

**Information security - IV**  
**Prof. M J Shankar Raman**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture – 54**  
**IP Tables Rules and Tool usage**

[FL] Welcome to this session on Network Security and Forensics. We have been discussing a lot about firewalls and the logs at the firewalls generate etcetera. So, what we will do is we try to go hands on into IP tables which is used in Linux and does the work of a firewall.

Before we go into the practical part of it where we look at the installation as well as the demo of IP tables; we will learn the basics of the architecture of IP tables and we will also once we look at the demo; we will also look at what are the precautions as a network security specialist or a network administrator, you have to take to ensure that you do not get your network blocked by putting this firewalls. See for example, you block yourself by putting a firewall you are going to have lot of problems later accessing the logs or other things.

And finally, in the last module we will look at the logs and different types of logs and a little bit of understanding about the log formats etcetera. Why we want to do is that we will be able to write some Linux scripts and automate the whole process of collecting the forensic evidence. So, let us start with IP tables we will the whole section is divided into two parts from our point of view. So, we are going to integrate this IP logs along with the IP tables along with the logs that we are going to see.

So, the next sections will include part of demo as well as if you want to do some kinds of hands you can do it and finally, we will look at some theoretical part of logging. So, coming to the objective ok; so, for this module we will look at what is IP table? What is the architecture of IP tables? And we will also look at what are called chains.

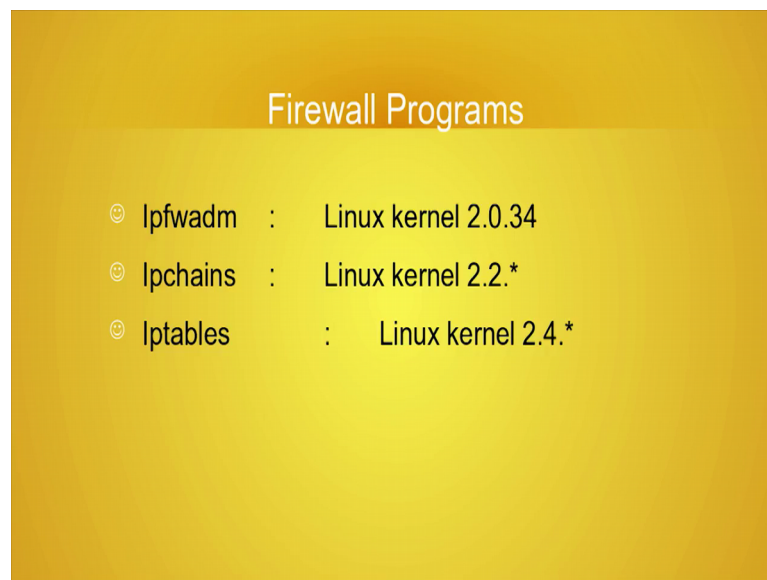
So, essentially the idea is that you give a bunch of rules and these rules form a sort of pipeline in IP tables. So, that the packets goes stage by stage from one route to the other route and depending on what you have given as the rule that accept drop or log whatever it is you will the firewall will behave so, according to what command you want to give.

We will also see loading and unloading the kernel driver models and all that will not be a part of this.

Because we will just look at the insulation part of it firewall which will do all most all the activity and practical is definitely you will look at a small example where we try to prevent a icing packets from pinging our system. Or when you try to ping the other system you ensure that you can filter out the IP ICMP packets and so, you do not get a response for the ping.

One of the place and we will also look at some common configurations ok. So, how to block Face book, how to block Face book for a particular time or a Gmail or whatever it is. So, this will be a sort of template which you can use in your organization to block certain websites ok. So, this is how the firewall programs progressed in Linux ok; initially we had in kernel 2.0.34 the IPFWAM ok. And then kernel 2.2.X you had something like IP chains then we moved to internal 2.4 we have IP tables and in kernel 2.6 we call it as something else ok.

