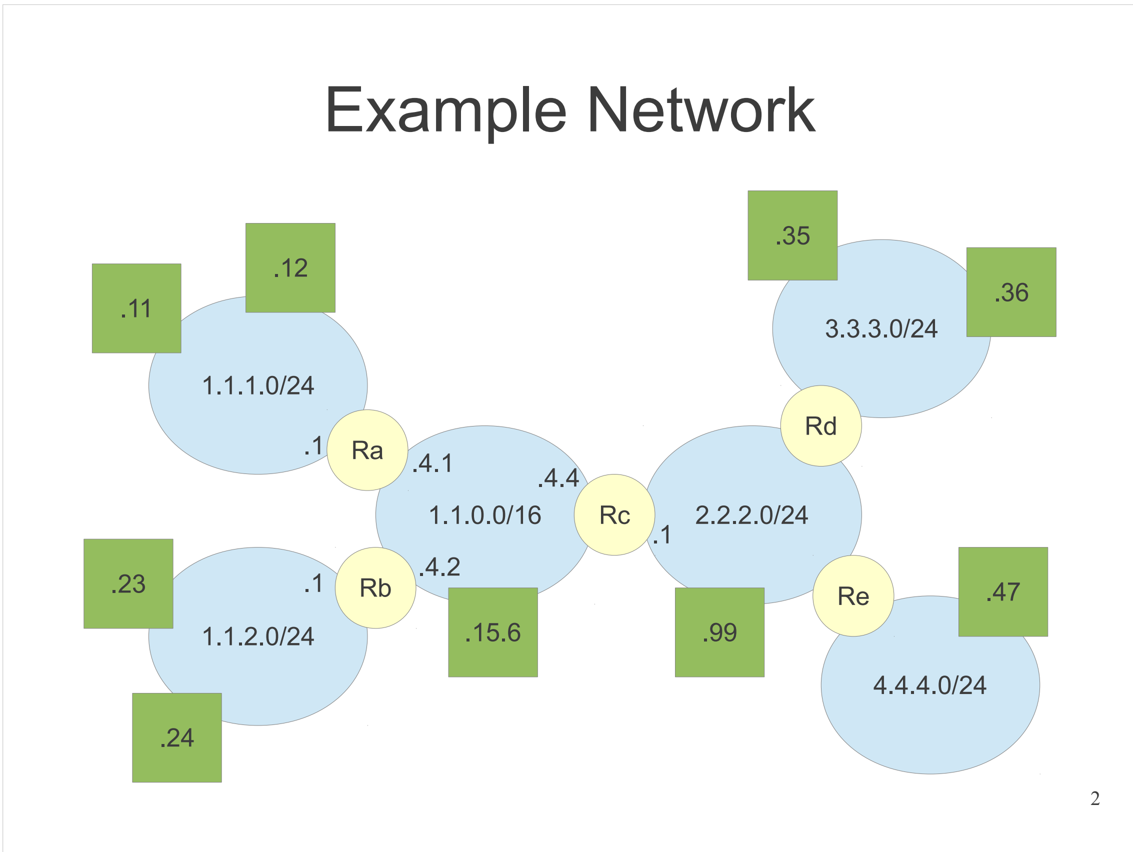
Using a firewall to control traffic in networks

1



Consider this example internet which has:

