Netfilter/ iptables

http://www.tecmint.com/how-to-install-and-configure-ufw-firewall/
http://www.tecmint.com/install-ipfire-firewall-distribution/
http://ipset.netfilter.org/iptables-extensions.man.html

service iptables [start | save | stop]
/sbin/iptables-save > filename ---- saves rules to STDOUT by default, so send to file
/sbin/iptables-restore < filename ---- restores rules from STDIN by default, so give it the file
/sbin/iptables ----primary ACL modifier utility

- Modular - modules to proxy or function with other OSI layers: [/usr/lib/iptables/*.so]
- To check if enabled: [grep -i config_netfilter /boot/config*] -- you'll find CONFIG_NETFILTER=y

Opening /etc/sysconfig/iptables-config:
- IPTABLES_SAVE_ON_RESTART and ON_STOP set to "yes" to save written and implemented rules.
- IPTABLES_SAVE_COUNTER turn on to keep packet counters going from where they left off after a stop or restart

Quick CLI Usage Overview:
iptables -t table <chain><action/direction><packet pattern><segment pattern> -j <fate>
Table: filter (default), NAT (change IP addresses/ports), mangle (alter packets/segments TOS/TTL, etc)
Actions: -A (append) -D (delete) -L (list) -F (flush) -I (insert) -R (replace) -N (new) -E (rename)
Direction: INPUT, OUTPUT, FORWARD; PREROUTING and POSTROUTING
Interface: -i [eth0 | eth1 | eth+] "+" is always wildcard, use !eth0 for negation. Use -o for output interface
Layer 3 Packet Pattern: -s ip-addr (source), -d ip-addr (destination)
Layer 4 Segment Pattern: -p [tcp | udp | icmp] ; -dport, -sport ---- port #, or any name in /etc/services
fate: DROP, ACCEPT, REJECT/DENY, REDIRECT (NAT prerouting chain- local ports), LOG (syslog)

4 default tables (can't remove) and their chains in processing order:
- Filter (default- inbound and outbound traffic rules) INPUT, FORWARD, OUTPUT
- NAT (change IP addresses, ports) PREROUTING, OUTPUT and POSTROUTING
- Mangle (packet alteration) PREROUTING, INPUT, FORWARD, (OUTPUT) and POSTROUTING
- Raw (special- rarely used) PREROUTING, OUTPUT

The **NAT** table is consulted when a packet that creates a new connection is encountered. Three builtins are PREROUTING (for altering packets as soon as they come in), OUTPUT (for altering locally-generated packets before routing), and POSTROUTING (for altering packets as they are about to go out).

The **mangle** table is used for specialized packet alteration. PREROUTING (for altering incoming packets before routing), INPUT (for packets coming into the box itself), FORWARD (for altering packets being routed through the box), OUTPUT (not recommended to use), and POSTROUTING (for altering packets as they are about to go out).

The **raw** table is used mainly for configuring exemptions from connection tracking in combination with the NOTRACK target. It registers at the netfilter hooks with higher priority and is thus called before ip_conntrack, or any other IP tables. It provides the following built-in chains: PREROUTING (for packets arriving via any network interface) OUTPUT (for packets generated by local processes)

For more information on this see: http://www.faqs.org/docs/iptables/traversingoftables.html

---

iptables -L -v --verbose shows bytes count matching for chain and any listed rules
iptables -L --line-numbers -----to enumerate lines (you need for insert and replace operations)
iptables -L PREROUTING -t mangle ---specify a chain to list it's access control entries (ACEs)

Sample output for default FILTER table:
Chain INPUT (policy ACCEPT) ---FORWARD and OUTPUT will also be listed in same format
target port opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
DROP all -- anywhere anywhere

iptables -L -t NAT ----to show the NAT table
PREROUTING, POSTROUTING, and OUTPUT chains will display
iptables -L -t mangle ----- show mangle table
PREROUTING, INPUT, FORWARD, OUTPUT, and POSTROUTING chains will display

Append= A - There is also insert (I), replace (R), and (D) delete
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport telnet -j DROP
Permit SSH, deny telnet.
-j = Jump to DROP, DENY, ACCEPT, REJECT, LOG (or custom chain)

iptables -A INPUT -j DROP ---- drop all inbound traffic
iptables -I INPUT 1 -p tcp --dport 22 -j DROP -----inserts this as rule #1 in the INPUT chain list
iptables -D INPUT 1 -----deletes rule #1 in the INPUT chain list
iptables -D INPUT -p tcp --dport telnet -j DROP ----- deletes first match of these rules
iptables -R INPUT 1 -p tcp --dport 22 -j ACCEPT ------replace rule 1 with this rule

iptables -Z INPUT --------Z option zeros out byte count for chain
iptables -Z PREROUTING -t mangle ---zero the PREROUTING chain byte count in the mangle table

iptables -F OUTPUT -------flush all rules from output filter chain
iptables -F ----- flush all chains

Drop all webserver connections:
iptables -A INPUT -m state --state NEW,ESTABLISHED -p tcp -m multiport --dport 80,443 -j DROP
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p tcp -m multiport --dport 80,443 -j DROP
(-m for matching)
/etc/sysconfig/iptables is where rules are stored (for manual editing of the file
The -A append puts items at the end of the file after a DROP command of packets not referred to earlier in file.

- Packet-processing is done top-down by chain order (so are rules in a chain)
- Rules that are more likely to be matched should be put in the chain higher up in the list than others (line numbers)
- Duplicate rules can reside in the same chain.

User-Defined Chains (subchains)
iptables -N INTRANET -t mangle -------- make a new chain in mangle table

iptables -E INTRANET MY_SUBNET ---- rename a chain
iptables -R INPUT 1 -s 192.168.1.0/24 -j MY_SUBNET

----- Replace an existing rule and modifying it - filter all packets from INPUT that came from my subnet and send to the MY_SUBNET chain for processing

iptables -A MY_SUBNET -p tcp --dport 22 -j DROP
------if this is NOT matched, it is passed back to (origin chain) INPUT and there, it starts filtering from the rule where it left off (the next rule)

Chain Policy
- Basic idea is to determine which services are needed, write allow rules for those ports, then, once fully audited, set the default policy to DENY instead of ALLOW:
iptables -P INPUT DROP
iptables -P FORWARD DROP

Default DROP policy may prevent typical TCP/UDP/ICMP communications i.e. 3-way handshakes (SYN-SYNACK-ACK). In example, this is because the SYNACK didn't match any of the rules in MY_SUBNET chain. Added iptables -A MY_SUBNET -s 192.168.1.0/24 -j ALLOW to allow all traffic from local subnet.

Set Default Chain Policies
The default chain policy is ACCEPT. Change this to DROP for all INPUT, FORWARD, and OUTPUT chains as shown below.
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
If you make both INPUT, and OUTPUT chain's default policy as DROP, for every firewall rule requirement you have, you should define two rules. i.e one for incoming and one for outgoing.

Allow ALL Incoming SSH on eth0 interface
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

Allow Incoming SSH only from a Specific Network
iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
You can do this 192.168.100.0/255.255.255.0 if you must.

$ vi rules.sh
# 1. Delete all existing rules
iptables -F

2. Set default chain policies to DROP
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

3. Allow incoming SSH
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

4. Allow outgoing SSH
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

5. Allow incoming HTTP
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT

Save chmod u+x rules.sh and run

Matches
-s, --src, --source ---for INPUT
-d, --dst, --destination ---for OUTPUT
-i for in and -o for out interface (eth0, eth1)

iptables -A INPUT -s 192.168.1.72 -j DROP --- block all traffic from IP
iptables -A OUTPUT -d 192.168.1.72 -j DROP --- block traffic to IP
iptables -A INPUT -i eth0 -j DROP --- block all traffic from eth0
iptables -A INPUT -i eth1 -s 10.0.0.0/24 -j DROP --- block all traffic on eth1 coming from subnet specified
iptables -A INPUT -i eth+ -p tcp --dport telnet -j DROP --- "+" is a wildcard. This matches eth0, eth1, eth2, etc.

Negation- block all traffic not from specified IP
iptables -A INPUT -s ! 192.168.1.72 -j DROP

TCP: Drop Packets example
-p sport, dport
--tcp-flags SYN, FIN, SYNACK, ACK
iptables -A INPUT -p tcp --dport 22 -j DROP

UDP:
TFTP- booting systems and updating infrastructure like Cisco (port 69)
Syslog- port 514
NTP (port 123)
DHCP (ports 67 and 68)
With UDP (unlike TCP), the source and destination ports are often the same

Example to restrict access to syslog:
iptables -A INPUT -p udp --dport 514 -s 192.168.1.1 -j ACCEPT
IP is the gateway, which should have access.
iptables -A INPUT -p udp --dport 514 -s ! 192.168.1.1 -j DROP
Deny everyone else.

Lock down NTP:
iptables -A INPUT -p udp --dport 123 --sport 123 -s 129.6.15.2+ -j ACCEPT ---put in real NTP servers first
(129.6.15.28 and 129.6.15.29 are NIST servers. Using "+" wildcard here)
iptables -A INPUT -p udp --dport 123 --sport 123 -s ! 192.168.1.0/24 -j DROP
Else, if they aren't from our subnet DROP em!

Supported ICMP types: echo-request/ reply
iptables -p ICMP --icmp-type
iptables -p icmp --help ----- lists types for us to view
iptables -A INPUT -p icmp --icmp-type echo-reply -j DROP ----deny from all hosts

iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP ----don't reply to all hosts
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j DROP ----don't reply to all hosts

---

Multiport Matches
- Multiple ports with fewer rules. combine 2 ports in one line:
iptables -A INPUT -p tcp -m multiport --dport 8080,23 -j DROP
iptables -A INPUT -p tcp -m multiport --dport 8080,23 -s ! 127.0.0.1 -j DROP
15 ports maximum per rule

MAC address filtering- more secure than IP address
iptables -A INPUT -p tcp -m mac --mac-source 40:6c:8f:47:84:d0 -dport 8080,23 -j DROP

State Machine
(IPTable's Statefullness) (TCP/UDP/ICMP)
NEW(syn), ESTABLISHED(syn-ack), RELATED, INVALID
Permit host to initiate (syn) but deny other hosts from initiating traffic to our hosts.

iptables -I INTRANET 3 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -j ACCEPT
The end implicit drop/deny would prevent any new connections from being initiated from another machine.

