So now if a source address is forged, then router will detect the unusual path that was taken for that source and destination. So a router can systematically learn IP address that it forwards. It can maintain this as the permitted list of IP addresses. Suddenly when the router sees huge traffic, it can stop learning the IP addresses and drop only those packets which are not there in the permitted list till the attack is under control.

4 Network Intrusion Prevention by Configuring ACLS on Routers Based on Snort IDS Alerts

In this attack on router is prevented by the use of Snort to generate alerts for attacks and configure ACLs to defend.

Snort can also be used to detect any intrusion in router and also took measures to take action for that intrusion using ACLs(Access Control Lists). Snort is an open source tool which works as an IDS (Intrusion Detection System). Rules are written in Snort and they are matched against the packets. An example of a rule is as follows

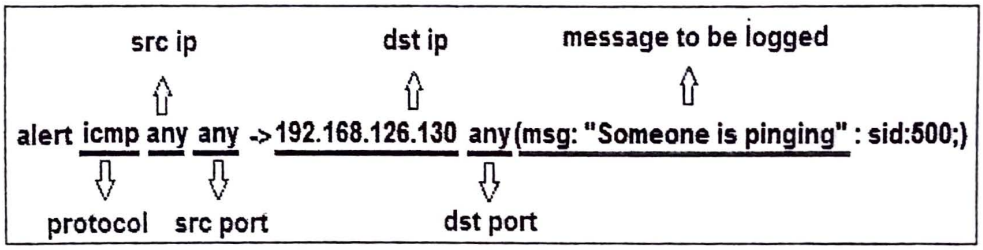

Figure 11.24 A Snort Rule

If a packet matches then messages are sent to the snort log. These logs now can be studied and appropriate access control lists can be generated for the router to curb the attack. Snort is an open source tool and is easily available. Access control lists is an in built feature in routers and very powerful when security is concerned. So the router is secured with very less cost overhead.

11.6 Hardening Linux Operating System for Network Security

We have provided the following guidelines for hardening the linux operating systems for network security

11.6.1 Workstation Security

Security begins with the first time you put that CD or DVD into your disk drive to install Red Hat Enterprise Linux. Configuring your system securely from the beginning makes it easier to implement additional security settings later.

Disk Partitions

Create separate partitions for /boot, /, /home, /tmp, and /var/tmp. The reasons for each are different and we will address each partition.

/boot –The boot loader and kernel images that are used to boot your system into RedHat Enterprise Linux are stored in this partition. This partition should not be encrypted. If this partition is included in / and that partition is encrypted or otherwise becomes unavailable then your system will not be able to boot.

/home - When user data (/home) is stored in / instead of in a separate partition, thepartition can fill up causing the operating system to become unstable. Also, when upgrading your system to the next version of Red Hat Enterprise Linux it is a lot easier when you can keep your data in the /home partition as it will not be overwritten during installation. If the root partition (/) becomes corrupt your data could be lost forever. By using a separate partition there is slightly more protection against data loss. You can also target this partition for frequent backups.

/tmp and /var/tmp - Both the /tmp and the /var/tmp directories are used to store datathat doesn't need to be stored for a long period of time. However if a lot of data floods one of these directories it can consume all of your storage space. If this happens and these directories are stored within / then your system could become unstable and crash. For this reason, moving these directories into their own partitions is a good idea.

To get mounted partition information use the following command. # df–lh, Where df : displays the amount disk space available on the file system, l: limit listening to local fiel system, h: print in human radable formate (i.e 22k) Utilize LUKS Partition Encryption: During the installation process an option to encrypt your partitions will be presented to the user. The user must supply a passphrase that will be the key to unlock the bulk encryption key that will be used to secure the partition's data [2].

Manually Encrypting Directories

single user mode.