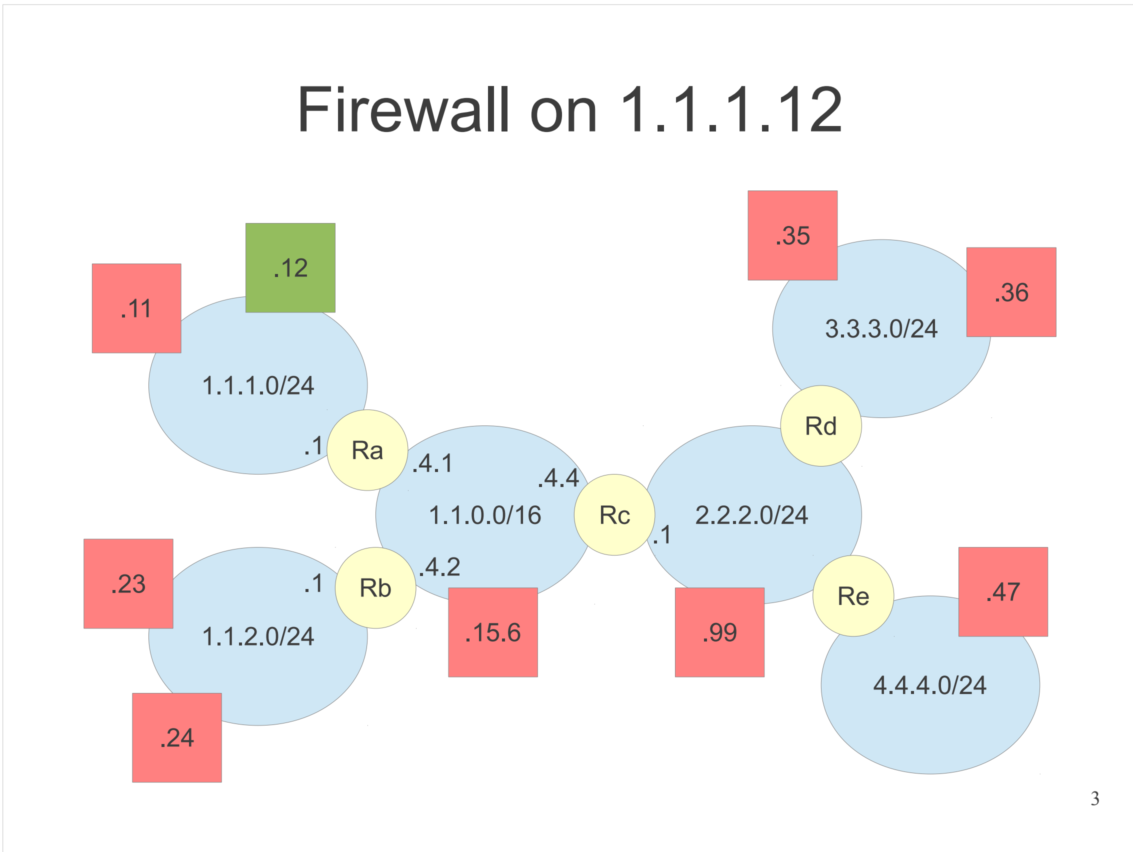
• 6 subnets (blue ovals), each with unique network addresses, e.g. 1.1.1.0/24.

• 5 routers (yellow circles), with names Ra, Rb, … and IP addresses on each interface

- For example, Ra has IP addresses 1.1.1.1/24 and 1.1.4.1/26

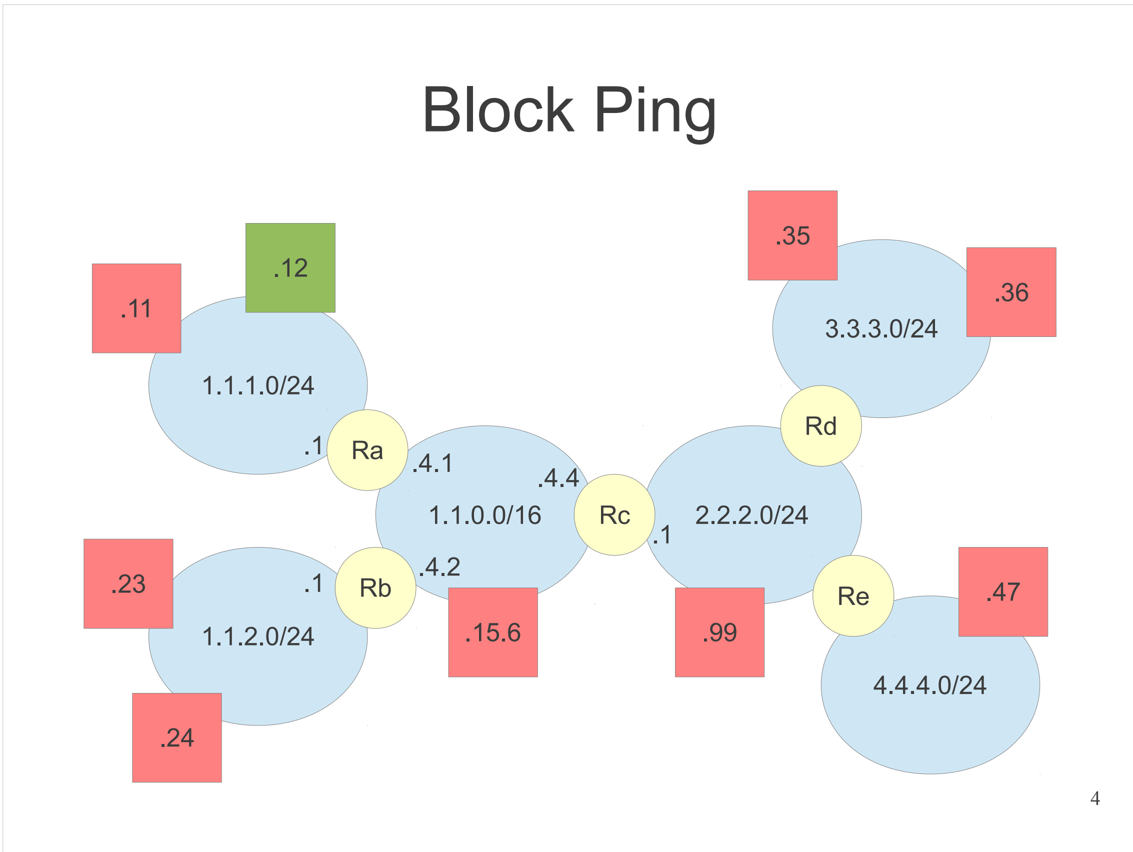
• 9 hosts (green squares), with IP addresses based on network address, e.g. 1.1.1.11/24 and 1.1.1.12/24 in the top left hosts.