NAT- port redirection
iptables -t nat -A PREROUTING -p tcp --dport 2323 -j REDIRECT --to-ports 23
More on NAT table later

Logging
Goes in /var/log/messages by default. Add exception "kern.none" to line logging anything over "info" level and add line to provision further instructions
/etc/syslog.conf
uncomment #kern.* /dev/console/ and change to /var/log/firewall

iptables -A INPUT -p tcp --dport telnet -j LOG
Put these at the top of the INPUT chain (or even better, a subchain pointed to there)
Default level is "warning"
iptables -A INPUT -p tcp -m multiport --dport 8080,23 -j LOG
iptables -A INPUT -p tcp -m multiport --dport ! 8080,23 -j LOG
iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "SSH ACCESS ATTEMPT: "
iptables -A INPUT -p tcp --dport 23 -j LOG --log-prefix "UNAUTHORIZED TELNET ACCESS ATTEMPT "

--log-level debug emerg etc
--log-tcp-options
--log-ip-options
--log-tcp-sequence

Forwarding/Routing
/sbin/sysctl -a lists kernel variables. Network vars are stored in /proc/sys/net/
To turn on forwarding, run "echo 1 > /proc/sys/net/ipv4/ip_forward" to set the boolean
Also, you can go to /etc/sysctl.conf and set the boolean to 1 as well for defaults

(/proc/sys/net/ip_conntrack is also a good file to check for troubleshooting rules in iptables)

On other machines, "route add -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.1.20"
Syntax: route add -net SUBNET netmask 255.255.255.0 gw GATEWAY --- where gateway is our IPTables router
netstat -rn and -na options (show routing and active connections respectively)
route

The following logs specific traffic on port, then allows all subnet hosts to access that specific service in both directions (to allow handshake, initialization, and established connections to said service)
iptables -P FORWARD DROP --- change chain default policy from ACCEPT to DROP
iptables -N LOGGINGFORWARD --create new chain to log all forwarding traffic
iptables -A FORWARD -j LOGGINGFORWARD ---- tell FORWARD to pass it to the new chain
iptables -A LOGGINGFORWARD -s 192.168.1.0/24 -d 10.0.0.50 -p tcp --dport 3389 -j LOG ---log term svcs packets

iptables -A FORWARD -s 192.168.1.0/24 -d 10.0.0.50 -p tcp --dport 3389 -j ACCEPT
iptables -A FORWARD -d 192.168.1.0/24 -s 10.0.0.50 -p tcp --sport 3389 -j ACCEPT
--- after LOGGINGFORWARD, traffic passes through back to FORWARD chain. Then permit in FORWARD terminal services packets in subnet (in both directions) to 10.0.0.50 (this gets replaced in second example block below)

------ADD MORE TO SAME RULESET
The first rules in the block rules in the previous example Re: port 3389 in the FORWARD chain
iptables -A FORWARD -m state --state NEW,ESTABLISHED -j ACCEPT ---permit new and established sessions
iptables -A FORWARD -m state --state ESTABLISHED -j ACCEPT --ensure syn-ack can be accepted
iptables -A FORWARD -s 10.0.0.0/24 -p tcp -m multiport --dport 8080,23 -j ACCEPT --- let subnet use http and SSL
iptables -A FORWARD -p udp --dport 53 -s 10.0.0.0/24 -d 192.168.1.0 -j ACCEPT
--- allow DNS queries BUT- ONLY from our local network (we don't want to make queries outside to strange servers)

################## IPTables NAT

NAT Masquerading
(on subnet1) 10.0.0.50 ---> Firewall masquerades address to its IP 192.168.1.20 ---> subnet2

SNAT/Masquerading outside traffic to destination = POSTROUTING
DNAT = PREROUTING/Masquerading
Local NAT = OUTPUT

NAT Masquerading example:
iptables -t nat -A POSTROUTING -j MASQUERADE -s 10.0.0.0/24 -d 192.168.1.0/24
--- this makes 192.168.1.30 think that 10.0.0.50's pings are coming from the netfilter firewall 192.168.1.20
Firewall masquerades the real destination address to be its own 192.168.1.20 address

iptables -t nat -R POSTROUTING 1 -j MASQUERADE -s 10.0.0.0/24 -d 192.168.1.10
--- this replaces the above- 192.168.1.30 think that 10.0.0.50's pings are coming from 192.168.1.10
Firewall masquerades the destination address as it’s peer’s IP: 192.168.1.10

--- no destination means use IP of netfilter firewall as default (primary interface's IP (of eth0)

Forcing TCP source-port range to 1024-10240
iptables -t nat -p tcp -R POSTROUTING 1 -j MASQUERADE --to-ports 1024-10240

--- with this, a telnet 192.168.1.10 22 from 10.0.0.50 looks like it came from 192.168.1.20 at port 1224 instead of 22 (with firewall's IP since we didn't define one for the masquerade operation)

---------- SNAT
Applies with static IPs. 1-to-1 or 1-to-many mappings.
iptables -t nat -p tcp -R POSTROUTING 1 -j SNAT --to-source 192.168.1.20:1024-10240 -s 10.0.0.0/24
--- this does the same as the masquerading port range above, but retains 10.0.0.0/24 IP

iptables -t nat -R POSTROUTING 1 -j SNAT --to-source 192.168.1.20 -s 10.0.0.0/24
--- removing the port information, pings appear to come from 192.168.1.20

Bind multiple addresses to eth0 (public/internet) interface (eth1 used for 10.0.0.1)
- SNAT 10.0.0.0/24 traffic using 192.168.1.21
[Preliminary- set up subinterface on eth0. Could not run system-config-network GUI due to BastilleLinux. Duplicated /etc/sysconfig/network-scripts/ifcfg-eth0. cp ifcfg-eth0 ifcfg-eth0\:1 for new subinterface. Edit ifcfg-eth0:1 and give new IP address of 192.168.1.21]

iptables -t nat -R POSTROUTING 1 -j SNAT --to-source 192.168.1.21 -s 10.0.0.0/24

At this point flush the NAT table.

Use source 192.168.1.21 when communicating with 192.168.1.10, and 192.168.1.22 (make eth0:2) when communicating with all other IPs
iptables -t nat -I POSTROUTING 1 -p tcp -j SNAT --to-source 192.168.1.21 -d 192.168.1.10 -s 10.0.0.0/24
iptables -t nat -A POSTROUTING 1 -p tcp -j SNAT --to-source 192.168.1.22 -s 10.0.0.0/24
Result: (telnet) packets from 10.0.0.0/24 to 192.168.1.10 appear as if coming from 192.168.1.21, packets from 10.0.0.0/24 to 192.168.1.30 appear as if coming from 192.168.1.22

---------- DNAT - before routing occurs
(on subnet1) 10.0.0.50 ---> Firewall IP 192.168.1.20 ---> subnet2 (connected to internet)
- permits connections to unprotected hosts behind the firewall

Demonstration: publish to the internet port 3389 and point it at Windows box (10.0.0.50)
iptables -t nat -A PREROUTING -p tcp --dport 3389 -j DNAT --to-destination 10.0.0.50 -d 192-168.1.20
-- the firewall is given as the destination, then to be DNAT'd
-- the forwarding chain will have to be able to allow the packet to pass for this to work! It does the actual routing!
-- it is important to not DNAT ports that the firewall is listening to (or vice versa)

iptables -t nat -A PREROUTING -p tcp --dport 3389 -d 192.168.1.21 -j DNAT --to-destination 10.0.0.50
--- with this, trying to telnet to 192.168.1.20:3389 will fail, but 192.168.1.21:3389 will work

----------Netmap NAT
Subnet 1 = 10.0.0.0/24 Subnet 2=192.168.1.0/24
Hide IPs on LAN- watch out for conflicts! Host octet is the same on both subnets specified. Is useful only when you can specify an uncommon subnet (applications such as VPNs)
iptables -t nat -A PREROUTING -s 10.0.0.0/24 -j NETMAP --to 192.168.1.0/24

----------IPTables DMZs

---Interfaces:
eth0 - 192.168.1.20 - Internet facing interface - subnet3 - external host
eth0:1, 192.168.1.21
eth0:2 192.168.1.22
eth1- 10.0.0.1 - Corporate internal subnet - subnet1
eth2- 172.16.75.1 DMZ - subnet2 with host 172.16.75.2 (need port 22 and 80 open) NAT to 192.168.1.20

eth2 needs 2 NAT entries for PAT (port address translation):
iptables -t nat -A PREROUTING -p tcp --dport 22 -d 192.168.1.202 -j DNAT --to-destination 172.16.75.2
iptables -t nat -A PREROUTING -p tcp --dport 80 -d 192.168.1.202 -j DNAT --to-destination 172.16.75.2
Ensure the ports are open, and attempt to connect from inside and outside of subnet

---------- DMZ Forwarding Awareness
Accessing DMZ host from a 192.168.1.10 (outside firewall)

Had to add to the routing table first:
- route add -net 172.16.75.0 netmask 255.255.255.0 gw 192.168.1.20
- was able to ssh into DMZ box (security risk) - iptables is forwarding regardless of position with DMZ- patch.
NO ACCESS FROM OUTSIDE INTO DMZ!
Netfilters need to be tweaked to allow only connections from trusted networks/subnet

iptables -P FORWARD DROP
iptables -A FORWARD -s 10.0.0.0/24 -j ACCEPT
iptables -A FORWARD -s 172.16.75.0/24 -m state --state established -j ACCEPT
-- use established only- new will leave a hole!

---------- Dual DMZ Configuration (tier1/tier2)
--- Layers:
"subnet3" (open to the internet)
Tier1- eth2- subnet2 - 172.16.75.0/24 - Web Tier (DMZ1) -- entry. (webserver, etc) Sources to DMZ2
Tier2- eth3- subnet4 - 172.17.76.0/24 - Middleware Tier (DMZ2) -- (websphere, jboss) Sources to internal subnet.
Tier3- eth1- subnet1 - 10.0.0.1 - Corporate internal subnet (perhaps RDBMS, etc)

-- Permit ONLY subnet2 (DMZ1) to talk to subnet4 (DMZ2)
Add to existing rules:
iptables -A FORWARD -s 172.16.75.0/24 -d 172.17.76.0/24 -j ACCEPT
iptables -A FORWARD -s 172.17.76.0/24 -d 172.16.75.0/24 -m state --state established -j ACCEPT
TEST IT

-- Permit subnet4 (DMZ2) to source to subnet1 (internal network)
iptables -A FORWARD -s 172.17.76.0/24 -d 10.0.0.0/24 -p tcp --dport 1433 -j ACCEPT (port for MS SQL)

#################### iptables6

/etc/rc.d/init.d/ip6tables ---------run-script
/etc/sysconfig/iptables-config -----systemwide config file

- uses IPv6-specific modules, but also some others, (eui64, trace, connection marking, queuing, etc.)

/sbin/ip6tables ----primary IPv6 ACL modifier utility
/sbin/ip6tables-restore ---- restores rules of current ip6tables instance
/sbin/ip6tables-save ---- saves rules to STDOUT by default (send to file)

/etc/sysconfig/iptables-config
adds loading unload of modules, as well as the save_on_stop/restart

-- Has filter and mangle tables
-- Instead of a NAT table ip6tables uses a Raw table to handle raw packets
---Raw has 2 chains: PREROUTING and OUTPUT

