



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Security in Transport Layer

1

Attacks in Transport Layer

Denial of Service in Transport Layer

Make a service unusable, usually by overloading the server or network

- Consume host resources
TCP SYN floods
- Consume bandwidth
UDP floods
- Crashing the victim
TCP options (unused, or used incorrectly)
- Forcing more computation
Taking long path in processing of packets

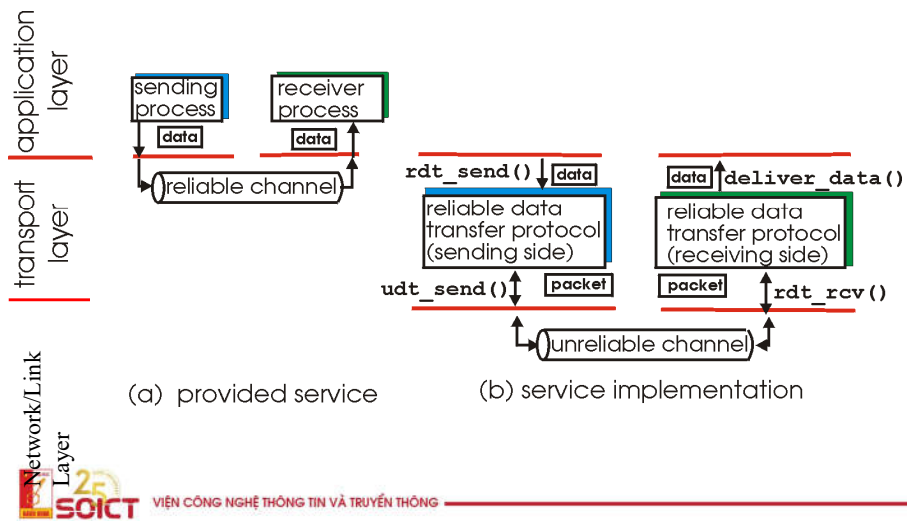


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

2

General Principles of Reliable data transfer

Important in app., transport, link layers



3



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TCP

Transmission Control Protocol

4

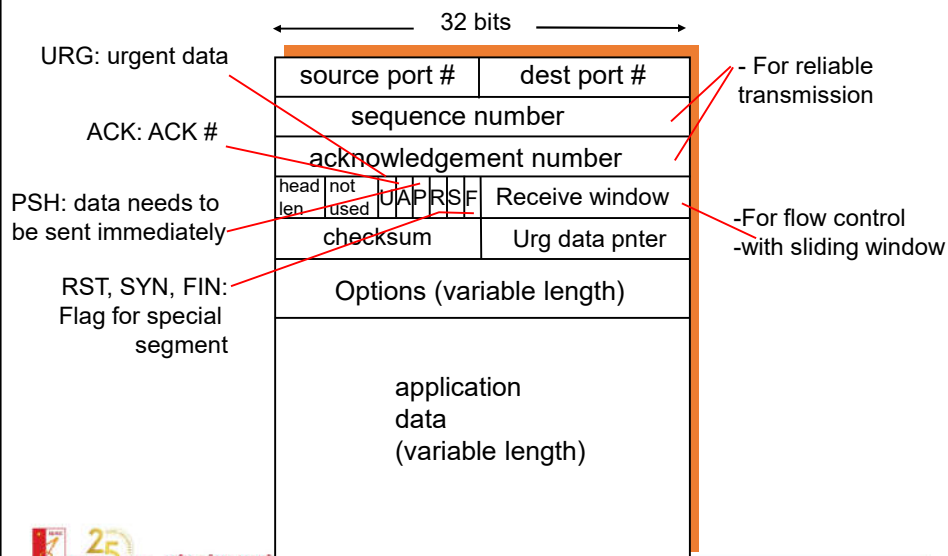
4

Overview of TCP

- Connection oriented
 - 3 steps hand-shake
- Data transmission in stream of byte, reliable
 - Use buffer
- Transmit data in pipeline
 - Increase the performance
- Flow control
 - Sliding windows
- Congestion control
 - Detect congestion and solve

5

TCP segment



6

How TCP provide reliable service?

- In order to assure if data arrives to destination:
 - Seq. #
 - Ack
- TCP cycle life:
 - Connection establishing
 - 3 steps
 - Data transmission
 - Close connection

7

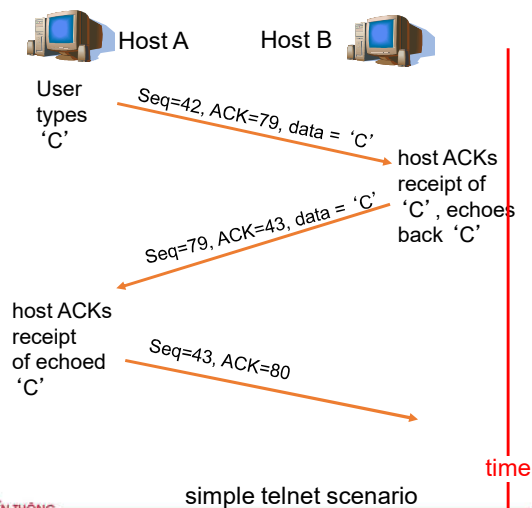
Acknowledgement in TCP

Seq. #:

- Index of the first byte of the segment in the data stream

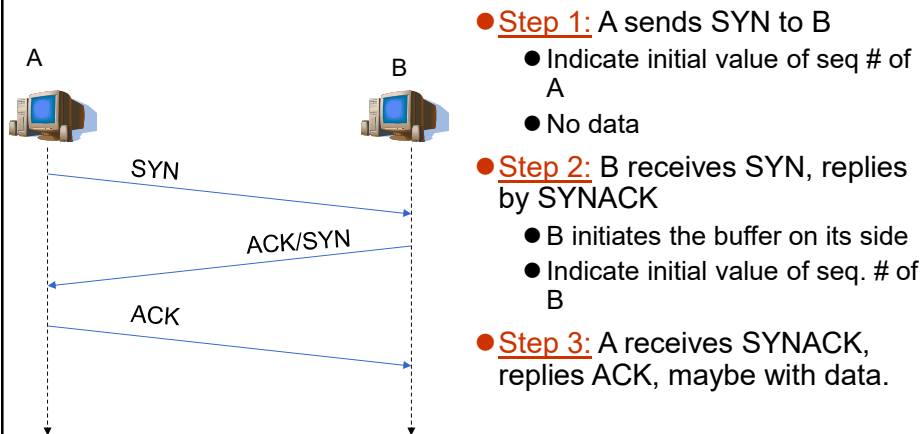
ACK:

