Laboratory works no. 3

Reconnaissance phase - resource & infrastructure Enumeration

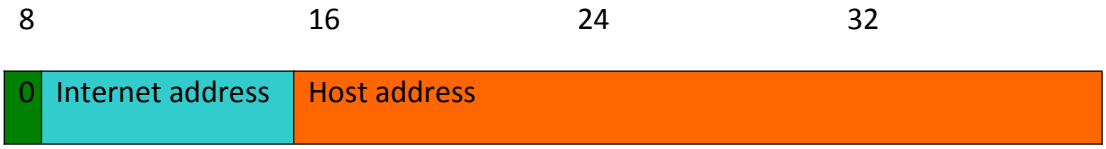
(footprinting)

1. **Reminder** addressing in the Internet From the category "friends know why ..." I have come to the conclusion that a brief recapitulation of some precursor concepts for this subject should be made. The ones that already knows well the subject can jump over this section.

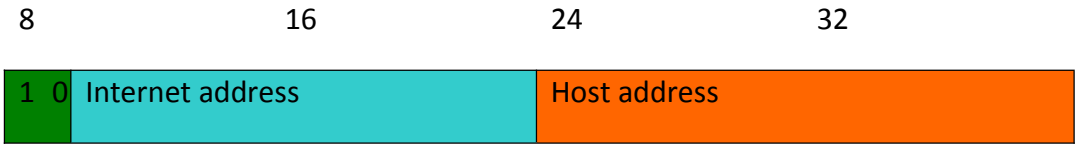
1.1. Enumerating network stations

Identifying hosts on the Internet is done by booking an address for each one, called an Internet address or IP address. IP addresses have a length of 32 bits that is divided into a network part and part of the host. This division led to the formation of five address classes.

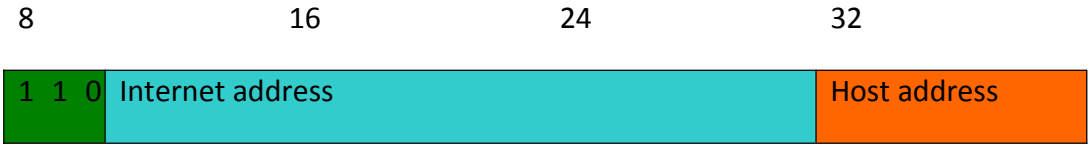
Class A has 7 bits for the network address and 24 bits for the host address. The most significant bit is 0. 128 Class A networks are possible.



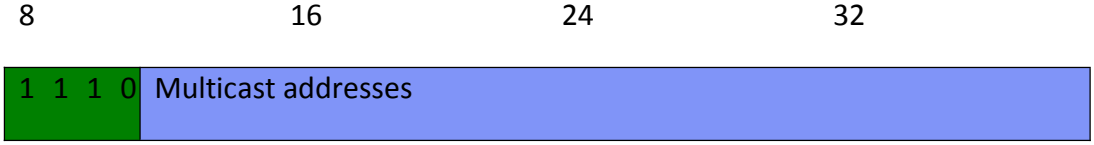
Class B has 14 bits for the network address and 16 bits for the host address. The most significant bits are 10. 16,384 Class B networks are possible



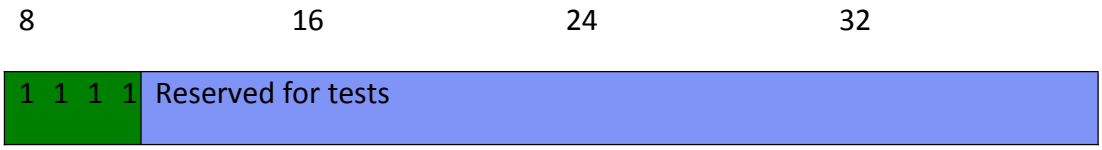
Class C has 21 bits for the network address and 8 bits for the host address. The most significant 3 bits are 110. 2,097,152 Class C networks are possible.



Class D is used for multicast addresses. The most significant 4 bits are 1110.



Class E is reserved for experiments. The most significant 4 bits are 1111.



A common notation for IP addresses divides the 32-bit into four groups of 8 bits, and each group is specified by the decimal numeric correspondent, each group being separated by the point. This is called decimal point notation.

Special Addresses

Address where all host bits are 0 is the network address.

