IT2654

SYSTEMS ADMINISTRATION & SECURITY

TOPIC 5: LINUX FIREWALL

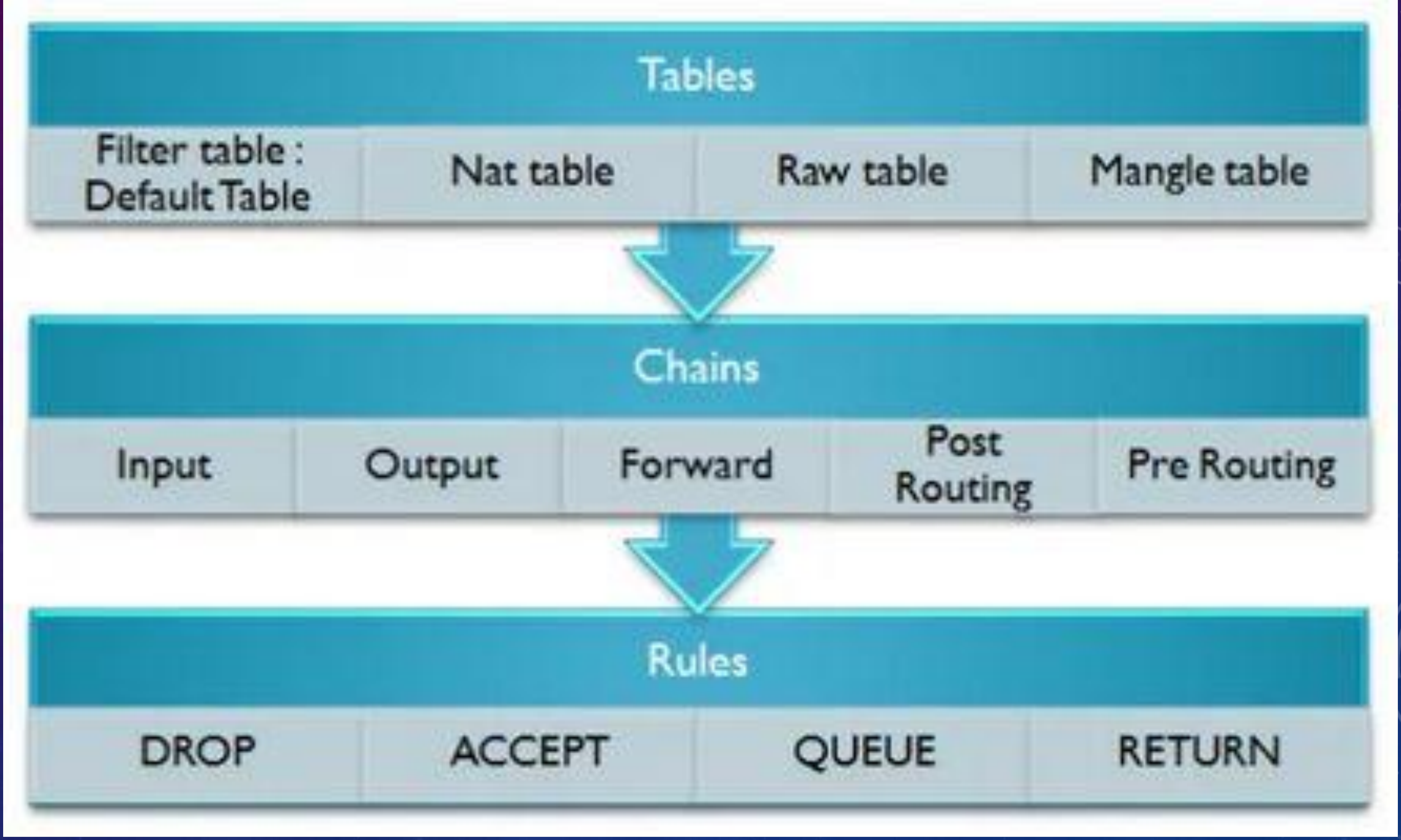
OBJECTIVES

- ☻ Upon completion of this unit, you should be able to:
  - ❖ Describe the Netfilter architecture
  - ❖ Understand the GUI and CLI tools in administering firewall rules
  - ❖ Configure packet filtering rules using CLI tool

BASIC CONCEPTS

- ☻ Netfilter – packet filtering software that is part of the Linux kernel (firewall)
- ☻ iptables - command line (CLI) tool to set up rules for the firewall (netfilter)
- ☻ Table – container to organize rules via chains
- ☻ Chain – a sequence of rules in a table to inspect the packet
- ☻ Target – a rule is matched to a target. A target decides the fate of a packet, such as allowing or rejecting it.
- ☻ By default, firewall rules are saved in the /etc/sysconfig/iptables or /etc/sysconfig/ip6tables files

FIREWALL STRUCTURE



Tables

Filter table : Default Table | Nat table | Raw table | Mangle table

Chains

Input | Output | Forward | Post Routing | Pre Routing

Rules

DROP | ACCEPT | QUEUE | RETURN

WAYS TO CONFIGURE FIREWALL

1) system-config-firewall
2) system-config-firewall-tui
3) iptable
4) Edit the /etc/sysconfig/iptables file