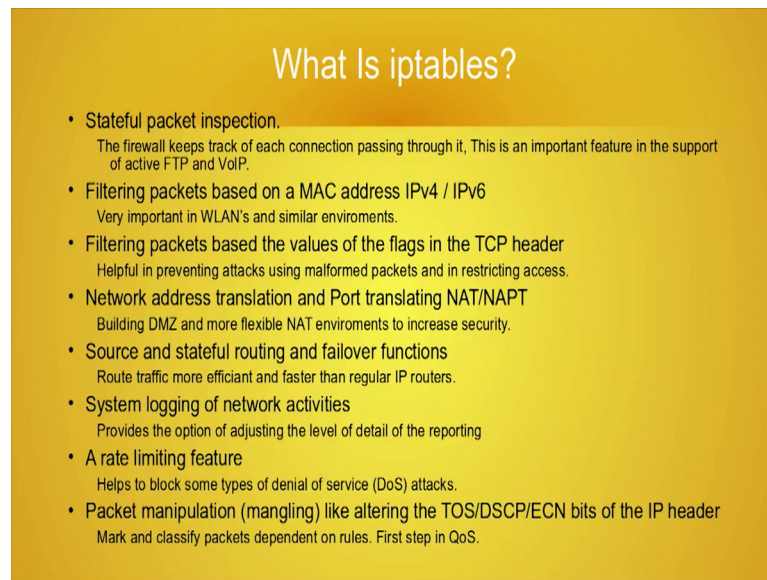
(Refer Slide Time: 03:47)



Firewall Programs	
☺ ipfwadm	: Linux kernel 2.0.34
☺ Ipchains	: Linux kernel 2.2.*
☺ Iptables	: Linux kernel 2.4.*

So, we I mean the essential in general it is called as IP tables ok.

(Refer Slide Time: 03:54)



### What Is iptables?

- **Stateful packet inspection.**  
The firewall keeps track of each connection passing through it, This is an important feature in the support of active FTP and VoIP.
- **Filtering packets based on a MAC address IPv4 / IPv6**  
Very important in WLAN's and similar environments.
- **Filtering packets based the values of the flags in the TCP header**  
Helpful in preventing attacks using malformed packets and in restricting access.
- **Network address translation and Port translating NAT/NAPT**  
Building DMZ and more flexible NAT environments to increase security.
- **Source and stateful routing and failover functions**  
Route traffic more efficient and faster than regular IP routers.
- **System logging of network activities**  
Provides the option of adjusting the level of detail of the reporting
- **A rate limiting feature**  
Helps to block some types of denial of service (DoS) attacks.
- **Packet manipulation (mangling) like altering the TOS/DSCP/ECN bits of the IP header**  
Mark and classify packets dependent on rules. First step in QoS.

So, what are IP tables? Ok I think we had seen briefly about IP tables it is a stateful packet inspection that firewall keeps track of each connection that flows through it ok. So, we will be able to take advantage of using this is you will be able to monitor VoIP or FTP where I mean TCP connections.

So, you will be able to filter take action based on what is that the flow that is there in your network ok. It also allows filtering the packets based on MAC addresses or IP v 4 or IP v 6 addresses; well there are two different kernels for IP v 4 we generally use IP tables and for IP v 6 you use IP 6 tables ok. And this is very useful when you are looking at the in network and out network then demilitarized zones ok. The other thing that you can do with IP tables is you can look at the flags of the TCP header and take actions based on flags.

You could also see you can have network address translation done with IP tables. So, that you mask of the real IP address for example, any of the data that goes via your natural interface I mean you can actually change the IP address of the interface. So, that the data goes to some other location the advantage of using port translation is that you can have a NAT environment and many of you know that if you have a NAT environment we are breaking the internet policies, but to some extent, but then this provides some extend what secure environment for the internal machines inside the organization.

You will also look at I mean this IP table also provides failure functions of course we will not see demo of it, but it is better to know that yes it provides some kind of failure functions it does it can do system logging and the log format has a certain format. So, that you it is easy for you to parts; it can also do a rate limiting see this is especially used when you have DoS attacks ok.

See DoS attacks actually send the lot of packets within a small period of time to ensure that your system fails or responds very slowly. So, you can also put a rate limiting feature with firewall to ensure that you do not land up with lot of DoS attacks. The other thing you can do is, it does something known as mangling like altering the type of survey is are the explicit conjection notification bits etcetera.

So essentially if your network supports or the end system support explicit conjection notification; you can actually set a bit on one of the IP packets and then the and the TCP packets and then there will be a end to end system will take care of this explicit conjection notification and then try to rate limit between themselves. So, this is one of the first limits when the issue of mangling helps you in developing quality of service and IP tables helps you achieve it.