1. Enter runlevel 1 by typing the following at a shell prompt as root: # init 1
2. Unmount your existing /home: # umount /sd7
3. If the command in the previous step fails, use fuser command to find processes hogging /home and kill them: # fuser -mvk /sd7

   Where m: mounted file system, v: verbose output, k: kill process access that file
4. Verify /home is no longer mounted:

   # grep sd7 /proc/mounts

5. Fill your partition with random data:
   # dd if= /dev/urandom of= /dev/sda7
   This process can take many hours to complete.
6. Initialize your partition:
   [root@prem /] # cryptsetup --verbose –verify-passphrase luksformat /dev/sda7
7. Open the newly encrypted device:
   [root@prem /] # cryptsetup luksopen /dev/sda7 sd7
8. Make sure the device is present: [root@prem /1 # ls -1 /dev/mapper ! grep sd7
9. Create a file system:mkfs.ext3 /dev/mapper/home [root@prem /] # mkfs.ext4 dev/mapper/sd7
10. Mount the file system:
    [root@prem sd7] # mont /dev/mapper/sd7 /sd7/
11. Make sure the file system is visible: [root@prem sd]# df -hl
12. Add the following to the /etc/crypttab file:
    # home /dev/sd7 none
13. Edit the /etc/fstab file, removing the old entry for /home and adding the following line:
    # /dev/mapper/home /home ext3 defaults 1 2
14. Reboot the machine:
    # shutdown -r now
15. The entry in the /etc/crypttab makes your computer ask your luks passphrase on boot.
16. Log in as root and restore your backup.

Install Minimal Software

It is very critical to look at the default list of software packages and remove unneeded packages or packages that don't comply with your security policy, it is a good practice not to have development packages, desktop software packages (e.g. X Server) etc. installed on production servers. Other packages like FTP and Telnet daemons should not be installed as well unless there is a justified business reason for it (SSH/SCP/SFTP should be used instead). A good approach is to start with a minimum list of RPMs and then add packages as needed.To get a list of all installed RPMs you can use the

following command:#rpm–qa

Where rpm : RPM package manager,-q : query, -a : all

If you want to know more about a particular RPM, run the following command:

 # rpm -qi <package_name>

Where rpm : RPM package manager,-q : query, -i : information

To check for and report potential conflicts and dependencies for deleting a RPM, run following command

 # rpm -e --test <package_name>

Where rpm : RPM package manager, e : erase, test : don't actually uninstall anything

[root@prem]# rpm –e --test rpm

Install Signed Packages: Package signing uses public key technology to prove that thepackage that was published by the repository has not been changed since the signature was applied. Verifying Signed Packages: All Red Hat Enterprise Linux packages are signed with the Red Hat GPG key. Assuming the disc is mounted in / mnt/cdrom, use the following command as the root user to import it into the keyring (a database of trusted keys on the system):

[root@prem cdrom]# rpm—import RPM-GPG-KEY-redhat-beta

Now, the Red Hat GPG key is located in the /etc/pki/rpm -gpg/ directory.

To display a list of all keys installed for RPM verification, execute the following command:

[root@prem cdrom] # rpm -qa gpq-pubkey*

Where rpm : RPM package manager,-q : query , -a : all

To verify all the downloaded packages at once, issue the following command: [root@ prem packages] # rpm –k rpcbind-0.2.0-8.el6.i686.rpm

Where rpm : RPM package manager -K : check the package key

For each package, if the GPG key verifies successfully, the command returns gpg md5 OK. Packages that do not pass GPG verification should not be installed, as they may have been altered by a third party.

BIOS and Boot Loader Security

Password protection for the BIOS (or BIOS equivalent) and the boot loader can prevent unauthorized users who have physical access to systems from booting using removable media or obtaining root privileges through single user mode. BIOS Passwords: The two primary reasons for password protecting the BIOS of a computer are:

1. Preventing Changes to BIOS Settings
2. Preventing System Booting If you forget the BIOS password, it can either be resetwith jumpers on the motherboard or by disconnecting the CMOS battery. For this reason, it is good practice to lock the computer case if possible.

Preventing BIOS Password Circumvention: Since Linux distributions can be runfrom any form of removable media (CDs, VDs, floppy drives, and USB devices), disabling the ability to boot from any form of removable media is advisable and will keep out many of the lower-level, script-kiddie- attackers.

Boot Loader Passwords: The primary reasons for password protecting a Linux bootloader are as follows [2].

