Edinburgh Napier UNIVERSITY

Networked Services
Essential Firewalls

Module Leader: Dr Gordon Russell
Lecturers: G. Russell

Edinburgh Napier UNIVERSITY

Firewalls

Firewalls

- Firewalls and the implementation of Management Policy plus proactive security methods
- Management Policy can include
  - Network Services allowed to run, and under what conditions
  - Client access to external resources
  - Quality of service
  - Security and maintaining availability

Edinburgh Napier UNIVERSITY

Corporate Firewalls

- Cisco ASA
- Cisco router ACLs
- Linux router iptables

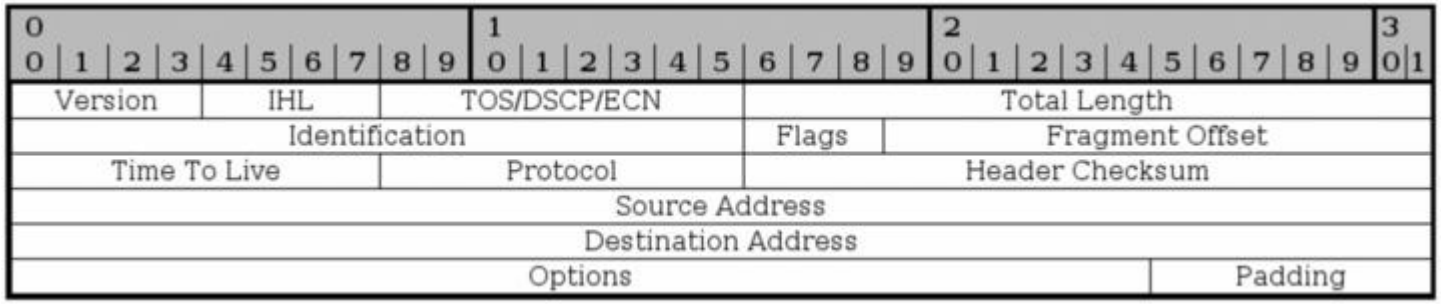
- ASA syntax is strange, yet ASA solutions are popular!
- ACLs are weak for policy implementations
- iptables is powerful but uncommon.

Edinburgh Napier UNIVERSITY

Linux Firewalls

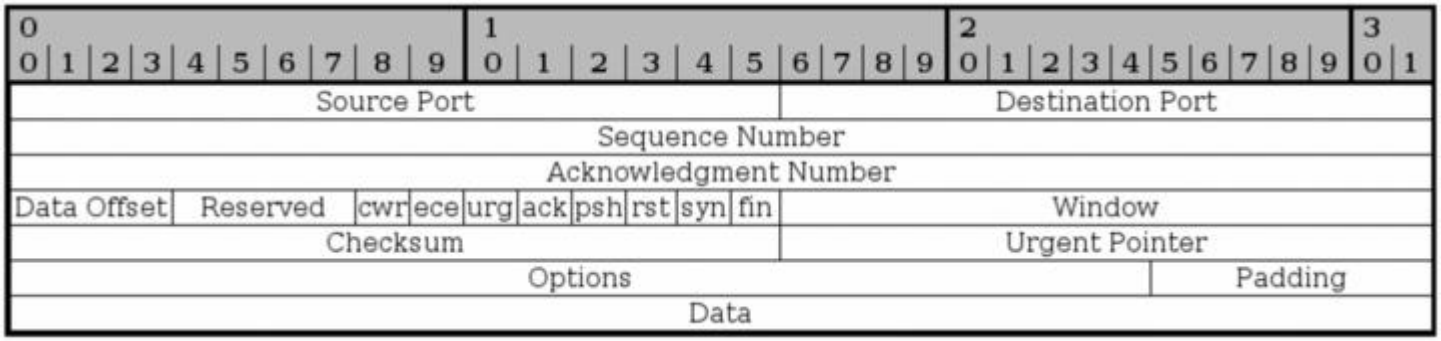
- Security issues which Linux firewalls can deal with:
  - Denial of service
  - Differences between external and internal users
  - Hiding local-only services
  - Traffic management
  - Virus and hacking protection
  - Implementing management policy
- Rules are needed in terms of packets arriving, leaving, and passing through.