(Refer Slide Time: 07:10)

Processing For Packets Routed By The Firewall 1/2

Queue Type	Queue Function	Packet transformation chain in Queue	Chain Function
Filter	Packet filtering	FORWARD	Filters packets to servers accessible by another NIC on the firewall.
		INPUT	Filters packets destined to the firewall.
		OUTPUT	Filters packets originating from the firewall
Nat	Network Address Translation	PREROUTING	Address translation occurs before routing. Facilitates the transformation of the destination IP address to be compatible with the firewall's routing table. Used with NAT of the destination IP address, also known as <b>destination NAT</b> or <b>DNAT</b> .

So, what IP tables does is it provides a bunch of queues on which you can do some kind of a processing ok. So, here are some examples the first queue type is the filter queue type and the function of this queue is packet filtering see you can have a NAT queue type

filter network address translation or you can have mangle queue type which actually does the TCP header modification that we discussed earlier. So, this filter packet filtering has I can do 3 transformation chain has 3 transformation chains in the queue.

(Refer Slide Time: 07:33)

**Processing For Packets Routed By The Firewall 2/2**

		POSTROUTING	Address translation occurs after routing. This implies that there was no need to modify the destination IP address of the packet as in pre-routing. Used with NAT of the source IP address using either one-to-one or many-to-one NAT. This is known as <b>source NAT</b> , or <b>SNAT</b> .
		OUTPUT	Network address translation for packets generated by the firewall. (Rarely used in SOHO environments)
Mangle	TCP header modification	PREROUTING POSTROUTING OUTPUT INPUT FORWARD	Modification of the TCP packet quality of service bits before routing occurs (Rarely used in SOHO environments)

So the first chain is the forward chain and the input chain and the output chain ok. So, the input chain actually filters packets that are coming to the firewall and the output chain filters packets that are going out of the firewall and this is what I am talking about the queue type filter ok.

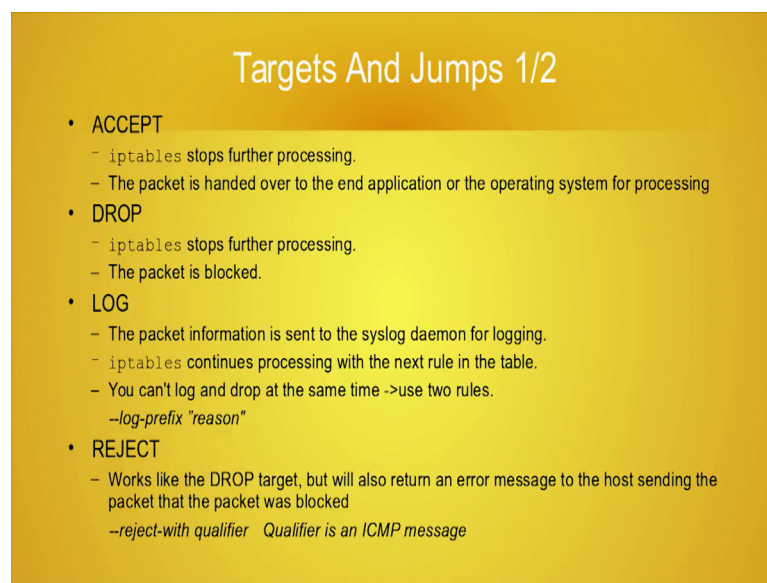
And forward is whether you can send it to some other NIC that is access or servers that are accessible to you by from the firewall I mean through another NIC. So, this is filter packets; so, servers accessible by another NIC on the firewalls. So, this is; so, one is I can whatever I can I process whatever I can send process and I can also provide it to someone else the next part is the next queue is the NAT queue usually I mean this is filtering queue is the one that you will use the most because this is where the TCP connection and the IP packets everything come to you.

Therefore, I use this connection for doing for lot of work for filtering work we will see we will use one of this output to ensure that whenever we do a ping, we do not get a response. We can block our ping packets coming to network address translation ok. So, this is as I told you is for increased security of your internal network ok. So, the address translation actually occurs before routing therefore, you are able to block your IP address

from getting known to the external world this is one of the. So, this is a actually the prerouting stuff; you can also do it postrouting this is known as the source network address translation. And you can also do it for the output for packet generated by the firewall this is not much using small office home office environments.

Similarly mangling is TCP header modification and you can do pre-routing, post routing, output input forward etcetera anyway ok. Going on now once you once I get the packet and then the packet has been filtered now what is the kind of action we can take?

