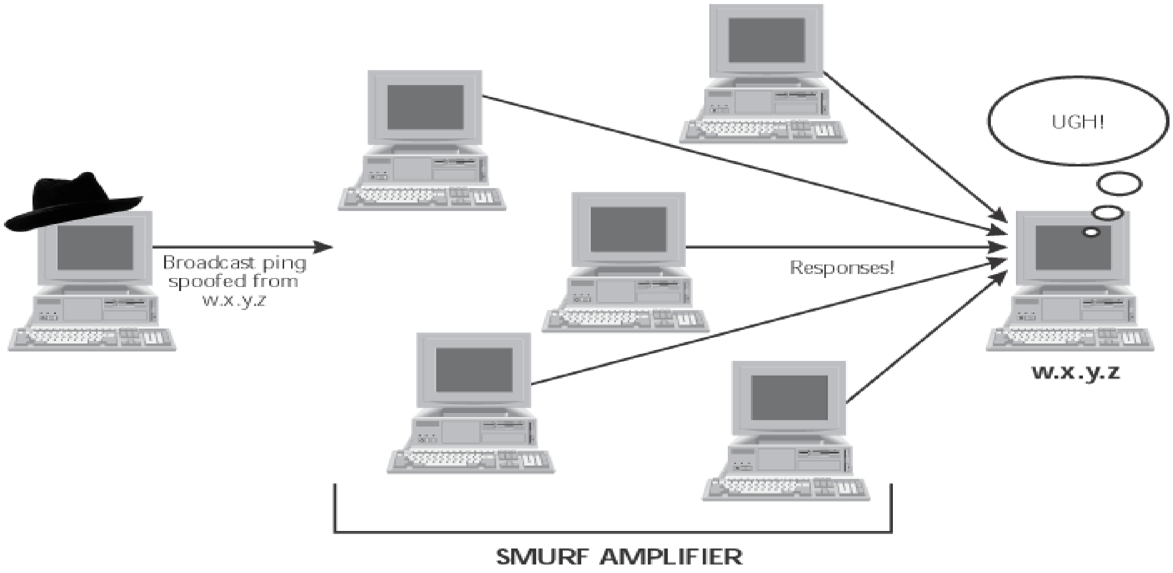
When the ACK (B, ISN$_B$) arrives, Bob applies the same funtion to the ACK packet to check if the value of ISN$_B$ is legitimate. If this is a valid ISN$_B$, the connection is established.

EVE sends spoofed packets from X, but Bob will never store information in the connection queue for these SYNs; instead, Bob sends SYN-ACKs with SYN cookies.

SYN cookies

Smurf Attacks

Aka directed broadcast attacks. Smurf attacks rely on an ICMP directed broadcast to create a flood of traffic on a victim. Attacker uses a spoofed source address of victim. Smurf attack is a DOS that consumes network bandwidth of victim.Smurf amplifier is a network that responds to directed broadcast messages



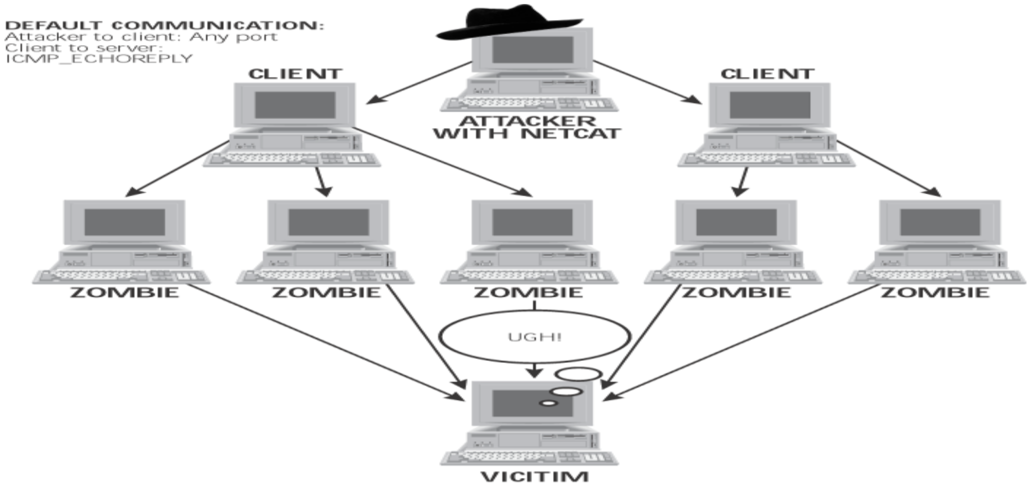
A Smurf attack results in a flood of the victim

