DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Taller 3

Port Scanning y DNS

Teoria de las comunicaciones
Segundo Cuatrimestre de 2020

Integrantes	LU	Correo electrónico
Luis Castro	422/14	castroluis1694@gmail.com
Felipe Mateo Curti	71/17	fmcurti@gmail.com
J. Martin Mamani Aleman	630/17	mr.tinchazo@gmail.com
Ian Luca Celestino Breitman	375/15	ian13celestino@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Índice

1. Introducción

En talleres anteriores realizamos análisis relativos a los paquetes transferidos hacia y desde nuestras redes. En éste nos centramos en una parte crucial y particular de este tipo de comunicaciones: los puertos.

Se nos pide hacer un análisis del estado de los puertos de los servidores de ciertas universidades públicas (una por cada integrante del grupo) mediante el código provisto por la cátedra y nuestras subsiguientes modificaciones. Dentro de ese análisis también responderemos a ciertas preguntas planteadas en el enunciado.

Consultaremos el estado de los puertos bien conocidos para 4 universidades públicas del mundo seleccionadas de entre las que habíamos usado en el taller anterior. De esas consultas comprobaremos las proporciones de puertos abiertos, cerrados y filtrados para ambos tipos (TCP y UDP).

Para la parte opcional tuvimos que recorrer los servidores de un dominio dado y encontrar el registro MX que indica el tipo de servidor mail.

2. Métodos y condiciones de los experimentos

2.1. Descripción de la estrategia

El hecho de que UDP no es orientado a conexión nos permite no pasarle un flag de sincronización a la función UDP() a la hora de crear el paquete a enviar. Sin embargo, este hecho tiene una desventaja: es posible poder generar una conexión con un puerto UDP pero que este no nos responda absolutamente nada. Por esto, si al enviar un paquete recibimos una respuesta nula, esto puede significar tanto que el puerto está abierto como que está filtrado por acción de firewalls o proxies. Es decir, ante una no respuesta, no podemos discernir en qué estado se encuentra un puerto UDP. En este caso, diremos que el estado es ”abierto|filtrado”.

También puede suceder que al estar abierto el puerto y poder establecer conexión, la respuesta sea un paquete con protocolo UDP. En tal caso, sin importar el contenido del paquete recibido, el puerto lo podemos considerar abierto.

Distinto es cuando la respuesta posee protocolo ICMP y tipo 3. Bajo este contexto, pueden darse dos escenarios dependiendo del código ICMP que contenga el paquete: según nmap[1], si el código es 3 (Port unreachable error) entonces el puerto está cerrado. En cambio, si el código es 1, 2, 9, 10 o 13 (other unreachable errors), entonces el puerto está filtrado.

Implementaremos estas 4 posibilidades, como veremos a continuación.

2.2. Código implementado

En el análisis de los puertos TCP, agregamos una posibilidad con respecto a la respuesta recibida: si la respuesta del paquete enviado al i-ésimo puerto tiene a ICMP como protocolo, entonces este puerto está cerrado. Luego, al final de cada iteración, escribimos una línea en un archivo csv que contenga el número de puerto, el protocolo (en este caso, TCP), el estado y los flags si es que el tipo de respuesta nos permitió conocerlos.

Con respecto al análisis de puertos UDP, primero creamos el paquete con protocolos IP y UDP y este genera una respuesta, que analizamos según los posibles casos explicados en la subsección anterior. Para revisar el caso en el que un puerto está filtrado, creamos la lista ICMP_FILTERED_CODES, que contiene a los códigos 1, 2, 9, 10 y 13. De esta manera, solo necesitamos revisar si el código ICMP está en dicha lista para saber si un puerto está filtrado. Como con los puertos TCP, antes de pasar a la siguiente iteración, escribimos una línea en el archivo csv pero sin el campo correspondiente a los flags. Cabe destacar que, ante la ambigüedad del estado abierto/filtrado, no nos inclinamos por ninguna y lo plasmamos así en el csv.

[1] `https://nmap.org/book/scan-methods-udp-scan.html`

3. Experimentación

A continuación, las 4 universidades analizadas y una abreviación para referirnos a cada una de ellas de aquí en adelante:

- University of California (UCLA). Dominio: ucla.edu
- Moscow State University (MSU). Dominio: msu.ru
- University of Toronto (UT). Dominio: utoronto.ca
- Hiroshima City University (HCU). Dominio: hiroshima-cu.ac.jp

Dada la naturaleza de los servidores que vamos a analizar, siendo universidades públicas, nuestra hipótesis sobre el estado de los puertos es que en su gran mayoría estarán filtrados. Teniendo únicamente disponibles algunos puertos específicos típicos en las comunicaciones. Por ejemplo el puerto 80 para la navegación web HTTP o el 443 que corresponde a la versión para comunicaciones seguras (HTTPS). Creemos que el motivo por el que la mayoría estén filtrados sea mayormente la seguridad de los datos públicos.