(Refer Slide Time: 09:44)



### Targets And Jumps 1/2

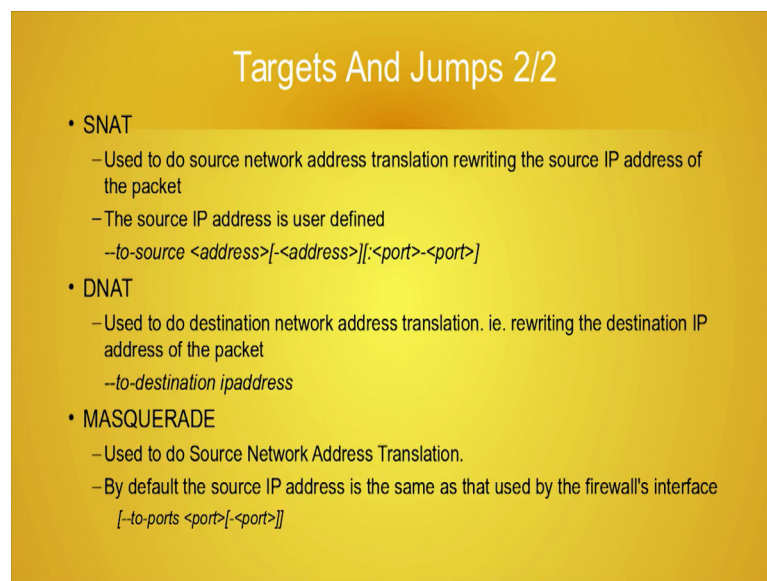
- ACCEPT
  - `iptables` stops further processing.
  - The packet is handed over to the end application or the operating system for processing
- DROP
  - `iptables` stops further processing.
  - The packet is blocked.
- LOG
  - The packet information is sent to the `syslog` daemon for logging.
  - `iptables` continues processing with the next rule in the table.
  - You can't log and drop at the same time ->use two rules.  
`--log-prefix "reason"`
- REJECT
  - Works like the DROP target, but will also return an error message to the host sending the packet that the packet was blocked  
`--reject-with qualifier` *Qualifier is an ICMP message*

Here are 4 actions that you can take one is known as the accept action ok. So, this action the IP tables stops further processing of the data the packet is handed over to end application ok. So; that means, it is like a pass through ok. So, I am accepting it I found no problems let us pass through all the chains then I take it.

The second one is drop the packets the packet is blocked it does not go to the any the external agency or the application ok. And the third one is log so, what it does is that it just logs and then the packet information is set information about the packet is same to `syslog` demo for login and IP tables continuous processing the next rule. So, it logs and then the next rule is going to be accept then it goes and accepted and the goes and accepted is next rule is going to deny just going to deny it or deny reject or drop ok.

So, the log you will have to put the log reason why you want to log the packet ok. And the last one is the reject it similar to drop ok, but will also a return will also send the message to the host sending the packet that I have blocked your packet. So, and you can say why you have dropped the packet ok. So, this is used on I mean usually people use drop ok. Drop means the vendors will not know why the something has not happened here the vendors will know that the packet has been dropped and icing pairing messenger has been returned.

(Refer Slide Time: 11:30)