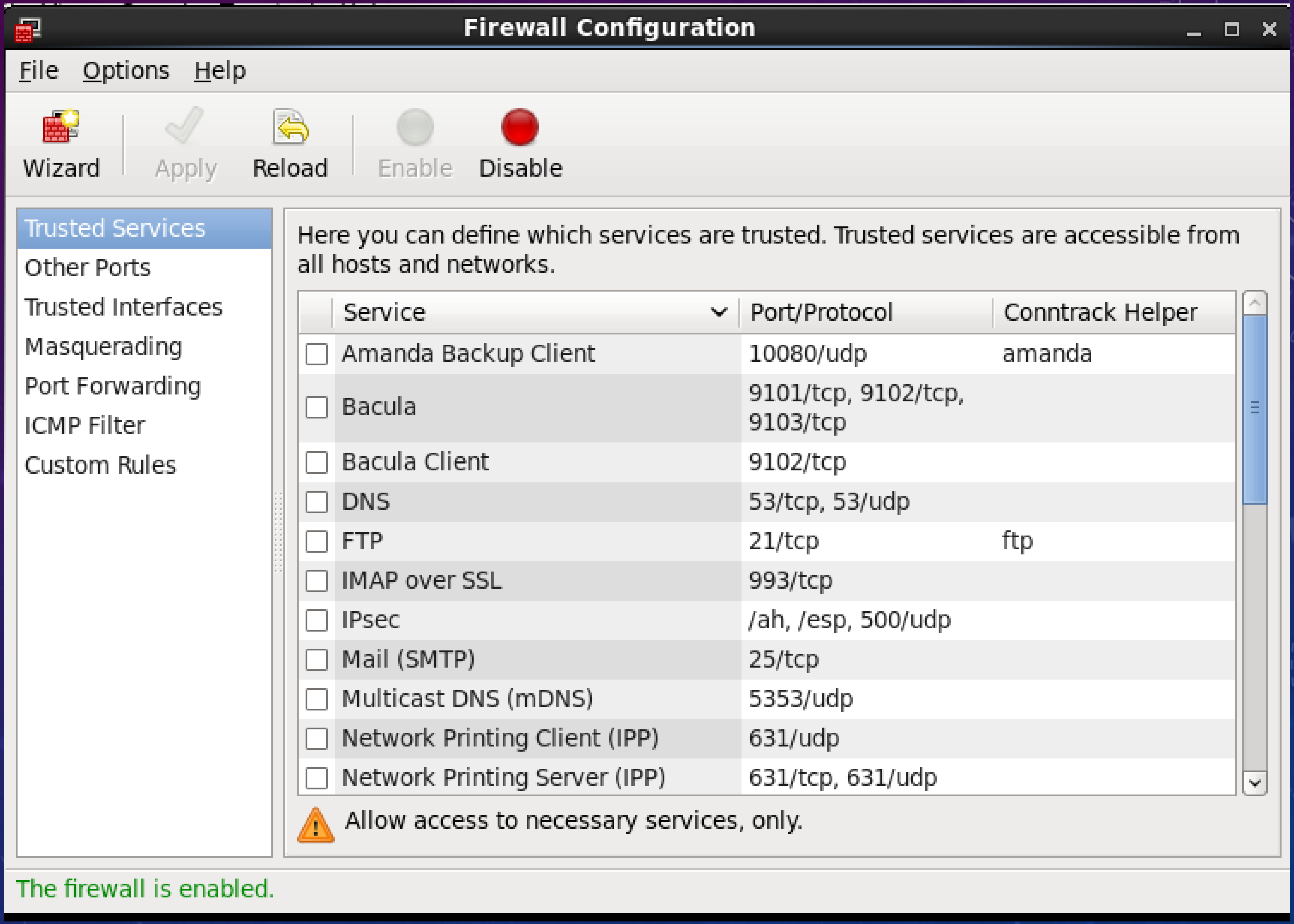
FIREWALL: SYSTEM-CONFIG-FIREWALL

- ☻ Provides a very basic GUI firewall interface
- ☻ Hides much of the complexity, and functionality, of Netfilter
- ☻ System->Administration->Firewall

If all you need to do is allow traffic into specific ports for all hosts and deny traffic from anyone else, you can avoid the complexities of the Netfilter command-line interface by using system-config-firewall instead.

The first setting one sees on the Firewall Options tab controls whether the firewall is active at all or not. Below that, one may specify ports to allow traffic on. Any enabled port will accept traffic from any hosts. Commonly-used ports are represented by the names of the associated services with checkboxes that can be used to enable or disable them.

SYSTEM-CONFIG-FIREWALL



Here you can define which services are trusted. Trusted services are accessible from all hosts and networks.

	Service	Port/Protocol	Conntrack Helper
□	Amanda Backup Client	10080/udp	amanda
□	Bacula	9101/tcp, 9102/tcp, 9103/tcp	
□	Bacula Client	9102/tcp	
□	DNS	53/tcp, 53/udp	
□	FTP	21/tcp	ftp
□	IMAP over SSL	993/tcp	
□	IPsec	/ah, /esp, 500/udp	
□	Mail (SMTP)	25/tcp	
□	Multicast DNS (mDNS)	5353/udp	
□	Network Printing Client (IPP)	631/udp	
□	Network Printing Server (IPP)	631/tcp, 631/udp	

⚠ Allow access to necessary services, only.

The firewall is enabled.

SYSTEM-CONFIG-FIREWALL

When the firewall is enabled via GUI, the following rules are put into place:

① Packets destined for the loopback device(lo) are accepted

② Packets destined for an enabled port are accepted from any source

③ Packets that are responses to the packets you have sent, such as the response to an http request, are accepted

④ ICMP packets are accepted

⑤ Packets destined for a few utility services, such as the CUPS print server, are accepted.

