PA2Q4

a) How does nmap work? (organizations, scripts, techniques....)

b) Explain how to perform portscan through firewalls (simple state filter).

c) Systematize how to map a network using nmap. What limitations? What can be obtained?

NMAP Tutorial for Beginners // Stealth Scan vs TCP Connect Scan // NMAP -sS -ST (Video link from professor)

Nmap portscan tutorial -> port-scanning-tutorial

What is a Port Scanner and How Does it Work?

---

a) How does nmap work ?

1. **Organizations**
   PT-BR Origin: Wikipédia, a enciclopédia livre.
   EN Origin: Wikipedia, the free encyclopedia
   (The content is similiar but not the same.)

Nmap (Network Mapper) is a -open source- *network scannner* created by Gordon Lyon (also know by his pseudonym Fyodor Vaskovich). Nmap is used to discover **hosts** and **services** on a **computer network** by sending packets and analyzing the responses.
Nmap is also widely used to evaluate computer security, and to discover services or servers on a computer network.

> Nmap is a CUI (Console User Interface) program, so it runs on the command line, but it has a graphical interface (GUI), NmapFE (Nmap Front End), which was replaced by Zenmap on October 11, 2007, as it is a portable version and provides a better interface for execution and especially for viewing and analyzing Nmap results.

- **Original author**: Gordon Lyon (Fyodor)
- **Plataform**: i386
- **Initial release**: September1997
- **Repository**: https://github.com/nmap/nmap.git
- **Written in**: C, C++, Lua, Python (GTK)
- **Operating System**: Windows, Mac OS X, Linux
- **License**: NPSL or modified, GPLv2 or proprietary, open-source
- **Website**: insecure.org/nmap

nmap is a port and network scanning tool focused on protection, it uses protocols such as tcp, udpip, ftp and can perform all these protocols completely or incompletely, according to the user's needs to reduce network noise.
For example: the -sS flag that only sends a Syn connection and waits for a response, without completing the TCP connection, being a silent way of scanning
another flag is -sU which uses the UDP protocol
Furthermore, it has flags such as s0 which uses IP protocol and -b for FTP protocol

For the organization, nmap expects an IP address, or a set of IP addresses to scan, that is, nmap can also scan networks and discover all the addresses there, for this it supports CIDR notation with the number of bits at the end.

To improve scanning, nmap is also able to continue a conversation with the port to detect the software running there and often the current version of it. Through this bit analysis it is sometimes possible to also detect the operating system that is running on the target. But obviously, when these scans are carried out, the exchange of information on your part is also large, generating logs and noise on the network

1. **Scripts**
   Nmap Scripting Engine (NSE)

> The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts (using the Lua programming language) to automate a wide variety of networking tasks. Those scripts are executed in parallel with the speed and effciency you expect from Nmap. Users can rely on the growing and diverse set of scripts distributed with Nmap, or write their own to meet custom needs.
> To reflect those different uses and to simplify the choice of which scripts to run, each script contains a field associating it with one or more categories. Currently defined categories are *auth, broadcast, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version* and *vuln.*
> ***Scripts are not run in a sandbox and thus could accidentally or maliciously damage your system or invade your privacy. Never run scripts from third parties unless you trust the authors or have carefully audited the scripts yourself.***

Script Categories

- **auth**
  The scripts with authentication credentials (or bypassing them) on the target system.
- **broadcast**
  Scripts in this category typically do discovery of hosts not listed on the command line by broadcasting on the local network.
- **brute**
  These scripts use brute force attacks to guess authentication credentials of a remote server.
- **default**
  These scripts are the default set. This category also be specified explicitly like any other.
- **discovery**
  These scripts try to actively discover more about the network by queying public registries, SNMP-enabled devices, directory services, and the like.
- **dos**
  Scripts in this category may cause a denial of service. Sometimes this is done to test vunerability to a denial of service method, but more commenly it is an undesired by necessary side effect of testing for a traditional vunerability. These tests sometimes crash vunerable services.

- **exploit**
  These scripts aim to actively exploit some vunerability.
- **external**
  Scripts in this category may send data to a third-party database or other network resource. Most scripts involve traffic strictly between the scanning computer and the client; any that do not are placed in this category.
- **fuzzer**
  This category contains scripts which are designed to send server software unexpected or randomized fields in each packet.
- **intrusive**
  These are scripts that cannot be classified in the safe category because the risks are too high that they will crash the target system, use up significant resources on the target host (such as bandwidth or CPU time), or otherwise be perceived as malicious by the target's system administrators.
- **malware**
  These scripts test whether the target platform is infected by malware or backdoors.
- **safe**
  Scripts which weren't designed to crash services, use large amounts of network bandwidth or other resources, or exploit security holes are categorized as safe.
- **version**
  The scripts in this special category are an extension to the version detection feature and cannot be slected explicitly. They are selected to run only if version detection was requested. Their output cannot be distinguished from version detection output and they do not produce service or host script results.
- **vuln**
  These scripts check for specific know vunerabilities and generally only report results if they are found.