Smurf-Attack Defenses

- http://www.pentic.net/denial-of-service/white-papers/smurf.cgi
- Install adequate bandwidth and redundant paths
- Filter ICMP messages at your border router
- Make sure that your network cannot be used as a Smurf amplifier
  - Test via http://www.powertech.no/smurf
  - Insert “no ip directed-broadcast” on Cisco border routers

Distributed Denial-of-Service Attacks (DDoS)

- More powerful than Smurf attacks. No limitation on number of machines used to launch attack. No limitation on bandwidth that can be consumed. Used against Amazon, eBay, Etrade, and Zdnet in Feb 2000.
- Before performing a DDOS flood, attack must take over a large number of victim machines (zombies) and install zombie software
- Attacker communicates with client machines which in turn send commands to zombies



DDoS Tools

- Tribe Flood Network
- TFN2K
- Blitznet
- MStream
- Trin00
- Trinity
- Shaft
- Stacheldraht (“barbed wire”)
  - Combines features of TFN and Trin00
- http://packetstorm/securify.com/distributed
- http://mixter.warrior2k.com
- Description of DDOS tools
- http://www.washington.edu/People/dad/

UDP Flooding

A **UDP flood** is a network **flood** and still one of the most common **floods** today. The attacker sends **UDP** packets, typically large ones, to single destination or to random ports.

Firewalls

a **firewall** is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules.

Firewalls are often categorized as either *network firewalls* or *host-based firewalls*. Network firewalls are a software appliance running on general purpose hardware or hardware-based firewall computer appliances that filter traffic between two or more networks. Host-based firewalls provide a layer of software on one host that controls network traffic in and out of that single machine.

Packet Filtering Firewall

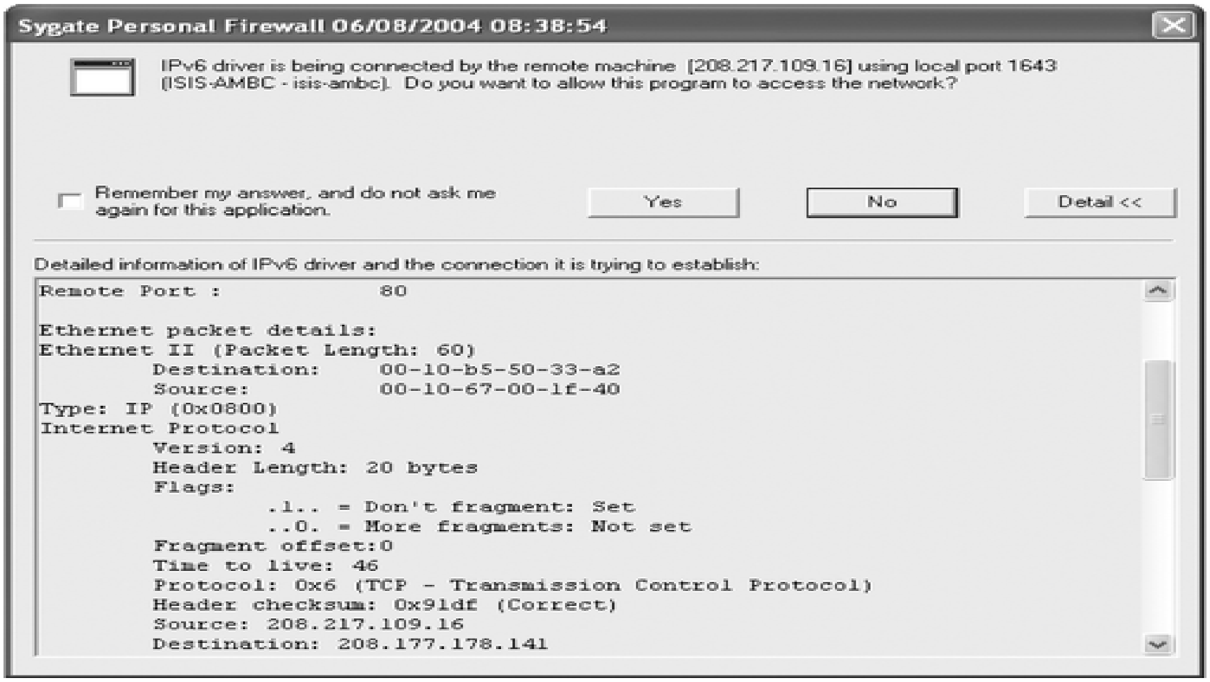
Packet filter: hardware or software designed to block or allow transmission of packets based on criteria such as port, IP address, protocol.To control movement of traffic through the network perimeter, know how packets are structured and what goes into packet headers.