⑥ All other packets are rejected with an ICMP “Host Prohibited” message.

SYSTEM-CONFIG-FIREWALL-TUI

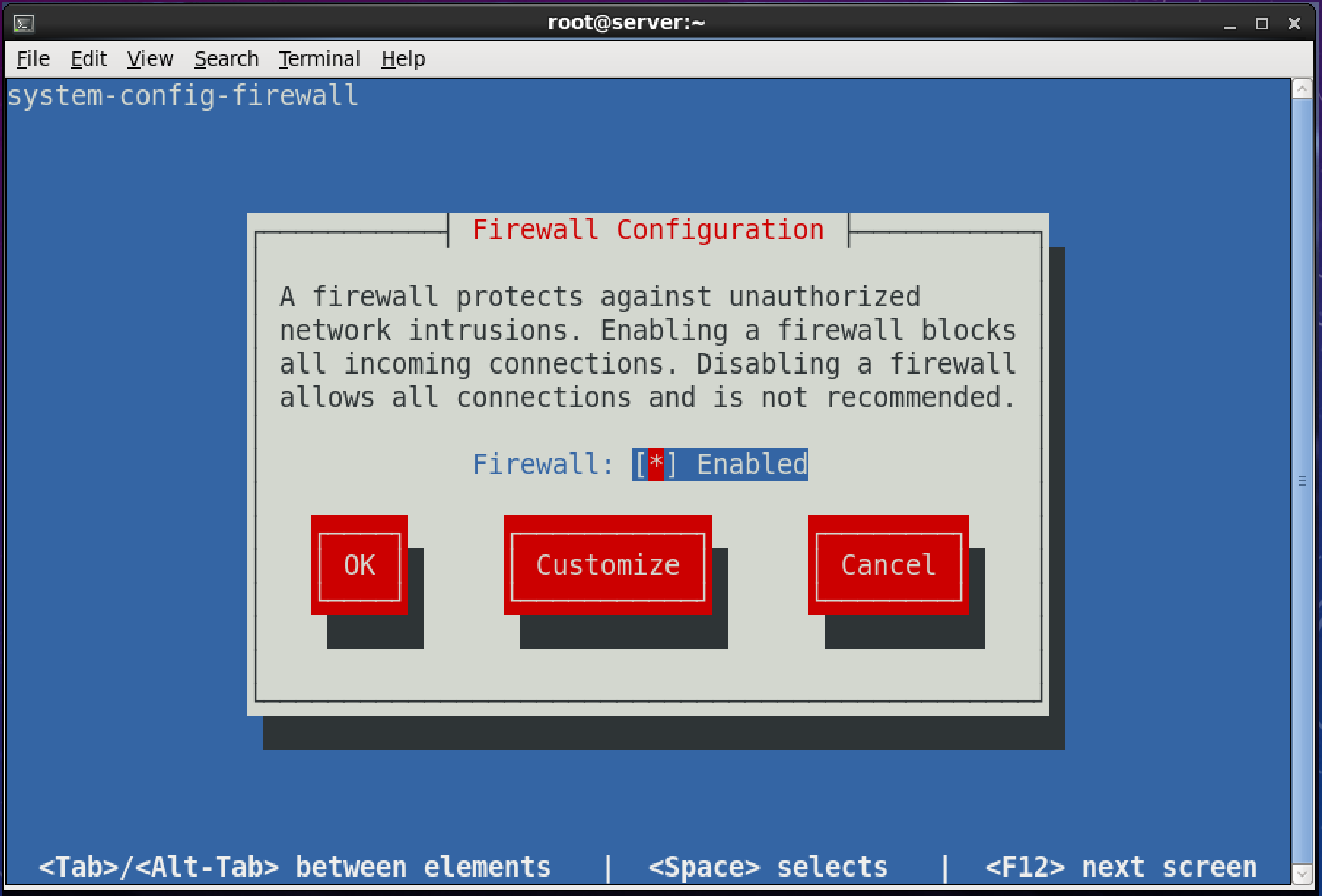
- This configuration may be sufficient in many cases, but Netfilter supports much more finely-tuned configurations, such as accepting or rejecting packets based on the source host or network, which system-config-firewall does not allow for.

- If system-config-firewall is executed in a non-graphical environment, or if system-config-firewall-tui is executed instead, a simpler, curses-based interface is presented. This interface can be navigated using the Tab key to move between fields and the spacebar toggle checkboxes and “click” buttons.

- The TUI interface even has a feature not present in the GUI interface the ability to designate “trusted” interfaces. If an interface is listed as trusted, all traffic received on that interface is allowed through the firewall.

- Use this feature with caution! Just because a network is internal doesn’t mean that hostile traffic can’t come from it. Sometimes, your internal hosts may be used to attacked your server once the hosts have been hijacked by hackers.