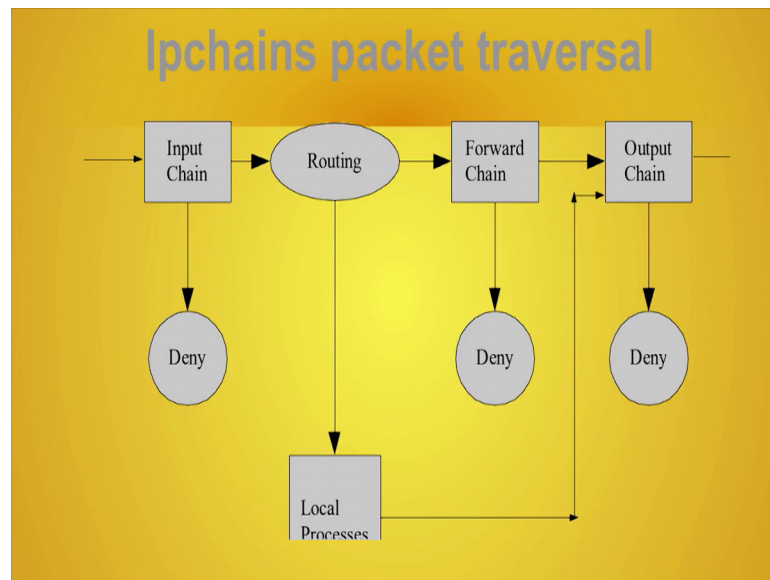


### Targets And Jumps 2/2

- SNAT
  - Used to do source network address translation rewriting the source IP address of the packet
  - The source IP address is user defined
  - `--to-source <address>[<-<address>];<port>-<port>]`
- DNAT
  - Used to do destination network address translation. ie. rewriting the destination IP address of the packet
  - `--to-destination ipaddress`
- MASQUERADE
  - Used to do Source Network Address Translation.
  - By default the source IP address is the same as that used by the firewall's interface
  - `[-to-ports <port>[-<port>]]`

As I told you in NATs you can have source NAT and you can have destination NAT. So, the source IP address is defined as two source address colon address and then you give what port to what port can be used for NAT. And finally, you can also have some masquerade where you used to do source address ok. So, by default the source IP address is same as the firewall interface, but then you cannot do a translation anyway ok. So, you will not look at these things all these things are slightly advanced that will itself take a course on IP tables, but we will just take a look at that IP tables can have this kind of features.

(Refer Slide Time: 12:10)



This is the typical block diagram. For example, I get an input chain then I can pass it to a deny state or I can pass it to a routing state. And the routing state can do some kind of local crossing and then send it to the output chain or the routing state can send it to a forward chain and the forward chain can also deny it or when it goes to the output chain the output chain can also deny.

So, it is a kind of any given commands; so, I give a line. So, this is the first chain. So, in the input chain what do I do? After I process the input chain what do I do? I mean either I deny it or I send it to another chain. So, then this chain what will I do? So, this is the if you if you see the configuration file you will be able to understand how you can pass from one chain to chain to another chain. So, if you look at this there are certain options for IP tables minus t tells you which table by default you have the filter table that we discussed earlier.



(Refer Slide Time: 13:07)

iptables command Switch	Description
-t <table>	If you don't specify a table, then the <code>filter</code> table is assumed. As discussed before, the possible built-in tables include: <code>filter</code> , <code>nat</code> , <code>mangle</code>
-j <target>	Jump to the specified target chain when the packet matches the current rule.
-A	Append rule to end of a chain
-F	Flush. Deletes all the rules in the selected table
-p <protocol-type>	Match protocol. Types include, <code>icmp</code> , <code>tcp</code> , <code>udp</code> , and <code>all</code>

You can also have the NAT table or you can also have the mangle table etcetera. And minus j tells you to which chain I have to jump after process after doing the current processing. So, this will usually be the last line of the rule usually. And minus A tells you that you have to append this rule to the end of the chain and minus F is used to flush all the rules ok.

We will see how minus F operates minus A operates etcetera and minus p tells you the protocol type which can include ICMP, TCP, UDP all of them. Then within a protocol I will be able to specify the port ok, I will be able to even target at this destination port or a source port I could also target the type of packets like syn synnat etcetera I can actually specify both packet port and type. So, I provide all these kinds of flexibility within my chain rule chain ok.

(Refer Slide Time: 14:04)

### Common ICMP (Ping) Match Criteria

Matches used with	Description
<code>--icmp-type</code>	
<code>--icmp-type &lt;type&gt;</code>	The most commonly used types are echo-reply and echo-request

- Allow ping request and reply
  - `iptables` is being configured to allow the firewall to send ICMP echo-requests (pings) and in turn, accept the expected ICMP echo-replies.

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

- Put limit on ping to prevent flood pings

```
iptables -A INPUT -p icmp --icmp-type echo-request \
-m limit --limit 1/s -i eth0 -j ACCEPT
```

So, for example, ICMP which we are going to look at commonly usually we do a echo request and a echo response ok. So, like I mean I do a ping of a machines. So, whether to find the machine is live or not. So, what could you do is you sends an ICMP echo request and if the machine is alive it sends the echo response. Now I can block this tool find out see because usually if we start doing a ping I can do a broadcast ping and all the try to identify the what are all the IP in machines in the network with the IP address with the different IP address that are all available.