The address where all host bits are 1 is the routing address.

Address 0.0.0.0 represents all networks.

Address 127.0.0.1 is the address of the local loop interface.

Address 255.255.255.255 is the general routing address

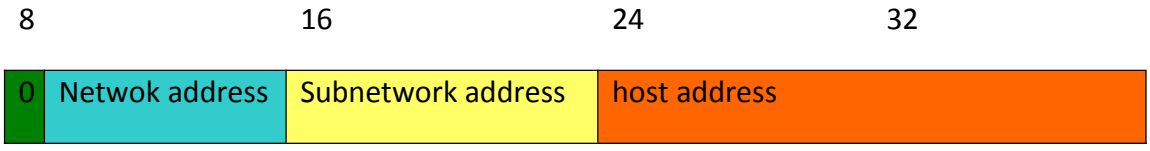
1.2 Sub-networks (subnets)

For technical or administrative reasons, many organisations have chosen to divide a network into several subnets. In these circumstances, an Internet address can be interpreted as follows:

<network address><subnetwork address><host address>

where*<host address>* it has at least one bit, *<subnetwork address>* has a constant length for a given network, and *<network address>* is the network address corresponding to classes A, B or C. If the field length is 0 then the network is not divided into Subnets.

An example of dividing a Class A network with a subnet field length of 8 would look like this:



For reasons of simplicity and effective implementation, it was considered at one time that most organizations would use a multiple subnet field length of 8. However, an implementation that supported variable length had to be prepared. To support Subnets, a further 32-bit amount called a network mask must be used. This is a bit mask with bits set in the fields corresponding to the network address and subnet. For example, on an 8-bit class A network for the subnet field, the network mask would be 255.255.0.0.

1.2. Subnetwork & routing

In the absence of subnets, there are only two possible types of routing in THE IP:

- routing to all hosts of a specific network or
- routing to all hosts on any network.

The last type of routing is used when a host does not know which network it is on.

- When Subnets are used, the situation becomes a little more complicated. First, there is the possibility of routing to a specific subnet. Second, routing to all hosts on a network divided into Subnets requires additional mechanisms. Finally, routing to all hosts on any network should be interpreted as routing only within the original network. Deployments must therefore recognize three types of routing addresses, in addition to their own host addresses:

- Physical network – a destination address with all bits of 1 (255.255.255.255) means a datagram transmitted by routing into the physical network; this datagram must not be submitted by any gate;
- Specific network – the destination address contains a valid network address; host address has all 1 bits (e.g. 36.255.255.255);
- The specific subnet – destination address contains a valid network address and a valid subnet address where the host address has all 1 bits (for example: 36.40.255.255).

1.3 addresses reservation for private networks

Hosts in organizations that use IP can be divided into three categories:

1. hosts that do not require access to hosts from other organizations or to the Internet;

2. hosts that require access to a limited set of services (e-mail, FTP, etc.) that can be provided by gates at the application level;

3. hosts that require network-level access outside the organization;

Hosts in the first category can use IP addresses that are not ambiguous within the organization but can be ambiguous between organizations. For many category two hosts, unrestricted external access may be unwanted for security reasons. Only third-category hosts require IP addresses that are not globally ambiguous. Many apps require connectivity only within your organization and don't require external connectivity for most internal hosts. In very large organizations it is often easy to identify a substantial number of hosts that use TCP/IP and do not need external connectivity. The Internet Number Assignment Authority (IANA) has reserved the following three blocks of IP address space for private networks :

10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

Use the first block terminology as the "24-bit block", the second as the "20-bit block" and the third as the "16-bit block". The first block is a single Class A number while the second block is a set of 16 continuous Class B numbers and the third block is a set of 255 continuous Class C numbers.
An organization that decides to use IP addresses from the address space defined in this document can do so without any coordination with IANA or other attribution authorities on the Internet. the address space can therefore be used by several organizations.
Addresses in the private address space will only be unique within that organization. Any organization that needs unique addresses globally must obtain them from an attribution authority on the Internet. An organization that requires IP addresses for external connectivity will never be assigned addresses from the private address space.
To use private addresses, an organization must determine which hosts do not need external connectivity in the near future. Such hosts will be called private hosts and will use the private address space defined in this document.
Private hosts can communicate with all other hosts within the organization, public or private. In any case they may not have IP connectivity with external hosts.
All other hosts will be publicly named and will use unique global addresses. Public hosts can communicate with other hosts within the organization, be they public or private, and may have IP connectivity with external public hosts.
Public hosts may not have connectivity with private hosts belonging to other organisations. Changing a host from public to private or vice versus involves a change of IP address. Because private addresses are of no global significance, routing information about private networks will not have to be propagated on links between organizations, and packages with the source address and private destination address should not be switched to such links.
Network routers that do not use private address space, in particular those of Internet providers, must be configured to reject (filter) routing information about private networks. If a router receives such information, its rejection should not be treated as a routing protocol error. Indirect references to such addresses should be contained within the organisation.
Obvious examples of such references are DNS resource records and other information related to internal private addresses. In particular, ISPs must take steps to prevent leaks of this kind.