SYSTEM-CONFIG-FIREWALL-TUI

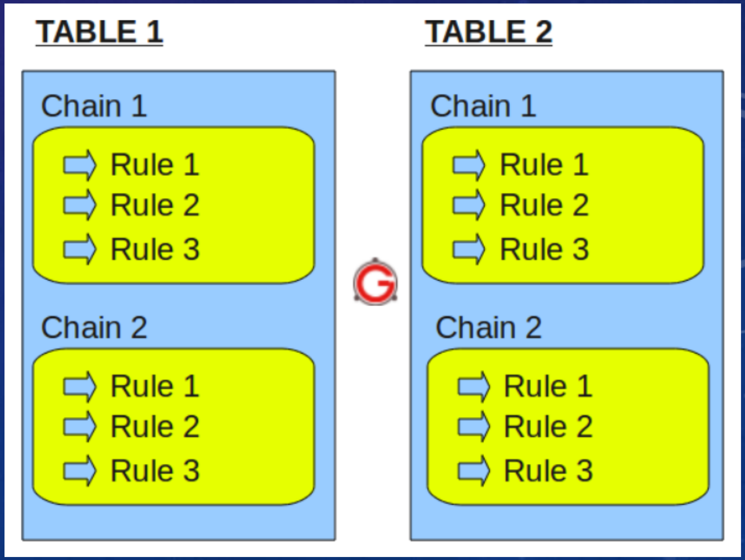


```
system-config-firewall

Firewall Configuration

A firewall protects against unauthorized
network intrusions. Enabling a firewall blocks
all incoming connections. Disabling a firewall
allows all connections and is not recommended.

Firewall: [*] Enabled
```

```
<Tab>/<Alt-Tab> between elements   |  <Space> selects   |  <F12> next screen
```

IPTABLES BASIC

- iptables firewall is used to manage packet filtering and NAT rules in Linux. IPTables comes with all Linux distributions.
- iptables tool is used to manage the Linux firewall rules.
- iptables contain multiple tables. Tables can contain multiple chains. Chains can be built-in or user-defined. Chains might contain multiple rules. Rules are defined for the packets.
- The structure is: iptables -> Tables -> Chains -> Rules. This is defined in the following diagram.

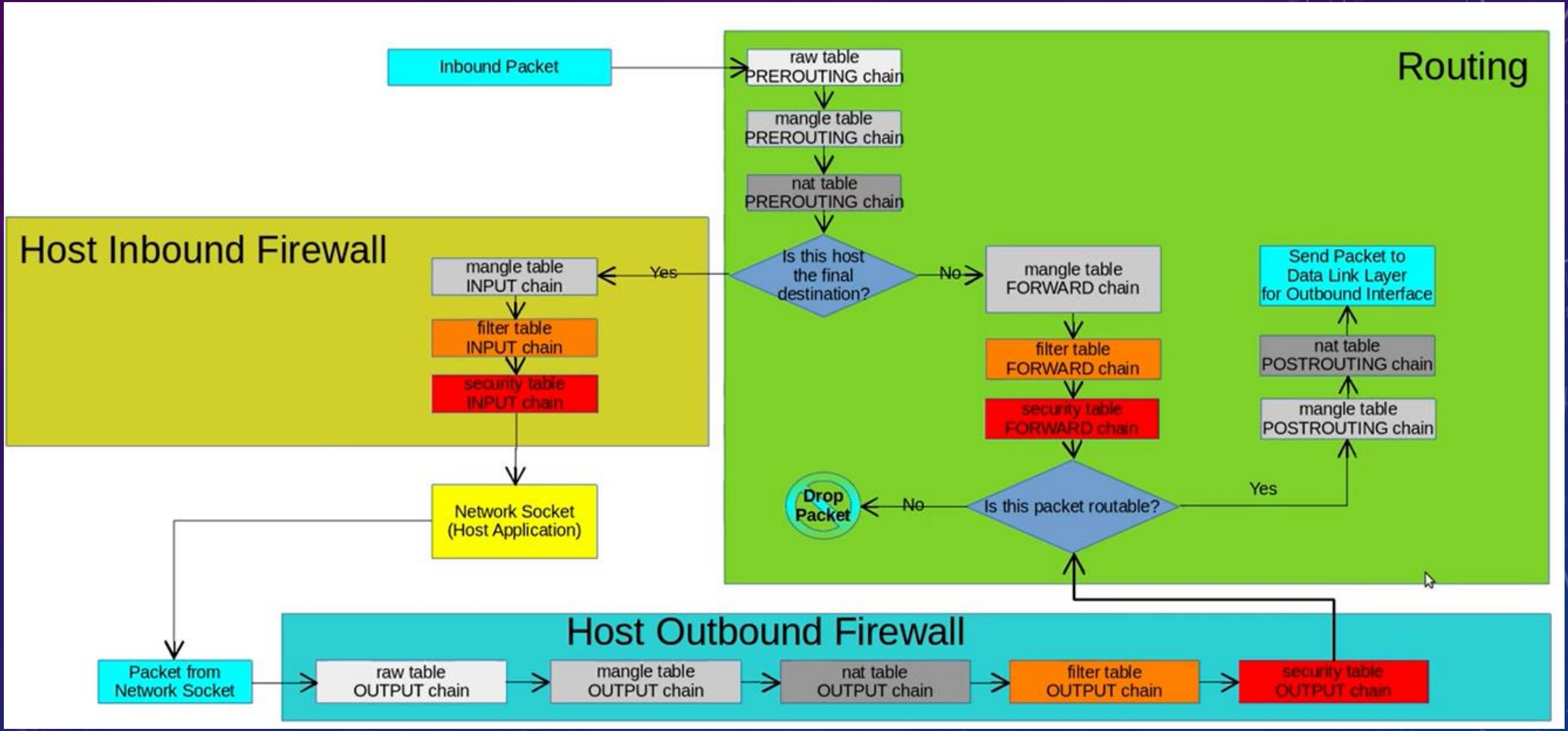


NETFILTER TABLES AND CHAINS

- There are five tables

Table	Role
filter	The default table for handling network packets
nat	Used to alter packets that create a new connection and used for NAT (Network Address Translation)
mangle	Used for specific types of packet alteration. This table is rarely used
raw	Used mainly for configuring exemptions from connection tracking in combination with the NOTRACK target
security	Used for Mandatory Access Control (MAC) networking rules, such as those enabled by the SECMARK and CONNSECMARK targets.

