



Practical Network Defense

Master's degree in Cybersecurity 2020-21

Networking 101 lab

Angelo Spognardi

spognardi@di.uniroma1.it

*Dipartimento di Informatica
Sapienza Università di Roma*

Main tasks

- Properly configure the topology provided in the lab packages
- Manual configuration
 - Via ip and via interfaces file
- Automatic configuration
 - Via DHCP
- Network debug
 - Fix configuration errors
- Reference link:
<https://www.debian.org/doc/manuals/debian-reference/ch05.en.html>

Assigning IP addresses

- Blocks of public addresses are allocated by IANA (Internet Assigned Numbers Authority) and RIRs (regional Internet registries)
 - To companies, institutions, universities and so on
- Private addresses can be used by anyone
 - Manually
 - Automatically (via DHCP Dynamic Host Configuration Protocol)

To do the activities

- We will use Kathará (formerly known as netkit)
 - A container-based framework for experimenting computer networking: <http://www.kathara.org/>
- A virtual machine is made ready for you
 - <https://drive.google.com/open?id=15WlXlTWXQnZuXEdYk2WSM5KLLFa9Fqx>
- For not-Cybersecurity students, please have a look at the Network Infrastructure Lab material
 - http://stud.netgroup.uniroma2.it/~marcos/network_infrastructures/current/cyber/
 - Instructions are for netkit, we will use kathara

The kathara VM

- It should work in both Virtualbox and VMware
- It should work in Linux, Windows and MacOS
- There are some alias (shortcuts) prepared for you
 - Check with `alias`
- All the exercises can be found in the git repository:
 - <https://github.com/vitome/pnd-labs.git>
- You can move in the directory and run `lstart`
 - **NOTE:** the first `lstart` attempt can (...will...) fail

Katharà main commands (aliases)

- All the commands should be used in the lab directory
- Start (restart) a lab exercise:
 - `lstart/lrestart`
- Stop a running lab exercise
 - `lclean`
- Wipe the kathara environment (when labs do not restart after a failure)
 - `kwipe`
- List virtual networks and virtual interfaces of VMs
 - `docker network list`



Lab activity: ex1



Exercise 1: pnd-labs/lab1/ex1

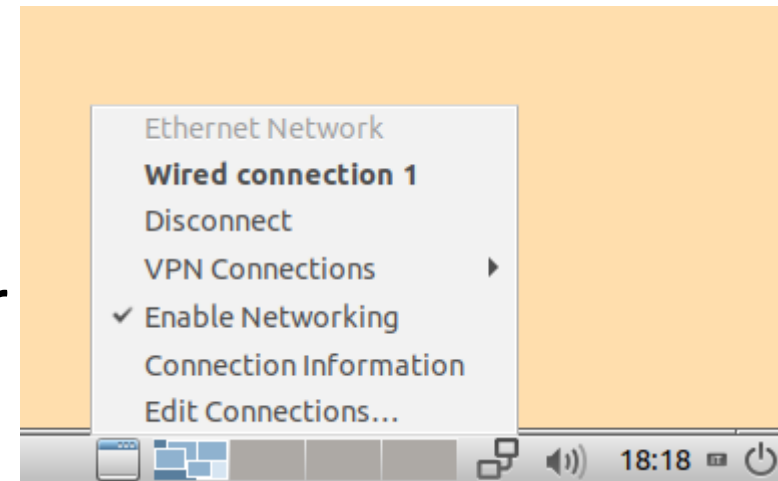
- Manually configure pc1, pc2 and pc3 in order to be in the same network than the r1 host, with IP 192.168.100.30/29
- Configure pc1 using the **interfaces** file (before starting the lab), within the pc1/etc/network/ directory (remember to add the **ifup** command in the startup file)
- Configure pc2 using the **ip** command
- Configure pc3 using the **ifconfig** command
- The DNS server can be the server used by the host machine
 - This should be used also in the **r1.startup** file
- The default gateway must be the r1 host
- Verify connectivity within the network and with the Internet (ex: **ping www.google.com**)