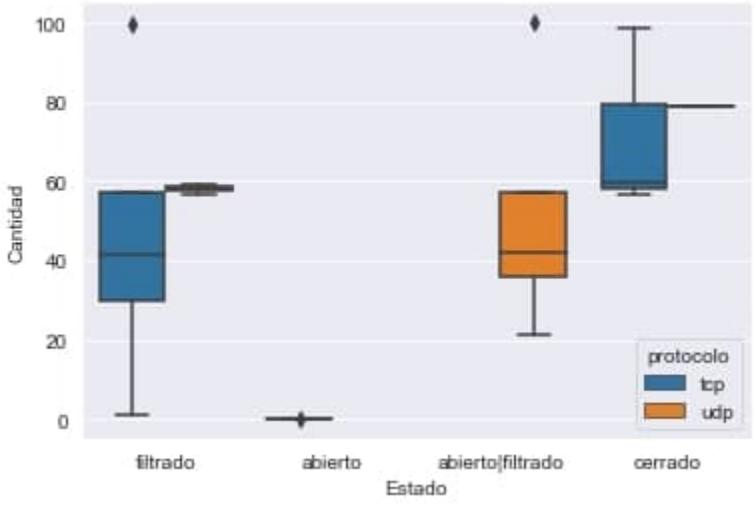

Figura 1: Distribución de puertos de las 4 universidades juntas según su estado y protocolo

Podemos ver que la distribución de los estados de los puertos parecería confirmar nuestra hipótesis inicial. Veamos entonces los números más en detalle. A continuación, responderemos las siguientes preguntas formuladas en el enunciado.

3.1. Preguntas

1. ¿Cuántos puertos abiertos aparecen? ¿A que servicios/protocolos (nivel de aplicación) corresponden?

En las 4 universidades, aparecen abiertos los puertos 80 (HTTP) y 443 (HTTPS). Además, el puerto 22 de MSU, correspondiente a SSH, también se encuentra abierto.

2. ¿Cuántos puertos filtrados tenían los sitios web que se probaron?

- UT

- TCP filtrados:1022
- UDP filtrados:0

- UCLA
  - TCP filtrados:440
  - UDP filtrados:583
- HCU
  - TCP filtrados:11
  - UDP filtrados:0
- MSU
  - TCP filtrados:407
  - UDP filtrados:608

Para el caso de UDP, no podemos estar seguros de que esas sean las cantidades exactas de puertos filtrados, ya que las no respuestas son ambiguas y pueden significar que el puerto está abierto o filtrado.

3. ¿Es posible darse cuenta si los hosts que se probaron están protegidos por un firewall?

No es posible determinar con total seguridad si los hosts están protegidos. En el caso de TCP, consideramos a un puerto como filtrado cuando no recibimos respuesta alguna. Esto puede suceder por la acción de un firewall pero también por la intervención de proxies u otros dispositivos de red como switches. De esta manera, a pesar de que obtuvimos una gran cantidad de puertos TCP filtrados en todas las universidades salvo en la de Hiroshima, no podemos afirmar si el causante fue un firewall. Por otro lado, con los puertos UDP, tenemos aún más incertidumbre dado el resultado ambiguo de una no respuesta. Sin embargo, asumiendo que las filtraciones se deben únicamente a un firewall, podemos afirmar que tanto los puertos TCP de la universidad de Toronto como los puertos UDP de las universidades de Hiroshima y Moscú se encuentran protegidos por un firewall dado que son más de la mitad de los 1024 puertos de su tipo.

4. Parte opcional

Esta vez, los paquetes a enviar tendrán protocolo DNS (con *qtype* MX), UDP (con puerto destino 53) e IP. En este último iremos variando la ip destino, comenzando por la correspondiente a b.root-servers.net y actualizándola con la IP de los respectivos name servers que hallaremos. Así, obtenemos una primera respuesta por parte de un root name server.