1. *Preventing Access to Single User Mode*
2. *Preventing Access to the GRUB Console*
3. *Preventing Access to Insecure Operating Systems*

Password Protecting GRUB: To do this, first choose a strong password, open a shell,log in as root, and then type the following command:

[root@prem log] # grub-md5-crypt

When prompted, type the GRUB password and press Enter. This returns an MD5 hash of the password. Next, edit the GRUB configuration file /boot/grub/grub.conf. Open the file and below the timeout line in the main section of the document, add the following line:

password --md5 *<password-hash>*

Replace <password-hash> with the value returned by /sbin/grub-m d5-crypt.

Whole Disk or Partition Encryption The best way to protect against data tampering or unintended disclosure is to implement one of the many whole disk or partition encryption methodologies available to Linux systems. This entails encrypting the entire contents of the hard drive, or partition, using a cryptographic encryption algorithm.

Password Security: For security purposes, the installation program configures the system to use Secure Hash Algorithm 512 (SHA512) and shadow passwords. The single most important thing a user can do to protect his account against a password cracking attack is create a strong password.

/etc/passwd file

[root@prem ~] #tail /etc/passwd /etc/shadow file

[root@prem ~] # tail /etc/shadow

1. Disallow Remote Root Login

Any actions requiring a direct log on to the system via _root' should be restricted to the local console. Edit ***/etc/securetty*** to reflect the following changes:

2. Disabling root access via any console device (tty)

To further limit access to the root account, administrators can disable root logins at the console by editing the /etc/securetty file.

To prevent the root user from logging in, remove the contents of this file by typing the following command at a shell prompt as root:

[root@prem]#echo > ***/etc/securetty***

11.6.2 Server Security

Disabling root SSH logins: To prevent root logins via the SSH protocol, edit the SSHdaemon's configuration file, /etc/ssh/sshd_config, and change the line that reads:#PermitRootLogin yes to read as follows:PermitRootLogin no

Disable CTRL-ALT-Delete: For those machines with poor or non-existent physical security, to disable the CTRL-ALT-Delete function that allows an attacker to shut down the machine. Edit /etc/inittab to comment out the following line.

 # ca::ctrlaltdel:/sbin/shutdown -t3 -r now

Save the change and to restart the service for it to take effect run the following command:

[root@hod BBM]# */sbin/init q*

Where q: tell init to re-exmine the /etc/inittab file

Thus We summarized the vulnerabilities, attacks and defense mechanisms.

When a system is used as a server on a public network, it becomes a target for attacks. Hardening the system and locking down services is therefore of paramount importance for the system administrator. Before delving into specific issues, review the following general tips for enhancing server security: Keep all services current, to protect against the latest threats., Use secure protocols whenever possible, - Serve only one type of network service per machine whenever possible, Monitor all servers carefully for suspicious activity.

Enhancing Security with TCP Wrappers

TCP Wrappers are capable of much more than denying access to services. This section illustrates how they can be used to send connection banners, warn of attacks from particular hosts, and enhance logging functionality. Refer to the hosts_options man page for information about the T CP Wrapper functionality and control language.

TCP Wrappers and Connection Banners: To implement a TCP Wrappers banner fora service, use the banner option. For this example, the file is called /etc/banners/vsftpd and contains the following lines:

220-Hello, %c

220-All activity on ftp.exam ple.com is logged.

220-Inappropriate use will result in your access privileges being removed.

The %c token supplies a variety of client information, such as the username and hostname, or the username and IP address to make the connection even more intimidating.

For this banner to be displayed to incoming connections, add the following line to the

/etc/hosts.allow file:

vsftpd : ALL : banners /etc/banners/

TCP Wrappers and Attack Warnings: If a particular host or network has beendetected attacking the server, TCP Wrappers can be used to warn the administrator of subsequent attacks from that host or network using the **spawn** directive.