NETFILTER PACKET FLOW



Inbound Packet

Routing

raw table PREROUTING chain

mangle table PREROUTING chain

nat table PREROUTING chain

Is this host the final destination?

Yes

No

Host Inbound Firewall

mangle table INPUT chain

filter table INPUT chain

security table INPUT chain

Network Socket (Host Application)

mangle table FORWARD chain

filter table FORWARD chain

security table FORWARD chain

Is this packet routable?

No

Drop Packet

Yes

mangle table POSTROUTING chain

nat table POSTROUTING chain

Send Packet to Data Link Layer for Outbound Interface

Host Outbound Firewall

Packet from Network Socket

raw table OUTPUT chain

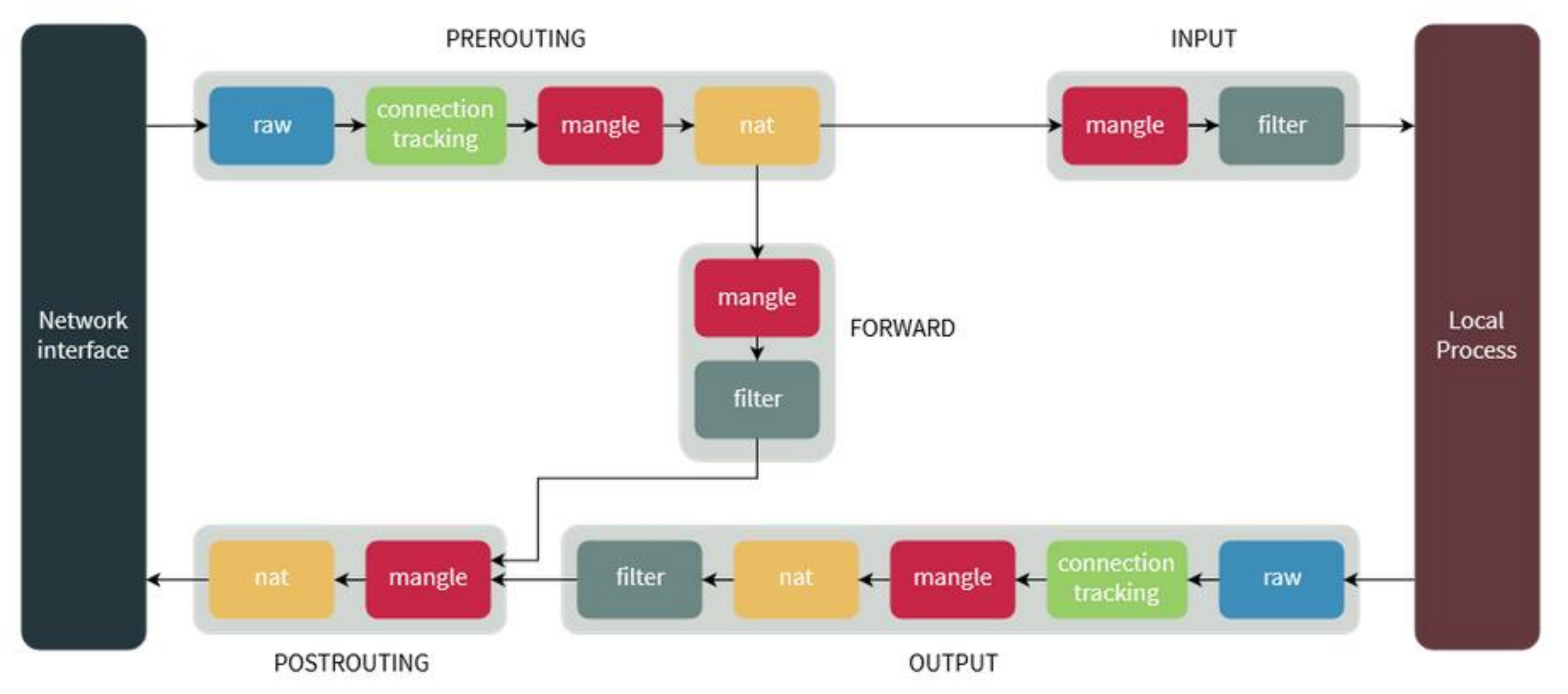
mangle table OUTPUT chain

nat table OUTPUT chain

filter table OUTPUT chain

security table OUTPUT chain

CHAINS & IPTABLES



Refer to below link for a description of the chains iptables relationship:
https://www.booleanworld.com/depth-guide-iptables-linux-firewall/

NETFILTER PACKET FLOW USING BUILT-IN CHAINS

- Packet filtering takes place within the kernel at the five packet filtering points (also known as Built-in Chains).
- Filtering points names are case-sensitive and are all in upper case.
- The five chains are:

1) INPUT Chain
2) PREROUTING Chain
3) FORWARD Chain
4) OUTPUT Chain
5) POSTROUTING Chain

NETFILTER PACKET FLOW USING BUILT-IN CHAINS

- ☻ FORWARD Chain: This filtering point handles packets being routed through the local system (mangle, filter). FORWARD Chain is present in the *mangle and filter tables. Only packets that neither* originate nor terminate at the local host traverse this chain.