2. *IP v 6*

IP version 6 (IPv6) is the successor to version 4 (IPv4). Changes from IPv4 to IPv6 can be divided into the following categories:

1. Possibilities for expanding the address space - IPv6 increases the ip size from 32-bit to 128-bit to support multiple levels of address hierarchy, a larger number of addressable nodes, and simple address self-configuration.

2. Simplify Header Format - Several IPv4 header fields have been removed or made optional to reduce header processing costs and limit bandwidth costs.

3. Improved support for extensions and options - Changes in IP header encoding mode allow for more efficient advancement, fewer limits on options length, and more flexibility to introduce new options in the future.

4. Possibility of traffic labeling - A new possibility is added to allow the labeling of packages belonging to particular traffic flows for which the transmitter requires special treatment, such as the quality of non-default services or real-time services.

5. Authentication and Privacy Possibilities - Extensions have been specified in IPv6 to support authentication, data integrity, and (optional) data privacy.

2.1. IPv6 addressing

IPv6 addresses are 128-bit identifiers for interfaces and interface sets. There are three types of addresses:

Unicast – an identifier for a single interface. A package sent to a unicast address is delivered to the interface identified by that address.

Anycast – an identifier for a set of interfaces (typically belonging to different nodes). A package sent to an anycast address is delivered to one of the interfaces identified by that address (the nearest, consistent with the distance measured by the routing protocol)

Multicast – an identifier for a set of interfaces (typically belonging to different nodes). A package sent to a multicast address is delivered to all interfaces identified by that address.

There are no broadcast addresses in IPv6, their functionality being replaced by multicast addresses.

All zero and one values are legal for any field, unless they are specifically excluded. IPv6 addresses of all types are assigned to interfaces, not nodes. Since each interface belongs to a single node, any of the uniquest addresses of that node interfaces can be used as an identifier for that node.

A unicast IPv6 address refers to a single interface.

There are three conventional forms for representing IPv6 addresses as text strings:

1. The preferred form is x:x:x:x:x:x:x,x,x," where each "x" represents a 16-bit hexadecimal value.

   Examples:

   FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

   1080:0:0:0:8:800:200C:417A

   Note that it is not necessary to write zeros that prefix an individual field, but there must be at least one numeric value in each field (except in the case described in point 2).

   2. Due to the method of booking some IPv6 address styles, it will be normal for addresses to contain long strings of bits of zero. To make writing addresses that contain

bits of zero easier, a special syntax is available to compress zeros. The sign "::" indicates multiple groups of 16 bits containing zeros. This sign can only appear once in an address. It can also be used to compress start or end zeros in an address.

For example, the following addresses:

1080:0:0:0:8:800:200C:417A – unicast address

FF01:0:0:0:0:0:0:43 – multicast address

0:0:0:0:0:0:0:1 – loopback address

0:0:0:0:0:0:0:0 – unspecified address

Unspecified/unknown address

can be represented as :

1080::8:800:200C:417A

FF01::43

::1

::

3. An alternative form that is sometimes more convenient when dealing with a mixed environment containing IPv4 and IPv6 nodes is x:x:x:x:x:x:d.d.d.d, where "x" represents the hexadecimal values of the first 6 16-bit fields and "d" represents the decimal values of the last 4 8-bit fields.

Examples:

0:0:0:0:0:0:13.1.68.3

0:0:0:0:0:FFFF:129.144.52.38

Or in compressed form:

::13.1.68.3

::FFFF:129.144.52.38

The specific type of an IPv6 address is indicated by its first bits. The variable length field that includes these bits is called a format prefix.