Although there are just 9 hosts shown, in some of the following examples we may assume there are many more hosts on each subnet.

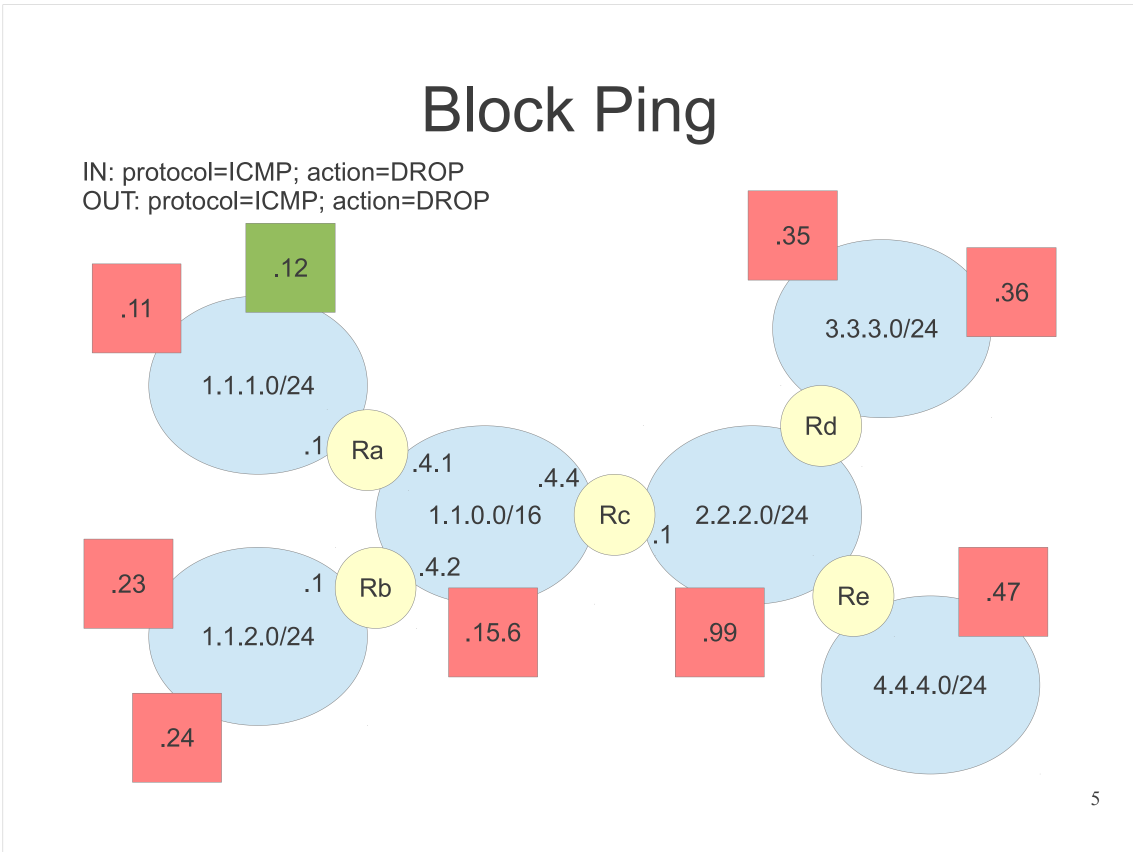


Consider an example of a host-based firewall. Assume 1.1.1.12/24 is your computer and it runs its own firewall. You want to provide some protection by blocking/accepting some traffic.