- ☻ INPUT Chain: This filtering point handles packets destined for the local system, after the routing decision. INPUT Chain is present in the *mangle and filter tables.* Only packets terminating on localhost traverse this chain.

- ☻ PREROUTING Chain : This filtering point deals with packets first upon arrival (conntrack , mangle , nat). PREROUTING Chain is present in the *conntrack, nat and mangle tables. Packets traverse this chain* **before a routing decision is made by the kernel.**

NETFILTER PACKET FLOW USING BUILT-IN CHAINS

- ☻ OUTPUT Chain: This filtering point handles packets after they have left their sending process and prior to POSTROUTING (conntrack, mangle, nat ,filter). OUTPUT Chain is present in the *conntrack, nat, mangle and filter tables. Only packets* originating on localhost traverse this chain.

- ☻ POSTROUTING Chain: This filtering point handles packets immediately prior to leaving the system (mangle, nat). POSTROUTING Chain is present in the *nat and mangle tables. Packets traverse this chain* **after a routing decision is made by the kernel.**

RULE MATCHING FEATURES IN NETFILTER

- Rules are in ordered list
- Packets tested against each rule in turn
- If the criteria is matched, it goes to the rules specified in the target (or) executes the special values mentioned in the target
- If the criteria is not matched, it moves on to the next rule.
- Rule may specify multiple criteria for match
- Every criterion in a specification must be met for the rule to match (logical AND)
- Chain Policy applies if no match

RULE TARGETS

- Built-in targets: DROP, ACCEPT, QUEUE, RETURN
- Extension targets: LOG, REJECT, custom chain
  - Reject sends a notice returned to sender
  - LOG connects to system log kernel facility
  - LOG match does not exit the chain

- Target is optional, but no more than one per rule and defaults to the chain policy if absent

- Rule targets determine what action to take when a packet matches the rule's selection criteria
- Target names qualify the –j option of the iptables command (think j as in jump).
- A target can be built in (Base) target, a custom chain or extension target.
- In the netfilter framework, it supports DROP and ACCEPT as the two base targets.

TARGET VALUES

Target – if the criteria in a rule is matched the target is executed. A target decides the fate of a packet, such as allowing or rejecting it.

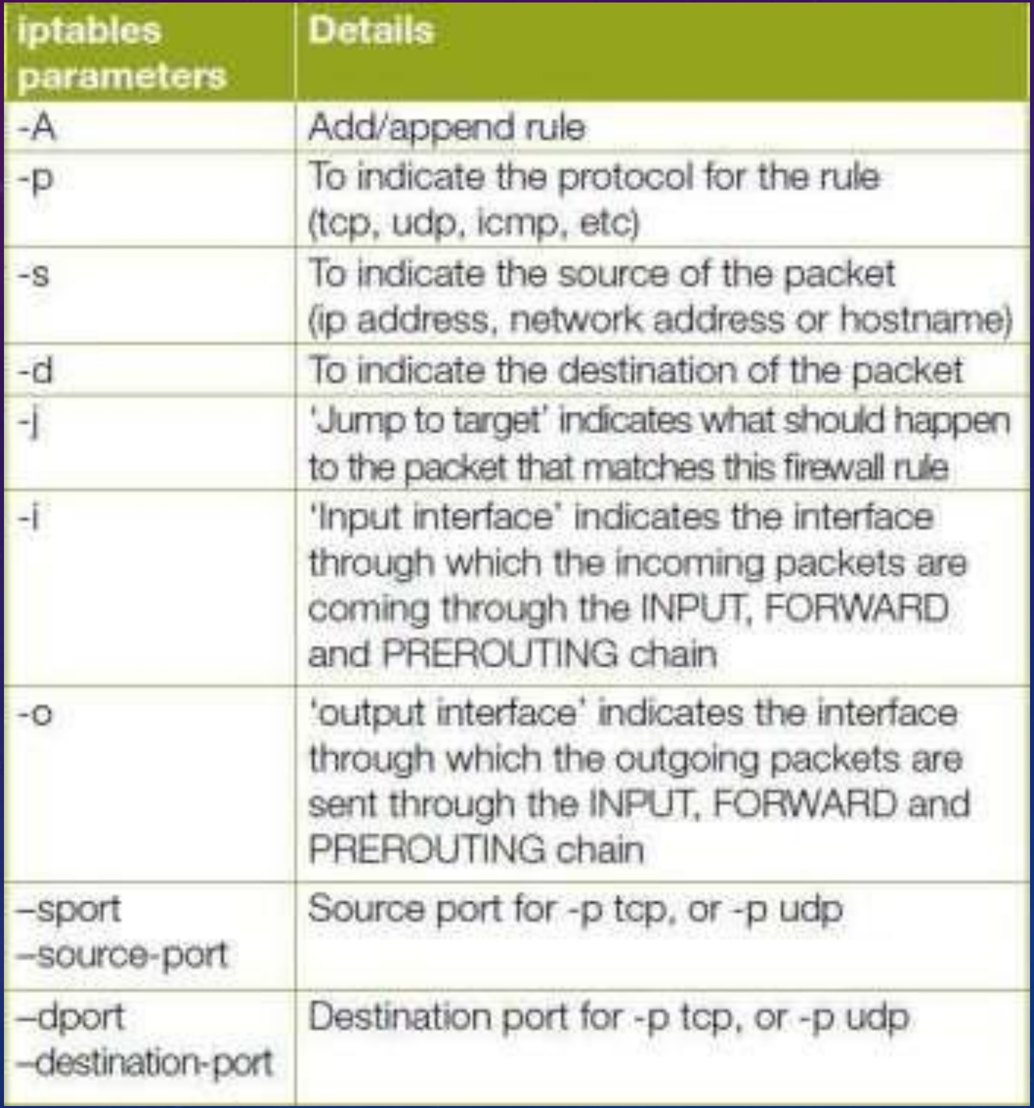
- ACCEPT – Firewall will accept the packet.
- DROP – Firewall will drop the packet.
- QUEUE – Firewall will pass the packet to the userspace.
- RETURN – Firewall will stop executing the next set of rules in the current chain for this packet. The control will be returned to the calling chain.