So, I can also I can prevent the ping flooding ok; so, because ping packets you can specify the packet size and that might actually take up the band without the network. So, all these things you can actually overcome using the filtering criteria that we discussed. So, here is an examples say for example, it is says a IP tables minus A output minus p ICMP minus ICMP type echo request minus j accept. So, it tells you that on the output queue if you have a ICMP packet and the packet type is echo request, then go the next rule you have to apply is the accept rule.

That means I am going to accept all the ICMP packets goes on the output queue. If you look at the next line it says IP tables minus A that append to the input queue same ICMP packet with the echo reply type and; that means, you accept. Now suppose I want to block I just have to you say for example, I want to send a block all the ICMP packets that goes out of my network then in the first command first rule where I say IP tables minus

A output; that means, it goes out from my firewalls. So, so I will say block or deny or whatever it is if you if you do that this packet will drops I put drop.

So, that the packet gets dropped and it does not go out; that means, you will be not able to send ping packet from your network to another network ok. Similarly I can actually put a limit on how much of ping packets I can say. So, for example, in this says I say limit and limit 1 by s ok. So, so what does it say is that 1 per second ok; so, minus i and then eth0.

So, you will also have to which interface you have to do it ok. So, here you are specifically saying that all these Ethernet 0 interface you do not send more than 1 packet per second kind of.

(Refer Slide Time: 16:49)

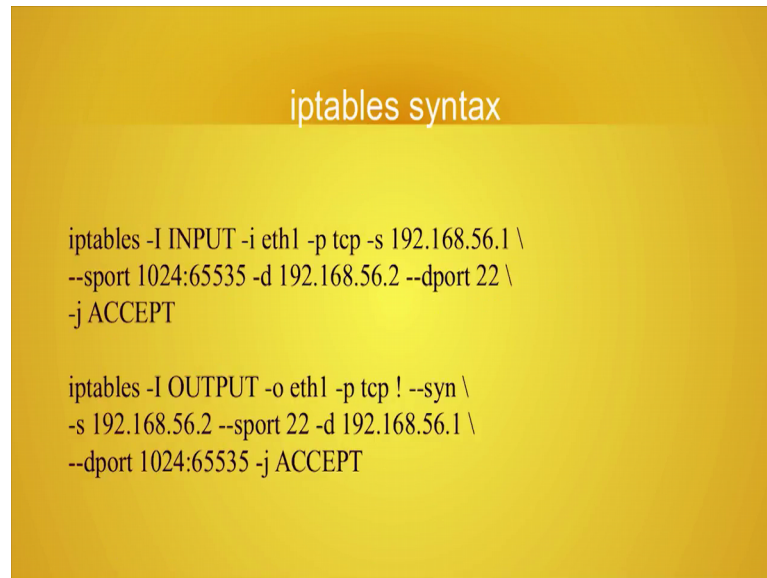
### Important Iptables Command Switch Operations 2/2

-s <ip-address>	Match source IP address
-d <ip-address>	Match destination IP address
-i <interface-name>	Match "input" interface on which the packet enters.
-o <interface-name>	Match "output" interface on which the packet exits

- **We try to define a rule that will accept all packages on interface eth0 that uses TCP and has destination address 192.168.1.1.**
- **We first define the MATCH criterias:**
  - Use default filter table (absence of -t )
  - Append a rule to end of INPUT chain (-A INPUT )
  - Match on source address can be any 0/0 address (-s 0/0 )
  - Input interface used is eth0 (-i eth0 )
  - Match on destination address 192.168.1.1 (-d 192.168.1.1)
  - Match Protocol TCP (-p TCP )
  - If all matches is fulfilled, then jump to ACCEPT chain. (-j ACCEPT )