From your computers perspective, all other hosts are potentially malicious (red).



Lets say you want to block ping from working. Recall that ping uses ICMP: when a computer pings another computer a ICMP Echo request is sent, and ICMP Echo replies are returned. So to stop ping from working, we will need our firewall to block ICMP packets to be sent out of our computer or if ICMP packets are received, block them from being delivered to an application.



We can write the specification of what we want the firewall to do in some strucutured format. On the slide the two rules say:

- For packets coming IN to the computer, if the protocol is ICMP then DROP the packet
- For packets going OUT of the computer, if the protocol is ICMP then DROP the packet

We will see that most firewall software uses rules using such conditional statements: if a packet matches some conditions, take some action.

Firewall contains rules

- Each packet is checked against firewall rules
- If conditions in rule are true then perform action on that packet (eg. DROP, ACCEPT)
- If no rules match, then perform default action
- Multiple rules are combined to create a table

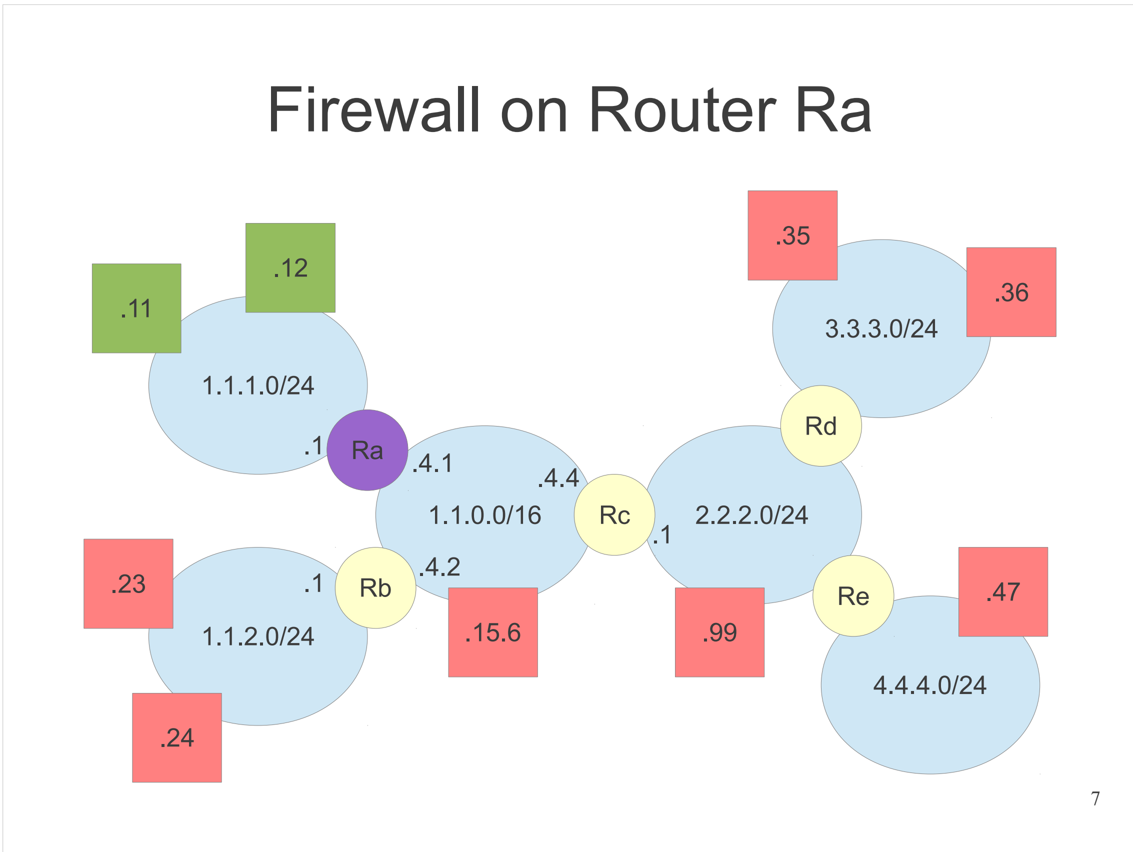
6

In simple terms, a firewall checks each packet that passes through it against a set of rules. The rules are created by the administrator of the firewall (you).

Rules are made up of conditions and actions. If the conditions are true for that packet, then the action is taken.

Normally rules are processed one-by-one, in order. If a packet matches all conditions, then normally the action is taken and no further rules are considered for that packet. If a packet doesn't match the rule conditions, then the next rule is checked. If the packet doesn't match any of the rules in the firewall, there should be some default action to take.