Assume that a cracker from the 206.182.68.0/24 network has been detected attempting to attack the server. Place the following line in the */etc/hosts.deny* file to deny any connection attempts from that network, and to log the attempts to a special file:

ALL : 206.182.68.0 : spawn /bin/echo `date` %c %d >> /var/log/intruder_alert

The %d token supplies the name of the service that the attacker was trying to access. To allow the connection and log it, place the spawn directive in the */etc/hosts.allow* file.

TCP Wrappers and Enhanced Logging: If certain types of connections are of moreconcern than others, the log level can be elevated for that service using the severity option. For this example, assume that anyone attempting to connect to port 23 (the T elnet port) on an FT P server is a cracker. To denote this, place an emerg flag in the log files instead of the default flag, info, and deny the connection. To do this, place the following line in /etc/hosts.deny: in.telnetd : ALL : severity emerg This uses

the default authpriv logging facility, but elevates the priority from the default value of info to emerg, which posts log messages directly to the console.

116.3 Network Security

Protect portmap With TCP Wrappers: It is important to use TCP Wrappers to limitwhich networks or hosts have access to the portmap service since it has no built-in form of authentication. Further, use only IP addresses when limiting access to the service. Avoid using hostnames, as they can be forged by DNS poisoning and other methods.

Securing FTP

The File Transfer Protocol (FTP) is an older TCP protocol designed to transfer files over a network. Red Hat Enterprise Linux provides three FTP servers [2].

1. gssftpd -A Kerberos-aware

2. xinetd - based FTP daemon that does not transmit authentication information over the network.

3. Red Hat Content Accelerator (tux) - A kernel-space Web server with FTPcapabilities. vsftpd — A standalone, security oriented implementation of the FTP service. The following security guidelines are for setting up the vsftpd FTP service.

Configuring vsftpd for Anonymous FTP: The vsftpd.conf file controls the vsftpd daemon. The vsftpd binary has only one ommandline option, which allows you to specify the location of the vsftpd.conf configuration file.

 # vsftpd /etc/vsftpd.conf

Following shows a simple configuration file for a secure stand-alone anonymous FTP server that allows only downloads.

The /etc/vsftpd.conf file

 # General Configuration

listen=YES

background=YES

listen_address=192.168.0.1

nopriv_user=ftp_nopriv

xferlog_enable=YES

 # Mode and Access rights

anonymous_enable=YES

local_enable=NO

write_enable=NO

188

cmds_allowed=PASV,RETR,QUIT

 # Security

ftpd_banner=Puppy.YourDomain.Net FTP Server connect_from_port_20=YES

hide_ids=YES pasv_min_port=50000 pasv_max_port=60000

 # DoS

ls_recurse_enable=NO max_clients=200 max_per_ip=4

Securing SSH: Many network services like telnet, rlogin, and rsh are vulnerable toeavesdropping which is one of several reasons why SSH should be used instead. The Restricting System Access from Servers and Networks shows how direct logins can be disabled for shared and system accounts including root. For securing SSH Use the following parameters: no, UsePrivilegeSeparation yes, Allow Tcp Forwarding no, Strict Modes yes Ensure that all host-based authentications are disabled. These methods should be avoided as primary authentication. Ignore Rhosts yes, Host based Authentication no, Rhosts RSA Authentication no, Disable sftp if it's not needed: #Subsystem sftp /usr/lib/misc/sftp-server After changing any directives make sure to restart the sshd daemon run the following command: root@prem] # /etc/init.d/sshd restart

ssh passphrase: Passwords aren't very secure, Anyone who gains access to your drivehas gained access to every system you use that key with. This is also a Very Bad Thing. The solution is obvious, add a passphrase.

Step by step procedure to create passphrase for ssh as follow.

- Ssh server is running sshd so it will allow remote ssh login and which stores public key.
- ssh client has ssh client which is used to get ssh access to ssh server . Passphrase is created on ssh client and store private key.
- Create user ***prem*** on both server and client.
- **On client**

1) Generate an RSA key pair by typing the following command at a shell prompt:

 # ssh-keygen –d

Where: -d: dsa It will create .ssh directory in user home directory. Then it will ask for the passphrase, Enter the passphrase, the two files are created in .ssh directory,

1 id_dsa (private key) id_dsa.pub (public key)

2) Login on server from client using ssh **On sever**

1) Create .ssh directory in home directory (/home/hod) of user prem. Use the following command. # mkdir .ssh
2) To Go inside .ssh directory use follwing command. # cd .ssh
3) Use the following command to copy public key into server.