- The index of the first byte expected to receive from the other-side
- Implicitly to confirm that the ACK senders have received well previous bytes



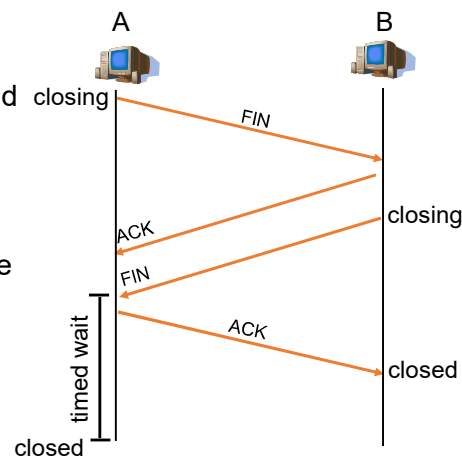
8

Connection establishing in TCP : 3 steps

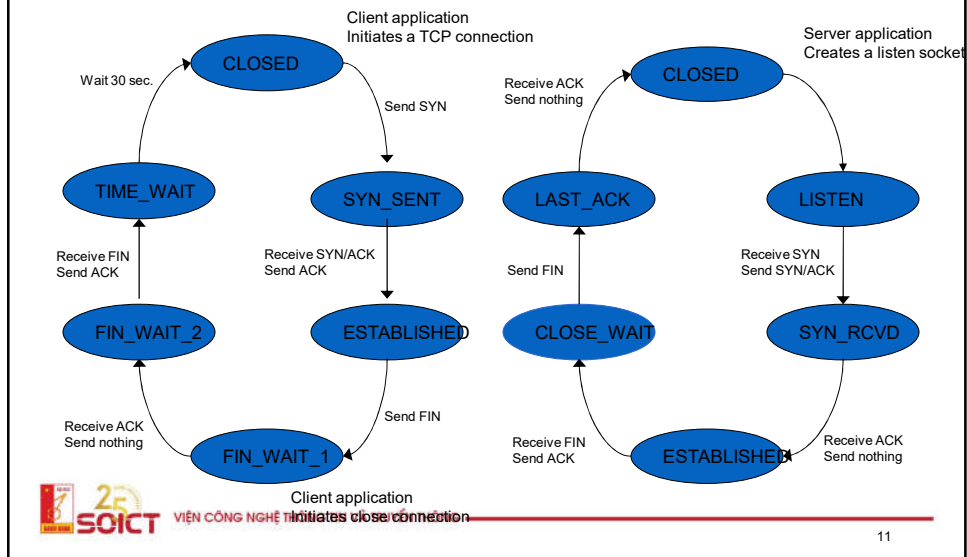


Close connection

- Step 1:** Send FIN to B
- Step 2:** B receives FIN, replies ACK, closes the connection and sends FIN.
- Step 3:** A receives FIN, replies ACK, go to "waiting".
- Bước 4:** B receives ACK. close connection



Simplified life cycle of TCP



11



25
SOICT

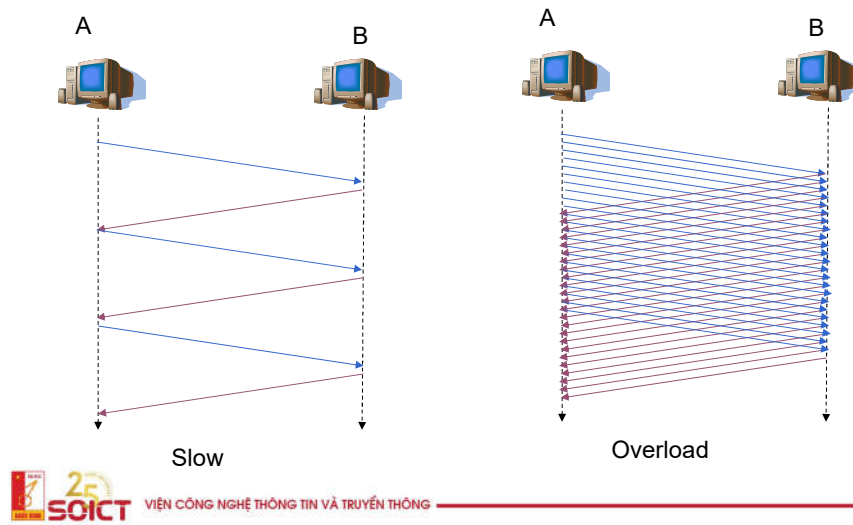
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Flow control

12

12

Flow control(1)



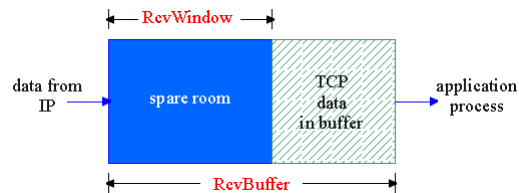
13

Flow control (2)

- Control the amount of data to be sent
 - Assure the best efficiency
 - Avoid overloading the receiver.
- Two windows
 - Rwnd: Receive window on receiver side
 - CWnd: Congestion window on sender side
- The maximum amount of data to be sent should be $\min(Rwnd, Cwnd)$

14

Flow control TCP



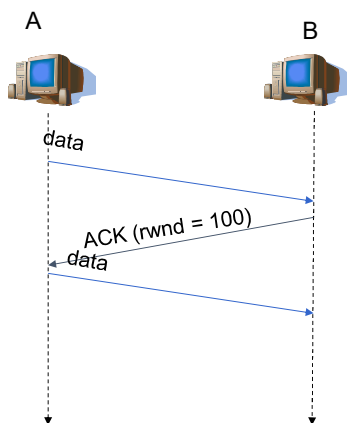
- Size of free buffer

= $Rwnd$

= $RcvBuffer - [LastByteRcvd - LastByteRead]$

15

Information exchanged on Rwnd



- Receiver inform regularly to senders the value of $Rwnd$ in acknowledgment segments

16

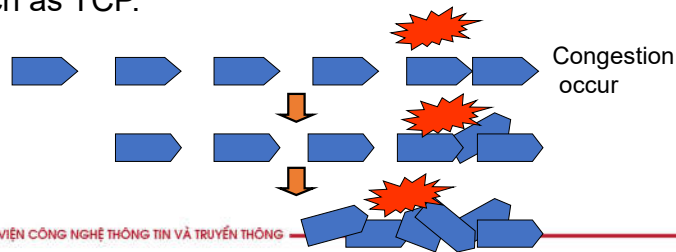
Congestion control in TCP

17

17

Overview of Congestion control

- When congestion happens?
 - Too many pairs of senders-receivers in the network
 - High traffic
- Consequence of congestion
 - Packet loss
 - Reduce of throughput, increase of delay
 - Network situation become worst with reliable protocol such as TCP.