Properly configure a host

- In order to (properly) use Internet a host has to receive 4 main pieces of information
 - The IP address
 - The netmask
 - The IP address of its default gateway
 - Namely the host of its local network able to access to the distribution layer
 - The IP address of a DNS (Domain Name Server)
 - Namely a remote host able to translate human intelligible names to IP addresses

Systems with a GUI

- Use the related tool
- Most used one: network-manager
 - Counterpart from command line: nmcli



```
angelo@lacr:~/teaching/Netdef2020$ nmcli connection show -a
NAME                UUID                                  TYPE  DEVICE
TIM-30962259        8397b8a4-ab2e-45e2-ae07-990a2652cd03  wifi  wlp2s0
tun0                 20cb14f6-bcc0-405a-b4c6-887514e3leec  tun   tun0
angelo@lacr:~/teaching/Netdef2020$
```

- Quite extensible with plugins
 - Example: for managing additional VPN types



Manual network configuration (linux)

(old school – legacy)

- **ifconfig** to assign the IP address
 - This command is used to configure network interfaces, or to display their current configuration. In addition to activating and deactivating interfaces with the “up” and “down” settings
- **route** to define the default gateway
 - The route command is the tool used to display or modify the routing table
- **/etc/resolv.conf** to specify the DNS server(s)



Manual network configuration using ip (preferred)

- **ip addr** to assign the IP address
 - This command is used to configure network interfaces, or to display their current configuration. In addition to activating and deactivating interfaces with the “up” and “down” settings
- **ip route** to define the default gateway
 - The route command is the tool used to display or modify the routing table
- **/etc/resolv.conf** to specify the DNS server(s)

Other details about ip command

- Show interfaces
 - `ip link show`
- Bringing interface up/down
 - `ip link set eth0 (up|down)`
- Set MAC address
 - `ip link set eth0 address 00:11:22:33:44:55`
- Show IP address
 - `ip address show [dev eth0]`
- Add/remove IP address
 - `ip address (add|del) 10.0.0.1/8 dev eth0`
- Flush any IP address (remove the assigned address/es)
 - `ip address flush [dev eth0]`

ip for routing purposes

- List/flush routing table
 - `ip route (list|flush)`
- Add/del routes
 - next hop
 - `ip route (add|del) 10.0.0.0/8 via 10.0.0.1`
 - default
 - `ip route (add|del) default via 10.0.0.1`
 - direct forwarding
 - `ip route (add|del) 10.0.0.0/24 dev eth0`

ip for ARP and more

- Show ARP cache
 - `ip neigh show [dev eth0]`
- Flush ARP cache
 - `ip neigh flush dev eth0`
- Add/del/change/replace ARP cache entry
 - `ip neigh (add|del|change|replace) to 10.0.0.2 lladdr 00:11:22:33:44:55 dev eth0 nud "state_name"`
 - (state_name: permanent, stale, noarp, reachable...)
- IP tunneling (IPinIP, IPinGRE, IPv6 tunneling)
 - `ip tunnel`



Lab activity: ex2

Exercise 2: pnd-labs/lab1/ex2

- Configure r1, pc1 and pc2 in order to receive their networking configuration from a DHCP server (r1)
 - The DNS server can be the server used by the host machine
 - The default gateway must be the r1 machine, with IP address 192.168.100.30/29
- Configure r1 in order to operate as a DHCP server on the eth1 interface
 - You can install **udhcpd** or any other server
 - (apt install udhcpd)
- Configure pc1 using the **interfaces** file
- Configure pc2 using the **dhclient** command (after run)
- Verify connectivity within the network and with the Internet (ex: ping www.google.com)