The set of rules for a firewall can be considered as a table: packets are checked row-by-row. We will see with different firewall software, there may be multiple tables for different purposes.



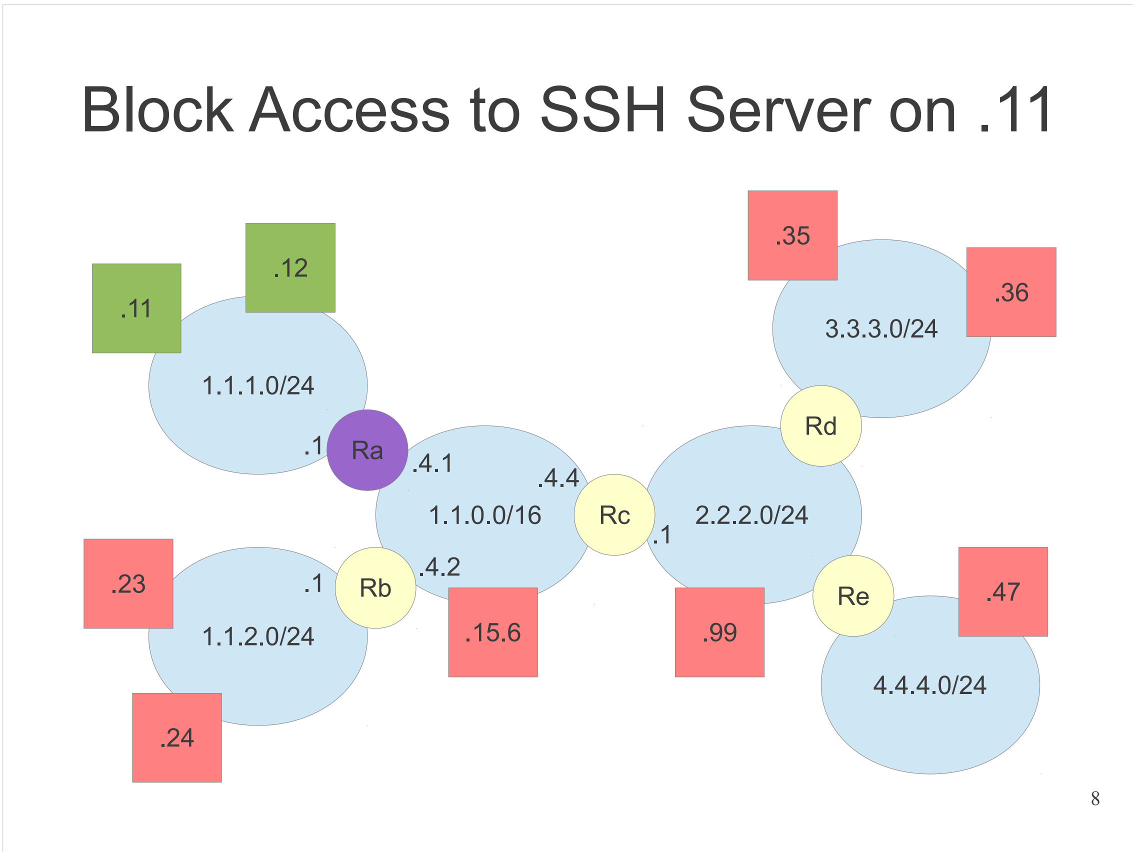
Now consider a different scenario. In the first example, the firewall was running on a host. But that means for an organisation (such as a company or SIIT), each host must run a firewall and the rules on all hosts must be configured and maintained. It is much easier to instead run a firewall on a router operated by the organisation where all traffic from the hosts pass through that router. Running a firewall on a router is very common for organisations; running firewalls on hosts is more common for home users.

In this example, assume the organisation owns the network 1.1.1.0/24, including the two hosts and router Ra. To control traffic going to/from the hosts, we will run a firewall on router Ra (purple). Now the network administrator can configure rules on just one firewall to implement the security policies for the entire organisation.