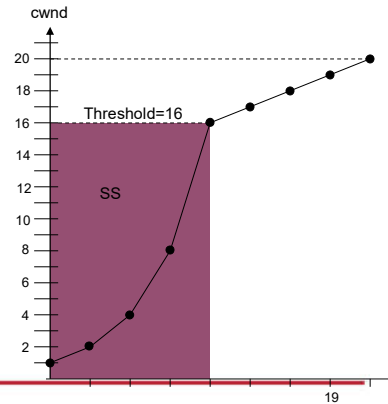


18

18

Principles of congestion control

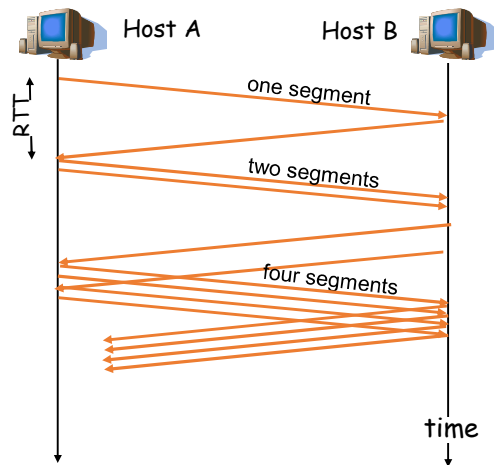
- Slow-start
 - Increases the transmission speed in exponential order
 - Increase until a threshold
- Congestion avoidance
 - Increase the transmission speed in linear order until congestion is detected
- How to detect the congestion?
 - By packets lost?



TCP Slow Start (1)

- Main idea
 - Initiate $cwnd = 1$ MSS (Maximum segment size)
 - Increase $cwnd = +1$ MSS after each reception of a ACK packet from the receiver.
 - Increase slowly but the speed increase in exponential order
- Increase until a threshold: $ssthresh$
- After that TCP move to congestion avoidance period

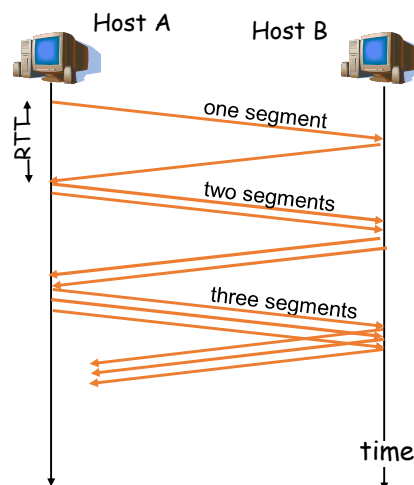
TCP Slow Start (2)



21

Congestion avoidance

- Main idea
 - Increase cwnd in additional order until cwnd reaches to congestion
- After each RTT, $cwnd = cwnd + 1 \text{ MSS}$



22

TCP reaction in congestion situation (1)

- Reduce the transmission speed
- How to detect the congestion?
 - If there are some re-transmits → There might be congestion
- When the source node need to re-transmit data?
 - Timeout!
 - When it receives multiple ACK for the same segment



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

23

23

TCP reaction in congestion situation(2)

- When sender reach timeout but still does not receive ACK for a segment
 - TCP sets ssthresh = $\frac{1}{2}$ current cwnd
 - TCP sets cwnd = 1 MSS
 - TCP move to slow start phase
- If sender receives 3 identical ACK
 - TCP sets ssthresh = $\frac{1}{2}$ current cwnd
 - TCP sets cwnd = ssthresh
 - TCP move to “congestion avoidance”

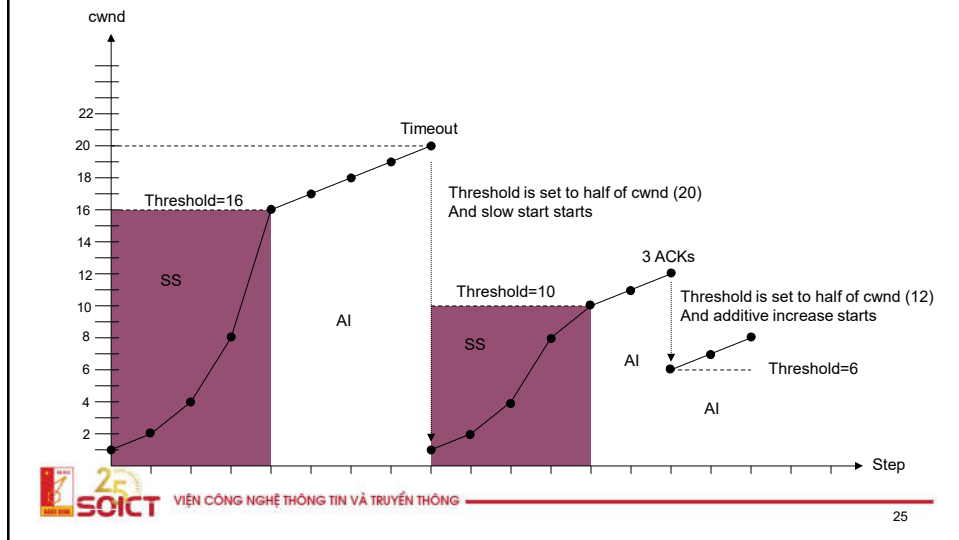


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

24

24

Congestion control – illustration



25

TCP SYN Flooding

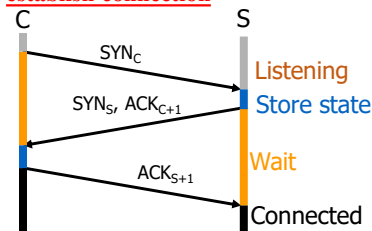
In a server that provides TCP connections for services such as Telnet, Web, Email, etc., lots of half-open TCP connections will cause a problem known as TCP SYN Flooding attack. This problem is due to the TCP 3-way hand-shaking protocol: A client initiates a TCP connection by sending a TCP SYN packet to the server in Step 1. The server upon receiving the TCP SYN packet replies with an ACK packet in Step 2. However, the client may not send an ACK packet to the server to complete the TCP 3-way hand-shaking protocol. If the client keeps sending the SYN packets, the server will eventually run out of resource to other TCP connection requests.