Filter packet-by-packet, making decisions to forward/drop a packet based on:

- source IP address, destination IP address
- TCP/UDP source and destination port numbers
- ICMP message type
- TCP SYN and ACK bits

Packet filter inspects packet headers before sending packets on to specific locations within the network. A variety of hardware devices and software programs perform packet filtering:

- Routers: probably most common packet filters
- Operating systems: some have built-in utilities to filter packets on TCP/IP stack of the server software
- Software firewalls: most enterprise-level programs and personal firewalls filter packets



Sygate Personal Firewall 06/08/2004 08:38:54

IPv6 driver is being connected by the remote machine [208.217.109.16] using local port 1643 (ISIS-AMBC - isis-ambc). Do you want to allow this program to access the network?

□ Remember my answer, and do not ask me again for this application.

Detailed information of IPv6 driver and the connection it is trying to establish:

```
Remote Port :           80

Ethernet packet details:
Ethernet II (Packet Length: 60)
        Destination:    00-10-b5-50-33-a2
        Source:         00-10-67-00-1f-40
Type: IP (0x0800)
Internet Protocol
        Version: 4
        Header Length: 20 bytes
        Flags:
                .1.. = Don't fragment: Set
                ..0. = More fragments: Not set
        Fragment offset:0
        Time to live: 46
        Protocol: 0x6 (TCP - Transmission Control Protocol)
        Header checksum: 0x91df (Correct)
        Source: 208.217.109.16
        Destination: 208.177.178.141
```

Packet Filtering Rules

Packet filtering: procedure by which packet headers are inspected by a router or firewall to make a decision on whether to let the packet pass. Header information is evaluated and compared to rules that have been set up (Allow or Deny) Packet filters examine only the header of the packet (application proxies examine data in the packet).

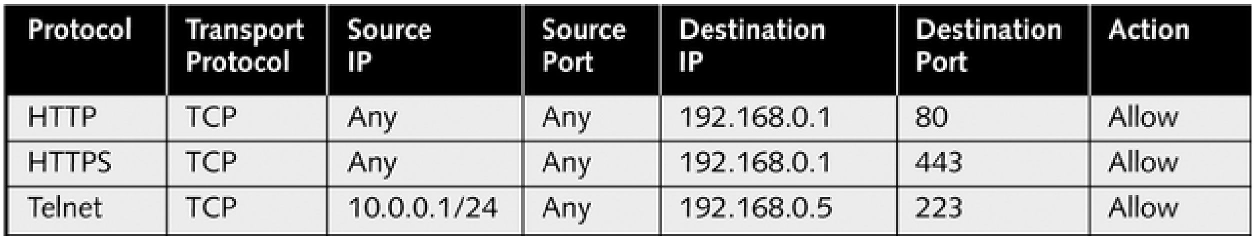
Drop all inbound connections; allow only outbound connections on Ports 80 (HTTP), 25 (SMTP), and 21 (FTP),Eliminate packets bound for ports that should not be available to the Internet (e.g., NetBIOS).Filter out ICMP redirect or echo (ping) messages (may indicate hackers are attempting to locate open ports or host IP addresses).Drop packets that use IP header source routing feature