RULES CHEATSHEET

Note, most of these expect a default drop (ruleset #2) with a second action. Some may need it added.

```
# 1. Delete all existing rules
iptables -F

# 2. Set default chain policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# 3. Block a specific ip-address
#BLOCK_THIS_IP="x.x.x.x"
#iptables -A INPUT -s "$BLOCK_THIS_IP" -j DROP

# 8. Allow outgoing SSH
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 9. Allow outgoing SSH only to a specific network
#iptables -A OUTPUT -o eth0 -p tcp -d 192.168.101.0/24 --dport 22 -m state --state NEW,ESTABLISHED -j
ACCEPT
#iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 4. Allow ALL incoming SSH
#iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
#iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 5. Allow incoming SSH only from a sepcific network
#iptables -A INPUT -i eth0 -p tcp -s 192.168.200.0/24 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
#iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 6. Allow incoming HTTP
#iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
#iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT

# Allow incoming HTTPS
#iptables -A INPUT -i eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
#iptables -A OUTPUT -o eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

# 7. MultiPorts (Allow incoming SSH, HTTP, and HTTPS)
iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,80,443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports 22,80,443 -m state --state ESTABLISHED -j ACCEPT
```

```
# 10. Allow outgoing HTTPS
iptables -A OUTPUT -o eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

# 11. Load balance incoming HTTPS traffic
#iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --every 3 --packet 0 -j
DNAT --to-destination 192.168.1.101:443
#iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --every 3 --packet 1 -j
DNAT --to-destination 192.168.1.102:443
#iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --every 3 --packet 2 -j
DNAT --to-destination 192.168.1.103:443

# 12. Ping from inside to outside
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

# 13. Ping from outside to inside
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT

# 14. Allow loopback access
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# 15. Allow packets from internal network to reach external network.
# if eth1 is connected to external network (internet)
# if eth0 is connected to internal network (192.168.1.x)
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT

# 16. Open DNS ports
#iptables -A INPUT -p tcp -m state --state NEW --dport 53 -j ACCEPT
#iptables -A INPUT -p udp -m state --state NEW --dport 53 -j ACCEPT

# 17. Allow NIS Connections
# rpcinfo -p | grep ypbind ; This port is 853 and 850
#iptables -A INPUT -p tcp --dport 111 -j ACCEPT
#iptables -A INPUT -p udp --dport 111 -j ACCEPT
#iptables -A INPUT -p tcp --dport 853 -j ACCEPT
#iptables -A INPUT -p udp --dport 853 -j ACCEPT
#iptables -A INPUT -p tcp --dport 850 -j ACCEPT
#iptables -A INPUT -p udp --dport 850 -j ACCEPT

# 18. Allow rsync from a specific network
iptables -A INPUT -i eth0 -p tcp -s 192.168.101.0/24 --dport 873 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 873 -m state --state ESTABLISHED -j ACCEPT

# 19. Allow MySQL connection only from a specific network
iptables -A INPUT -i eth0 -p tcp -s 192.168.200.0/24 --dport 3306 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 3306 -m state --state ESTABLISHED -j ACCEPT

# 20. Allow CUPS printing service udp/tcp ports 631 for LAN users
```

```
iptables -A INPUT -s 192.168.200.0/24 --dport 631 -p udp -m udp -j ACCEPT
iptables -A INPUT -s 192.168.200.0/24 --dport 631 -p tcp -m tcp -j ACCEPT

# 21. Allow NTP synch for LAN users
iptables -A INPUT -p udp -s 192.168.200.0/24 --dport 123 -m state --state NEW -j ACCEPT

# 22. Open Samba for LAN users
#iptables -A INPUT -s 192.168.200.0/24 -p tcp --dport 137 -m state --state NEW -j ACCEPT
#iptables -A INPUT -s 192.168.200.0/24 -p tcp --dport 138 -m state --state NEW -j ACCEPT
#iptables -A INPUT -s 192.168.200.0/24 -p tcp --dport 139 -m state --state NEW -j ACCEPT
#iptables -A INPUT -s 192.168.200.0/24 -p tcp --dport 445 -m state --state NEW -j ACCEPT

# 23 Allow

# 23 Allow SMTP and Secure SMTP
iptables -A INPUT -i eth0 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 465 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 465 -m state --state ESTABLISHED -j ACCEPT

# 24. Allow IMAP and IMAPS
iptables -A INPUT -i eth0 -p tcp --dport 143 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 143 -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 993 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 993 -m state --state ESTABLISHED -j ACCEPT

# 25. Allow POP3 and POP3S
iptables -A INPUT -i eth0 -p tcp --dport 110 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 110 -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 995 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 995 -m state --state ESTABLISHED -j ACCEPT

# 26. Prevent DoS attack
iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-burst 100 -j ACCEPT

# 27. Port forwarding 422 to 22
iptables -t nat -A PREROUTING -p tcp -d 192.168.102.37 --dport 422 -j DNAT --to 192.168.102.37:22
iptables -A INPUT -i eth0 -p tcp --dport 422 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 422 -m state --state ESTABLISHED -j ACCEPT

# 28. Log dropped packets
iptables -N LOGGING
iptables -A INPUT -j LOGGING
iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables Packet Dropped: " --log-level 7
iptables -A LOGGING -j DROP
```

Firewalld Stuff From Vugt RHCE

firewalld replaces iptables as the RHEL default frontend to netfilter in RHEL7
(Many utilities write directly to firewalld)
Interfaces are assigned to zones (public, DMZ, private)
Services are connected to zones

Executables include:
firewall-cmd, firewall-config (GUI), firewalld, firewall-offline-cmd
firewall-cmd --get-zones, --get-services, -- list-services commands
get- lists all, list- show only currently active
firewall-cmd --list-all --- dumps all the info on current zone

/usr/lib/firewalld/services contains XML files.. like ldap.xml, ftp.xml
They are system default services, contain usually:

```
<service>
        <short>FTP</short>
        <description>A description here</description>
        <port protocol="tcp" port="21"/>
        <module name="nf-conntrack_ftp"/>      (optional kernel module needed to load
</service>
```

You can add or change services, but they need to be put in /etc/firewalld/services. Order of xml tags in <service> block doesn't really matter

Nothing in firewall-cmd is persistent after reboot unless you add "--permanent"
In the GUI there is also a choice between "runtime" and permanent"
You can issue firewall-cmd --state and --reload in place of systemctl/service commands

In /etc/firewalld/ there is firewalld.conf, lockdown-whitelist.xml, and then services, zones and icmp-types directories

firewalld.conf elements:
DefaultZone=public
MinimalMark=100
CleanUpOnExit
Lockdown=no
IPv6_rpfilter=yes

systemctl stop iptables
systemctl mask iptables --- mask keeps something from being enabled by accident
systemctl unmask filewalld
systemctl start filewalld

```
firewall-cmd --get-active-zones
public
        interfaces: eno133433
firewall-cmd --get-active-zones --list-all
public (default, active)
        interfaces: eno133433
        sources:
        services: dhcpv6-client  ssh
        ports:3260/tcp
        masquerade: no
        forward-ports:
        icmp-blocks:
        rich-rules:
firewall-cmd --get-active-zones --list-all --zone=dmz
dmz
        interfaces: eno133433
        sources:
        services: ssh
```

ports:
masquerade: no
forward-ports:
icmp-blocks:
rich-rules:

firewall-cmd --add-service=vnc-server --zone=dmz ---if not specified default zone, becomes active immediately
firewall-cmd --add-service=vnc-server --zone=dmz --permanent
firewall-cmd --reload --remember that if you use the permanent option, you **don't** activate it immediately
firewall-cmd --add-source=10.0.0.0/24
firewall-cmd --add-port= 8000/tcp <---for ports, you need to specify type

Direct rules are not recommended- use rich rules instead

man firewalld-rich-language

Order of filtering:

port forwarding and masquerading
login rules
allow rules
deny rules

add-rich-rule remove-rich-rules create-rich-rules list-rich-rules

firewall-cmd --permanent --zone=public --add rich-rule='rule family=ip4 source address=10.0.0.100/32 reject'
firewall-cmd --permanent --add rich-rule='rule service name=http limit value=3/m accept'
--3 packets PER minute
firewall-cmd --permanent --add rich-rule='rule protocol value=cbt accept'
-it refers to the /etc/protocol listing for protocols
firewall-cmd --permanent --zone=public --add rich-rule='rule family=ip4 source address=10.0.0.1/24 port port=7936-7985 protocol=tcp accept'
-- specify port range
firewall-cmd --permanent --add rich-rule='rule service name="ssh" log prefix="ssh" level='notice" limit value=2/m accept'

```
[root@server1 ~]# firewall-cmd --list-all
public (default, active)
  interfaces: eno16777736
  sources: 10.0.0.0/24
  services: dhcpv6-client ssh vnc-server
  ports: 3260/tcp 8000/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
        rule family="ipv4" source address="10.0.0.0/24" port port="7900-7905" protocol="tcp" accept
        rule service name="ssh" log prefix="ssh" level="notice" limit value="2/m" accept
        rule family="ipv4" source address="10.0.0.100/32" reject
        rule service name="http" accept limit value="3/m"
        rule protocol value="cbt" accept
```

NAT AKA Masquerading- EASY!
firewall-cmd --permanent --zone=public --add-masquerade
firewall-cmd --reload
--- DONE ---

firewall-cmd --permanent --zone=public --add-forward-port=888:proto=tcp:toport=80:toaddr=10.0.0.10
So with the masquerading, 888 listening on firewall, sending it out through port 80

Exercise 7

- Perform all tasks on server1
- Configure a service with the name myservice; this service should be configured for accessing port 2022
- Configure SSH to listen on port 2022 and verify from server2 that SSH is accessible on server1
- On server1, configure port forwarding to make the SSHD process is available on port 2222 as well
- In case of trouble, use nmap to check

cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/myservice.xml

- edit and change to listen on port 2022
- vim /etc/ssh/sshd_config, change listening port
- systemctl restart sshd; doesn't work
- semanage port –l | grep ssh
- semanage port –a –t ssh_port_t –p tcp 2022
- systemctl restart sshd
- firewall-cmd --get-services
- firewall-cmd --add-service=myservice --permanent
- firewall-cmd --reload
- firewall-cmd --permanent --zone=public --add-forward-port=port=2222:proto=tcp:toport=2022
- firewall-cmd –reload
- (from server2) ssh –p 2222 server1
- (from server2) nmap server1
- (from server1) netstat -tulpen

Note no IP on the port forwarding- local port forwarding only.

firewall-cmd --get-default-zone
home
firewall-cmd --set-default-zone dmz
success
firewall-cmd --permanent --add-service=ssh
firewall-cmd --permanent --add-service=ntp
firewall-cmd --permanent --add-service=ftp
firewall-cmd --permanent --add-service=vnc-server
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=dns
firewall-cmd --list all
won't show - we added them as perm and not runtime
systemctl restart firewalld
firewall-cmd --list all
now works

Orphaned FirewallD stuff - (some IPtables stuff at the end) - from firewalld.rtf

Both firewalld and the iptables service use netfilter framework in the kernel through the same interface, not surprisingly, the iptables command. However, as opposed to the iptables service, firewalld can change the settings during normal system operation without existing connections being lost.

Instead of INPUT OUTPUT AND FORWARD chains, there are zones

You can use either firewall-config tool (GUI version) or firewall-cmd command (command line version) to set up your firewall. Systemd manages FirewallD by using firewalld.service unit file.

yum info firewalld firewall-config
yum install firewalld firewall-config

Operating System : CentOS Linux release 7.0.1406 (Core)
IP Address :192.168.0.55
Host-name : server1.domain-namelocal.com

Make sure iptables isn't running, do the same with ip6tables:
systemctl status iptables
systemctl stop iptables
systemctl mask iptables

Iptables put it's config into /etc/sysconfig/iptables. Firewalld stores its configuration across two directories, /usr/lib/firewalld and /etc/firewalld:
ls /usr/lib/firewalld /etc/firewalld
Zone files: /etc/firewalld/zones/zone.xml and /usr/lib/firewalld/zones/zone.xml

Firewalld comes with some default zones that can be copied and customized:
Drop Zone: Any incoming packets are dropped, same as "iptables -j drop". No reply, only outgoing network connections will be available.
Block Zone: Incoming network connections are rejected with an icmp-host-prohibited and icmp6-adm-prohibited for IPv6. Only network connections initiated from within the system are possible.
Public Zone: For use in public areas. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.
External Zone: For use on external networks with masquerading enabled especially for routers. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.
DMZ Zone: For computers that are publicly-accessible with limited access to your internal network. Only selected incoming connections are accepted.
Work Zone: You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.
Home Zone: You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.
Internal Zone: You mostly trust the other computers on the networks to not harm your computer. Only selected incoming connections are accepted.
Trusted Zone: All the traffic is accepted.

systemctl status -l firewalld (-l is full, don't truncate output)
firewall-cmd --get-zones
firewall-cmd --get-default-zone
firewall-cmd --list-all-zones OR # firewall-cmd --get-active-zones
If the zones have any rich-rules, enabled services or ports will be also listed with respective zone information
firewall-cmd --set-default-zone=internal
firewall-cmd --get-zone-of-interface=enp0s3

firewall-cmd --get-icmptypes
firewall-cmd --get-services
Services which are enabled, will be automatically loaded when the Firewalld service up and running