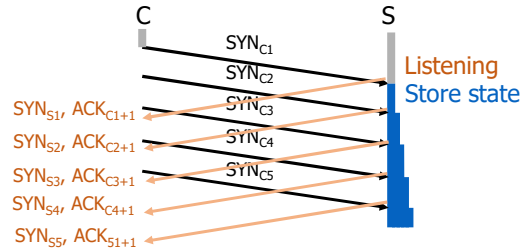
26

TCP SYN Flooding - Continue

TCP Three Way Handshake for establish connection



SYN Flooding



- The backlog queue is a large memory structure used to handle incoming packets with the SYN flag set until the moment the three-way handshake process is completed.
- An operating system allocates part of the system memory for every incoming connection. Every TCP port can handle a defined number of incoming requests. The backlog queue controls how many half-open connections can be handled by the operating system at the same time.
- When a maximum number of incoming connections is reached, subsequent requests are silently dropped by the operating system.

27

TCP SYN Flooding - Continue

Identifying the Attacker

The IP address of an attacking system is hidden because the source addresses in the SYN packets are often falsified. When the packet arrives at the server, there is no way to determine its true source IP address. Since the network forwards packets based on destination address, the only way to validate the source of a packet is to use input source filtering in the client side.

Attack Detection

Most of the operating systems provide a command line tool “netstat” to display protocol statistics and current TCP/IP network connections. The following command running on a Window 7 machine lists network connections. Pay attention to the state column. If there are lots of “SYN_RECEIVED” connections, the system is under attack. The SYN_RECEIVE state indicates that a connection request has been received from the network.

28

How to detect a TCP SYN attack

- The netstat command shows how many connections are currently in the half-open state. The half-open state is described as SYN_RECEIVED in Windows and as SYN_RECV in Unix systems.

```
# netstat -n -p TCP
```

```
- tcp    0    0 10.100.0.200:21      237.177.154.8:25882    SYN_RECV
- tcp    0    0 10.100.0.200:21      236.15.133.204:2577    SYN_RECV
- tcp    0    0 10.100.0.200:21      127.160.6.129:51748    SYN_RECV
- tcp    0    0 10.100.0.200:21      230.220.13.25:47393    SYN_RECV
- tcp    0    0 10.100.0.200:21      227.200.204.182:60427  SYN_RECV
- tcp    0    0 10.100.0.200:21      232.115.18.38:278      SYN_RECV
- tcp    0    0 10.100.0.200:21      229.116.95.96:5122     SYN_RECV
- tcp    0    0 10.100.0.200:21      236.219.139.207:49162  SYN_RECV
- tcp    0    0 10.100.0.200:21      238.100.72.228:37899   SYN_RECV  - ...
```

- How many half-open connections are in the backlog queue at the moment can be counted. In the example below, 769 connections (for TELNET) in the SYN_RECEIVED state are kept in the backlog queue.

```
# netstat -n -p TCP | grep SYN_RECV | grep :23 | wc -l 769
```

- The other method for detecting SYN attacks is to print TCP statistics and look at the TCP parameters which count dropped connection requests.

- TcpHalfOpenDrop parameter on a Sun Solaris machine.

```
# netstat -s -P tcp | grep tcpHalfOpenDrop      tcpHalfOpenDrop = 473
```

- It is important to note that every TCP port has its own backlog queue, but only one variable of the TCP/IP stack controls the size of backlog queues for all ports.

29

Built-in Protection for SYN Flooding

The most important parameter in Windows 2000 and also in Windows Server 2003 is SynAttackProtect. Enabling this parameter allows the operating system to handle incoming connections more efficiently. The protection can be set by adding a SynAttackProtect DWORD value to the following registry key:

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

- When a SYN attack is detected the SynAttackProtect parameter changes the behavior of the TCP/IP stack. This allows the operating system to handle more SYN requests. It works by disabling some socket options, adding additional delays to connection indications and changing the timeout for connection requests. When the value of SynAttackProtect is set to 1, the number of retransmissions is reduced. The recommended value of SynAttackProtect is 2, which additionally delays the indication of a connection to the Windows Socket until the three-way handshake is completed.
- By enabling the SynAttackProtect parameter we don't change the TCP/IP stack behavior until under a SYN attack. But even then, when SynAttackProtect starts to operate, the operating system can handle legitimate incoming connections.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

30

Built-in Protection for SYN Flooding - Continue

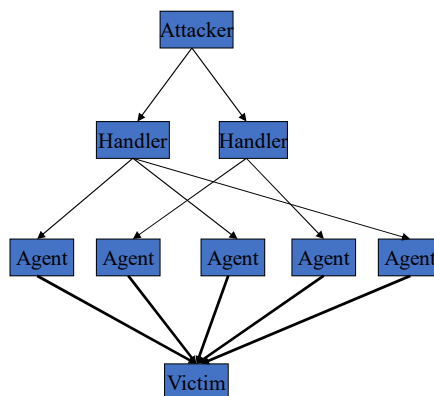
The operating system enables protection against SYN attacks automatically when it detects that values of the following three parameters are exceeded. These parameters are `TcpMaxHalfOpen`, `TcpMaxHalfOpenRetried` and `TcpMaxPortsExhausted`. To change the values of these parameters, first we have to add them to the same registry key as we made for `SynAttackProtect`.

- ❖ **TcpMaxHalfOpen** registry entry defines the maximum number of SYN RECEIVED states which can be handled concurrently before SYN protection starts working. The recommended value of this parameter is 100 for Windows 2000 Server and 500 for Windows 2000 Advanced Server.
- ❖ **TcpMaxHalfOpenRetried** defines the maximum number of half-open connections, for which the operating system has performed at least one retransmission, before SYN protection begins to operate. The recommended value is 80 for Windows 2000 Server, and 400 for Advanced Server.
- ❖ **TcpMaxPortsExhausted** registry entry defines the number of dropped SYN requests, after which the protection against


31

Distributed DoS

- The handlers are usually very high volume servers
 - Easy to hide the attack packets
- The agents are usually home users with DSL/Cable
 - Already infected and the agent installed
- Very difficult to track down the attacker
- How to differentiate between DDoS and Flash Crowd?
 - Flash Crowd: Many clients using a service legitimately
 - Slashdot Effect
 - Victoria Secret Webcast
 - Generally the flash crowd disappears when the network is flooded
 - Sources in flash crowd are clustered



32



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG


UDP User Datagram Protocol

33

33

“Best effort” protocols

- Why UDP?
 - No need to establish connection (cause delay)
 - Simple
 - Small header
 - No congestion control → send data as fast as possible
- Main functionality of UDP?
 - MUX/DEMUX
 - Detect error by checksum