DHCP

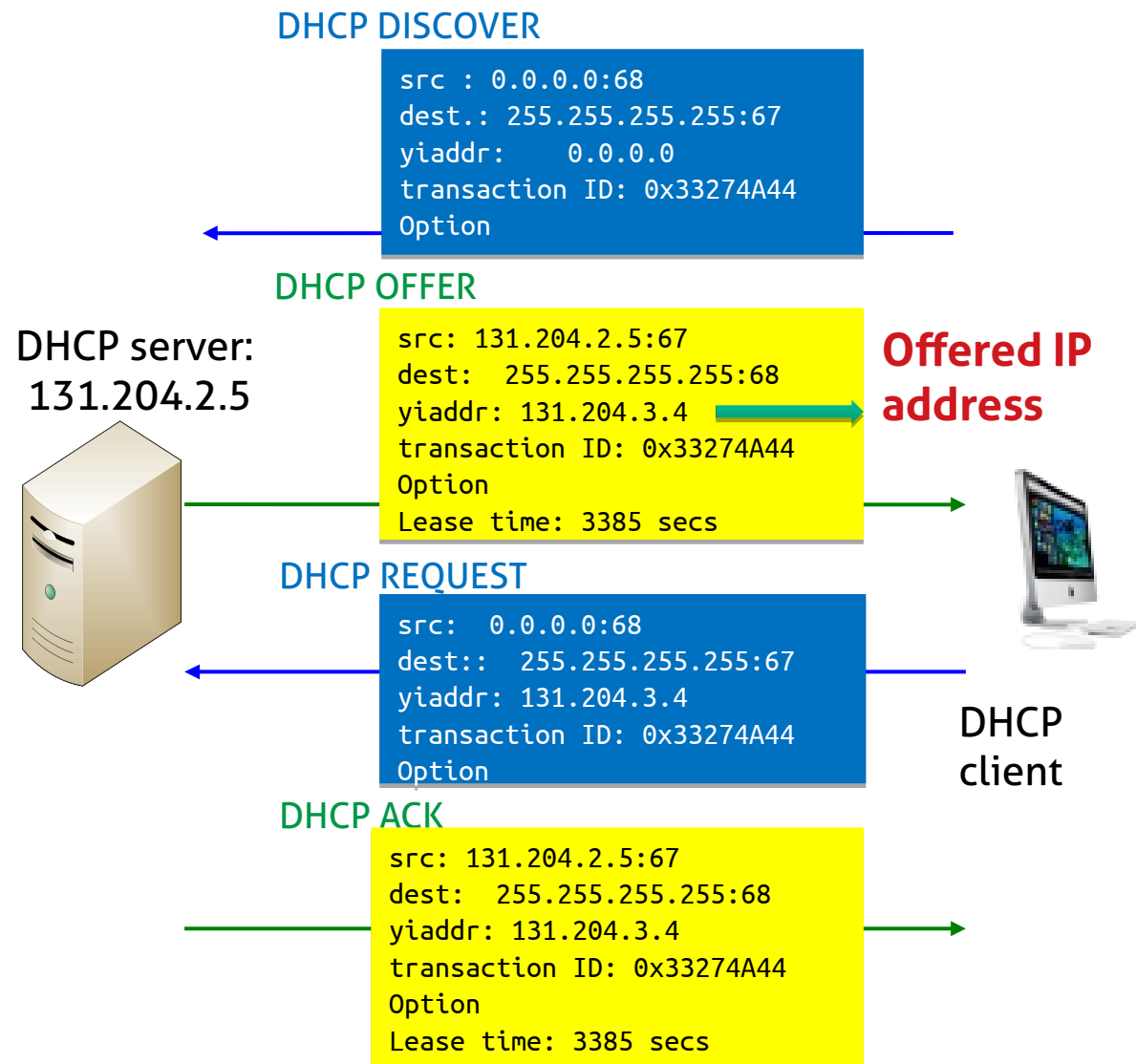
- Client-server mechanism
- Server has a pool of IP addresses to distribute, together with the network configuration
- Client requesting a new IP address receive a proposal and accept it
- Once accepted, the IP is reserved for a “leasing time”
- Observations?

