Taller 3: Port Scanning y DNS

Teoría de las Comunicaciones
Departamento de Computación
FCEN - UBA

01.06.2021

1. Objetivos

Implementar estrategias para detectar el estado de los puertos en un sistema operativo. Probar la herramienta `nmap` e implementar métodos similares para entender cómo funcionan los protocolos de transporte más usados del modelo de Internet: TCP[1] y UDP[2]. En la parte optativa se explorará la resolución de nombres con el protoclo DNS [3] [4].

2. Normativa

- Fecha de entrega: hasta el 15/06/2021.
- El trabajo práctico se deberá enviar por correo electrónico con el siguiente formato:
  to: tdc-doc at dc uba ar
  subject: debe tener el prefijo [tdc-nmap]
  body: nombres de los integrantes y las respectivas direcciones de correo electrónico
  attachments: el informe en formato pdf + el código fuente en formato zip.

3. Enunciado

3.1. Introducción

3.1.1. *Port Scanning* y `nmap`

Port scanning (escaneo de puertos) es el nombre que se le da a un conjunto de técnicas y métodos que sirven para obtener información acerca de un *host* o de un *router* conectado a la red. Ejecutar un escaneo de puertos en una red o en un servidor revela qué puertos están abiertos y escuchando (listos para recibir información) y también revela la presencia (o no) de firewalls entre el emisor y el blanco. Esto en sí no representa un proceso malicioso; de hecho, la mayoría de los *port scans* se usan simplemente para determinar qué servicios se encuentran disponibles en un host remoto. El uso de estas técnicas es muy útil para detectar vulnerabilidades en la seguridad de una red. Debido a esta funcionalidad, también es usado por atacantes como una herramienta de reconocimiento para buscar puntos débiles que permitan acceder a un sistema y comprometer la información disponible en el mismo.

`nmap`[5] ("Network Mapper") es una de las herramientas de port scanning más conocidas disponibles. Es una utilidad libre y de código abierto para exploración de redes y realización de auditorías de seguridad. Es útil, por nombrar algunos usos, para monitorear el estado de una red, comprobar los servicios que se están prestando y tomar mediciones de hace cuánto un servicio o host se encuentran encendidos (*uptime*). `nmap` usa el stack de TCP/IP en formas novedosas para determinar qué hosts se encuentran disponibles en una red, qué servicios brindan (nombres de aplicación y versión), qué sistemas operativos están corriendo (*OS fingerprinting*), qué tipo de firewalls o filtros están en uso y otras muchas características. Fue diseñado para

funcionar eficientemente en grandes redes pero también se puede utilizar para auditar hosts individuales. Es una utilidad extremadamente poderosa pero al mismo tiempo, compleja.

Acá vemos una ejecución típica de `nmap` tomando como blanco a un host que está puesto a disposición para este fin:

```
$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-20 04:41 -03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 995 closed ports
PORT      STATE    SERVICE    VERSION
22/tcp    open     ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol
    2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
25/tcp    filtered smtp
80/tcp    open     http       Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Go ahead and ScanMe!
9929/tcp  open     nping-echo Nping echo
31337/tcp open     tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/
    submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.54 seconds
```

Es notable que `nmap` haya llegado a la pantalla grande en películas taquilleras, incluyendo "The Matrix Reloaded", "The Bourne Ultimatum y "Duro de Matar 4".[6]

3.2. Primera consigna: programando un *port scanner*

A continuación se presenta un ejemplo para realizar un *port scanner* con **scapy**:

```python
#!/usr/bin/env python3

import sys
from scapy.all import *

ports = [19, 20, 21, 22, 23, 53, 80]
ip = sys.argv[1]

for i in ports:
    p = IP(dst=ip)/TCP(dport=i, flags='S')
    print(i ,end='')

    resp = sr1(p, verbose=False, timeout=1.0)
    if resp is None:
        print(" filtrado")
    elif resp.haslayer(TCP):
        tcp_layer = resp.getlayer(TCP)
        if tcp_layer.flags == 0x12:
            print(" abierto", tcp_layer.flags)
            sr1(IP(dst=ip)/TCP(dport=ports, flags='AR'), verbose=False, timeout=1)
        elif tcp_layer.flags == 0x14:
            print(" cerrado", tcp_layer.flags)
```

taller3.py

Para ejecutar el código, necesitamos pasar una IP destino como argumento, de la siguiente manera, por ejemplo:

```
$ python taller3.py 157.92.15.1
```