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

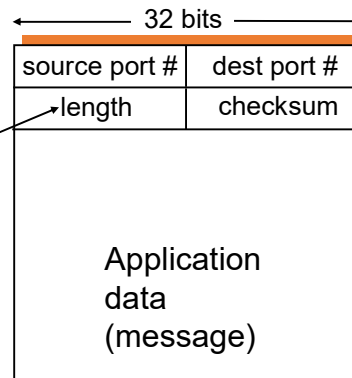
34

34

Datagram format

- Data unit in UDP is called datagram

Length of the datagram in byte



35

Issues of UDP

- No congestion control
 - Cause overload of the Internet
- No reliability
 - Applications have to implement themselves mechanisms to control errors

36

DOS Attack: UDP Flooder

- ☐ During the year 1998-2000, security specialist discovered "DoS attack with UDP flooding" vulnerabilities in many of the Systems including Microsoft products.
- ☐ In less than two months, 1,500 separate Internet Protocol addresses were attacked using UDP flooder
- ☐ These attacks have been significantly larger than anything we've seen
- ☐ Under a more common DOS attack, a network of bots, or compromised PCs commandeered by remote attackers, directly inundates a victim's Web server, name server or mail server with a multitude of queries. The goal of a DOS attack is to crash the victim's system, as it tries to respond to the requests.

Source: http://news.zdnet.com/2100-1009_22-6050688.html



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

37

DOS Attack: UDP Flooder

- ☐ Vulnerabilities were discovered in ACE/Server in its port 5000 against Fraggle attack
- ☐ Cisco has also discovered vulnerabilities of its IOS software in routers against diagnostic port where attacker used two ports namely diagnostics ports and chargen port as attacking media to attack using UDP flooding.
- ☐ Although DoS attack with UDP flooding are not new, there is still a significant risk exists of such attack as the new technique of DoS attack are being invented by the hackers.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

38

DOS Attack: UDP Flooder

- ❑ In this latest spate of DOS attacks, bots are sending queries to DNS (domain name system) servers with the return address pointed at the targeted victim. As a result, the DNS server, rather than the bot, makes the direct attack on the victim. The net result is a stronger attack and an increased difficulty in stopping it, Silva said.

Source: http://news.zdnet.com/2100-1009_22-6050688.html

- ❑ Denial of Service (DoS) attack is coordinated attacks performed by hackers to disable a particular computer service through manipulation of techniques those are used to provide the services.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

39

DOS Attack: UDP Flooder

• Motivation of DoS Attack

- The motivation for DoS attacks is not to break into a system but to make the target system deny the legitimate user giving service. This will typically happen through one of the following ways:
 - Crashing the target host system.
 - Disabling communication between systems.
 - Make the network or the system down or have it operated at a lower speed to reduce productivity.
 - Freeze the system, so that there is no automatic reboot, so that, production is disrupted.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

40

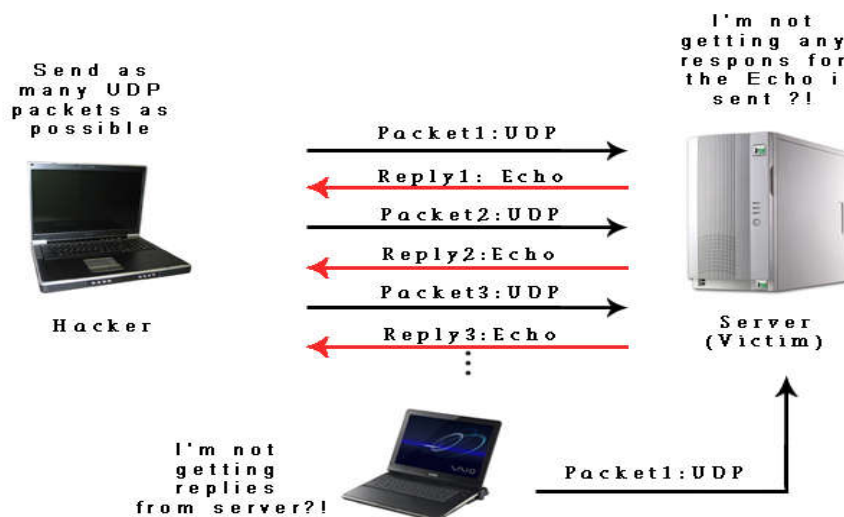
DOS Attack: UDP Flooder

❑ DoS Attack Classes:

- Bandwidth Depletion attack - floods a victim network and thereby prevents authorized traffic from reaching and getting the service of the targeted victim.
 - Resource Depletion attack - bind the resources of the target victim's system (such as processor) making the victim unable to process valid requests for services.
- ** Among the other flooding tools, UDP flooding is also used to deplete bandwidth or resources of the victim system.
- ** Flood attacks are being launched either with UDP or ICMP packets.

41

DOS Attack: UDP Flooder



42

DOS Attack: UDP Flooder

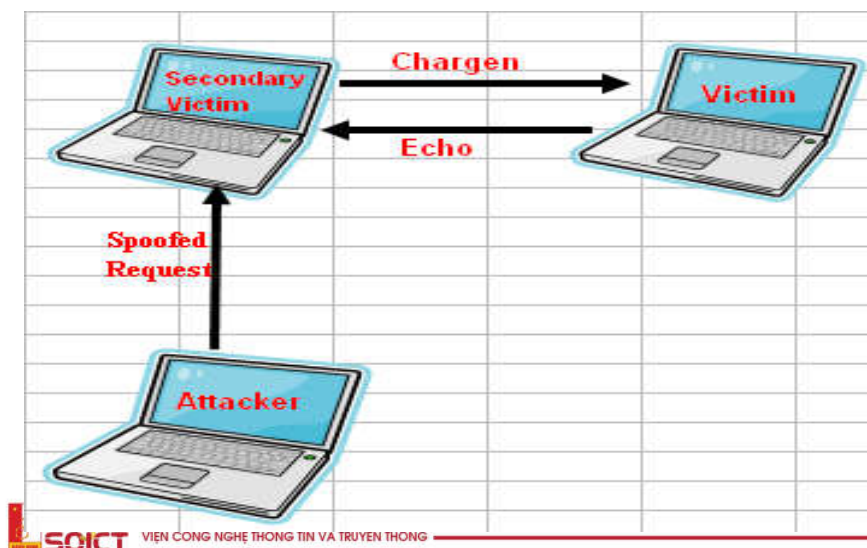
❑ Flood Attack

- In this kind of attack, the network of the victims system is flooded with a large number of packets by the attacker to deplete the network bandwidth and thereby making the victim's systems performance degradation or sometimes system crash. Due to saturation of the network bandwidth of the victim's system, the legitimate users of the system are prevented from accessing the system.