2. **Techniques**
    1. **Default Nmap Scan:**
       nmap (ip of device) or nmap (subnet of scan)
    2. **OS Enumeration**, what devices are we scanning and what OS its running:
       nmap -o (ip of device) or nmap -o (subnet of scan)
    3. **Ping Scan**, used for network discovery when you don't want to run an actual port scan yet you want to know what device is on the networ:
       nmap -sP (ip of device) or nmap -sP (subnet of scan)
    4. **Scan for a Specific Port**, check if a specific port is open on a device or a network:
       nmap -p (ip of device) or nmap -p (subnet of scan)

5. **Service Version Identification**, look up the version of the server that is running on devices:
   nmap -sV (ip of device) or nmap -sV (subnet of scan)
6. **Scan Using the All Flag**, a combination of all listed above:
   nmap -A (ip of device) or nmap -A (subnet of scan)
7. **Using Nmap Script**, tell Nmap to use a pre-written script:
   nmap -script Script Name (ip of device) or nmap -script Script Name (subnet of scan)
8. # Nmap | Seven Must Know Techniques in Seven Minutes

3. **More**
   Nmap features include:
   - Fast scan
   - Host discovery
   - Port scanning
   - Version detection
   - Ping Scan
   - TCP/IP stack fingerprinting
   - Scriptable interaction with the target
     Typical uses of Nmap:
   - Auditing the security of a device or firewall by identifying the network connection which can be made to, or through it.
   - Identifying open ports on a target host in preparation for auditing.
   - Network iventory, network mapping, maintance and asset management.
   - Auditing the security of a network by identifying new servers.
   - Generating treffic to hosts on a network, response analysis and response time measurement.
   - Finding and exploiting vunerabilities in a network.
   - DNS queries and subdomain search.

b) Explain how to perform portscan through firewalls (simple state filter).

Nmap has standard support for some techniques for some search techniques to cross the firewalls, ome of which are:
-f flag: fragments the packets sent from scan to confuse the firewall.
-D flag: uses the scan between serveral decoys, meaning that the firewall does not know which one is doing the scan and may let the scan be carried out. However, with deeper analysis, the target is able to discover the origin of the attack.
Furthermore, Nmap provides serveral ways to change the packets sent in the desired way,

including source IP, port, and TTL, allowing the user to disguise the packet as they wish. In addition to being able to create proxies to tunnel the connection. In short, it is possible to change the way the scan is performed to perform more specific scans such as IDLE or FTP bounce

c) Systematize how to map a network using nmap. What limitations? What can be obtained?

Scanning with nmap doesn't necessarily imply lots of network traffic, probes against huge port ranges and setting off intrusion detection alerts. You can also use it to quickly, easily and stealthily generate a listing of all sustems on a particular subnet.
To start the process of discovering the servers in a network you ca use a "skip port scan" to quickly outline each subnet that he was about to manage. As the name suggests, this nmap scan doesn't scan ports. Instead, it is just a "ping scan" or "ping sweep".
nmap -sP (ip of device or subnet of scan)
Output example:

```
Starting Nmap 4.60 ( http://insecure.org ) at 2010-02-17 10:01 EST
Host 10.1.2.1 appears to be up.
MAC Address: 00:06:31:7B:48:0C (Cisco Systems)
Host 10.1.2.2 appears to be up.
MAC Address: 00:03:BA:42:DE:49 (Sun Microsystems)
Host 10.1.2.3 appears to be up.
MAC Address: 00:03:BA:55:26:BA (Sun Microsystems)
```

You can see which IP addresses in the subnet are in use and the MAC address of each system.
The output above indicates that the first address in the subnet is a Cisco switch.
Then it moves on the servers and finds some older Sun systems.
By the end of the scan, we have the number of systems and composition of the subnet in terms of architecture.

Another nmpa command that costs virtually nothing in terms of network activity and intrusiveness is the list scan. This scan uses DNS to flesh out a network and doesn't send any packets to the system. Thus, it provides another way of finding out what your name server thinks is on the subnet. In this type of scan, nmap uses reverse lookups to populate system names and doesn't go any further is determining wheter the system is running or even present.

nmap -sL (ip of device or subnet of scan)
Output example:

```
Starting Nmap 4.60 ( http://insecure.org ) at 2010-02-17 10:54 EST
Host server1 (10.1.2.1) not scanned
Host 10.1.2.2 not scanned
Host 10.1.2.3 not scanned
Host server4 (10.1.2.4) not scanned
Host server5 (10.1.2.5) not scanned
Host server6 (10.1.2.6) not scanned
```

Due to the "no impact" nature of these scans, particular the list scan, you needn't be concerned that your gentle probing of network space is going to register as a problem. This is not true of more rigorous and comprehensive types of scans.
For more instrusive scanning of any network, you should always habe permission and that the responsible for managing the networks you are scanning is well aware of your activity.
Your are likely to set off alarms or get someone's attention when your port scan.
source