Example:

```
iptables -t filter -A INPUT -s 192.168.10.1 -j DROP
iptables -A INPUT -s 192.168.10.1 -j DROP
```

iptable Syntax

https://www.informit.com/articles/article.aspx?p=421057&seqNum=4



iptables parameters	Details
-A	Add/append rule
-p	To indicate the protocol for the rule (tcp, udp, icmp, etc)
-s	To indicate the source of the packet (ip address, network address or hostname)
-d	To indicate the destination of the packet
-j	'Jump to target' indicates what should happen to the packet that matches this firewall rule
-i	'Input interface' indicates the interface through which the incoming packets are coming through the INPUT, FORWARD and PREROUTING chain
-o	'output interface' indicates the interface through which the outgoing packets are sent through the INPUT, FORWARD and PREROUTING chain
–sport –source-port	Source port for -p tcp, or -p udp
–dport –destination-port	Destination port for -p tcp, or -p udp

EXAMPLE

- ☻ An INPUT rule for the filter table:

iptables –t filter –A INPUT –s 192.168.10.1 –j DROP

- ○ In this example , the ‘-A’ is for append, indicating that a single rule will be appended (in this case) to the INPUT chain of the filter table.
- ○ This rule causes any packet with a source address (hence ‘-s’) of 192.168.10.1 to match and “jump” to its target which is DROP, causing the packet to be discarded.

BASIC CHAIN OPERATIONS

- List rules in a chain or tables (-L or vL)
- Append a rule to the chain(-A)
- Insert a rule to the chain(-I)
  - -I Chain (inserts as the first rule)
  - -I chain 3 (inserts as rule 3)
- Delete an individual rule(-D)
  - -D CHAIN 3 (deletes rule 3 of the chain)
  - -D CHAIN RULE(deletes rule explicitly)

BASIC CHAIN OPERATIONS

- To list the contents of the chain (rules and policy), use –L .
- Using –v with this option displays packet and byte counters, interfaces and protocols as well

```
[root@stationX ~]# iptables -t filter -L OUTPUT
```

- List the contents of the table (rules and policies) for all the chains

```
[root@stationX ~]# iptables -t filter -L
```

BASIC CHAIN OPERATIONS

Two other options might be useful when listing rules: -n and –line numbers. –n prevents time consuming reverse lookups of IP addresses, while –line numbers will display line numbers that could then be used to determine the rule number to be used with –D(delete) or –I (insert)

```
[root@stationX ~]# iptables –t filter –nvL --line-numbers
```

Several operations may be performed on a chain.
Use –A to append a rule to the end of an existing chain.
If the table is not specified, then the filter table is assumed.

```
# iptables –A INPUT –s 12.34.12.34 –j DROP
```

Insert a rule into a chain as the first or at a given point with –I
Use –D to delete rule form a chain based on its sequence number, or explicit specification. Rules are numbered from one

COMMON MATCH CRITERIA

- IP address or network
  - -s 192.168.0.0/24
  - -d 192.168.0.1
- Network interface
  - -i lo
  - -o eth1
- Criteria can be inverted with !
  - -i ‘!’ eth0 –s 192.168.0.0/24

COMMON MATCH CRITERIA

- Most rules in the filter table involve allowing or denying packets based on their source or destination.

- A packet’s source or destination can be specified with –s or –d respectively. The option should be followed by an IP address or IP/Netmask combination or hostname.

- Netmasks can use CIDR (192.168.0.0/24) or VLSM (192.168.0.0/255.255.255.0) notation.
- Using a hostname is not recommended because it will just be translated into an IP when the rule is stored anyway.

COMMON MATCH CRITERIA

The following example would allow packets from any address on the 192.168.0.x network through the firewall.

```
#iptables –I INPUT -s 192.168.0.0/24 –j ACCEPT
```

Packets can also be matched based on the physical network interface they are arriving on or leaving through. This is done with the –I and –o options respectively. The following command would only allow packets destined for the local network to leave via eth0 (assuming all other packets are denied by default)

```
# iptables –I OUTPUT –o eth0 –d 192.168.0.0/24 –j
ACCEPT
```

Another common interface-based rule is the following, which allows all packets arriving on the system’s loop back interface through the firewall

```
# iptables –I INPUT –i lo –j ACCEPT
```

Since only local processes have access to the loopback interface, traffic on lo is usually unfiltered.