   # scp hod@client_ip:.ssh/id_dsa.pub authorized_keys

   Where : scp : secure copy command
4) Set owner permissions on /home/hod/.ssh directory and authorized_keys file.

5) To set permission use the following commands.

   # chmod 700 /home/prem/.ssh ,# chmod 600 authorized_keys

   Where ***: chmod :*** change the file or directory permissions

The Network Manager Daemon

The Network Manager daemon runs with root privileges and is usually configured to start up at boot time. You can determine whether the Network Manager daemon is running by entering this command ***as root:# service NetworkManager status:***

NetworkManager (pid 1527) is running. The service command will report Network Manager is stopped if the Network Manager service is not running. To start it for the current session: ***#service Network Manager start*** Run the chkconfig command to ensure that NetworkManager starts up every time the system boots: ***#chkconfig Network Manager on***

Security Enhanced Communication Tools

As the size and popularity of the Internet has grown, so has the threat of communication interception. Over the years, tools have been developed to encrypt communications as they are transferred over the network. Red Hat Enterprise Linux 6 ships with two basic tools that use high-level, public-key-cryptography-based encryption algorithms to protect information as it travels over the network.

OpenSSH - A free implementation of the SSH protocol for encrypting network communication. Gnu Privacy Guard **(GPG)** — A free implementation of the PGP (Pretty Good Privacy) encryption application for encrypting data. OpenSSH is a safer way to access a remote machine and replaces older, unencrypted services like telnet and rsh. OpenSSH includes a network service called sshd and three command line clientapplications: **ssh** — A secure remote console access client. **scp** — A secure remote copy command. **sftp** — A secure pseudo-ftp client that allows interactive file transfer sessions.

Enable TCP SYN Cookie Protection: A "SYN Attack" is a denial of service attackthat consumes all the resources on a machine. Any server that is connected to a network is potentially subject to this attack. To enable TCP SYN Cookie Protection, edit the */etc/sysctl.conf* file and add the following line:

net.ipv4.tcp_syncookies = 1

Disable IP Source Routing: Source Routing is used to specify a path or route throughthe network from source to destination. This feature can be used by network people for diagnosing problems. However, if an intruder was able to send a source routed packet into the network, then he could intercept the replies and your server might not know that it's not communicating with a trusted server. To enable Source Route Verification, edit the */etc/sysctl.conf* file and add the following line: ***net.ipv4.conf.all.accept_source_route = 0***

Disable ICMP Redirect Acceptance: ICMP redirects are used by routers to tell theserver that there is a better path to other networks than the one chosen by the server. However, an intruder could potentially use ICMP redirect packets to alter the hosts's routing table by causing traffic to use a path you didn't intend.

To disable ICMP Redirect Acceptance, edit the /etc/sysctl.conf file and add the following line: ***net.ipv4.conf.all.accept_redirects = 0***

Enable IP Spoofing Protection: IP spoofing is a technique where an intruder sends outpackets which claim to be from another host by manipulating the source address. IP spoofing is very often used for denial of service attacks. To enable IP Spoofing Protection, turn on Source Address Verification. Edit the /etc/sysctl.conf file and add the following line:

net.ipv4.conf.all.rp_filter = 1

Enable Ignoring to ICMP Requests: If you want or need Linux to ignore pingrequests, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.icmp_echo_ignore_all = 1

This cannot be done in many environments.

Enable Ignoring Broadcasts Request: If you want or need Linux to ignore broadcast requests, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.icmp_echo_ignore_broadcasts = 1

Enable Bad Error Message Protection: To alert you about bad error messages in thenetwork, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.icmp_ignore_bogus_error_responses = 1

Enable Logging of Spoofed Packets, Source Routed Packets, Redirect Packets: Toturn on logging for Spoofed Packets, Source Routed Packets, and Redirect Packets, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.conf.all.log_martians = 1