Packet Information: IP



- Decisions can be made about packets based on their IP header:
  - Packets from 10.0.0.1 are to be trusted
  - Packets sent to 10.1.1.1 should be blocked
  - ICMP protocol packets should be blocked
  - etc

Packet Information: TCP



- Decisions can be made about TCP packets based on their TCP header:
  - Packets sent to port 80 (http requests) should be allowed
  - Packets sent from port 57890 should be blocked
  - SYN packets with FIN (an invalid combination) should be blocked.
  - etc

Edinburgh Napier UNIVERSITY

iptables

- Since kernel 2.4, the standard interface for firewall configuration is iptables.
- It implements its rules using many “tables”
  - Filter – handles standard “firewall” things
  - NAT – rewriting of source/destination IPs
  - Mangle – specialised hacking of packet info
  - RAW – low-level modifications to packets
- Most firewalls only need to be involved with the filter table

Edinburgh Napier UNIVERSITY

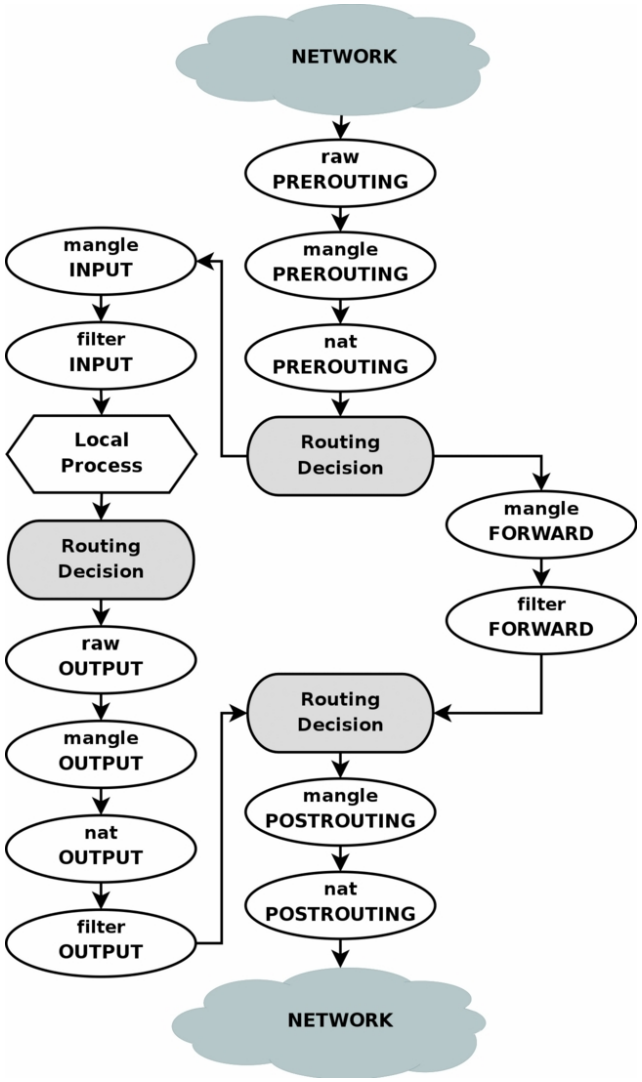
Chains

- Within each table is a number of chains.
- Think of each table as containing different types of rules you might want to use (like “filter” rules).
- Think of chains as defining WHEN a packet will have those rules applied to them.
- Chains are done in a particular order.
- Some packets only go to some chains and not others (depending on how the packet was made).