Luego de ejecutar el código y esperar unos instantes debería obtenerse una salida como la siguiente:

```
19 filtrado
20 filtrado
21 filtrado
22 abierto SA
23 filtrado
53 filtrado
80 abierto SA
```

Adaptar el código anterior para que también analice puertos UDP, sabiendo que cuando un puerto (UDP o TCP) está cerrado en un sistema operativo, éste usualmente responde con un mensaje ICMP de tipo *Port Unreachable*. Tener en cuenta que no siempre se podrá saber con certeza si el estado de los puertos UDP es **abierto**, **cerrado** o **filtrado**. La presencia de *firewalls* y/o *proxies* en el camino puede dificultar conocer el verdadero estado de los puertos.

3.3. Segunda consigna: Experimentación e Informe

Usando la herramienta desarrollada, analizar todos los puertos menores a 1025 (UDP y TCP) de sitios web de universidades en cualquier lugar del mundo (**un sitio web por cada integrante del grupo**). Analizar el estado de los puertos tratando de determinar si el sitio web se encuentra protegido por un *firewall*. Para esto, tener en cuenta el estado de la mayoría de los puertos escaneados.

El informe debe seguir la siguiente estructura, intentando cumplir con los límites de palabras sugeridos:

- **Introducción (máximo 200 palabras):** Breve explicación de los experimentos que se van a realizar.
- **Métodos y condiciones de los experimentos (máximo 400 palabras):** Explicación del código implementado y descripción de la estrategia para analizar puertos UDP.
- **Resultados de los experimentos (máximo 600 palabras):** En esta sección deben presentarse figuras y/o tablas que muestren de manera integral los resultados observados. A modo de sugerencia pueden mostrar la proporción entre los estados de cada puerto (UDP y/o TCP) para cada sitio analizado.
- **Conclusiones (máximo 200 palabras):** Breve reseña que sintetice las principales dificultades y descubrimientos.

A continuación se enumeran cuestiones que deben responderse una vez obtenido el estado de puertos. Es importante transcribirlas y destacarlas en el informe y se valorará significativamente el planteo de nuevas preguntas.

- ¿Cuántos puertos abiertos aparecen? ¿A que servicios/protocolos (nivel de aplicación) corresponden?
- ¿Cuántos puertos filtrados tenían los sitios web que se probaron?
- ¿Es posible darse cuenta si los hosts que se probaron están protegidos por un *firewall*?
- ¿Existen otros puertos *bien conocidos* que puedan estar abiertos en los hosts que se probaron?

3.4. Tercera consigna (OPCIONAL): DNS

DNS es un sistema distrubuido de resolución de nombres. La idea es que cuando una aplicación necesita conectarse con un dispositivo en Internet, usualmente se hace a través de un nombre de dominio en parte debido al gran dinamismo con el que pueden cambiar las direcciones IP y también debido que a los humanos se nos da mejor con los nombres que con los números. Para esto, todos los proveedores de servicios de Internet ofrecen el servicio de resolución de nombres mediante servidores denominados **Resolvers**, dedicados específicamente a este proceso. El proceso de resolución de nombres consiste en sucesivas consultas y respuestas por parte de todos los servidores DNS involucrados. Las consultas suelen ser **recursivas** cuando las PC quieren resolver un nombre y le preguntan al Resolver local y suelen ser **iterativas** cuando los Resolvers le pasan las consultas a los servidores **Autoritativos** responsables de cada zona. Por esa razón, en una consulta determinada, puede haber subconsultas recursivas e iterativas.