Thus many firewall rulesets begin with a rule like the above

Any match criterion may be negated by prepending ‘!’ (with the quotes) to the value. While the quotes are not strictly necessary in these examples, they are considered best practice when dealing with a special character the bash shell might desire to expand. The following example would block all traffic except packets from 192.168.0.1

```
# iptables –I INPUT –s ‘!’ 192.168.0.1 –j DROP
```

COMMON MATCH CRITERIA

- Transport protocol and port
  - -p tcp --dport 80
  - -p udp --sport 53

- ICMP Type
  - -p icmp –icmp-type host-unreachable

  - Packets can also be matched by their source or destination ports. Because port numbers are ambiguous unless associated with a transport protocol (since, e.g , tcp port 53 is distinct from udp port 53) references to port must always specify a layer -4 protocol with the –p option.

  - Destination ports are matched with --dport and source ports with -- sport. Ranges of port can be listed as “start_port:end_port” . If end_port is left out, it is assumed to be the highest possible port.

COMMON MATCH CRITERIA

The following example would allow tcp packets coming from port 123 of 192.168.0.1 to port 1024 or above of 192.168.0.2

```
# iptables –I INPUT –p tcp –s 192.168.0.1 –sport
123 –d 192.168.0.2 --dport 1024: –j ACCEPT
```

ICMP packets, which include ping request and responses, destination-unreachable messages from routers and many other types of network diagnostic messages can be selectively filtered by specifying icmp as the protocol and using the –icmp-type to match specific types. The following examples would explicitly deny ping requests and explicitly allow destination unreachable messages, respectively:

```
# iptables –I INPUT –p icmp –icmp –type Destination-
unreachable –j ACCEPT
```

While some networks choose to block ping requests , denying all ICMP packets is not recommended. Certain types, such as destination unreachable messages represent important information that network clients should receive.

BASIC IPTABLES SYNTAX REVIEW 1

- Add or delete a rule

```
iptables [-t table] -[AD] chain rule-spec [options]
```

- Examples:

```
iptables -t filter -A INPUT -p tcp --dport 22 -j
ACCEPT
iptables -D INPUT -p tcp --dport 22 -j ACCEPT
```

- Note the -A option means “append” — the rule is added to the **end of** the chain.

BASIC IPTABLES SYNTAX REVIEW 2

- Insert a rule into a chain

```
iptables [-t table] -I chain [rulenum] rule-
specs  [options]
```

- Example:

```
iptables -I INPUT 2 -p tcp --dport 110 -j ACCEPT
```

- This inserts a rule to accept incoming TCP traffic on port 110 directly before the existing rule number 2.

BASIC IPTABLES SYNTAX REVIEW 3

- Delete a rule from a chain by rule number

  iptables [-t table] -D chain [rulenum] [options]

- Example:

  ```
  iptables -D INPUT 2
  ```

- This deletes the rule number 2. Note that you would need to use iptables --line-numbers -L to get the number.

BASIC IPTABLES SYNTAX REVIEW 4

Flush (delete) all rules from a chain

- iptables [-t table] -F chain [options]

Examples:

- `iptables -t filter -F INPUT`
- `iptables -t nat -F POSTROUTING`

You can also add the -Z switch to zero the packet counters as well.

Note that all chains in the specified table will be flushed if you do not specify a chain, and remember that the default chain is **filter if one is not** specified.

BASIC IPTABLES SYNTAX REVIEW 5

- Set the default chain policy

  ```
  iptables [-t table] -P chain target [options]
  ```

- Example:

  ```
  iptables -t filter -P INPUT DROP
  ```

- The chain policy sets the default action to take on the packet if it does not match any of the rules in the chain it traverses.

BASIC IPTABLES SYNTAX REVIEW 6

- Create a custom chain

  iptables [-t table] -N chain

- Example:

  iptables -t filter –N Place

- This creates a custom chain called *Place in the* ***filter table. You would*** jump to it with something like this:

  iptables -t filter -A INPUT -j Place

BASIC IPTABLES SYNTAX REVIEW 7

- Delete a custom chain

  iptables [-t table] -X chain

- Examples:

  ```
  iptables -t filter -X Place
  ```

- This deletes the custom *Place chain we just created.* Note that there must not be any other rules that jump to a custom chain in order to remove it.

CHAIN POLICY REVIEW

- A chain’s policy decides what happens to packets when they “fall off” the chain; that is, if a packet does not match any of the rules that it sees, the chain policy is applied to it.
- Whether you should do a default **ACCEPT** or **DROP** policy depends on your needs, but generally speaking, **DROP** policy is the better option for filter table chains (except perhaps OUTPUT), and ACCEPT policy is better on other tables’ chains.

RULES ORDER REVIEW

- The order of rules is very important. Rules are applied to the packets in the order in which they were added (within the context of each individual table and chain).

- As an example, if you append a rule to the filter table's INPUT chain to DROP packets on port 22, and then append another rule to ACCEPT packets on port 22 from a specific IP address, the packets will still be dropped because they will match the DROP rule before they match the ACCEPT rule.
- The first matching rule "wins".

SUMMMARY

- We can configure Linux firewall via:
  - system-config-firewall
  - system-config-firewall-tui
  - iptables <options>
  - Edit the /etc/sysconfig/iptables file
- There are 5 tables – filter, nat, mangle, raw, security
- There are 5 types of chains – INPUT, PREROUTING, FORAWRD, POSTROUTING, OUTPUT
- iptables tool to configure filter table – chain policy & rules & target