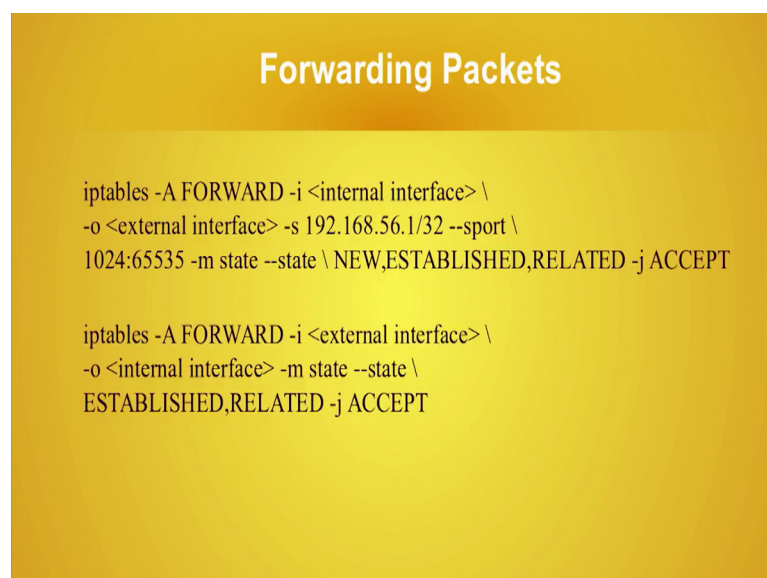
So, essentially what happens is that here are some of the parameters other parameters that you have ok; I think we the previous example actually tells you about how you should give a rule here is another example ok. So, where we can say that ok; so, use a filter table by default if minus t is absent then it is actually the filter table that you will used then it says append to the input chain, then it can says minus s 0 bar 0 ok, so, network address bar mass and so, on. So, this is the way you actually give a command.

(Refer Slide Time: 17:25)



So, here are some examples. So, here input minus I the input queue and then it also says interface is e t h 1; the protocol is TCP source address is 192 168 56 1 and then the source codes are from 1024 to 65535, the destination address is 192 168 56 point 2; the destination port is 22. And then you accept all these connections if you get this establish from here ok.

(Refer Slide Time: 18:00)



Similarly what to do with input ok; so, if you want to forward packets here is an example. So, it can forward packet to the internal from the internal interface to an

external interface ok. And the source address is whatever is given, the port address are this from 1024 to 65535 and then the states that you want to have is new establish related and when you get all these states you just go ahead and accept and forward it ok.

(Refer Slide Time: 18:25)

```
Raw iptables log output

Jun 25 09:05:11 hebe kernel: IN=eth1 OUT= MAC=00:00:92:a7:df:05:02:07:01:23:5e:29:08:00
SRC=10.90.10.112 DST=10.90.10.116 LEN=44 TOS=0x00
PREC=0x00 TTL=60 ID=7276 PROTO=TCP SPT=47785 DPT=10003 WINDOW=16384 RES=0x00 SYN URG=0
Jun 25 09:05:11 hebe kernel: IN=eth1 OUT= MAC=00:00:92:a7:df:05:02:07:01:23:5e:29:08:00
SRC=10.90.10.112 DST=10.90.10.116 LEN=44 TOS=0x00
PREC=0x00 TTL=60 ID=7276 PROTO=TCP SPT=47785 DPT=10003 WINDOW=16384 RES=0x00 SYN URG=0
Jun 25 09:05:12 hebe kernel: IN=eth1 OUT= MAC=ff:ff:ff:ff:ff:00:06:5b:d1:24:bb:08:00
SRC=10.90.50.251 DST=10.90.255.255 LEN=241 TOS=0x00 PREC=0x00 TTL=128 ID=547 PROTO=UDP
SPT=138 DPT=138 LEN=221
Jun 25 09:05:12 hebe kernel: IN=eth1 OUT= MAC=ff:ff:ff:ff:ff:00:06:5b:d1:24:bb:08:00
SRC=10.90.50.251 DST=10.90.255.255 LEN=241 TOS=0x00 PREC=0x00 TTL=128 ID=547 PROTO=UDP
SPT=138 DPT=138 LEN=221
Jun 25 09:05:12 hebe kernel: IN=eth1 OUT= MAC=ff:ff:ff:ff:ff:00:50:04:74:0b:81:08:00
SRC=10.90.10.6 DST=10.90.255.255 LEN=78 TOS=0x00 PREC=0x00 TTL=64 ID=44852 PROTO=UDP SPT=137
DPT=137 LEN=58
Jun 25 09:05:12 hebe kernel: IN=eth1 OUT= MAC=ff:ff:ff:ff:ff:00:50:04:74:0b:81:08:00
SRC=10.90.10.6 DST=10.90.255.255 LEN=78 TOS=0x00 PREC=0x00 TTL=64 ID=44852 PROTO=UDP SPT=137
DPT=137 LEN=58
Jun 25 09:05:15 hebe kernel: IN=eth1 OUT= MAC=ff:ff:ff:ff:ff:00:60:cf:20:2d:37:08:00
SRC=10.90.10.104 DST=10.90.255.255 LEN=78 TOS=0x00 PREC=0x00 TTL=1 ID=60733 DF PROTO=UDP
SPT=137 DPT=137 LEN=58
```

And here is some of the logs that come out of it ok; so, you see this log has a timestamp and then it talks about the what is the kernel that you are using ok; what is the input, what is the output, what is the MAC addresses that are used SR source address, destination address, length and the type of service etcetera.