Además de los servidores autoritativos de cada zona, el sistema DNS no podría funcionar si no existieran servidores por encima de toda la jerarquía de zonas que funcionen como punto de partida para comenzar las consultas iterativas. Estos servidores se llaman **Root Name Servers** [9] y tienen direcciones IP asignadas fijas, que nunca cambian de manera que no haga falta hacer una consulta DNS para resolverlos porque sino no se podría empezar. Estos servidores y sus direcciones IP están listados en la siguiente tabla:

Nombre del Servidor	Direcciones IP (IPv4, IPv6)	Entidad propietaria
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

También se pueden hacer consultas DNS usando `dig` [7] (o `nslookup` [8] en Windows). Este comando es de especial importancia para testear la respuesta de servidores DNS si por alguna razón se sospecha que algo puede no estar funcionando bien. A continuación se muestra la salida del comando `dig` en Linux.

```
$ dig www.dc.uba.ar
; <<>> DiG 9.16.1-Ubuntu <<>> www.dc.uba.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47504
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.dc.uba.ar.     IN  A

;; ANSWER SECTION:
www.dc.uba.ar.    600 IN  CNAME www-1.dc.uba.ar.
www-1.dc.uba.ar.  599 IN  CNAME dc.uba.ar.
dc.uba.ar.    599 IN  A 157.92.27.128

;; Query time: 46 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
```

```
;; WHEN: mie oct 14 01:18:44 -03 2020
;; MSG SIZE  rcvd: 92
```

Ese sería el resultado de una consulta **recursiva** al Resolver local. Sin embargo, también se puede realizar una consulta DNS a un servidor en particular. En `dig` se puede especificar este servidor utilizando el caracter `@` previo a la dirección IP a la que se desea solicitar la consulta. A continuación se muestra el resultado de ejecutar este comando, con la dirección del root server b.

```
$ dig @199.9.14.201 www.dc.uba.ar
; <<>> DiG 9.16.1-Ubuntu <<>> @199.9.14.201 www.dc.uba.ar
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61338
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 15
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: cd7336edf3bba94d8df821c55f867d48831147948cc77538 (good)
;; QUESTION SECTION:
;www.dc.uba.ar.     IN  A

;; AUTHORITY SECTION:
ar.     172800  IN  NS  c.dns.ar.
ar.     172800  IN  NS  f.dns.ar.
ar.     172800  IN  NS  b.dns.ar.
ar.     172800  IN  NS  a.dns.ar.
ar.     172800  IN  NS  d.dns.ar.
ar.     172800  IN  NS  e.dns.ar.
ar.     172800  IN  NS  ar.cctld.authdns.ripe.net.

;; ADDITIONAL SECTION:
a.dns.ar.   172800  IN  A 200.108.145.50
b.dns.ar.   172800  IN  A 200.108.147.50
c.dns.ar.   172800  IN  A 200.108.148.50
d.dns.ar.   172800  IN  A 192.140.126.50
e.dns.ar.   172800  IN  A 170.238.66.50
f.dns.ar.   172800  IN  A 130.59.31.20
ar.cctld.authdns.ripe.net. 172800 IN  A 193.0.9.59
a.dns.ar.   172800  IN  AAAA  2801:140::10
b.dns.ar.   172800  IN  AAAA  2801:140:11::50
c.dns.ar.   172800  IN  AAAA  2801:140:10::10
d.dns.ar.   172800  IN  AAAA  2801:140:dddd::50
e.dns.ar.   172800  IN  AAAA  2801:140:eeee::50
f.dns.ar.   172800  IN  AAAA  2001:620:0:ff::20
ar.cctld.authdns.ripe.net. 172800 IN  AAAA  2001:67c:e0::59

;; Query time: 156 msec
;; SERVER: 199.9.14.201#53(199.9.14.201)
;; WHEN: émi oct 14 01:23:36 -03 2020
;; MSG SIZE  rcvd: 517
```

Como se puede ver, la respuesta no es la misma, sino que este servidor nos responde con el primer nivel de la jerarquía de zonas para que se pueda continuar con la resolución de nombres.