To get the list of all the default available services, ls the following directory:
cd /usr/lib/firewalld/services/
Each XML file simply lists name, description, and specifies protocol and port. Just copy and edit one to customize.
cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/
mv ssh.xml rtmp.xml
Open nano, add port 1935 and adjust info
firewall-cmd --reload
firewall-cmd --get-services ---to see if it was added

firewall-cmd --state
firewall-cmd --get-active-zones
Public zone for interface enp0s3, this is the default interface, which is defined in /etc/firewalld/firewalld.conf file as DefaultZone=public.
To list all available services in this default interface zone.
firewall-cmd --get-service OR # firewall-cmd --zone=public --list-all

firewall-cmd --add-service=rtmp
firewall-cmd --zone=public --remove-service=rtmp

The above step was temporary. To make it permanent run
firewall-cmd --add-service=rtmp --permanent
firewall-cmd --reload

Do both for immediate effect. The following specifies a different zone than default:
firewall-cmd --zone=MyZone --add-service=http
firewall-cmd --zone=MyZone --permanent --add-service=http
firewall-cmd --zone=MyZone --add-service=https
firewall-cmd --zone=MyZone --permanent --add-service=https
firewall-cmd --reload

To define rules for a network source range and open any of the ports. For example, 192.168.0.0/24 and tcp 1935:

firewall-cmd --permanent --add-source=192.168.0.0/24
firewall-cmd --permanent --add-port=1935/tcp
Make sure to reload firewalld service after adding or removing any services or ports.
firewall-cmd --reload
firewall-cmd --list-all

Adding Rich Rules for Network Range
To allow the services http, https, vnc-server, PostgreSQL, use the following rules.
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="http" accept'
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="http" accept' --permanent
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="https" accept'
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="https" accept' --permanent
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="vnc-server" accept'
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="vnc-server" accept' --permanent
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="postgresql" accept'
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.0.0/24" service name="postgresql" accept' --permanent
firewall-cmd --reload
firewall-cmd --list-all

firewall-cmd --permanent --zone=public --add-port=80/tcp
firewall-cmd --zone=public --remove-port=80/tcp

firewall-cmd --zone=public --list-ports

Pre-defined services can also be added or removed by name:
firewall-cmd --zone=public --add-service=ftp
firewall-cmd --zone=public --remove-service=ftp
firewall-cmd --zone=public --list-services

Panic Mode- Block Incoming and Outgoing Packets
The following rule will drop any existing established connection on the system.
After enabling panic mode, try to ping any domain (say google.com) and check whether the panic mode is ON using '–query-panic' option as listed below.
firewall-cmd --panic-on
ping google.com -c 1
firewall-cmd --query-panic

Disable the panic mode and then once again ping and check.
firewall-cmd --panic-off
firewall-cmd --query-panic
ping google.com -c 1

IP / Port forwarding
Let's say you use the box not only as a software firewall, but also as the actual hardware-based one that sits between two distinct networks.

Here is the iptables method:

IP forwarding must have been already enabled in your system. If not, you need to edit /etc/sysctl.conf and set the value of net.ipv4.ip_forward to 1, as follows:
net.ipv4.ip_forward = 1
then save the change, close your text editor and finally run the following command to apply the change:
sysctl -p /etc/sysctl.conf
You may have a printer installed at an internal box with IP 192.168.0.10, with the CUPS service listening on port 631 (both on the print server and on your firewall). In order to forward print requests from clients on the other side of the firewall, you should add the following iptables rule:
iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 631 -j DNAT --to 192.168.0.10:631

Here is how to do that in firewalld:

Is masquerading is enabled?
firewall-cmd --zone=MyZone --query-masquerade
firewall-cmd --zone=MyZone --add-masquerade
firewall-cmd --zone=external --add-forward-port=port=631:proto=tcp:toport=631:toaddr=192.168.0.10

http://www.tecmint.com/shorewall-a-high-level-firewall-for-configuring-linux-servers/

http://www.tecmint.com/firestarter-a-high-level-graphical-interface-iptables-firewall-for-linux-systems/

1. Have you heard of iptables and firewall in Linux? Any idea of what they are and for what it is used?
Answer : I've been using iptables for quite long time and I am aware of both iptables and firewall. Iptables is an application program mostly written in C Programming Language and is released under GNU General Public License. Written for System administration point of view, the latest stable release if iptables 1.4.21.iptables may be considered as firewall for UNIX like operating system which can be called as iptables/netfilter, more accurately. The Administrator interact with iptables via console/GUI front end tools to add and define firewall rules into predefined tables. Netfilter is a module built inside of kernel that do the job of filtering.

Firewalld is the latest implementation of filtering rules in RHEL/CentOS 7 (may be implemented in other distributions which I may not be aware of). It has replaced iptables interface and connects to netfilter.
2. Have you used some kind of GUI based front end tool for iptables or the Linux Command Line?
Answer : Though I have used both the GUI based front end tools for iptables like Shorewall in conjugation of Webmin in GUI and Direct access to iptables via console.And I must admit that direct access to iptables via Linux console gives a user immense power in the form of higher degree of flexibility and better understanding of what is going on in the background, if not anything other. GUI is for novice administrator while console is for experienced.

3. What are the basic differences between between iptables and firewalld?
Answer : iptables and firewalld serves the same purpose (Packet Filtering) but with different approach. iptables flush the entire rules set each time a change is made unlike firewalld. Typically the location of iptables configuration lies at '/etc/sysconfig/iptables' whereas firewalld configuration lies at '/etc/firewalld/', which is a set of XML files.Configuring a XML based firewalld is easier as compared to configuration of iptables, however same task can be achieved using both the packet filtering application ie., iptables and firewalld. Firewalld runs iptables under its hood along with it's own command line interface and configuration file that is XML based and said above.

4. Would you replace iptables with firewalld on all your servers, if given a chance?
Answer : I am familiar with iptables and it's working and if there is nothing that requires dynamic aspect of firewalld, there seems no reason to migrate all my configuration from iptables to firewalld.In most of the cases, so far I have never seen iptables creating an issue. Also the general rule of Information technology says "why fix if it is not broken". However this is my personal thought and I would never mind implementing firewalld if the Organization is going to replace iptables with firewalld.

5. You seems confident with iptables and the plus point is even we are using iptables on our server.

What are the tables used in iptables? Give a brief description of the tables used in iptables and the chains they support.
Answer : Thanks for the recognition. Moving to question part, There are four tables used in iptables, namely they are:

- Nat Table
- Mangle Table
- Filter Table
- Raw Table

Nat Table : Nat table is primarily used for Network Address Translation. Masqueraded packets get their IP address altered as per the rules in the table. Packets in the stream traverse Nat Table only once. ie., If a packet from a jet of Packets is masqueraded the rest of the packages in the stream will not traverse through this table again. It is recommended not to filter in this table. Chains Supported by NAT Table are PREROUTING Chain, POSTROUTING Chain and OUTPUT Chain.

Mangle Table : As the name suggests, this table serves for mangling the packets. It is used for Special package alteration. It can be used to alter the content of different packets and their headers. Mangle table can't be used for Masquerading. Supported chains are PREROUTING Chain, OUTPUT Chain, Forward Chain, INPUT Chain, POSTROUTING Chain.

Filter Table : Filter Table is the default table used in iptables. It is used for filtering Packets. If no rules are defined, Filter Table is taken as default table and filtering is done on the basis of this table. Supported Chains are INPUT Chain, OUTPUT Chain, FORWARD Chain.

Raw Table : Raw table comes into action when we want to configure packages that were exempted earlier. It supports PREROUTING Chain and OUTPUT Chain.
6. What are the target values (that can be specified in target) in iptables and what they do, be brief!
Answer : Following are the target values that we can specify in target in iptables:

ACCEPT : Accept Packets
QUEUE : Pass Package to user space (place where application and drivers reside)
DROP : Drop Packets
RETURN : Return Control to calling chain and stop executing next set of rules for the current Packets in the chain.

7. Lets move to the technical aspects of iptables, by technical I mean practical.

How will you Check iptables rpm that is required to install iptables in CentOS?.
Answer : iptables rpm are included in standard CentOS installation and we do not need to install it separately. We can check the rpm as:

```
# rpm -qa iptables

iptables-1.4.21-13.el7.x86_64
```

If you need to install it, you may do yum to get it.

```
# yum install iptables-services
```

8. How to Check and ensure if iptables service is running?
Answer : To check the status of iptables, you may run the following command on the terminal.

```
# service status iptables                    [On CentOS 6/5]
# systemctl status iptables                          [On CentOS 7]
```

If it is not running, the below command may be executed.

```
---------- On CentOS 6/5 ----------
# chkconfig --level 35 iptables on
# service iptables start

---------- On CentOS 7 ----------
# systemctl enable iptables
# systemctl start iptables
```

We may also check if the iptables module is loaded or not, as:

```
# lsmod | grep ip_tables
```

9. How will you review the current Rules defined in iptables?
Answer : The current rules in iptables can be review as simple as:

```
# iptables -L
```

Sample Output

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
```

```
ACCEPT     icmp --  anywhere            anywhere
ACCEPT     all  --  anywhere            anywhere
ACCEPT     tcp  --  anywhere            anywhere            state NEW tcp dpt:ssh
REJECT     all  --  anywhere            anywhere            reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source              destination
REJECT     all  --  anywhere            anywhere            reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source              destination
```

10. How will you flush all iptables rules or a particular chain?
Answer : To flush a particular iptables chain, you may use following commands.

```
# iptables --flush OUTPUT
```

To Flush all the iptables rules.

```
# iptables --flush
```

11. Add a rule in iptables to accept packets from a trusted IP Address (say 192.168.0.7)
Answer : The above scenario can be achieved simply by running the below command.

```
# iptables -A INPUT -s 192.168.0.7 -j ACCEPT
```

We may include standard slash or subnet mask in the source as:

```
# iptables -A INPUT -s 192.168.0.7/24 -j ACCEPT
# iptables -A INPUT -s 192.168.0.7/255.255.255.0 -j ACCEPT
```

12. How to add rules to ACCEPT, REJECT, DENY and DROP ssh service in iptables.
Answer : Hoping ssh is running on port 22, which is also the default port for ssh, we can add rule to iptables as:To ACCEPT tcp packets for ssh service (port 22).

```
# iptables -A INPUT -s -p tcp - -dport  -j ACCEPT
```

To REJECT tcp packets for ssh service (port 22).

```
# iptables -A INPUT -s -p tcp - -dport  -j REJECT
```

To DENY tcp packets for ssh service (port 22).

```
# iptables -A INPUT -s -p tcp - -dport  -j DENY
```

To DROP tcp packets for ssh service (port 22).

```
# iptables -A INPUT -s -p tcp - -dport  -j DROP
```

13. Let me give you a scenario. Say there is a machine the local ip address of which is 192.168.0.6. You need to block connections on port 21, 22, 23, and 80 to your machine. What will you do?
Answer : Well all I need to use is the ‘multiport‘ option with iptables followed by port numbers to be blocked and the above scenario can be achieved in a single go as.

```
# iptables -A INPUT -s 192.168.0.6 -p tcp -m multiport --dport 22,23,80,8080 -j DROP
```

The written rules can be checked using the below command.