DOS Attack: UDP Flooder

- ❑ In a UDP Flood attack, numerous amounts of UDP packets are sent to either **random** or **specified** ports on the victim system.
- ❑ To determine the requested application, the victim system **processes** the incoming data.
- ❑ In case of absence of the requested application on the requested port, the victim system sends a "*Destination unreachable*" message to the sender (attacker).
- ❑ In order to hide the identity of the attacker, the attacker often *spoofs* the **source IP** address of the attacking packets.
- ❑ UDP flood attacks may also depletes the bandwidth of network around the victim's system.
- ❑ Thereby, the systems around the victim are also impacted due to the UDP flooding attack.

DOS Attack: UDP Flooder



45

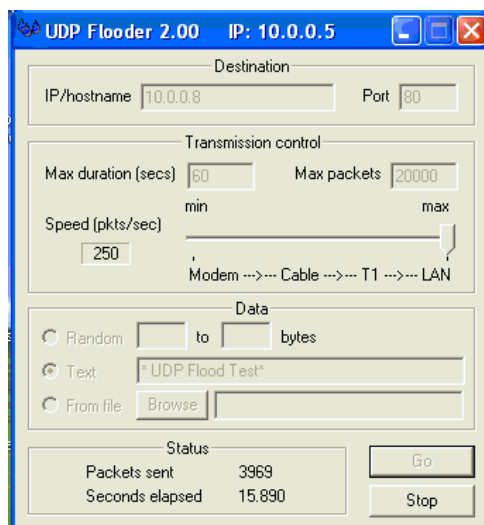
DOS Attack: UDP Flooder

❑ The Fraggle Attack

- This type of attack is usually used in UNIX and its family of OS platform, as well as, in network Routers and similar products. There are at least two service ports available (1) Echo (port #7 and (2) Chargen (port # 19) in this kind of OS or devices. Attacker sends UDP ECHO packets to the port that supports character generation (chargen port), with the return address spoofed to the victim's echo service (echo port) creating an infinite loop.
- The UDP ECHO packet (called as UDP Fraggle packet) targets the character generator (chargen port) of the systems reached by the broadcast address.
- The chargen port generates a character and sends the same to the echo service (echo port) of the victim's system. The victim's system echo port then sends an echo packet back to the chargen port - the process repeats and generates a loop. Packet generation loop created in this fashion generates damaging traffic and cause severe damage in the system.

46

DOS Attack: UDP Flooder



47

DOS Attack: UDP Flooder

❑ UDP Flooder (handy attacking tool)

- UDP flooder is a handy attacking tool for Windows Platform. The tool can send a numerous number of UDP packets (chosen by attacker) at a selected speed from a host to another host. It uses a specific port to attack and also uses some imaginary source address.

48

DOS Attack: UDP Flooder

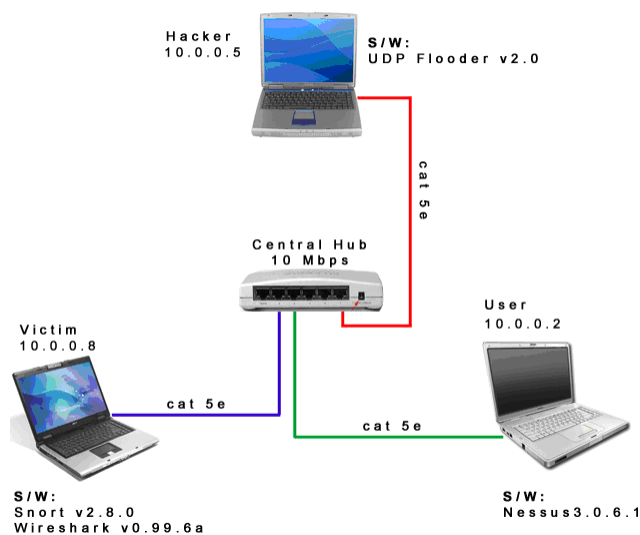
- ❑ While testing with this tool, we have used three thread of the flooder and flooded the target computer with three different ports. And the result was two ways.
 - Tie-up the CPU that resulted to crash (shut-down of the victim system).
 - Reduced the network speed (communication between third computer and attacking host was very slow)



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

49

The laboratory setup



50

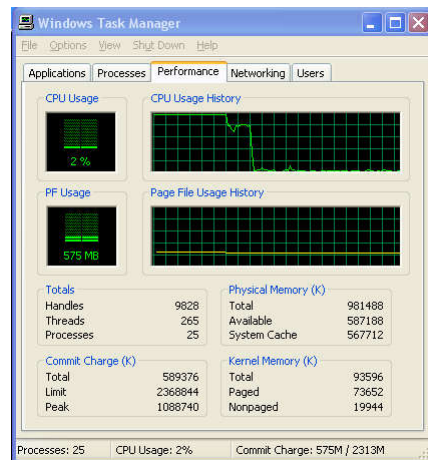
Snort

Snort is free and open source software for network intrusion detection. It is capable of performing packet logging and real-time traffic analysis.

→ To detect if there is a flooding attack depending on the **threshold** value set in snort rule.

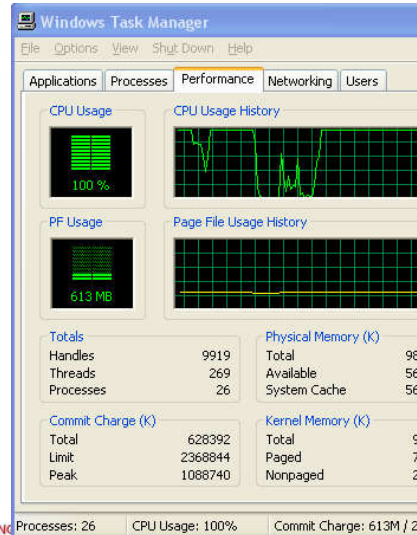
51

Before the UDP Flooding attack



52

During the UDP Flooding attack



53

Test result:

- The Snort generates an alert. Before the UDP Flooding attack, the IDS computer CPU utilization is 2%, during the UDP Flooding attack, the CPU utilization is 100%. Victim System is crashed when the 100% CPU utilization up to 5 minutes.

54

DoS Countermeasures

- Classified in three categories

1. Protection

2. Detection

3. Reaction

Attack Protection

- It focuses in eliminating the possibility of the attack.
 - Cryptography: the basis for even more protection (data integrity protection, authentication, protection from eavesdropping, and so on.)
 - Protection approaches patch existing vulnerabilities, correct bad protocol design, manage resource usage, and reduce the incidence of intrusions and exploits.