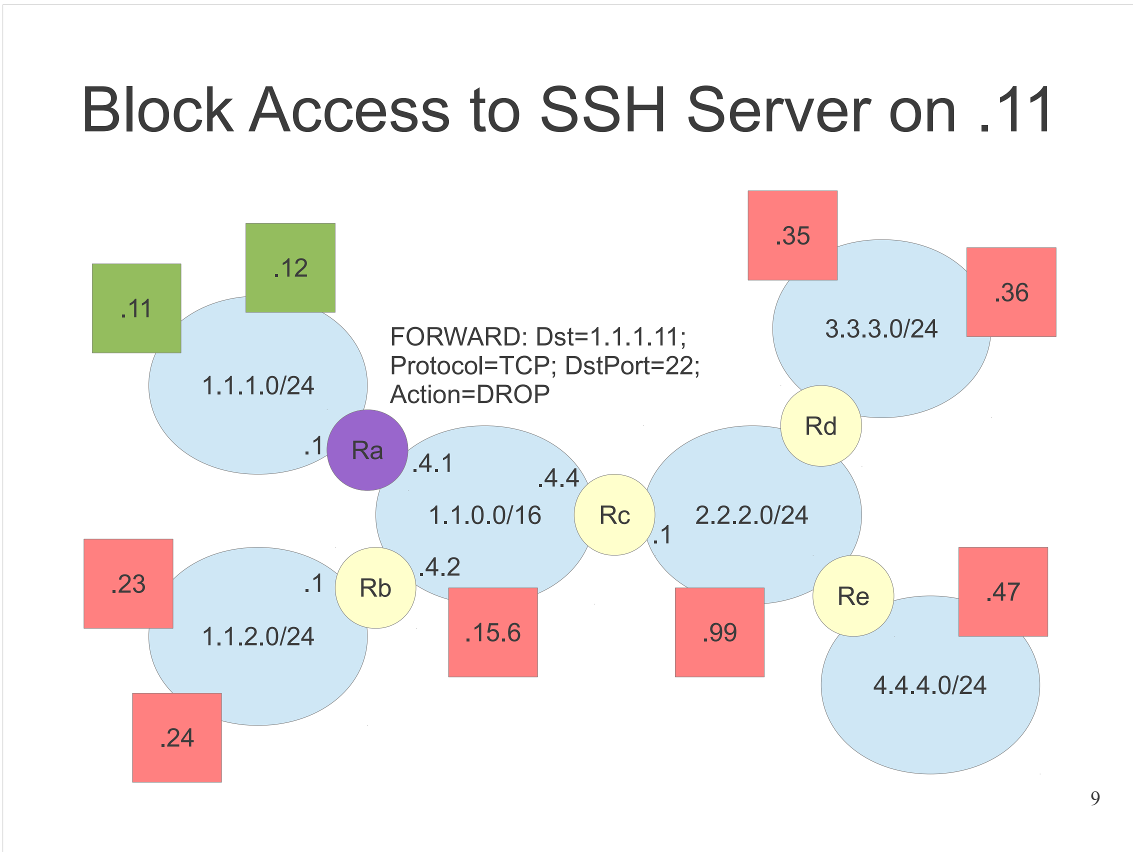
We can thing as devices on the organisations network 1.1.1.0/24 as *internal* (green), while all devices on other networks are *external* (red). There are two types of policies the organisation may implement:

• Stop external devices from accessing internal resources, e.g. stop computers on the Internet from accessing an internal web server.

• Stop internal users from access external resources, e.g. stop employees from accessing Facebook.



Assume hosts 1.1.1.11 runs a SSH server. It is only for other internal hosts to connect to; we don't want any external hosts connecting. Therefore the firewall on Ra should be configured to block external hosts from access the internal SSH server on 1.1.1.11.



The rule on the slide for the firewall says:

If a packet is being forwarded through this router, and the destination is 1.1.1.11, and the transport protocol is TCP, and the destination port is 22, then drop the packet.

Going backwards on the conditions:

• SSH servers by default received packets on port 22; therefore if an external host tries to connect the SSH server on an internal hosts, then the destination port will be 22.

• SSH is an application layer protocol that uses TCP as the transport protocol.

• We want to block access to the SSH server on 1.1.1.11; we don't necessarily want to block access to SSH servers on other internal hosts (like 1.1.1.12).

• Routers normally forward packets: that means the packet is going through the router (the router is not the final destination, nor the original source). We will see (on the next slide) that firewall software can usually have different rules depending on how the packet will be processed (forward, in or out).

Note that this firewall rule blocks packets going to the SSH server. Most Internet applications are client/server based, where the client initiates communications by sending a packet to the server. Therefore to stop the application from working, all we need to do is block that first packet from getting to the server.