Lab job

To start playing with arping by Thomas Habets, (you must bring it, compile it and install it on the lab stations. Because I saw the lab last month that the nmap documentation seems to be too bushy you have a summary and some examples to make sure you can use it for RedBlue.

3. NMAP

Nmap ("Network Mapper") is used for network exploration and security audit Everything is not an option (or its argument) in the command line of Nmap is treated as a specification of a target. The simplest case is to specify the IP address or name of the computer to be scanned. If you want to fully scan a network for adjacent computers. For example, 192.168.10.0/24 will scan the 256 computers between

192.168.10.0 and *192.168.10.255*

11000000 10101000 00001010 00000000 and 11000000 10101000 00001010 11111111

Given the name www.xxx.xxx with IP address 205.217.153.62, the www.xxx.xxx/16 specification will scan the 65,536 IP addresses between 205.217.0.0 and 205.217.255.255. The lowest allowed value is /1, which will scan half of the IPv4 addresses on the Internet. The largest is 32, which will only scan the computer specified by name or IP because all bits are fixed.

-iL <input file>(taking from file)

Read target specifications from input file. DHCP server can export a list of 10.000 associated IP's .

-iR <targets number>(choosing random targets)

Targets number instruct nmap to generate random targets acordingly

-exclude<host1[,host2][,host3],...>(in this case the explicitly given (as parameters) hosts/networks are excluded from scanning)

-excludefile <exclude_file>(in this case exclusion list is loaded from referred file)

Host Discovery It is often called ping scanning, but it goes far beyond the simple stage of ICMP packets that require a response associated with a ping. Users can skip the step involving ping with a list scan (-sL) or by turning off ping (-P0), or by engaging other arbitrary combinations of TCP SYN/ACK, UDP, and ICMP multiport samples.
The purpose of these tests is to request answers demonstrating whether an IP address is really active (it is used by a device on the network). If host discovery options are not provided, Nmap sends an ACK TCP package for port 80 and an ICMP query package with a response request to each target machine.
An exception to these is that ARP scanning is used for any target on a local network. For underprivileged users with shell accounts on UNIX systems, SYN packages are sent instead of ack ones using the systemconnect() call.

Option -P* (which selects the ping type) can be combined.

It can increase the chances of penetration of strict firewalls by sending more samples using different ports/flags (indicators in packages) TCP and ICMP codes.

The arp (-PR) discovery is executed by default against targets on a local network even if you specify another P* option, as it is almost always faster and more efficient.

-sL(list based scanning)

List based scanning is a degenerate form of host discovery that lists each host of the specified network without sending any packages to targets. by default, however, Nmap performs a reverse DNS resolution to find out the names of the targets. ??? - Useful information can give us the simple names of the hosts

.**-sP(ping based scanning)**

This option tells nmap to perform only a ping scan (host discovery) and then display the available hosts that respond to the scan. This step is more intrusive than list scanning and can often be used for the same purpose. Allows a discovery of network targets without attracting too much attention.

Knowing how many hosts are active is a much more valuable information to an attacker than the mere list provided by the list scan of each IP address and host names. The –sP option sends a response-requested ICMP package and a TCP package to port 80 by default.

When run by an unprivileged user, a SYN package is sent to the target's port 80.

When a privileged user tries to scan targets from a local network, ARP (-PR) requests are used only if is possible

-send-ip(destination IP) was given

-P0(without ping)

This option no longer performs the discovery phase. By default, Nmap performs advanced scanning such as port scanning, version detection, and operating system detection only for hosts found active. Disabling host discovery with -P0 causes Nmap to try advanced scanning techniques for each IP address specified as the target.

-PS [ports list](Ping TCP SYN)

This option sends an empty TCP package with the SYN flag set. The default destination port is 80 but another port can be specified as a parameter. Even a comma-separated list of ports can be specified (for example-PS22,23,25,80,113,1050,35000), in which case the test packages will be sent to each port in parallel.

The SYN flag suggests to the target that we want to establish a connection. normally the destination port will be closed and an RST (reset) package is sent back. If the port happens to be open, the target will take the second step of a three-step protocol) by responding with a SYN/ACK TCP package.