Attack Protection Cont.

- **Hygiene approaches:** try to close as many opportunities for DoS attacks in your computers and networks as possible, enhance security by keeping your network simple, well organized, and well maintained.
- **Fixing Host Vulnerabilities:** The first step in maintaining network hygiene is keeping software packages patched and up to date.
- **Fixing Network Organization:** Well-organized networks have no bottlenecks or hot spots that can become an easy target for a DoS attack.

Attack Protection Cont.

- A good way to organize a network is to spread critical applications across several servers, located in different subnetworks. The attacker then has to overwhelm all the servers to achieve denial of service.
- A good network organization not only repels many attack attempts, it also increases robustness and minimizes the damage when attacks do occur.
- Critical services are replicated throughout the network, machines affected by the attack can be quarantined and replaced by the healthy ones without service loss.

Attack Protection Cont.

- **Filtering Dangerous Packets:** the most standard approach against DOS attacks is the use of Ingress/Egress filtering techniques.
 - **Ingress filtering:** is from the point of view of the Internet. Here an Internet Service Provider (ISP) filters out packets with illegitimate source address, based on the ingress link by which the packet enters the network.
 - **Egress filtering:** is from the point of view of the customer network and the filtering occurs at the exit point of a customer domain. Here a router checks whether the source addresses of packets actually belong to the customer's domain. Packets with invalid source addresses are dropped.

Attack Protection Cont.

- UDP flooding is done using specific or random ports of the victims system. Therefore, it is recommended that you:
 - Disable and filter all unused UDP services
 - Disable unused ports/services for IP infrastructure devices such as routers and switches.
 - Disable the service port (port #7) and chargen port (port # 19) and filter the chargen and echo services.

Attack Detection

- If protection approaches cannot make DoS attacks impossible, then the defender must **detect** such attacks before he can respond to them.
- Two major goals of attack detection are **accuracy** and **timeliness**.
 - Accuracy is measured by how many detection errors are made. A detection method can misstep in two ways. It can falsely detect an attack in a situation when no attack was actually happening. This is called a *false positive*.

Attack Detection Cont.

- The other way for a detection method to misstep is to miss an attack. This is called a *false negative*.
- The performance of the whole DDoS defense system depends on the **timeliness** of the detection. Attacks that are detected and handled early may even be transparent to ordinary customers and cause no unpleasant disruptions.
- Three main approaches to attack detection are **signature**, **anomaly**, and **misbehavior** detection.

Attack Detection Cont.

- **Singature Detection:** Signature detection builds a database of attack characteristics observed in the past incidents—attack signatures. All incoming packets are compared against this database, and those that match are filtered out.
- **Anomaly Detection:** Anomaly detection takes the opposite approach from signature detection. Detects computer intrusions and misuse by monitoring system activity and classifying it as either *normal* or *anomalous*. The classification is based on rules, rather than patterns or signatures, and will detect any type of misuse that falls outwith normal system operation.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

63

Attack Detection Cont.

- **Misbehaviour Detection:** At one extreme, misbehavior modeling is the same as signature-based detection: Receiving a sufficiently large number of a particular type of packet on a particular port with a particular pattern of source addresses may be both a misbehavior model and a signature of the use of a particular attack toolkit.
- At the other extreme, misbehavior modeling is no different than anomaly modeling: If it is not normal, it is DoS. But misbehavior modeling, by trying to capture the characteristics of only DoS attacks, characterizes all other types of traffic, whether they have actually been observed in the past or not, as legitimate. True misbehavior modeling falls in the range between these extremes.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

64

Attack Reaction

- The goal of attack reaction is to improve the situation for legitimate users and mitigate the DoS effect. There are three major ways in which this is done:
 1. Traffic policing
 2. Attack Traceback
 3. Service differentiation

Attack Reaction Cont.

- **Traffic policing:** The most straightforward and desirable response to a DoS attack is to drop offending traffic. This makes the attack transparent both to the victim and to its legitimate clients, as if it were not happening. Since attack detection and characterization are sometimes inaccurate, the main challenge of traffic policing is deciding what to drop and how much to drop.
- **Attack traceback:** Has two primary purposes: to identify agents that are performing the DoS attack, and to try to get even further back and identify the human attacker who is controlling the DoS network.

Attack Reaction Cont.

- The first goal might be achievable, but is problematic when tens of thousands of agents are attacking. The latter is nearly impossible today, due to the use of stepping stones. These factors represent a major challenge to traceback techniques.
- Effective traceback solutions probably need to include components that automatically police traffic from offending machines, once they are found.
- **Service differentiation:** Many protection techniques can be turned on dynamically, once the attack is detected, to provide differentiated service. Clients are presented with a task to prove their legitimacy, and those that do receive better service.

Attack Reaction Cont.

- Service Differentiation approach offers a good economic model. The server is generally willing to serve all the requests. At times of overload, the server preserves its resources and selectively serves only the clients who are willing to prove their legitimacy and provides best-effort service to the rest. A challenge to this approach is to handle attacks that generate a large volume of bogus legitimacy proofs. It may be necessary to distribute the legitimacy verification service to avoid the overload.

Other Attacks in Transport Layer

1. Session hijacking

This kind of attack occurs after a source and destination computer have established a communications link. A third computer disables the ability of one the computers to communicate, and then imitates that computer. Because the connection has already been established, the third computer can disrupt the C-I-A (confidentiality integrity and availability) triad.

Protection against session hijacking

- Use SSL/HTTPS encryption for the entire web site, and you have the best guarantee that no man in the middle attacks will be able to sniff an existing client session cookie
- use some sort of encryption on the session value itself that is stored in your session cookie



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

69

Protection against session hijacking Using HTTPS/SSL

HTTPS is a combination of the standard HTTP protocol and the cryptographic security of the SSL protocol. The HTTPS protocol contains mechanisms for secure identification of the server and encryption of the client-server communication.

- Obtain an SSL certificate from a Certificate Authority (CA). A CA is third party that the client trusts to verify that the site using the certificate is indeed the owner of the certificate. There are many CAs to choose from, Google provides a list of popular CAs. Then configure Internet Information Service (IIS) so that the site uses the certificate.
- Make Sure That the Session Cookie is Sent Over an Encrypted Connection
- In order to fully safeguard against session hijacking, make sure that all communication where the session cookie is sent is encrypted. There are two options to achieve this:

Option 1 - Force SSL at All Times

Option 2 – Only Send Session Cookies Over SSL

By setting `requireSSL="true"` on the forms-element in web.config, specify that the session cookie should only be sent when using the HTTPS protocol.