So, you can log whatever you want and the idea is it writes into log and then you can just use our regular techniques to identify these whatever has happening. You can also do a log analysis there are certain tools for this at the end of the lecture will tell you there are certain tools which you can do you can have lot of graphs based on firewalls.

(Refer Slide Time: 18:56)

```
log_analysis output

3 Chain: input Interface: eth0 >> 211.39.225.244 1559 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 211.44.96.76 1659 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 24.209.129.7 2846 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 4.41.13.124 1537 => 192.168.56.2 TCP 27374
3 Chain: input Interface: eth0 >> 61.255.229.7 3714 => 192.168.56.2 TCP 27374
3 Chain: input Interface: eth0 >> 64.231.21.254 2361 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 65.24.46.200 1992 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 65.33.176.170 1328 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 65.43.103.123 3672 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 66.188.158.191 3064 => 192.168.56.2 TCP 27374
3 Chain: input Interface: eth0 >> 80.224.203.178 4697 => 192.168.56.2 TCP 27374
3 Chain: input Interface: eth0 >> 12.220.98.42 1380 => 192.168.56.2 TCP 27374
3 Chain: input Interface: eth0 >> 193.205.135.94 2498 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 198.83.120.42 1711 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 202.108.234.155 3877 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 202.140.162.42 19914 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 205.158.95.87 1367 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 208.2.225.43 3818 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 212.118.71.3 1429 => 192.168.56.2 TCP 1433
4 Chain: input Interface: eth0 >> 61.85.33.8 2113 => 192.168.56.2 TCP 27374
4 Chain: input Interface: eth0 >> 61.99.45.198 4515 => 192.168.56.2 TCP 27374
3 Chain: input Interface: eth0 >> 62.90.204.2 3798 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 63.231.101.56 61428 => 192.168.56.2 TCP 1433
3 Chain: input Interface: eth0 >> 66.28.45.209 4268 => 192.168.56.2 TCP 1433
```

So, for example, someone wants to query; what are all the issues that we are try that uses internal users are try to contact. Suppose someone in a organization wants to stamps of the whether users are gone to some websites which they are not supposed to go. So, you can actually develop a network of websites there are tools available and these tools may make use of this logs that are generated, you can do an analysis of the logs it can tells you what port people have connected, what is the communication that has happened etcetera.

(Refer Slide Time: 19:36)

### Firewall Optimization

- ☉ Place loopback rules as early as possible.
- ☉ Place forwarding rules as early as possible.
- ☉ Use the state and connection-tracking modules to bypass the firewall for established connections.
- ☉ Combine rules to standard TCP client-server connections into a single rule using port lists.
- ☉ Place rules for heavy traffic services as early as possible.

One of the things that you should do with firewalls is actually when you configure firewalls; you should follow certain rules for optimized performance. You know if the rules are too long I mean you can actually slow down your performer. So, here are some thumb rules that you can follow place loop back rule as early as possible because you want to pass them its loop back is that is locally generated. So, you have to pass them earlier as early as possible. Similarly since you are not going to do any processing with the any forwarding rules; so, keep them as early as possible.

So, the delay from the time you capture the packet and send it to someone else you should be as minimal as possible that is why you have the second result. So, these are all essentially common sense ok; so, if you want to do it much faster use the state and connection track in modules to bypass the firewall for established connections that is the third one.

So, place rules for heavy traffic services as early as possible because the rules are followed one by one and if you face this if you place the heavy traffic services. So, rules at the end; then, the traffic has to go through all the rules that will slow down your firewall. So, these are all some steps which you have to take to ensure that your firewall works well and you get optimized performance. These are all just thumbs some thumb rules at the end of the day you might have to use your common sense to identify a how to work with this firewalls.

So, what we will do is we now understood what is the firewall, what is the IP tables and so, what we will do then next session we are try to put a small demo where we it will tell you how to install firewall on a on Ubuntu machine. And then once the install firewall how do I specify the rules, how do I configure it; we will show a very simple configuration of webbing and if and how to flush the firewall and so on.

Thank you very much.