-PA [ports list](Ping TCP ACK)

TCP ACK ping is similar to PING SYN. An ACK package claims to carry data within an ACK connection already established, but there is no such connection. So targets must always respond with an RST package, revealing their existence in this process. The –PA option uses the same default port as SYN samples and can also retrieve a list of destination ports in the same format.

-PU [ports list](Ping UDP)

Another host discovery option is the UDP ping, which sends an empty package (only if the option--data-length is not specified) UDP to the specified port. port list has the same format as previously discussed in options-PS and –PA (default port 31338). After sending the test package to a closed target port, an unavailable ICMP port package must be obtained. This signals the Nmap that the machine is active and available.

-PE;-PP;-PM(various ICMP ping types)

Nmap can send standard ICMP packages of type 8 (response request) to the target IP address, waiting for a type 0 package (response) instead from the available hosts. many hosts and firewalls block these packages, instead of responding according toRFC1122.

The ICMP standard (RFC792) also specifies the timeprint request, information and network mask corresponding to codes 13,15 and 17. Timeprint requests and network mask can be sent using PP and PM options, respectively. A timeprint response (ICMP code 14) or a network mask response (code 18) reveals an available host.

-PR(Ping ARP)

One of Nmap's most common usage scenarios is to scan an entire local network (LAN). in many LANs, especially those that use the private address space specified in RFC1918, the vast majority of IP addresses are not used at any given time. If it receives a response, Nmap doesn't even consider IP-based pings since it already knows the host is active. This makes ARP scanning much faster and more accurate than IP-based scans.

-n(does not use DNS resolution)

Transmits to Nmap that it never achieves the DNS reverse resolution for the active IPs found. Since DNS is often slow, this option can increase scanning speed.

-R(use DNS resolution for all targets)

Transmits to Nmap as always to achieve DNS resolution for target IPs.

Port scanning basis

The target nmap command scans more than 1660 target TCP ports. Nmap divides ports into six states:

1. Open - An application actively accepts TCP connections or UDP packages to that port.
2. Closed - A closed port is accessible (receives and responds to a sample package sent by Nmap), but there is no application that listens to it. They can be useful in revealing the status of the host or as part of the detection of the operating system.
3. Filtered - Nmap cannot determine whether the port is open due to a package filter that prevents packages from reaching the destination port. Filtering can come from a dedicated firewall, the rules of a router, or a target software firewall. Because it provides very little information Nmap will resend test packages several times in case the package has been lost due to network congestion and not because of filtering.
4. Unfiltered - this status means that the port is accessible, but Nmap cannot determine whether the port is closed or open. Only the ACK scan, used to map firewall rules, classifies the port in this state.
5. Open|filtered - Nmap places ports in this category when it cannot determine whether the port is open or filtered. These occur for types of scans in which open ports provide no response.
6. Closed|filtered - This state is used when Nmap is unable to determine whether the port is closed or filtered. It is only used by IPID Idle scanning. These states are not intrinsic properties of ports, but describe how they are seen by Nmap.

Port scanning strategies

Even if Nmap tries to produce the most accurate results, it should be taken into account that it is based on packages returned by the target machine (or the firewall in front of it).

-sS(TCP SYN scanning)

SYN scanning is the default. It can be run quickly by scanning thousands of ports per second on a network without a firewall. SYN scans are relatively invisible, since they never establish a TCP connection.

-sT(TCP connect() scanning)

TCP Connect() scanning is the default when SYN is not a viable option. This is if the user does not benefit from the possibility of sending raw packages or scans IPv6 networks. instead of writing raw packages as most scan types do, Nmap requires lower operating system levels to establish a connection with the target machine and the desired port by making a connect system call().

-sU(UDP scanning)

DNS, SNMP and DHCP (registered ports 53, 161/162 and 67/68) are three of the most common ports. Because UDP scanning is generally slow and more difficult than TCP, some security experts ignore these ports. Closed ports are often an even bigger problem. usually send back an inaccessible ICMP error message. Nmap detects the limit rate and slows the scanning according to it to prevent flooding of the network with unnecessary packages that the target machine will ignore.