```
# iptables -L

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere               anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere              anywhere             state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited
DROP       tcp  --  192.168.0.6          anywhere            multiport dports ssh,telnet,http,webcache

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

http://ktaraghi.blogspot.com/2013/10/what-is-firewalld-and-how-it-works.html
http://manpages.ubuntu.com/manpages/saucy/man5/firewalld.zone.5.html

```
        <rule [invert="ipv4|ipv6"]/>
          [ <source address="address[/mask]" [invert="bool"]/> ]
          [ <destination address="address[/mask]" [invert="bool"]/> ]
          [
            <service name="string"/> |
            <port port="portid[-portid]" protocol="tcp|udp"/> |
            <protocol value="protocol"/> |
            <icmp-block name="icmptype"/> |
            <masquerade/> |
            <forward-port port="portid[-portid]" protocol="tcp|udp" [to-port="portid[-portid]"] [to-addr="address"]/>
          ]
          [ <log [prefix="prefixtext"] [level="emerg|alert|crit|err|warn|notice|info|debug"]/> [<limit value="rate/duration"/>]
</log> ]
          [ <audit> [<limit value="rate/duration"/>] </audit> ]
          [ <accept/> | <reject [type="rejecttype"]/> | <drop/> ]
        </rule>
```

SELINUX

Basic Concepts

- Integrated into the kernel, Implemented/enabled in RHEL by default
- Compiled into the kernel and supported via loadable security modules (LSMs)
- Restricts access by subjects (users and/or processes) to objects (files)
- Provides Mandatory Access Controls (MACs) which extend default Discretionary Access Controls (DACs)
- (DACs are basic linux permissions based solely on user identity)
- DACs are prone to malware, daemons, malicious activity. Plus, setuid/gid is vulnerable
- MAC-based checks happen AFTER the DAC-based checks
- Stores MAC permissions in extended attributes of file system
- Separates users, processes and objects via labeling and monitors/controls their interaction
- Implements sandboxes for subjects and objects
- Creates sandboxes (domains) for 'targeted' daemons and one major sandbox for everything else (unconfined_t)
- i.e., httpd and ntpd have their own sandboxes that cannot interact with eachothers sandbox
- SELinux denies interaction between subjects and objects by default (targeted)
- SElinux gets a mount point like /proc (i.e., a pseudofilesystem) when it is running, doesn't when it is not
- There are also non-discretionary access controls (nDAC) like a hybrid of MAC and DAC (admin controls DAC for entire system)
- nDAC not covered here

Modes

- Multiple modes (can be applied on startup or on running system):
    - permissive - permission is granted, but denials are logged to /var/log/messages (for testing)
    - enforcing - strictly enforces 'targeted' policy rules (default)
    - disabled - only basic DACs are used

Rules for SELinux are held in the access vector cache (AVC)

Policies

SELinux strict policy forces one to assign object types to everything since everything is jailed explicitly

- Two classes on the system: Subjects (users, processes) and Objects (files).
- Subjects are grouped by roles and granted permissions to objects by policies.
- Type Enforcement (TE) ties subjects to objects, and allows the creation of domains (assigning subjects to group)
- Granularity is demonstrated by different object types: i.e., r: httpd_t would control access to some processes, httpd_log_t would control access to logs, httpd_etc_t would allow Apache to grant a processID in certain instances

Security labels (tuples) = context - user_u:object_r:file:_t ----user, role, type
Type applied to a subject (i.e., httpd, processes and programs) is called a domain, and type applied to an object (install.log) is simply called a type.

Various Management Items

- system-config-selinux will bring up the SELinux configuration GUI
- also found in System>Administration>SELinux Management (See also system-config-securitylevel)
- /usr/sbin/sestatus - lists status, mode , policy name (default is 'targeted,' 'strict' can also be used)
    - "-v" for verbose lists policies and whether it is disabled
    - displays items as user_u:system_r:processname_t (contexts: user, role. _t is the domain/type sandbox)
    - object_r:httpd_config_t is another one for object type which domain binds to

To see if SELINUX was indeed compiled, you can check /boot/config-2.6.9.EL file for lines with selinux. These will tell you also if it is disabled there.

policycoreutils package:
/usr/bin/secon
/usr/sbin/fixfiles
/usr/sbin/genhomedircon
/usr/sbin/load_policy
/usr/sbin/restorecon
/usr/sbin/restorecon_xattr
/usr/sbin/semodule
/usr/sbin/sestatus
/usr/sbin/setfiles

/usr/sbin/setsebool
/etc/selinux/ -----primary config directory
/etc/sysconfig/selinux/ ---- This is a link to /etc/selinux/config
/etc/sysconfig/selinux/config -----dictates default mode and policy
/etc/sysconfig/selinux/restorecond.conf ----used to restore contexts on objects
/etc/sysconfig/selinux/semanage.conf ---- for the semanage utility
/etc/sysconfig/selinux/targeted/ top level container for the policies 'targeted'
getenforce
setenforce 0 (permissive) or 1 (enforce) use to switch modes on the fly -
Also recommend is using "echo 1 > /selinux/enforce"

Which booleans are set? see etc/selinux/targeted/modules/active/booleans.local
Also see etc/selinux/targeted/booleans
Don't edit it! use libsemanage

You can use semanage to make changes to policy.
- File context (semanage fcontext) - you can also type semanage-fcontext
- Network port (semanage port - Use -l to list)
- Network interface (semanage interface)
- Booleans (semanage boolean)
- others...

Note that semanage only changes the policy: Use chcon or restorecon to actually label the filesystem
semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?" to change folder and it's contents
restorecon -R -v /web

For changes to survive a reboot, semanage, then restorecon (from RHSummit 2018 35:05)

Boot Process and Disabling SELinux
INIT selinux invocation
BIOS-->GRUB-->Kernel-->init
Init first checks to see if /proc/filesystems lists selinuxfs (wont start if not found), then checks if enforcing=0 or 1 in config file.
GRUB/LILO directives for enforcing/permissive/disabled override the config file if they differ!
Then, loads selinux after mounting, and init reloads itself into the unconfined_t domain (by default- on more secure systems could be tightened).

To disable SELinux after boot: nano /etc/grub.conf and add selinux=0 to the end of kernel line inside appropriate boot block
Test after reboot with "ps -efZ"

Labeling of objects to support type enforcement (TE)
ALL objects MUST be properly labeled!!! Any files that violate this will be unprotected!
When moving a file, it's labels are preserved.

2 ways to re-lable the filesystem:
touch / .autorelabel && reboot (create this hidden file at the root of the filesystem- gets init to do it on startup)
If you don't want to reboot, run /sbin/fixfiles, BUT if you do this, you may have to restart daemons that are reliant on files that are affected or are using the files.
/sbin/fixfiles -l fixfiles_logfile relabel -----run on filesystem and spit out log
/sbin/fixfiles -R package_name restore ---- this will pertain to contents of a given rpm package that needs labels restored
TCameron says this requires a reboot- this is not entirely accurate. He also warns "Don't do it in runlevel 5 since it deletes everything in /tmp including files the X server needs" Fixfiles prompts you as per whether you want to clobber the tmp directory, and that if you do so, a reboot will be required- that is what he was getting at- but you do have a choice.

Labeling ~/public_html files (in user directories)
After setting up DAC permissions
/usr/bin/chcon -R -t type file ---- -R is recursive
chcon -R -t httpd_user_content_t public_html/

Vugt: Using chcon is a really bad idea:
if selinux detects that at any time the filesystem has been unmonitored, it will determine it can't rely on the filesystem anymore and will apply default contexts to every single file on the filesystem (relabel)
From Red Hat: "changes made with the chcon command do not survive a file system relabel, or the execution of the restorecon command"

Restorecon
/sbin/restorecon -nv ../ --- n option looks for files that have been changed but doesn't make changes
(note- the video contradicts this at one point and says "display changes that would be applied")
Restores to factory default permissions(labels), i.e, this in /etc/selinux/targeted/policy/
-Rv for recursive
If a file is moved and restorecon is run on it, it will be given permissions of it's parent directory

SELinux Basic Commands using -Z
These require permissive or enforcing policy running to work with -Z
id - reveals current security context of user
ps - reveals the various sandboxes/domains of programs (subjects)
ls - reveals the context of files/directories (objects)
mv - keep current context when moving (don't inherent new parent directory's context)
cp - preserve current context when copying (don't inherent new parent directory's context)
mkdir
netstat (ss?)

Target Policy Context Elements:
User labels:
Non-privileged user = user_u
Privileged user = root_u
Role-based Access Control (RBAC)
Non-privileged and users = system_r
Type/domain labels:
12 default protected daemons: httpd, ntpd, dhcpd, mysqld, named, nscd, portmap, postgres, snmp, squid, winbind, syslogd
All others (unless customized) get unconfined_t domain

Disabling Specific (Targeted) Policies While Running
We need to edit the items in the /booleans directory and toggle the boolean:
echo "1 1" > /selinux/booleans/http_disable_trans
The file commit_pending_bools is monitored by selinux to see if it needs to refresh the policies
echo "1" > /selinux/commit_pending_bools
Restart the affected service:
/sbin/service httpd restart
Apache is now running in the unconfined_t domain

Source Policy
The directory /etc/selinux contains binary and text versions of the targeted policy
You have to operate on the source of the targeted policy to make customizations
Installing the source also helps to understand the targeted policy and make these changes
Custom policies would be needed if a new program is added to the system that is not protected by default (i.e., httpd, etc.)

Contents of /etc/selinux/targeted: booleans file, plus the policy and context directories
Policy directory contains compiled binaries of policies
The contexts directory contains exactly that, i.e. the file default_type cats out system_r:unconfined_t
Two subdirectories: users holds contexts for users (i.e. root) and files, which has media (drives) and file_contexts

The /etc/selinux/targeted/contexts/file_contexts file is key- it is what maps directories and files to labels.
Syntax for file_contexts content= regexp [-type] (context | <<none>>)
Examples (-d stands for directory) :
/home/[^/]+ -d system_u:object_r:user_home_dir_t

/home/[^/]+/.+ system_u:object_r:user_home_t
/mnt/[^/]*/.* <<none>>

A "--" means a file instead of a directory (-d), and there is also a -c type (not explained but was shown referring to /dev/agpgart (AGP device)

Installing Source Policy
rpm -qa | grep selinux
Brings us selinux-policy-targeted-1.17.30 ---these are just the binaries again- we need selinux-policy-targeted-source-1.17.30.rpm
rpm -Uvh selinux-policy-targeted-source-1.17.30.rpm

/etc/selinux/targeted/src/policy/domains/program/dhcp.te --- for example, is in source… te files= type enforcement, fc are file context files

Key Startup Utility for SELinux Protected Daemons
/usr/sbin/run_init ensures protected daemon isolation
When daemons are restarted (service http restart) they will no longer be protected by SElinux until you run the run_init program!
Run_init is what is run when init specifies to run selinux on startup and after the selinux filesystem has been located- it is what initiates setting up the sandboxes (proper contexts).
If a process is running outside of proper contexts, (ps -axZ), kill the parent process (the one with smallest number), then run "/usr/sbin/run_init /etc/init.d/httpd"

If you make ANY changes to selinux that are global (i.e. booleans or installing new selinux binaries), you will have to restart all processes, so it is acceptable to reboot the machine

Source Policy - Generating file_contexts File
/etc/selinux/targeted/contexts/file_contexts
Created from many input files:

/etc/selinux/targeted/src/policy --- source tree --- contains .fc (file context) and .te (type enforcement) files
/etc/selinux/targeted/src/policy/file_contexts --- has the source info for building the file_contexts file
/etc/selinux/targeted/src/policy/file_contexts/program/ contains fc files for programs

Example content: ping.fc
#ping
/usr/ping.* -- system_u:object_r:ping_exec_t
/usr/sbin/hping2 -- system_u:object_r:ping_exec_t

FC files should be checked for instances such as Apache domains- be sure files and folders match to where they live for complex programs so they match.
Only files with a corresponding tc file are enforced! Otherwise fc file will not be implemented!
If a file (object) does not have specific fc/te specified, it inherts that of the enclosing directory.
/etc/selinux/targeted/src/policy/file_contexts/domains/ contains te files for the programs corresponding fc files

Review:
TE files enforce type:
- describe what domains are able to do
- types that domains are able to access
- system-related calls (link, unlink, read, write, tcp_open, udp_open, etc.)
Also,
- enables actions on objects
- provides booleans (see boolean file- /etc/selinux/targeted/booleans)
- defines/supplies labels for the domains (types_t) and whether to allow or deny
Apache is likely to need to do the following:
- read files (config, content, and log files)
- bind to network ports (TCP:80, TCP:443)
- write to logs files
- execute scripts

Example entry in TE file:
Syntax is allow | neverallow subject object:object_class {permissions} (curly braces needed only with multiple permissions or objects)

```
allow httpd_suexec_t self:capability { setuid setgid };
 bool httpd_enable_ftp_server false;
 if (httpd_enable_ftp_server) {
 allow httpd_t ftp_port_t:tcp_socket name_bind;
 }
allow { httpd_t httpd_sys_script_t httpd_suexec_t }
# execute perl
allow httpd_t { bin_t sbin_t }:dir r_dir_perms;
can_exec(httpd_t, { bin_t sbin_t })
allow httpd_t bin_t:lnk_file read;
```

Star utility for backup (SELinux TAR)
- Tar does not archive security context labels
- Star can be a general replacement for tar
- Star is installed separately - star-1.5a25-6.i386.rpm

Usage:

```
star -xattr -H=exustar -c -f newarchive.star foldername/    ---extended attributes, create, f for normal
star -xattr -x -f newarchive.star  ---to extract
```

Optionally you can backup without SELinux context and let SELinux to reinstate context using fixfiles, or "touch /.autorelabel && reboot"

SELinux-relevant Items in Logs
- look for AVC related entries (access vector cache) in /var/log/messages
- will show if policies were loaded, if setenforce was modified, if files were relabeled

Sample denial in logs: (scontext and tcontext are source and target context)
avc: denied {action} for PID 3742 exe=/usr/sbin/ httpd path=/home/username/public_html
scontext=user_u:system_r:httpd_t tcontext="system_u:object_r:user_home_dir_t tclass=dir
Some errors like this can be generated by DACs, so that must be kept in mind.