Edinburgh Napier UNIVERSITY

Chain Names

- Some chain names you might see are
  - PREROUTING
  - INPUT
  - OUTPUT
  - FORWARD
  - POSTROUTING

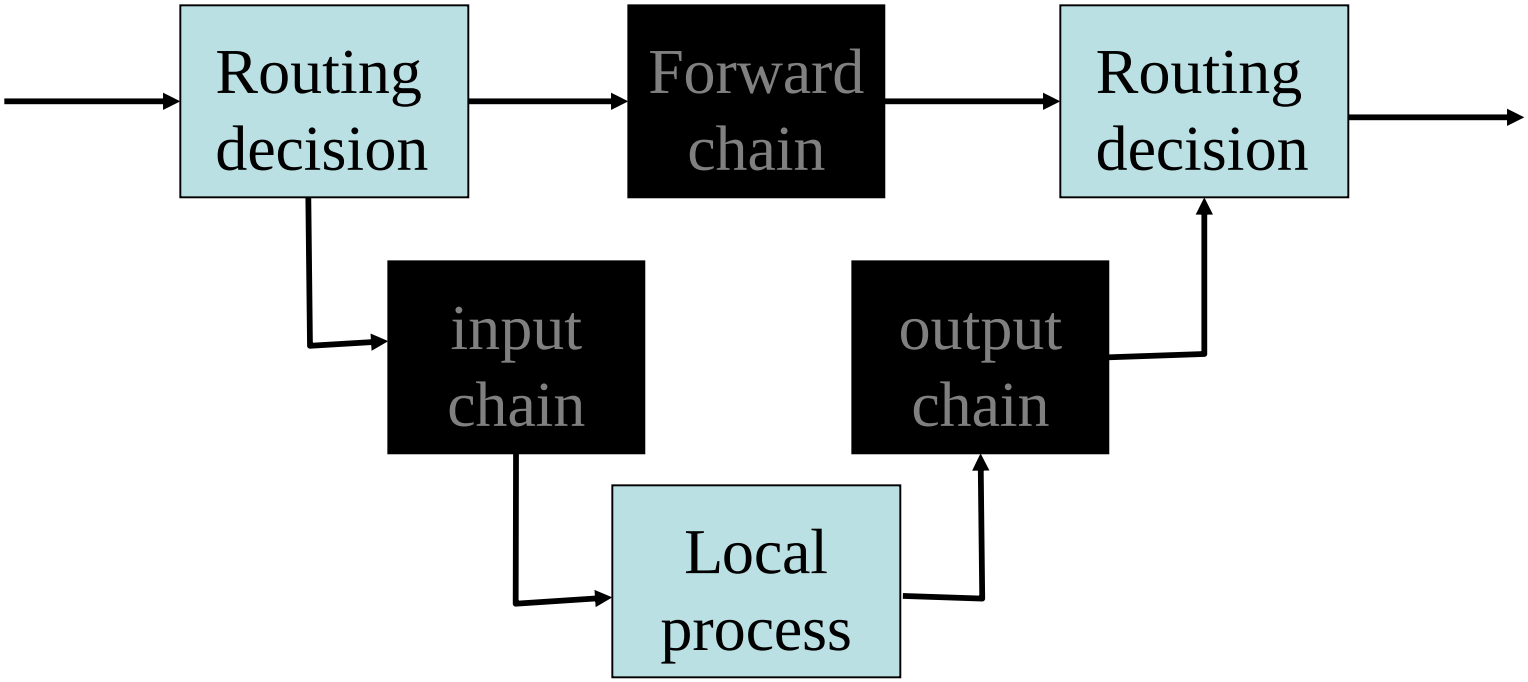


- Summary of the tables and chains a packet will traverse in iptables.

Source:
http://iptables-tutorial.frozentux.net/iptables-tutorial.html

Edinburgh Napier UNIVERSITY

FILTER TABLE CHAINS: INPUT, OUTPUT, FORWARD



Edinburgh Napier UNIVERSITY

FILTER TABLE

- In the filter table a packet will only go through 1 chain.
  - Packets created within the computer go through OUTPUT
  - Packets which need to be routed from one eth device to another eth device go through FORWARD.
  - Packets received by the computer for processing locally go through INPUT

Edinburgh Napier UNIVERSITY

A Chain

- Each chain is made up of 0 or more rules.
- A rule is a set of tests and an action.
- If the tests are all true then the action is performed.
- If a test is partially or completely false then the next rule is looked at instead.
- If all the rules are used up without an action taking place, then the chain POLICY is done instead (i.e. a default rule).