-sN;-sF;-sX(TCP Null, FIN, și Xmas scanning)

These three types of scanning exploit a loophole in the RFC TCP to differentiate between open and closed ports. There are three types of scans:

1. Null Scan (-sN): Do not set any bit (tcp header flag is 0)
2. FIN Scan (-sF): Sets only the TCP FIN.
3. Xmas Scan (-sX): Sets FIN, PSH, and URG flags

The main advantage of these types of scanning is that they can sneak through certain non-statefull firewalls and packet filtering routers. Another advantage of these types of scanning is that they are even more discrete than a SYN scan although many modern IDSs can be configured to detect them.

Not all systems comply with RFC 793. Some send an RST response to samples regardless of whether the port is open or not. This causes ports to be marked as closed.

-sA(TCP ACK scan)

This scan is different from the others discussed so far in the sense that it can never determine an open or open port|filtered It is used to check the rules of the firewall.

-sW(TCP Window scan)

Window scanning (window) is similar to ACK scanning, except as it exploits a detail of implementing certain systems to differentiate open ports from closed ports.

-sM(TCP Maimon scan)

The TCP Maimon scan is named after its discoverer, Uriel Maimon 1996). The technique is similar to null, FIN, and Xmas scans except that the sample is FIN/ACK. However, it has been observed that many systems derived from BSD ignore the package if the port is open.

--scanflags(custom TCP scan)

The option allows you to create your own types of scanning by specifying TCP flags. Any combination of URG, ACK, PSH, RST, SYN, andFIN may be used. For example,--scanflags UGACKPSHRSTSYNFIN sets all bits, although it is not very useful for scanning.

Services & versions detection

If nmap is run on a machine and it reports that ports 25/tcp, 80/tcp, and 53/udp are open. Using the datanmap-services database of approximately 2,200 known services, Nmap will report that those ports correspond to a mail server (SMTP), web server (HTTP) and DNS server (53), respectively.
This recognition is usually correct – most of the services that listen to the TCP port 25 are mail servers.

After TCP and/or UDP ports are discovered using one of the scanning methods, version detection queries those ports to determine more about what is actually running on them.
Nmap tries to determine the service protocol (e.g. ftp, ssh, telnet, http), application name (e.g. ISC Bind, Apache httpd, Solaris teletin), version number, host name, machine type (e.g. printer, router), operating system family (e.g. Windows, Linux) and sometimes other information as diverse as if there is an X server open to connections.
When RPC services are discovered, the RPC evaluator in Nmap (-sR) is automatically used to determine the RPC program and version number. Some UDP ports are left in the open|filtered state after a UDP scan that could not determine whether the ports are open or filtered. Version detection will try to get a response from these ports (as it does with open ports), and change their status in open if it succeeds. Open|filtered TCP ports are treated in the same manner. The –A option enables version detection among other things.

Examples

Nmap workstation numeric address

nmap -v www.xxx.xxx

This command line scans all reserved TCP ports on the www.xxx.xxx machine. The –v option enables real-time display of results.

nmap -sS -O www.xxx.xxx/24

Launch a SYN scan against all 255 cars in the C-class network where the host takes its place. It also tries to determine the operating system of each active host. This release requires root privileges because of SYN scanning and operating system detection.

nmap -sV -p 22,53,110,143,4564 198.116.0-255.1-127

Launches host enumeration and a TCP scan In the first half of the 255 198,116 B-class subnets. For each of the ports found open, version detection is put into operation to determine which application is running.

nmap -v -iR 100000 -P0 -p 80

Ask the Nmap to randomly choose 100,000 targets and scan them in Search of web servers (port 80). Host enumeration is disabled with -P0 since sending a couple of samples to determine if the host is active is a waste of time as long as a single port of each target is searched.

nmap -P0 -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap 216.163.128.20/20

Scans for web servers (no pings) and saves results in the Grep command format and XML format.

host -l www.xxx.xxx | cut -d -f 4 | nmap -v -iL -

Performs a DNS zone transfer to find all hosts in the www.xxx.xxx and then provides Nmap addresses. The above command is valid for a Linux system

Lab job

- Test each command with different parameter combinations (when possible). - no copy pasta please (I had “careful” people testing 50 of Internet addresses). It will be a good idea to limit your tests at the laboratory workstation including the one dedicated to be scanned (from my desk).

Homework:

Starting from the previous analysis (theme 2) create a map with the information extracted with the nmap for each station in the graph from the theme 2 Those who do not stay in the dorm disconnect the link with the isp and do analysis on computers in the local network (mother, father, desktop etc.) until they get used then anyway will do on real case in redblue.