This approach enables to use SSL only on parts of the site (edit/admin for example) and allow non-encrypted communication when browsing the public parts of the site.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

70

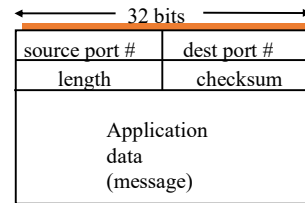
2. TCP Connection Spoofing

At the Transport layer, either a UDP or TCP header is added to the message. By knowing the UDP or TCP header fields and lengths, the ports that are used for communications between a source and destination computer can be identified, and that information can be corrupted or exploited.

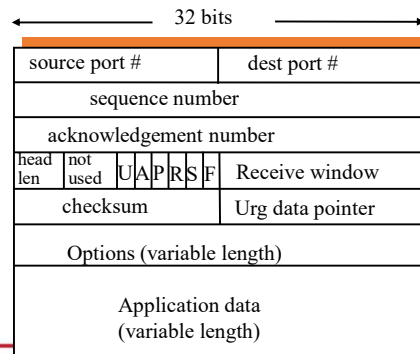
- If attacker knows initial seq # and amount of traffic sent, it can estimate likely current values
- Send a flood of packets with likely seq numbers
- Attacker can inject packets into existing connection



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



UDP segment format



TCP segment format

71

How TCP connection spoofing works

Injecting IP packets which seems to originate from another host is insufficient to impersonate that host during a TCP connection, because every TCP segment has a 32bit sequence number. A segment with a sequence number which is out of line will be ignored. That means in order to successfully insert a TCP segment into an existing transmission it needs to guess the next sequence number, otherwise the segment will be discarded.

- This isn't so hard when the attacker can eavesdrop at least on the client; otherwise, it can only brute-force the sequence number.
- With more simple transport protocols, like UDP for example, the attacker doesn't have problem. UDP has no sequence numbers, so unless an upper protocol layer replicates the functionality similar to sequence numbers, it can just insert additional segments which will then be treated as if they were coming from the real host.
- If the attacker doesn't know existing connection and instead want to establish a new TCP connection which appears to originate from another host, a 3-way handshake is required (client sends SYN, server sends ACK, client sends SYNACK). The ACK by the server includes a random number which the attack needs for an acceptable SYNACK. So when it can only send IP packets but not receive any of the packets intended for the spoofed host, attacker will have to guess this random number.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

72

Ports and Protocols

- A port is simply a logical communication endpoint
- Ports are classified as either **inbound** or **outbound** ports
 - An inbound port is used when your computer or server is listening for a connection
 - An outbound port is opened by a computer whenever it wants to connect to a server

Port groups

- Ports can be any number between 0 and 65,535
- 1. Well-Known ports
 - any ports that are between 0 and 1023
 - they are designated by IANA (Internet Assigned Numbers Authority)

Protocol	Port
HTTPS	443
Telnet	23
SSH	22
DNS	53
BGP	179
SMTP	25

Port groups (2)

2. Registered ports

- any ports that are between 1024 and 49,151
- they are registered with IANA prior to using them

Protocol	Port
SQL server	1433
RDP	3389
OpenVPN	1194
MQTT	1833
Docker REST API (SSL)	2376
MySQL	3306

Port groups (3)

3. Dynamic and private ports

- any ports that are between 49,152 and 65,535
- These ports can be used by any application at any time without having to be registered with IANA first

Memorization of Ports

- There are 35 ports that you simply should memorize

Protocol		Port	
FTP	TCP	21	transfer files from host to host
SSH, SCP, SFTP	TCP/UDP	22	Secure Shell used to remotely administer network devices Secure Copy used to securely copy files over a network
Telnet	TCP/UDP	23	used to remotely administer network devices
SMTP	TCP	25	used to send email over the Internet
DNS	TCP/UDP	53	used to resolve host names to IP and vice versa
TFTP	UDP	69	Trivial FTP is a simplified version of FTP
HTTP	TCP	80	used to transmit webpage data for a client

77

Memorization of Ports (1)

Protocol		Port	
Kerberos	TCP/UDP	88	used for network authentication
POP3	TCP	110	used to receive mail from a mail server
NNTP	TCP	119	Network News Transfer Protocol used to transport Usenet articles to a client
RPC	TCP/UDP	135	Remote Procedure Call is used to locate DCOM ports to request a service from a program on another computer
NetBIOS	TCP/UDP	137, 138, 139	used to conduct name querying, sending of data, etc.. over a NetBIOS connection
IMAP	TCP	143	used to receive email from a mail server
SNMP	UDP	161	used to remotely monitor network devices
SNMPTRAP	TCP/UDP	162	used to send Trap and InformRequests to the SNMP Manager on a network

78

Memorization of Ports (2)

Protocol		Port	
LDAP	TCP	389	used to maintain directories of users / objects
HTTPS	TCP	443	used to transmit web pages to a client
SMB	TCP	445	Server Message Block is used to provide shared access to files and other resources
SMTPs	TCP	465/ 587	used to send email
Syslog	UDP	514	used to conduct computer message logging
LDAPs	TCP/UDP	636	used to maintain directories of users / objects
iSCSI	TCP	860	for linking data storage facilities over IP
FTPs	TCP	989/990	transfer files from host to host
IMAP4s	TCP	993	used to receive email from a mail server
POP3s	TCP	995	used to receive mail from a mail server

79

Memorization of Ports (3)

Protocol		Port	
SQL server	TCP	1433	to receive SQL database queries from its clients
RADIUSs	UDP	1812/1813	Remote Authentication Dial-In used for authentication and authorization
L2TP	UDP	1701	Layer 2 Tunneling Protocol is used as an underlying VPN protocol
PPTP	TCP/UDP	1723	Point-to-Point Tunneling Protocol
FCIP	TCP/UDP	3225	Fibre Channel IP is used to encapsulate Fibre Channel frames within TCP/IP packets
RDP	TCP/UDP	3389	used to remotely view and control other Windows systems
iSCSIs	TCP	3260	for linking data storage facilities over IP
Diameter	TCP	3868	a more advanced AAA protocol than RADIUS
Syslog	TCP	6514	used to conduct computer message logging

80

Unnecessary ports

- There are 2 methods to close an unnecessary port
 - Stop the service that uses that port
 - GUI
 - Command line
 - Block the ports at your firewall
 - software or hardware-based firewall

81



82