Packet-Filtering Methods

Determines whether to block or allow packets—based on several criteria—without regard to whether a connection has been established. Also called static packet filtering. Useful for completely blocking traffic from a subnet or other network

Filtering on IP Header Criteria

- Packet’s source IP address.
- Destination or target IP address.
- Specify a protocol for the hosts to which you want to grant access.
- IP protocol ID field in the header.



Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	Any	Any	192.168.0.1	80	Allow
HTTPS	TCP	Any	Any	192.168.0.1	443	Allow
Telnet	TCP	10.0.0.1/24	Any	192.168.0.5	223	Allow

Stateful Packet Filtering

- Performs packet filtering based on contents of the data part of a packet and the header. Filter maintains a record of the state of a connection; allows only packets that result from connections that have already been established
- More sophisticated and secure. Has a rule base and a state table

stateful inspection) is a **firewall** that keeps track of the state of network connections (such as TCP streams, UDP communication) traveling across it.

Advantages:

- Simple
- Low cost
- Transparent to user

Disadvantages:

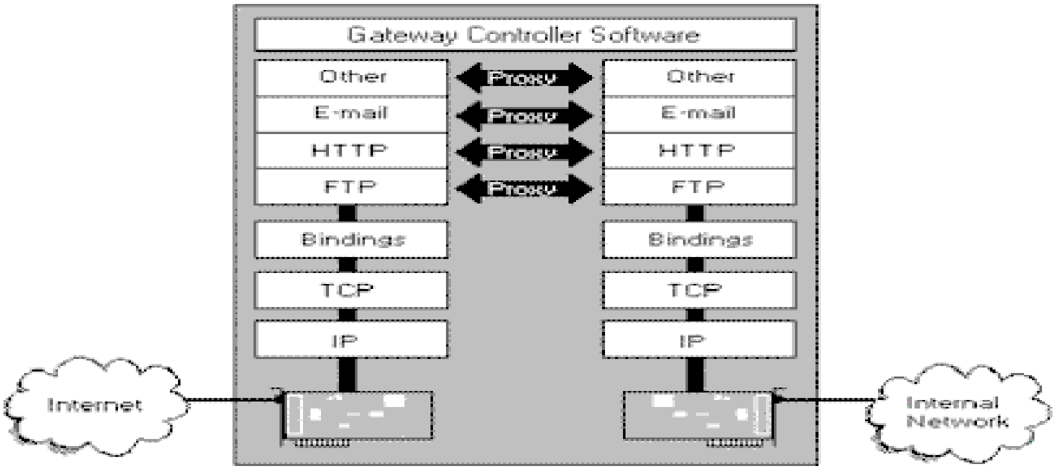
- Hard to configure filtering rules

- Hard to test filtering rules
- Don't hide network topology(due to transparency)
- May not be able to provide enough control over traffic
- Throughput of a router decreases as the number of filters increases

Application Proxy Firewalls

A proxy firewall is a network security system that protects network resources by filtering messages at the application layer. A proxy firewall may also be called an application firewall or gateway firewall.

Application Level Gateways (Proxy Server)



Firewall/Proxy Server

Advantages:

- complete control over each service (FTP/HTTP…)
- complete control over which services are permitted
- Strong user authentication (Smart Cards etc.)
- Easy to log and audit at the application level
- Filtering rules are easy to configure and test

Disadvantages:

- A separate proxy must be installed for each application-level service
- Not transparent to users.