Edinburgh Napier UNIVERSITY

Tests

- Each rule has 0 or more tests.
- There are many tests possible, such as:
  - Is this TCP?
  - Is this from 10.0.0.5?
  - Is this from port 22?
  - Is this going to port 23?
  - Is this going to ip 50.0.0.1?
  - Am I receiving packets faster than 10 per second?
  - Which interface is it going out/in on?
- Remember you can combine these tests, e.g. TCP and from port 22.

Edinburgh Napier UNIVERSITY

Actions

- If all the tests are true then the action is carried out.
- Some actions are terminating, meaning that once they are done no more rules are looked at.
- A few actions are non-terminating. This is useful if you want to say print a message if the test is true, but continue on with the next rule anyway.

- Example actions are:
  - DROP – delete a packet immediately and terminate
  - ACCEPT – the packet is good and terminate
  - REJECT – delete the packet and terminate, but send back an ICMP message to the sender
  - LOG – print to syslog a message and move onto the next rule.

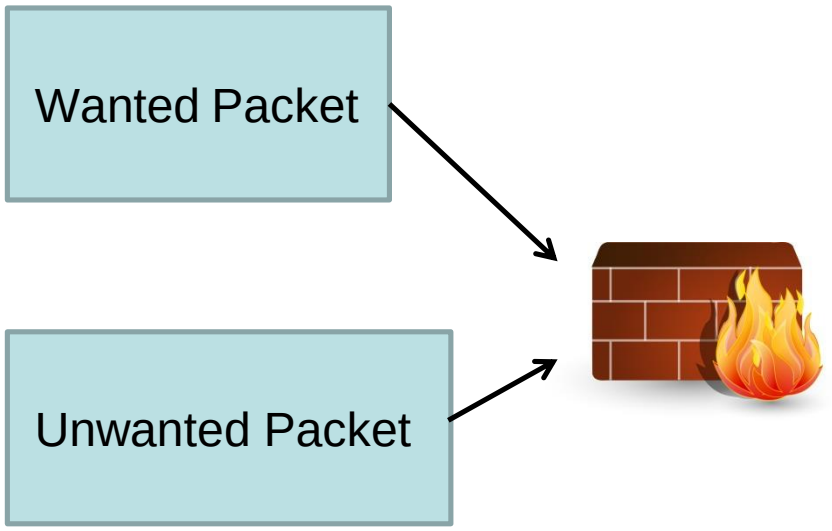
Edinburgh Napier UNIVERSITY

Some tests:

• Is this TCP ?	-p tcp
• Is this from 10.0.0.5?	-s 10.0.0.5
• Is this from port 22?	--sport 22
• Is this going to port 23?	--dport 23
• Is this going to ip 50.0.0.1?	-d 50.0.0.1
• Is this going out on eth0?	-o eth0
• Is this coming in from eth0?	-i eth0

Edinburgh Napier UNIVERSITY

Default Policy



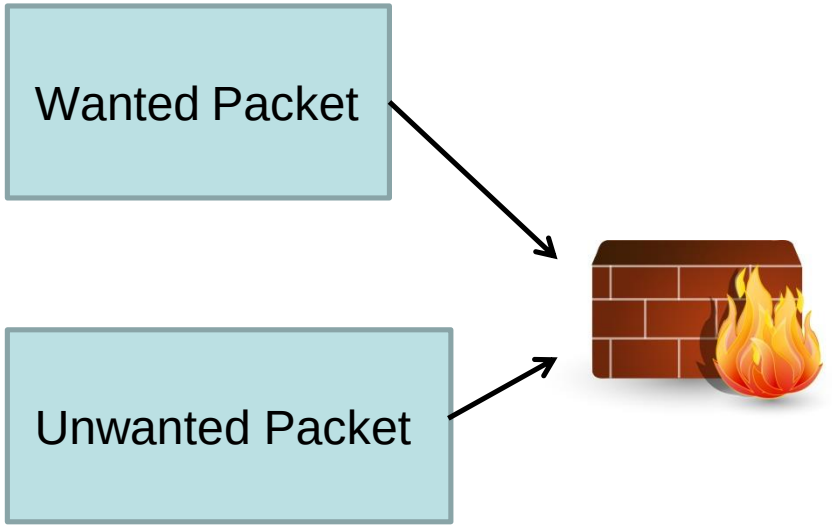
Rules:

1) Accept Packet “Wanted”
2) Drop Packet “Unwanted”

What about packets where there are no rules which match?