- grep AVC /var/log/audit/audit.log

```
[root@localhost web]# grep AVC /var/log/audit/audit.log
type=AVC msg=audit(1425660974.563:416): avc:  denied  { getattr } for  pid=3726 comm="httpd" path="/web/index.html" dev="dm-1" ino=2
8668713 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file
type=AVC msg=audit(1425660974.563:417): avc:  denied  { getattr } for  pid=3726 comm="httpd" path="/web/index.html" dev="dm-1" ino=2
8668713 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file
type=AVC msg=audit(1425661712.592:436): avc:  denied  { getattr } for  pid=3724 comm="httpd" path="/web/index.html" dev="dm-1" ino=2
8668713 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file
type=AVC msg=audit(1425661712.592:437): avc:  denied  { getattr } for  pid=3724 comm="httpd" path="/web/index.html" dev="dm-1" ino=2
8668713 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file
type=AVC msg=audit(1425661968.584:446): avc:  denied  { getattr } for  pid=3725 comm="httpd" path="/web/index.html" dev="dm-1" ino=2
8668713 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file
type=AVC msg=audit(1425661968.584:447): avc:  denied  { read } for  pid=3725 comm="httpd" name="index.html" dev="dm-1" ino=28668713
scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file
type=AVC msg=audit(1425661968.584:447): avc:  denied  { open } for  pid=3725 comm="httpd" path="/web/index.html" dev="dm-1" ino=2866
8713 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file
```

Deciphering the last line above:
Type of message (AVC) followed by epoch timestamp, there has been an avc:denied on {operation}
Open was denied, for PID, comm is related commend (httpd), path related to the deny is given, "dev=" gives device for the path, followed by "ino" for inode. scontext is source context, tcontext is target context. Both of those contexts are SELinix labels format (*_u, *_r, *_t for user, role, and type), the source being the program here (httpd) and type being the item of attempted access; at the end of those is tclass=file (it's a file)
So httpd has access of system_u:system_r:httpd_t:s0, but the file is unconfined_u:object_r:default_t (doesn't match)
Usually the focus is on type, so you could either allow httpd to access items with default_t (not a good idea- defeats the purpose of limiting access to only web-accessible items) or make the default_t items you want to access compatible with the httpd_t source target

Enabling Auditing for More Verbose Logs
Modify /etc/grub.conf - add "audit=1" to kernel line

This method puts logs into /var/log/messages/firewall.log
Returns UID, GID, inodes
Don't use it if you don't need to- it has a bit of overhead on performance

SELinux Management Tools
Shell-based:
setools-1.5.1-5.i386.rpm
- seinfo
    /usr/bin/seinfo /etc/selinux/targeted/policy/policy.file
    Returns number of roles, types, booleans, etc etc in policy
- avcstat
    /usr/sbin/avcstat 3 --- 3 second interval on lookups (optional)
    Displays metrics of lookups, hits, misses, allocs, reclaims, free
- sesearch - source, destination, or class type, which rules go with what
    /usr/bin/sesearch -a -t http_user_content_t /etc/selinux/targeted/policy/policy.file --all rules, type, policy
    /usr/bin/sesearch -a -s http_t /etc/selinux/targeted/policy/policy.file --all rules, source, policy
    Searches the policy for rules that match type and returns matching rules
    Useful for when wanting to mimic a existing rule and get it's info fast

GUI:
- apol - analyze policy- is much like seinfo and sesearch with some extras
- seaudit (dropped in the latest version?? see below)
    seaudit -l /var/log/firewall.log -p /etc/selinux/targeted/policy/policy.file

From the setools v4 page (https://github.com/TresysTechnology/setools/wiki)
Version 4 is a Python rewrite and seems to be missing tools listed on it's page
Graphical tools
apol A Qt graphical analysis tool. Use it to perform various types of analyses.
Command-line tools
sediff Compare two policies to find differences.
sedta Perform domain transition analyses.
seinfo List policy components.
seinfoflow Perform information flow analyses.
sesearch Search rules (allow, type_transition, etc.)

Version 3 had these
apol - analyze an SELinux policy.
seaudit - analyze audit messages from SELinux.
seaudit-report - generate highly-customized audit log reports.
sechecker - command line tool for performing modular checks on an SELinux policy.
sediff - semantic policy difference tool for SELinux.
secmds - command-line tools to analyze and search SELinux policy.
    secmds includes these:
seinfo is a tool for looking at an SELinux policy and viewing various component elements and statistics.
sesearch is a tool to search for rules (such as allow, type_change, and range_transition) in an SELinux policy.
findcon is a tool for performing an SELinux file context search upon a filesystem, a file_contexts file, or a database generated by indexcon. This tool allows searches for files that match a particular user, type, path, and so forth. The search string can specify complete contexts, partial contexts, and shell globbing style wildcards
replcon is a tool for replacing file contexts. This tool uses the same searching paramaters as findcon, but will then replace the context or part of the context on the matched filesystem objects.
indexcon is a tool for creating a snapshot of security contexts for SELinux filesystem entities.

On seaudit: "Policy analysis with apol is many magnitudes more complex than audit log analysis with seaudit"
https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/SELinux_Guide/rhlcommon-section-0104.html

---

~]# semanage login -l

Login Name	SELinux User	MLS/MCS Range
_default__	unconfined_u	s0-s0:c0.c1023
root	unconfined_u	s0-s0:c0.c1023
system_u	system_u	s0-s0:c0.c1023

Output may differ slightly from system to system. The Login Name column lists Linux users, and the SELinux User column lists which SELinux user the Linux user is mapped to. For processes, the SELinux user limits which roles and levels are accessible. The last column, MLS/MCS Range, is the level used by Multi-Level Security (MLS) and Multi-Category Security (MCS).

level
The level is an attribute of MLS and MCS. An MLS range is a pair of levels, written as lowlevel-highlevel if the levels differ, or lowlevel if the levels are identical (s0-s0 is the same as s0). Each level is a sensitivity-category pair, with categories being optional. If there are categories, the level is written as sensitivity:category-set. If there are no categories, it is written as sensitivity.

If the category set is a contiguous series, it can be abbreviated. For example, c0.c3 is the same as c0,c1,c2,c3. The /etc/selinux/targeted/setrans.conf file maps levels (s0:c0) to human-readable form (ie. CompanyConfidential). Do not edit setrans.conf with a text editor: use semanage to make changes. Refer to the semanage(8) manual page for further information. In Red Hat Enterprise Linux, targeted policy enforces MCS, and in MCS, there is just one sensitivity, s0. MCS in Red Hat Enterprise Linux supports 1024 different categories: c0 through to c1023. s0-s0:c0.c1023 is sensitivity s0 and authorized for all categories.

MLS enforces the Bell-La Padula Mandatory Access Model, and is used in Labeled Security Protection Profile (LSPP) environments. To use MLS restrictions, install the selinux-policy-mls package, and configure MLS to be the default SELinux policy. The MLS policy shipped with Red Hat Enterprise Linux omits many program domains that were not part of the evaluated configuration, and therefore, MLS on a desktop workstation is unusable (no support for the X Window System); however, an MLS policy from the upstream SELinux Reference Policy can be built that includes all program domains. For more information on MLS configuration, refer to Section 5.12, “Multi-Level Security (MLS)”.

RHEL SELinux Guide: http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/
Fedora Project SELinux Documentation: http://fedoraproject.org/wiki/SELinux
fedora-selinux-list (mailing list): https://www.redhat.com/mailman/listinfo
Red Hat Training - Red Hat Enterprise SELinux Policy Administration: http://red.ht/aoRDyr
Dan Walsh's blog: http://danwalsh.livejournal.com/

```
[TM@localhost ~]$ rpm -ql policycoreutils
/etc/pam.d/newrole
/etc/pam.d/run_init
/etc/rc.d/init.d/restorecond
/etc/rc.d/init.d/sandbox
/etc/selinux/restorecond.conf
/etc/selinux/restorecond_user.conf
/etc/sestatus.conf
/etc/sysconfig/sandbox
/etc/xdg/autostart/restorecond.desktop
/sbin/fixfiles
/sbin/load_policy
/sbin/restorecon
/sbin/setfiles
/usr/bin/secon
/usr/bin/semodule_deps
/usr/bin/semodule_expand
/usr/bin/semodule_link
/usr/bin/semodule_package
/usr/sbin/genhomedircon
/usr/sbin/load_policy
```

/usr/sbin/open_init_pty
/usr/sbin/restorecond
/usr/sbin/run_init
/usr/sbin/semodule
/usr/sbin/sestatus
/usr/sbin/setsebool

From Vugt:
Module 4: Managing Network Services: 21: Managing HTTP Services
/sbin/httpd
rpm -qc http ---show conf files

/etc/sysconfig/httpd ---startup stuff

/etc/httpd/ contains
conf conf.d conf.modules.d logs modules run
/etc/httpd/conf/httpd.conf --main
/etc/httpd/conf/httpd.conf.rpmsave - saved so settings won't get lost during update
NOTE THE NEW DIR STRUCTURE HTTPD/CONF/HTTPD.CONF

/var/www/html for DocumentRoot - SELinux will change this.

systemctl start httpd
systemctl enable httpd

sealert -a /var/log/audit/audit.log
SEAlert will analyze the last message and return a simplified translation, offer a solution. Example:

```
Raw Audit Messages
type=AVC msg=audit(1425661968.584:446): avc:  denied  { getattr } for  pid=3725 comm="httpd" path="/web/index.html" dev="dm-1" ino=2
8668713 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file


type=SYSCALL msg=audit(1425661968.584:446): arch=x86_64 syscall=stat success=yes exit=0 a0=7fc8bd6803a0 a1=7fff4dd61400 a2=7fff4dd61
400 a3=7fc8b1925792 items=0 ppid=3722 pid=3725 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(
none) ses=4294967295 comm=httpd exe=/usr/sbin/httpd subj=system_u:system_r:httpd_t:s0 key=(null)

Hash: httpd,httpd_t,default_t,file,getattr

-------------------------------------------------------------------------------

SELinux is preventing /usr/sbin/httpd from read access on the file .

*****  Plugin catchall_labels (83.8 confidence) suggests   ******************

If you want to allow httpd to have read access on the  file
Then you need to change the label on $FIX_TARGET_PATH
Do
# semanage fcontext -a -t FILE_TYPE '$FIX_TARGET_PATH'
where FILE_TYPE is one of the following: NetworkManager_exec_t, NetworkManager_tmp_t, abrt_dump_oops_exec_t, abrt_etc_t, abrt_exec_t
, abrt_handle_event_exec_t, abrt_helper_exec_t, abrt_retrace_coredump_exec_t, abrt_retrace_spool_t, abrt_retrace_worker_exec_t, abrt
```