A continuación se presenta un ejemplo para realizar una consulta a uno de los servidores DNS Root. La consulta es por el registro A del dominio `www.dc.uba.ar` y se hace a uno de los servidores DNS Root que aparecen en la tabla anteriormente.

```
#!/usr/bin/env python3
```

```
from scapy.all import *

dns = DNS(rd=1,qd=DNSQR(qname="www.nasa.gov"))
udp = UDP(sport=RandShort(), dport=53)
ip = IP(dst="199.9.14.201")

answer = sr1( ip / udp / dns , verbose=0, timeout=10)

print(answer[DNS])

if answer.haslayer(DNS) and answer[DNS].qd.qtype == 1:
    print("Additional Records")
    for i in range(answer[DNS].arcount):
        print(answer[DNS].ar[i].rrname, answer[DNS].ar[i].rdata)
    print("Name Servers")
    for i in range(answer[DNS].nscount):
        print(answer[DNS].ns[i].rrname, answer[DNS].ns[i].rdata)
    print("Answer")
    for i in range(answer[DNS].ancount):
        print(answer[DNS].an[i].rrname, answer[DNS].an[i].rdata)
```

dns.py

Adaptar el código anterior de manera que, a través de sucesivas consultas iterativas se obtenga el registro MX de un dominio dado. Para esto, tener en cuenta que en cada consulta DNS puede tener 3 tipos de respuestas: i) nos devuelven los servidores DNS a los cuales seguir preguntando, ii) nos devuelven la respuesta a la consulta que estamos haciendo o iii) nos devuelven el registro SOA de la zona indicando que el registro solicitado no forma parte de la base de datos de nombres de la zona.

Usando la herramienta desarrollada, consultar por los servidores de mail que atienden los correos del dominio de una universidad (su nombre de dominio) en algún lugar del mundo. **Se deben probar tantos dominios como integrantes en el grupo participen de la parte optativa**. Analizar si los servidores de mail tienen nombres en el mismo dominio que el de la universidad o pertenecen a otro dominio. Si es posible, averiguar también si dichos servidores de mail se encuentran en la misma **zona geográfica**, indicando de forma aproximada si es el mismo país, misma región o mismo continente.

A continuación se enumeran cuestiones que deben responderse una vez obtenidas y analizadas las respuestas de los servidores. Es importante transcribirlas y destacarlas en el informe y se valorará significativamente el planteo de nuevas preguntas.

- ¿Cuántos niveles de servidores DNS se recorrieron en las sucesivas consultas hasta obtener la información solicitada?
- ¿Todos los servidores DNS Autoritativos que aparecen en las sucesivas respuestas responden a las consultas realizadas?
- ¿Cuántos nombres de servidores de mail encontraron? ¿Tienen nombres en el mismo dominio que la universidad?
- ¿Cuántas direcciones IP distintas hay? ¿Estas direcciones IP corresponden a dispositivos que están prendidos? (*Hint*: probar con **ping** si responden)
- ¿Coinciden las IPs de los servidores de correo con las IPs de los servidores Web?

Referencias

[1] (TCP) https://tools.ietf.org/html/rfc793

[2] (UDP) https://tools.ietf.org/html/rfc768

[3] (Domain Names - Concepts and Facilities) https://tools.ietf.org/html/rfc1034

[4] (Domain Names - Implementation and Specification) https://tools.ietf.org/html/rfc1035

[5] (nmap) https://nmap.org/

[6] (nmap en películas) https://nmap.org/movies

[7] Comando dig https://en.wikipedia.org/wiki/Dig_(command)

[8] Comando nslookup https://en.wikipedia.org/wiki/Nslookup

[9] Internet Assigned Numbers Authority (IANA) https://www.iana.org/domains/root/servers