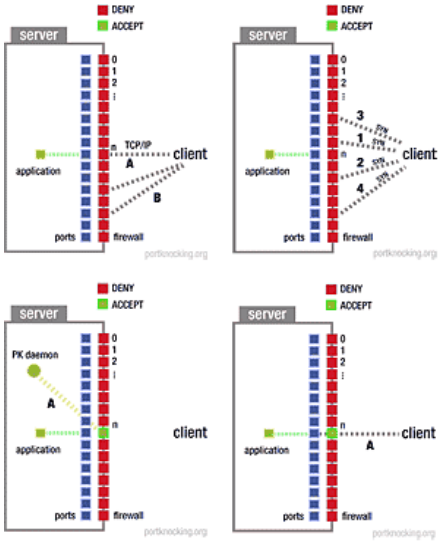

UDP Port Knocking Project

s19358 Aysenur Ozgur 12c

- Definition

UDP is a Transport Layer protocol. It is **unreliable and connectionless protocol.** So, there is no need to establish connection prior to data transfer. It has no handshaking dialogues. There is no guarantee of delivery, ordering, or duplicate protection. UDP is suitable for purposes where error checking and correction are either not necessary or are performed in the application.

Port knocking is the method of opening the desired ports according to the connection requests made to the predetermined ports. When the connection request is sent to the predetermined ports again within the predetermined time interval, the firewall rules are dynamically changed and the ports of the desired service are actually opened. This application is very useful in cases where it is wanted to be hidden from outside service for security reasons. How does Port Knocking work?

Step 1: The client cannot connect to the application listening on port n on the server.

Step 2: The client sends connection requests (SYN packets) to the predefined ports.

Step 3: The port knocking program running on the server recognizes the correct connection request and opens the desired port (where n is the port) for that client.

Step 4: The client connects to the server normally.

-Project Description

There are 2 main classes Server and client .This project is on a custom protocol which enables server to open many UDP ports and starts waiting UDP datagrams from clients .

In the server side : it opens many udp ports and gets datagrams, when a proper sequence of datagrams sent from the same IP address and port then the server opens a randomly selected UDP port between **49,152** and **65,535** (Dynamic ports are numbered from **49,152** through **65,535**). Server will send this port number also name and length of a file with content to the client and continue to wait another udp datagrams.

In the client side : first client sends a datagram to the server and wait for response, when client receives informations from server then it will create a file with same as received name and length .

-Implementation

Run server class and first enter how many ports you want to use, then enter the port numbers.Server will open this ports.Next , run the client class ,then you should enter the same port numbers.Otherwise you will get timeoutException because in the client class there is a time like 10 seconds.A call to receive() for DatagramSocket will block for only this amount of time. Same story will happen for wrong port sequence.

-Some difficulties…

I had some problems with getting port numbers from user 2 times.I've tried to get them one time and use them in both classes.I know it is really simple to fix it but idk why I didnt take those numbers from another class. I also had problems with taking file name from server . It was getting random port number instead of filename.Then I easily handled it.