Note that this has a percentage of confidence attached to it- below this is a "solution with a low confidence percentage, which we look at to reveal a really dumb way to deal with things.
audit2allow is receiving the piped output of the log to create a policy module file to allow all of the denies listed there for httpd (as said, very stupid) I wanted to mention what the program is and was there, as a warning:

So, understanding that is an incredibly stupid thing to do, moving on it uses a program worth mentioning "semodule -i mypolicy.pp" If you had a module that was meaningful, this is how you'd load it.
In the directory with the *.pp file (same working directory we were in) there is mypolicy.te and here are it's contents:

```
module mypolicy 1.0
require {
        type httpd_t;
        type default_t;
        class file { read getattr open };}
#=================httpd_t=============
allow httpd_t default_t:file { read getattr open };
```

read getattr and open are permissions. This allows this access to every damn file on your filesystem with default_t through Apache

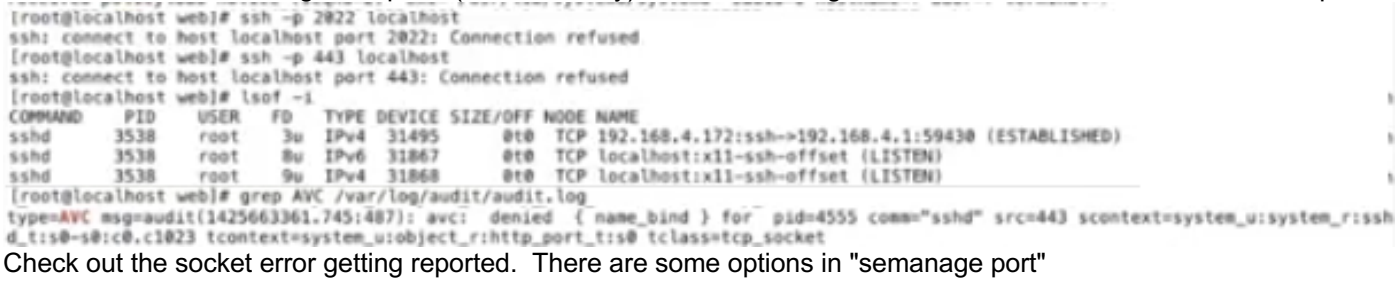
SELinux Modules is a mess!

Run "semodule -l" and you willl see a GIANT list of modules SELinux is supporting/are loaded

Guess who gets to clean this mess up with "semodule -r modulename" on each one? You do!

Good question: How much of a problem is this?

Here is another look at logs and ports (back to reality). SSH is refusing connections. Here is some trimmed output:



Check out the socket error getting reported. There are some options in "semanage port"
-a is add, if you want to modify so it is 443 its semanage -m -t sshd_t -p tcp 443

```
List all port defitions
# semanage port -l
Allow Apache to listen on tcp port 81
# semanage port -a -t http_port_t -p tcp 81
Allow sshd to listen on tcp port 8991
# semanage port -a -t ssh_port_t -p tcp 8991
```

Rather than take the advice of using blanket allow policies that are suggested (with module to allow all traffic of a particular type) if we look at /var/log/messages, we sometimes get more data
Here is an example telling use to find more info in the database: "run sealert -l 2asgfdagh9q-weqtr3y5"

```
Mar  6 12:40:22 localhost dbus[868]: [system] Successfully activated service 'org.fedoraproject.Setroubleshootd'
Mar  6 12:40:23 localhost setroubleshoot: Plugin Exception restorecon_source
Mar  6 12:40:23 localhost setroubleshoot: SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket . For complet
e SELinux messages. run sealert -l 08dc1d25-8b9e-4a0f-ad9e-44120b8fe9ac
Mar  6 12:40:23 localhost python: SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket .

*****  Plugin catchall (100. confidence) suggests   **************************

If you believe that sshd should be allowed name_bind access on the  tcp_socket by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep sshd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Mar  6 12:40:23 localhost setroubleshoot: SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket . For complet
e SELinux messages. run sealert -l 08dc1d25-8b9e-4a0f-ad9e-44120b8fe9ac
```

So we go to the link, and it is going to give us some real solution, right? This time it doesn't.

```
SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket .

*****  Plugin catchall (100. confidence) suggests   **************************

If you believe that sshd should be allowed name_bind access on the  tcp_socket by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep sshd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Additional Information:
Source Context                system_u:system_r:sshd_t:s0-s0:c0.c1023
Target Context                system_u:object_r:http_port_t:s0
Target Objects                [ tcp_socket ]
Source                        sshd
Source Path                   /usr/sbin/sshd
Port                          443
Host                          localhost.localdomain
Source RPM Packages           openssh-server-6.4p1-8.el7.x86_64
Target RPM Packages
Policy RPM                    selinux-policy-3.12.1-153.el7.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
Host Name                     localhost.localdomain
Platform                      Linux localhost.localdomain 3.10.0-123.el7.x86_64
                              #1 SMP Mon Jun 30 12:09:22 UTC 2014 x86_64 x86_64
Alert Count                   11
First Seen                    2015-03-06 12:35:19 EST
Last Seen                     2015-03-06 12:40:21 EST
Local ID                      08dc1d25-8b9e-4a0f-ad9e-44120b8fe9ac

Raw Audit Messages
type=AVC msg=audit(1425663621.647:542): avc:  denied  { name_bind } for  pid=4819 comm="sshd" src=443 scontext=system_u:system_r:ssh
d_t:s0-s0:c0.c1023 tcontext=system_u:object_r:http_port_t:s0 tclass=tcp_socket


type=SYSCALL msg=audit(1425663621.647:542): arch=x86_64 syscall=bind success=no exit=EACCES a0=3 a1=7fff7394905d0 a2=1c a3=7fff2c7ef7
64 items=0 ppid=1 pid=4819 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm=ss
hd exe=/usr/sbin/sshd subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 key=(null)

Hash: sshd,sshd_t,http_port_t,tcp_socket,name_bind

[root@localhost web]#
```

Here is the same stupid message waiting for some dumb sysadmin to go do what it says: make a policy module that allows a bunch of junk permissions that shouldn't be there. AND it's 100% confident this is the right answer, after all! This shows the danger this can cause!

Setting up a VSFTP server to allow anon uploads. Edit the conf. grepping passwd is a quick way to find the directory location to tweak (since it shows the user and home directory:

```
[root@localhost ~]# vim /etc/vsftpd/vsftpd.conf
[root@localhost ~]# grep ftp /etc/passwd
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
[root@localhost ~]# cd /var/ftp
[root@localhost ftp]# ls -ldZ .
drwxr-xr-x. root root system_u:object_r:public_content_t:s0 .
[root@localhost ftp]# semanage fcontext -a -t public_content_rw_t "/var/ftp(/.*)?"
libsemanage.dbase_llist_query: could not query record value (No such file or directory).
[root@localhost ftp]# cd /
[root@localhost /]# semanage fcontext -a -t public_content_rw_t "/var/ftp(/.*)?"
[root@localhost /]# restorecon -Rv /var/ftp
restorecon reset /var/ftp context system_u:object_r:public_content_t:s0->system_u:object_r:public_content_rw_t:s0
restorecon reset /var/ftp/pub context system_u:object_r:public_content_t:s0->system_u:object_r:public_content_rw_t:s0
```

If it is about file context, auditlog in very informative, but if it isn't about file context, but booleans, then var/log/messages is where we will need to look

```
[root@localhost /]# lftp localhost
lftp localhost:~> ls
drwxrwxrwx    2 0        0               6 Jun 10  2014 pub
lftp localhost:/> cd pub
lftp localhost:/pub> put /etc/hosts
put: Access failed: 553 Could not create file. (hosts)
lftp localhost:/pub> exit
[root@localhost /]# getenforce
Enforcing
[root@localhost /]# setenforce Permissive
[root@localhost /]# lftp localhost
lftp localhost:~> ls
drwxrwxrwx    2 0        0               6 Jun 10  2014 pub
lftp localhost:/> cd pub
lftp localhost:/pub> put /etc/hosts
158 bytes transferred
lftp localhost:/pub> exit
[root@localhost /]# grep AVC /var/log/audit/audit.log
type=AVC msg=audit(1425664094.116:558): avc:  denied  { write } for  pid=5031 comm="vsftpd" name="pub" dev="dm-1" ino=20671706 scont
ext=system_u:system_r:ftpd_t:s0-s0:c0.c1023 tcontext=system_u:object_r:public_content_rw_t:s0 tclass=dir
type=AVC msg=audit(1425664132.620:560): avc:  denied  { write } for  pid=5054 comm="vsftpd" name="pub" dev="dm-1" ino=20671706 scont
ext=system_u:system_r:ftpd_t:s0-s0:c0.c1023 tcontext=system_u:object_r:public_content_rw_t:s0 tclass=dir
type=AVC msg=audit(1425664132.620:560): avc:  denied  { add_name } for  pid=5054 comm="vsftpd" name="hosts" scontext=system_u:system
_r:ftpd_t:s0-s0:c0.c1023 tcontext=system_u:object_r:public_content_rw_t:s0 tclass=dir
type=AVC msg=audit(1425664132.620:560): avc:  denied  { create } for  pid=5054 comm="vsftpd" name="hosts" scontext=system_u:system_r
:ftpd_t:s0-s0:c0.c1023 tcontext=system_u:object_r:public_content_rw_t:s0 tclass=file
[root@localhost /]# cat /var/log/messages
*****  Plugin catchall (6.38 confidence) suggests   **************************

If you believe that vsftpd should be allowed write access on the  directory by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep vsftpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Mar  6 12:48:53 localhost setroubleshoot: SELinux is preventing /usr/sbin/vsftpd from write access on the d
ELinux messages. run sealert -l c3b4b1d5-9060-42a8-87a1-dc5bc670da57
Mar  6 12:48:53 localhost python: SELinux is preventing /usr/sbin/vsftpd from write access on the directory

*****  Plugin catchall_boolean (47.5 confidence) suggests   ******************

If you want to allow ftpd to anon write
Then you must tell SELinux about this by enabling the 'ftpd_anon_write' boolean.
You can read 'None' man page for more details.
Do
setsebool -P ftpd_anon_write 1

*****  Plugin catchall_boolean (47.5 confidence) suggests   ******************

If you want to allow ftpd to full access
Then you must tell SELinux about this by enabling the 'ftpd_full_access' boolean.
You can read 'None' man page for more details.
Do
setsebool -P ftpd_full_access 1

*****  Plugin catchall (6.38 confidence) suggests   **************************

If you believe that vsftpd should be allowed write access on the  directory by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep vsftpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Mar  6 12:49:03 localhost dbus-daemon: string index out of range
Mar  6 12:49:03 localhost dbus-daemon: 'list' object has no attribute 'split'
Mar  6 12:50:01 localhost systemd: Starting Session 10 of user root.
Mar  6 12:50:01 localhost systemd: Started Session 10 of user root.
```

One of the booleans looks likely- lets check them out. Anon write is off:

```
[root@localhost /]# getsebool -a | grep ftp
ftp_home_dir --> off
ftpd_anon_write --> off
ftpd_connect_all_unreserved --> off
ftpd_connect_db --> off
ftpd_full_access --> off
ftpd_use_cifs --> off
ftpd_use_fusefs --> off
ftpd_use_nfs --> off
ftpd_use_passive_mode --> off
httpd_can_connect_ftp --> off
httpd_enable_ftp_server --> off
sftpd_anon_write --> off
sftpd_enable_homedirs --> off
sftpd_full_access --> off
sftpd_write_ssh_home --> off
tftp_anon_write --> off
tftp_home_dir --> off
[root@localhost /]#
```