Firewall's security policy

Embodied in the filters that allow or deny passages to network traffic

Filters are implemented as proxy programs.

- Application-level proxies
    - one for particular communication protocol
    - E.g., HTTP, FTP, SM
    - Can also filter based on IP addresses

Batch File Programming

Batch file programming is nothing but the Windows version of Unix Shell Programming. Let's start by understanding what happens when we give a DOS command. DOS is basically a file called command.com It is this file (command.com) which handles all DOS commands that you give at the DOS prompt---such as COPY, DIR, DEL etc. These commands are built in with the Command.com file. (Such commands which are built in are called internal commands.).

DOS has something called external commands too such as FORMAT, UNDELETE, BACKUP etc. So whenever we give a DOS command either internal or external, command.com either straightaway executes the command (Internal Commands) or calls an external separate program which executes the command for it and returns the result (External Commands.)

For example if you create a Batch file and save it with the filename batch.bat then all you need to execute the batch file is to type: C:\windows>batch.bat

Now let's execute this batch file and see what results it shows. Launch command.com (DOS) and execute the batch file by typing: C:\WINDOWS>batch_file_name You would get the following result: C:\WINDOWS>scandisk And Scandisk is launched. So now the you know the basic functioning of Batch files, let' move on to Batch file commands.

The REM Command The most simple basic Batch file command is the REM or the Remark command. It is used extensively by programmers to insert comments into their code to make it more readable and understandable. This command ignores anything there is on that line. Anything on the line after REM is not even displayed on the screen during execution.

It is normally not used in small easy to understand batch programs but is very useful in huge snippets of code with geek stuff loaded into it. So if we add Remarks to out first batch file, it will become: REM This batch file is my first batch program which launches the fav hacking tool;

Telnet telnet The only thing to keep in mind while using Remarks is to not go overboard and putting in too many of them into a single program as they tend to slow down the execution time of the batch commands.

ECHO: The Batch Printing Tool. The ECHO command is used for what the Print command is in other programming languages: To Display something on the screen. It can be used to tell the user what the bath file is currently doing. It is true that Batch programs display all commands it is executing but sometimes they are not enough and it is better to also insert ECHO commands which give a better description of what is presently being done.

Say for example the following batch program which is full of the ECHO command deletes all files in the c:\windows\temp directory: ECHO This Batch File deletes all unwanted Temporary files from your system

ECHO Now we go to the Windows\temp directory.

cd windows\temp ECHO Deleting unwanted temporary files....

del *.tmp ECHO Your System is Now Clean.

Now let's see what happens when we execute the above snippet of batch code. C:\WINDOWS>batch_file_name

C:\WINDOWS>ECHO This Batch File deletes all unwanted Temporary files from your system C:\WINDOWS>ECHO Now we go to the Windows\temp directory.

Now we go to the Windows\temp directory. C:\WINDOWS>cd windows\temp Invalid directory C:\WINDOWS>ECHO Deleting unwanted temporary files Deleting unwanted temporary files... C:\WINDOWS>del *.tmp C:\WINDOWS>ECHO Your System is Now Clean Your System is Now Clean The above is a big mess!

The problem is that DOS is displaying the executed command and also the statement within the ECHO command. To prevent DOS from displaying the command being executed, simply precede the batch file with the following command at the beginning of the file: ECHO OFF.

Useful References:

1. Kenneth C.Brancik, “Insider Computer Fraud”, Auerbach Publications Taylor & Francis, Group 2008.
2. Ankit Fadia, “Ethical Hacking”, Second Edition Macmillan India Ltd, 2006.
3. https://healholistic.files.wordpress.com/2013/08/batch-file-programming-ankit-fadia.pdf.
4. https://www.princeton.edu/~rblee/ELE572F02presentations/**DDoS**.**ppt**