El código principal consiste de un ciclo que se ejecuta hasta encontrar al menos un mail name server. Al principio, imprimimos la información que se encuentra en la respuesta haciendo uso de las funciones *print_authority()*, *print_name_servers()* y *print_answers()*. Luego, tenemos los 3 posibles casos que se detallan en el enunciado:

i) Si recibimos name servers a los cuales seguir preguntando, revisamos si todos los name servers listados están disponibles para responder una consulta mediante la función *check_if_each_name_server_answers()*, que sigue el ejemplo proporcionado por Scapy[2] y envía un paquete a cada name server pero usando como IP destino a 8.8.8.8 para ver rápidamente si hay respuesta. Luego, actualizamos la respuesta a analizar en la próxima iteración enviando un nuevo paquete, esta vez con la ip del próximo name server a visitar (vamos probando en el orden que aparezcan listados en el name server actual). Sin embargo, aquí puede suceder que la ip del próximo name server no esté en el actual, o que esté pero no responda. En tales casos, seguimos probando con el próximo name server listado y terminamos la iteración.

[2] `https://scapy.readthedocs.io/en/latest/usage.html#dns-requests`

ii) Si en la sección Answer obtuvimos registros de tipo MX, entonces usamos la función *process_mail_server_names()* para agregar cada dominio y su ip a un diccionario y además actualizar el contador de mail servers que tienen el mismo dominio que la universidad. Luego, llamamos a *print_summary()* para imprimir la información recolectada a lo largo de la ejecución.

iii) Si recibimos el registro SOA de la zona, entonces el registro buscado no se encuentra en la base de datos de esta zona. En tal caso, imprimimos dicha conclusión y no hacemos nada más.

Cabe destacar que para la parte opcional consideramos que no era correspondiente hacer uso de una exhibición gráfica de los datos así que decidimos plasmar la información recolectada únicamente por medio de valores numéricos impresos por pantalla. Es posible replicar cada análisis para una ip dada ejecutando sudo python3 dns.py <ip>

Para cada universidad, contestaremos en orden las siguientes 4 preguntas:

1. ¿Cuántos niveles de servidores DNS se recorrieron en las sucesivas consultas hasta obtener la información solicitada?

2. ¿Todos los servidores DNS Autoritativos que aparecen en las sucesivas respuestas responden a las consultas realizadas?

3. ¿Cuántos nombres de servidores de mail encontraron? ¿Tienen nombres en el mismo dominio que la universidad?

4. ¿Cuántas direcciones IP distintas hay? ¿Estas direcciones IP corresponden a dispositivos que están prendidos? (Hint: probar con ping si responden)

4.1. UCLA

1. Niveles recorridos: 3
2. Todos contestaron: True
3. Cantidad de nombres de servidores de mail encontrados: 1

   Cantidad de nombres en el mismo dominio de la universidad: 1
4. Direcciones ip de los servidores de mail:

   {'mx.smtp.ucla.edu.': '169.232.46.172'}

4.2. MSU

1. Niveles recorridos: 3
2. Todos contestaron: True
3. Cantidad de nombres de servidores de mail encontrados: 2

   Cantidad de nombres en el mismo dominio de la universidad: 2
4. Direcciones ip de los servidores de mail:

   {'mx.msu.ru.': '93.180.0.1', 'nss.msu.ru.': '93.180.0.1'}

4.3. UT

1. Niveles recorridos: 3
2. Todos contestaron: False
3. Cantidad de nombres de servidores de mail encontrados: 1
   Cantidad de nombres en el mismo dominio de la universidad: 0
4. Direcciones ip de los servidores de mail:
   {'utoronto-ca.mail.protection.outlook.com.': None}

4.4. HCU

1. Niveles recorridos: 3
2. Todos contestaron: True
3. Cantidad de nombres de servidores de mail encontrados: 2
   Cantidad de nombres en el mismo dominio de la universidad: 0
4. Direcciones ip de los servidores de mail:
   {'cluster6.us.messagelabs.com.': None, 'cluster6a.us.messagelabs.com.': None}

En todas las subsecciones, las direcciones IP de los servidores de mail que han resultado "None" corresponden a dispositivos que no están prendidos, es decir, no responden al comando ping. El resto de los dispositivos están prendidos.

5. Conclusiones

A lo largo de este trabajo, hemos conocido en profundidad el método para establecer conexiones entre distintos hosts a nivel de aplicación. Se ha manifestado claramente la diferencia entre un protocolo orientado a conexión y uno no orientado a conexión. Sobre esto último, un posible trabajo a futuro es desambiguar el estado "abierto|filtrado" obtenido con el protocolo UDP.

Sobre las búsquedas iterativas a lo largo de DNS, tuvimos la oporunidad de implementar los distintos casos según la respuesta obtenida por parte de un name server. Esto solidificó los conocimientos teóricos obtenidos durante la materia y nos permitió completar mentalmente el modelo en el que se basa Internet.