Fix with setbool -P ftpd_anon_write on   (-P for persistent)
/etc/sysconfig/selinux - to set defaults- change to enforcing and targeted
set MLS (multi Level Security) here if needed

----added stuff from Thomas Cameron
Policies can be added to determine access between users, files, directories, sockets, tcp/udp ports - **but also memory areas!**

Policies:
Targeted: default
        only targeted policies are protected by SELinux
        Everything else is unconnfined
MLS - Multi-Level Multi-Category Security
        Out of scope for this talk
        can get very complex, typically used only in TLA (3 letter acronym) government agencies

/usr/sbin/sestatus, getenforce

```
[root@w541 ~]# /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      30
```

/etc/selinux/config (symlink in /etc/sysconfig/selinux)

```
[root@w541 ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are pro
tected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted


[root@w541 ~]# ls -l /etc/sysconfig/selinux
lrwxrwxrwx. 1 root root 17 Oct 29  2015 /etc/sysconfig/selinux -> ../selinux/con
fig
```

Files and directories, labels are stored as extended attributes on the filesystem (the fs needs xattr)
For processes, ports etc, the kernel manages these labels
"ext2, ext3, ext4, JFS, Squashfs, Yaffs2, ReiserFS, XFS, Btrfs, OrangeFS, Lustre, OCFS2 1.6 and F2FS[8] filesystems support extended attributes (abbreviated xattr) when enabled in the kernel configuration
https://en.wikipedia.org/wiki/Extended_file_attributes

Labels - SELinux user- not GEKOS user
user:role:type:level(optional)
Items other than type are usually not covered, since we use labeling and type enforcement
Those are definitely used a lot in MLS/MCS

Apache types
httpd_sys_content_t, httpd_config_t, httpd_log_t
netstat -tulpZ show labels -- try with ss utility works??

```
[root@w541 ~]# semanage port -l | grep [h]ttp
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
[root@w541 ~]#
```

Contexts are usually set on files when created, based on parent directory (with a few exceptions)
RPMs can set policies as part of the install process (in RPM spec file)
The login process sets the default context (unconfined in the targeted policy)

"File transitions" defined by policy - aren't really transitions at all just govern the type given to file being created.
If application foo_t creates a file in a directory labeled bar_t, policy can require a transition so that the file is created with the baz_t label (so that it doesn't inherit from creator or directory containing it.

dontaudit option

Where things go wrong:
Mislabling
Boolean or policy module issue
Could be a bug in policy
Could even be a security violation/ indicator

getsebool -a -----list booleans

```
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
auditadm_exec_content --> on
authlogin_nsswitch_use_ldap --> off
authlogin_radius --> off
authlogin_yubikey --> off
awstats_purge_apache_log_files --> off
boinc_execmem --> on
cdrecord_read_content --> off
cluster_can_network_connect --> off
cluster_manage_all_files --> off
cluster_use_execmem --> off
cobbler_anon_write --> off
cobbler_can_network_connect --> off
cobbler_use_cifs --> off
cobbler_use_nfs --> off
collectd_tcp_network_connect --> off
condor_tcp_network_connect --> off
conman_can_network --> off
cron_can_relabel --> off
:
```

yum -y install setroubleshoot setroubleshoot-server
- for policy module development machines
- Also tools to diagnose and fix probs (makes some things human readable)
- Reboot or restart auditd process after install
  - systemctl restart auditd.service will give an error!
  - service auditd restar WORKS
    - This is NOT a bug!
    - https://bugzilla.redhat.com/show_bug.cgi?id=1026648

Example-
User wants to homepage in /home/fred/public_html
- enable UserDir in /etc/httpd/conf/httpd.conf, restart Apache
- chmod o+x /home/fred/public_html
- Doesn't work
- SOP is check /var/log/access_log and error_log which just say 403 error
- /var/log/messages - get sealert which actually is good instruction:
- setsebool -P httpd_read_user_content 1 (and httpd_enable_homedirs)

```
Jun 26 23:54:34 w541.tc.redhat.com setroubleshoot[4437]: failed to retrieve rpm info for /home/fred/public_html/index.html
Jun 26 23:54:34 w541.tc.redhat.com setroubleshoot[4437]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /home/fred/
s. run sealert -l c36627c9-7b99-44c9-a78c-a9a737ff119b
Jun 26 23:54:34 w541.tc.redhat.com python3[4437]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /home/fred/public_

                                                  ***** Plugin catchall_boolean (24.7 confidence) suggests ******************

                                                  If you want to allow httpd to read home directories
                                                  Then you must tell SELinux about this by enabling the 'httpd_enable_homedirs' boolean.
                                                  You can read 'None' man page for more details.
                                                  Do
                                                  setsebool -P httpd_enable_homedirs 1

                                                  ***** Plugin catchall_boolean (24.7 confidence) suggests ******************

                                                  If you want to allow httpd to read user content
                                                  Then you must tell SELinux about this by enabling the 'httpd_read_user_content' boolean.
                                                  You can read 'None' man page for more details.
                                                  Do
                                                  setsebool -P httpd_read_user_content 1
```

getsebool -a
Temp is /etc/selinux/targeted/modules/active/booleans.local
It says don't edit it - but it doesn't do anything if you edit it:
When you run setsebool -P the entire /etc/selinux/targeted directory is deleted and rebuilt!

restorecon --reference /var/www/html /mynew/webpage/
Gives slaps the target label from the first onto the second item
semanage fcontext -e /var/www/ /foo

- equals (-e) does the same thing
restorecon -vR ---------- uses /etc/selinux/targeted/contexts/
- Remember to use restorecon when you make changes

Note: journalctl -b -0 for msgs on the current boot -1, -2 for previous boots

semanage to set the policy, restorecon to relabel the filesystem as such

In the case that a boolean or labeling does not fix the issue, you may have to make a policy module
Gets to audit.log -
Switches to /var/log/messages/

Set SELinux as permissive, setenforce = 0, the run the troubled application through all of the paces
(in this case it is SquirrelMail, and paces means sending and receiving messages) This way you log all of the denials as it goes through the motions, then be able to go through the logs to catch when there is a problem and address each one

e.g. - first example is "preventing /usr/sbin/httpd from name_connect access on the tcp_socket"
grep httpd /var/log/audit/audit.log | audit2allow -M squirrellocal (example says mypol)
semodule -i squirrellocal.pp
-- here it is explained that he adds local to differentiate from the normal policy module names; at xyz.com maybe name it squirrelxyz

```
[root@armitage ~]# sealert -l f64ca3e4-4fe2-4998-85eb-de402ba79db2
Gtk-Message: Failed to load module "pk-gtk-module": libpk-gtk-module.so: cannot
open shared object file: No such file or directory
SELinux is preventing /usr/sbin/httpd from name_connect access on the tcp_socket
 .

*****  Plugin catchall (100. confidence) suggests  **************************

If you believe that httpd should be allowed name_connect access on the  tcp_sock
et by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp
```

Admittedly, this could have been done with a boolean. Here is the content of the policy module allowing the server to connect over a tcp socket to the smtp port.

```
[root@armitage ~]# cat squirrellocal.te

module squirrellocal 1.0;

require {
        type httpd_t;
        type smtp_port_t;
        type pop_port_t;
        class tcp_socket name_connect;
}

#============= httpd_t ==============
#!!!! This avc can be allowed using one of the these booleans:
#     httpd_can_sendmail, allow_ypbind, httpd_can_network_connect

allow httpd_t pop_port_t:tcp_socket name_connect;
#!!!! This avc can be allowed using one of the these booleans:
#     httpd_can_sendmail, allow_ypbind, httpd_can_network_connect

allow httpd_t smtp_port_t:tcp_socket name_connect;
[root@armitage ~]#
```

Enabling SELinux:
edit /etc/selinux/config and set SELINUX=permissive
T Cameron says: "... if it has been turned off in the past. Do not set it to enforcing as it will more than likely hang at boot time" First: set permissive, then
touch /.autorelabel
OR run "fixfiles relabel" BUT don't run it in runlevel5 - it deletes /tmp contents and X fonts stuff
AND you'd still have to reboot anyway.
After reboot, setenforce 1 (or change permissive to enforcing)

For gui install policycoreutils-gui to "ssh -X" in and run system-config-selinux

The /selinux pseudofilesystem
(like /proc) mentioned earlier:
The following example shows sample contents of the /selinux/ directory:
-rw-rw-rw- 1 root root 0 Sep 22 13:14 access
dr-xr-xr-x 1 root root 0 Sep 22 13:14 booleans
--w------- 1 root root 0 Sep 22 13:14 commit_pending_bools
-rw-rw-rw- 1 root root 0 Sep 22 13:14 context
-rw-rw-rw- 1 root root 0 Sep 22 13:14 create
--w------- 1 root root 0 Sep 22 13:14 disable
-rw-r--r-- 1 root root 0 Sep 22 13:14 enforce
-rw------- 1 root root 0 Sep 22 13:14 load
-r--r--r-- 1 root root 0 Sep 22 13:14 mls
-r--r--r-- 1 root root 0 Sep 22 13:14 policyvers
-rw-rw-rw- 1 root root 0 Sep 22 13:14 relabel
-rw-rw-rw- 1 root root 0 Sep 22 13:14 user

https://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-selinux.html

VUGT RHCSA RECAP
syscalls are filtered through selinux, through policy to see if allowed.
If not allowed, avc:denied, goes through auditd, which writes event to /var/log/audit/audit.log
 (config file is in /etc/audit/auditd.conf)
If SELinux is in enforcing, action stopped, permissive allowed, but logged.

web browser asks Apache for a doc in /foo, a getattr /foo/index.html syscall is issued. If it has the wrong label, selinux enforcing stops there. You can also see the getattr referred to in the AVC alerts in audit.log

If disabled, you must restart before setting enforcing in /ect/sysconfig/selinux will take effect (same with switching from enforcing to disabled)
Switching between enforcing and permissive does not have that limitation
getenforce/ setenforce
Context is user_u:objectRole_r:type_t - since we mostly concern with types, it is "type enforcement"

getsebool -a | grep httpd / setsebool for booleans
setsebool not persistent without -P - **setsebool -P ftp_home_dir on**

ps auxZ

Moving (mv) a file maintains the previous context, while copying (cp) a file creates a new file so it inherits the parent dir

semanage fcontext, ports, boolean, and more
Setting context for all files in /web - **semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"**
This doesn't write to the filesystem, but the policy so you have to write it to the filesystem with **restorecon -R -v /web**

semanage fcontext -l | grep httpd

Many SELinux man pages not installed by default
yum provides */sepolicy tells us what to get
yum -y install policycoreutils-devel
Running **sepolicy manpage -a** generates the manpages for all of our services (throws them into /tmp)
Put in the proper place- **mv /tmp/*.8 /usr/share/man/man8** Running **mandb** will index them
man -k _selinux | grep httpd

Better way would have been to run this:
sepolicy manpage -a -p /usr/share/man/man8

Don't use chcon! use semanage!
chcon problems: writes to the inode, not the policy
chcon -R --type=httpd_sys_content_t /foo
So if you do a .autorelable restart or restorecon, it will zap your changes

setroubleshoot
systemctl status auditd
grep AVC /var/log/audit/audit.log - shows the error
It is syslog's /var/log/messages to show the sealert items with database code to lookup

Audit log will show AVC message with source context (like httpd_t) and target context (like default_t)

If you are dealing with common items, check semanage man page to see name of the proper other man page like semanage-fcontext, look up that man page, and the example for http_t is right there!