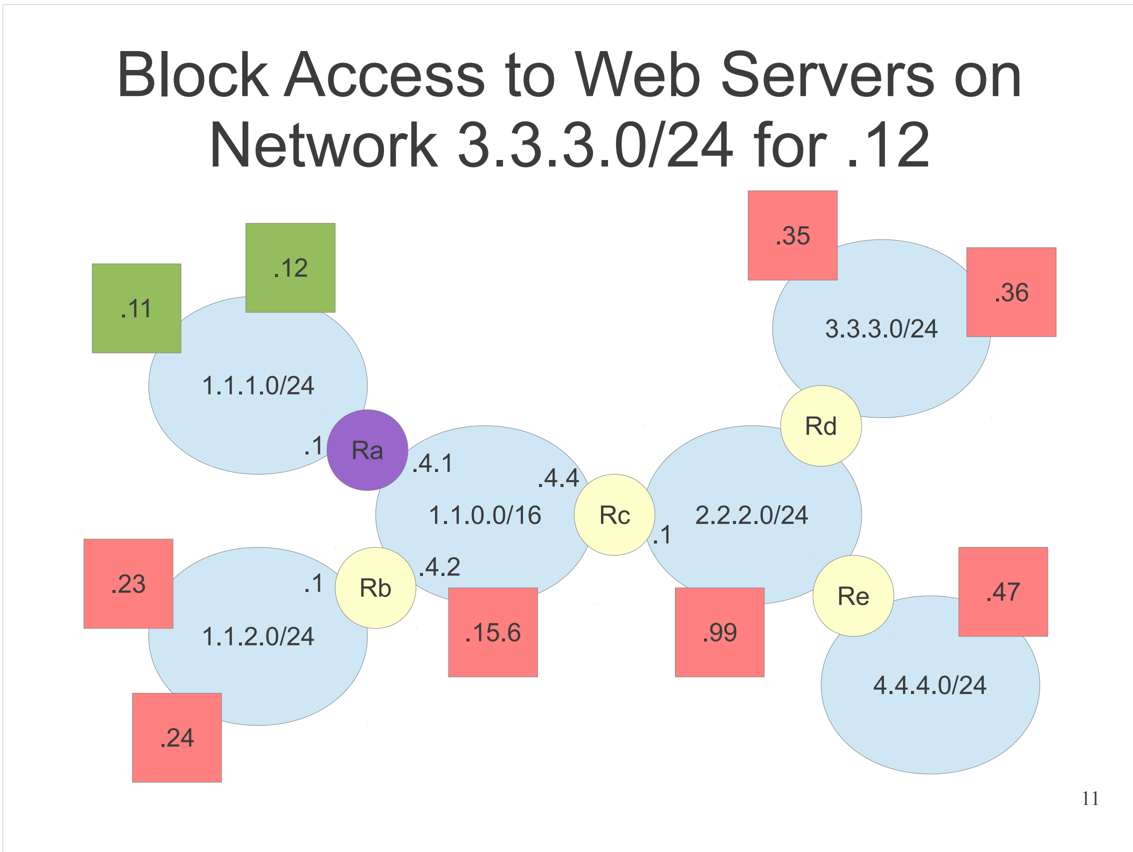
Firewall can have different rules

- INPUT: Applies only to packets destined to this computer
- OUTPUT: Applies only to packets created by this computer
- FORWARD: Applies only to packets going through this computer
- These are called *chains*

