Slide 2:

A port is an addressable network location implemented in an operating system to help differentiate traffic destined for different services or applications. A port is always associated with an IP address of a host and the protocol type for the communication.

​Ports are specified by a number ranging from 1 to 65535. There are different categories of ports.

0-1023 – are “Well Known Ports” associated with services considered as critical or essential. They are also referred to as System Ports.

1024-49151 – are Registered Ports or User Ports and as the name indicates, can be reserved by sending a request to the Internet Assigned Numbers Authority (IANA).

49152-65535 – are called Dynamic Ports and are suggested for private use. They also referred to as the Private Ports.

A service is said to be “listening” on a port when it is binding to a port/protocol/IP address combination in order to wait for requests from clients of the service. Upon receipt of the request, it establishes a one-to-one server client dialog using the same port number.

Slide 3:

iptables provide us with granular control over traffic.

The iptables firewall operates by comparing network traffic against a set of rules. The rules define the characteristics that a packet must have to match the rule, and the action that should be taken for matching packets.

There are many options to establish which packets match a specific rule. You can match the packet protocol type, the source or destination address or port, the interface that is being used, its relation to previous packets, etc.

When the defined pattern matches, the action that takes place is called a target. A target can be a final policy decision for the packet, such as accept, or drop. It can also be move the packet to a different chain for processing, or simply log the encounter. There are many options.

These rules are organized into groups called chains. A chain is a set of rules that a packet is checked against sequentially. When the packet matches one of the rules, it executes the associated action and is not checked against the remaining rules in the chain.

A user can create chains as needed. There are three chains defined by default. They are:

INPUT: This chain handles all packets that are addressed to your server.

OUTPUT: This chain contains rules for traffic created by your server.

FORWARD: This chain is used to deal with traffic destined for other servers that are not created on your server. This chain is basically a way to configure your server to route requests to other machines.

Each chain can contain zero or more rules, and has a default policy. The policy determines what happens when a packet drops through all of the rules in the chain and does not match any rule. You can either drop the packet or accept the packet if no rules match.

append this rule to the input chain (-A INPUT) so we look at incoming traffic

check to see if it is TCP (-p tcp).

if so, check to see if the input goes to the SSH port (--dport ssh).

if so, accept the input (-j ACCEPT).

Slide 6:

OpenSSH is a powerful collection of tools for the remote control of, and transfer of data between, networked computers. You will also learn about some of the configuration settings possible with the OpenSSH server application and how to change them on your Ubuntu system.

OpenSSH is a freely available version of the Secure Shell (SSH) protocol family of tools for remotely controlling, or transferring files between, computers. Traditional tools used to accomplish these functions, such as telnet or rcp, are insecure and transmit the user’s password in cleartext when used. OpenSSH provides a server daemon and client tools to facilitate secure, encrypted remote control and file transfer operations, effectively replacing the legacy tools.

The OpenSSH server component, sshd, listens continuously for client connections from any of the client tools. When a connection request occurs, sshd sets up the correct connection depending on the type of client tool connecting. For example, if the remote computer is connecting with the ssh client application, the OpenSSH server sets up a remote control session after authentication. If a remote user connects to an OpenSSH server with scp, the OpenSSH server daemon initiates a secure copy of files between the server and client after authentication. OpenSSH can use many authentication methods, including plain password and public key.

Slide 7:

Using the iptable command we discussed earlier how will we allow incoming ssh requests?

How do you remove public keys?

public keys give ssh access to anyone with no password so we want to remove those

Slide 8:

File Transfer Protocol (FTP) is a TCP protocol for downloading files between computers.

FTP works on a client/server model. The server component is called an FTP daemon. It continuously listens for FTP requests from remote clients. When a request is received, it manages the login and sets up the connection. For the duration of the session it executes any of the commands sent by the FTP client.

Access to an FTP server can be managed in two ways:

Anonymous

Authenticated

In the Anonymous mode, remote clients can access the FTP server by using the default user account called “anonymous” or “ftp” and sending an email address as the password. In the Authenticated mode a user must have an account and a password. This latter choice is very insecure and should not be used except in special circumstances. If you are looking to transfer files securely see SFTP in the section on OpenSSH-Server. User access to the FTP server directories and files is dependent on the permissions defined for the account used at login. As a general rule, the FTP daemon will hide the root directory of the FTP server and change it to the FTP Home directory. This hides the rest of the file system from remote sessions.

Slide 10:

OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography standards required by them.

The openssl program is a command line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. It can be used for Creation and management of private keys, public keys and parameters and Creation of X.509 certificates

Slide 11:

Remove <Anonymous> blocks to prevent anonymous access to the server.