DHCP Client/Server

DHCP procedure:

1. Host broadcasts "DHCP Discover"
2. DHCP server responds with "DHCP Offer"
3. Host requests IP address: "DHCP Request"
4. DHCP server sends address: "DHCP ACK"

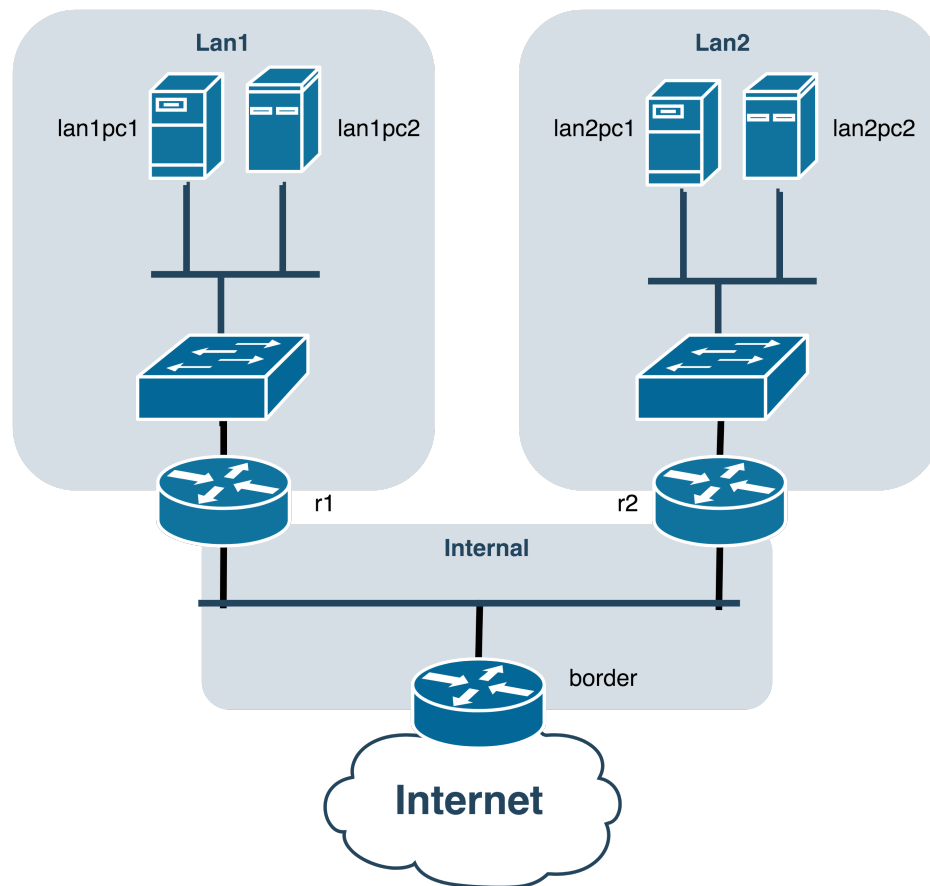
yiaddr: (offered) your IP address





Lab activity: ex3

Exercise 3: pnd-labs/lab1/ex3



- The border r0 is already configured to act as the gateway.
- eth0 on r0 has not to be configured.
- PC from the two lans have to be able to reach each other and to reach internet

ex3 activity

- Configure the 4 pc and the three routers so that the two lans are reachable and all can reach the Internet
 - You have to use the 172.16.0.0/16 network and assign subnetworks to all the LANs in the topology. Think about the most suitable approach.
 - r0 has to be the default gateway of the whole network. It is already set up to act as the default gateway. It is connected to the internet via eth0.
 - r1 and r2 have to be the default gateways for "lan1" and "lan2", respectively. They have to have a default route towards r0 and static routes to reach lan1 or lan2.
 - the DNS server can be the server used by the host machine (this has to be set in all the pcs of the lab)
 - the PCs can be configured as you prefer



That's all for today

- **Questions?**
- See you next on Monday!