You need a default action…

Default Policy: ACCEPT?



Rules:

~~1) Accept Packet "Wanted"~~

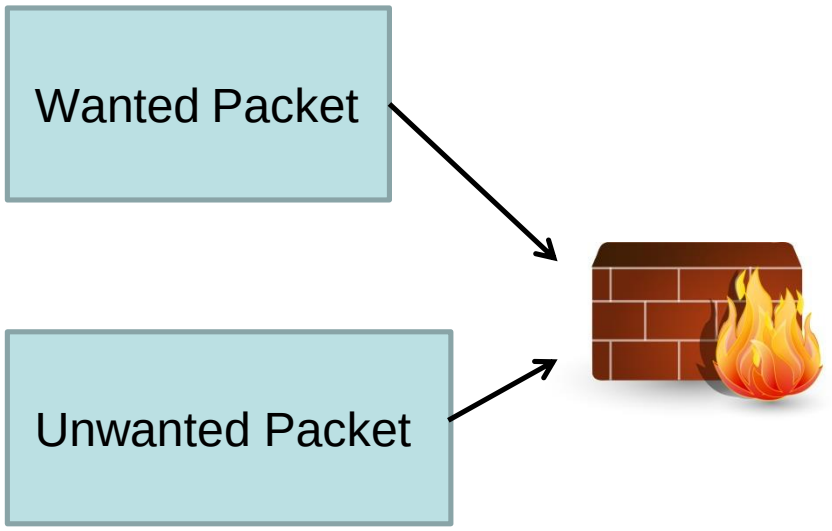
2) Drop Packet "Unwanted"

With default ACCEPT, all packets are accepted unless you have a rule to DROP or REJECT them.

You must have a rule for any bad packets you could receive.

Edinburgh Napier UNIVERSITY

Default Policy: DROP?



Rules:

1) Accept Packet “Wanted”
2) ~~Drop Packet “Unwanted”~~

With default DROP, all packets are DROPPED unless you have a rule to ACCEPT them.

You must have a rule for any good packets you could receive.

More secure but easier to lock yourself out by accident

Edinburgh Napier UNIVERSITY

Setting the policy

```
$ iptables -P INPUT ACCEPT
$ iptables -P OUTPUT ACCEPT
$ iptables -P FORWARD DROP
```

- This is a typical unsecured machine configuration. Typical machines only have 1 eth device, so don't forward. Otherwise, all packets are allowed.
- Better to have INPUT DROP and have the ACCEPT rules, but remember INPUT DROP without rules then drops everything!

```
$ iptables -P INPUT DROP
```

Edinburgh Napier UNIVERSITY

Editing firewalls

- iptable does allow you to edit firewalls dynamically.
- However, this is very problematic and difficult.
- Instead, I recommend putting all your rules in a file and running that file to change the firewall.
- This allows you to use your favourite editor to write the firewall.
- At the start of the file, delete all current firewall rules in each table using “-F”.

Edinburgh Napier UNIVERSITY

```
$ touch firewall
$ chmod +x firewall
$ vi firewall
/sbin/iptables -F INPUT
/sbin/iptables -F OUTPUT
/sbin/iptables -F FORWARD

# Set the default policies for the chains
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD DROP
```

Edinburgh Napier UNIVERSITY

- To load these rules do

$ ./firewall

- However, don't do that yet. The default is DROP for INPUT. Without more rules you will be kicked out of the server never to return…
- This is bad if the server is 5 minutes walk away. But if it is 500miles away you are in trouble!
- This type of firewall is INGRESS ONLY. No rules for going out (OUTPUT/EGRESS). Kind of like the XP firewall…

Edinburgh Napier UNIVERSITY

Discussion

- How would iptables firewall rules, combined with a Linux based router, compare with a custom firewall appliance, such as a Cisco ASA5500?

Edinburgh Napier UNIVERSITY

Discussion

- If a packet arrives at a linux server, and it needs to be delivered onto a local process on that server, then which chain in the filter table would it traverse?