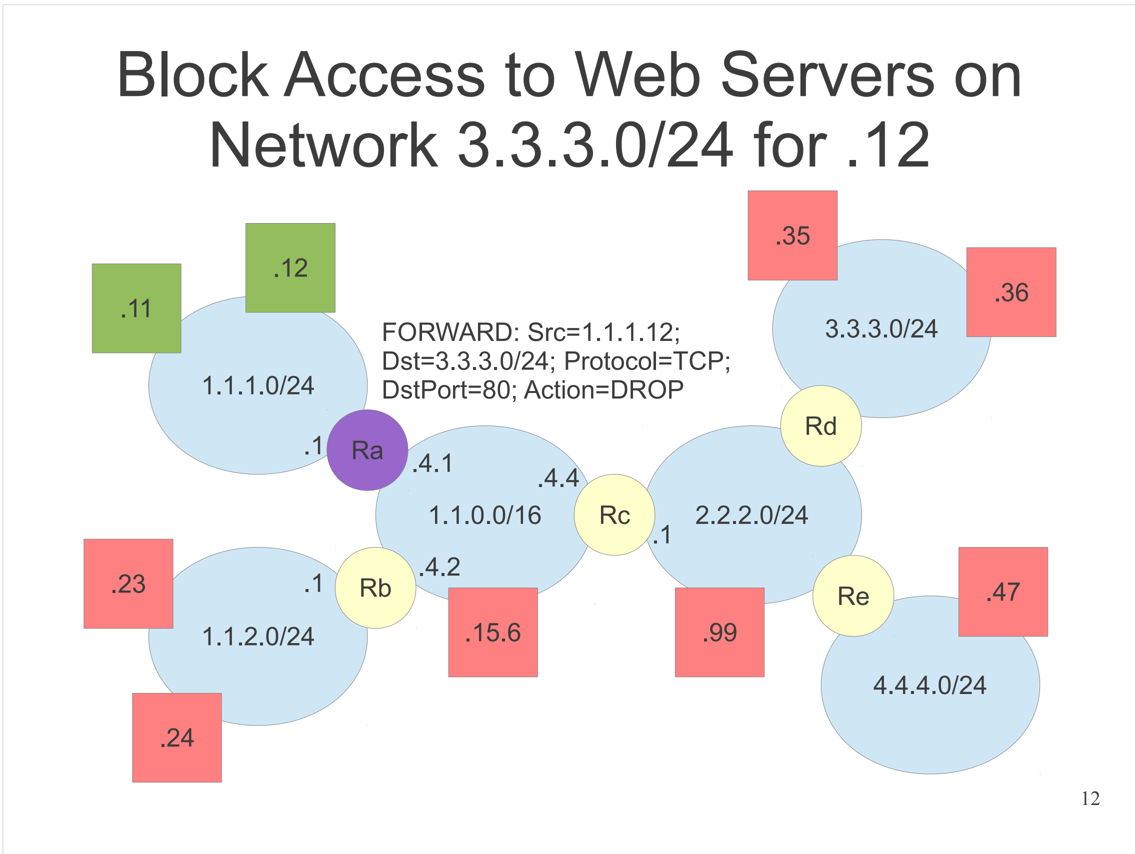
10

iptables is the firewall software used in Linux. It allows separates sets of rules depending on how the packet is processed by the operating system. The three main processing operations, called chains by iptables, are listed above (INPUT, OUTPUT, FORWARD).

This allows us to configure rules in a firewall specific to how it will be processed. Normally, if the firewall is on a host, the INPUT and OUTPUT chains are used. If the firewall is on a router, the FORWARD chain is used (although INPUT and OUTPUT may also be used).



Now lets consider another example with the firewall on the router. Another policy of the organisation is to prevent the internal hosts 1.1.1.12 to access web servers on network 3.3.3.0/24. Why? Maybe 3.3.3.0 is the network for Facebook, and the person using 1.1.1.12 has been wasting their time on Facebook, when they should be working. So the organisation will use the firewall to block access to Facebook for that specific host only.



The rule says:

If a packet is being forwarded through this router, and it is coming from 1.1.1.12, and it is going to any host on network 3.3.3.0/24, and the transport protocol is TCP, and the destination port is 80, then drop the packet.

Again we use the destination port number to identify the application (web browsing). Web servers receive data on port 80 by default, so if a web browser on 1.1.1.12 sends a HTTP request to a web server on network 3.3.3.0/24, then the destination port will be 80.

Firewall Rules Viewed as Table

Firewall table for FORWARD:

Rule	Source	Dest.	Protocol	Action
1	*	1.1.1.11:22	TCP	DROP
2	1.1.1.12:*	3.3.3.0/24:80	TCP	DROP
Default	*	*	*	ACCEPT

When packet arrives at firewall, rules are checked row-by-row. If a rule matches, the ACTION is taken and no further rules are checked.

Separate tables for INPUT, OUTPUT and FORWARD chains.

13

Now look at our firewall rules on the router, assuming both policies (block access to SSH on 1.1.1.11, and block web access to 3.3.3.0) are to be implemented. The firewall rules are for the FORWARD chain. There are two rules. The above slide summarizes the rules in the form of a table.

As a packet comes into the router, and the operating system determines it is a packet to be forwarded, it is passed to the firewall. The firewall checks the packet against the FORWARD rules above. If a rule matches, the action is taken. If a rule does not match, then the next rule in the table is checked.

The syntax of * (wildcard) is used to indicate any value. For example, rule 1 says if the source of the packet is any value. Also, for convenience, the IP address and port numbers are combined, separated by a : (colon).

Note that the last (3rd) rule in the table we did not create in the previous examples. It is a default action to take if no other rules match. In this example, if the first 2 rules do not match, then by default the packet will be accepted.

Most concepts demonstrated through these examples are common to different firewall software/hardware. However, firewall software may implement them differently. In this lab the firewall software used is called iptables – it is the main firewall for Linux operating systems. Other handouts will show how to implement firewall rules using iptables.