Taller 3 - Port Scanning y DNS

Contenidos

Integrantes

LU	Nombre	Mail
72/18	Manuel Panichelli	panicmanu@gmail.com
76/16	Luciano Strika	lucianostrika44@gmail.com
692/12	Elías Cerdeira	eliascerdeira@gmail.com

Introducción

En este trabajo se desarrolla una herramienta para detectar el estado de los puertos en un sistema operativo, implementando un subconjunto de las funcionalidades de nmap. En concreto, la herramienta efectuará un *port scan* sobre una cierta IP enviando paquetes TCP y UDP a cada puerto y analizando las respuestas para determinar su estado.

En particular, se evalúan los *well-known ports* (1 a 1024) de las páginas web de 3 universidades diferentes: Osaka, MIT y Moscow. A partir de los resultados obtenidos se pretende descubrir qué servicios están disponibles en cada host y si están protegidos por un *firewall*.

Métodos y condiciones de los experimentos

La herramienta implementada envía paquetes TCP y UDP a la IP especificada utilizando la biblioteca `scapy`. Analiza las respuestas y las clasifica en tres grandes categorías: `filtrado`, `abierto` y `cerrado`.

- Para TCP, se inicia una conexión mediante un paquete TCP con flag SYN sobre IP al socket elegido y se analiza la respuesta:
    - Si no hubo respuesta, se devuelve `filtrado`.
    - Si hubo respuesta del tipo TCP,
        - Si los flags de la respuesta fueron SYNC ACK (SA), se devuelve `abierto`.
        - Si los flags fueron RESET ACK (RA), se devuelve `cerrado`.
    - Si hubo respuesta del tipo ICMP
        - Se retorna `filtrado (icmp t: <type> c: <code>)`.
- Para UDP, se envía un paquete UDP vacío sobre IP al socket elegido y se analiza la respuesta siguiendo la guía de interpretación especificada por `nmap`
    - Si no hubo respuesta, se devuelve `abierto|filtrado`
    - Si hubo respuesta UDP, `abierto`.
    - Si hubo respuesta ICMP de tipo 3 (*Destination Unreachable*)

        - Si tiene code 3 (*Port Unreachable*) se devuelve `cerrado`
        - Sino, se devuelve `filtrado (icmp t: 3 c: <code>)`

Se ejecutó la herramienta sobre los sitios web de tres universidades, a las 19:00 del sábado 06/06/2021.

- MIT: `23.37.251.54`
- Osaka University: `133.1.138.1`
- Moscow University: `188.44.51.94`

Se ejecutó también para otros sitios, pero los resultados no variaron significativamente.

Resultados de los experimentos

A continuación presentamos los resultados obtenidos para cada universidad, enumerando primero las respuestas TCP y luego las UDP.

- **Osaka** (Elías)

Protocolo	Respuesta	Cantidad
tcp	filtrado	1022
tcp	abierto (SA)	2
udp	abierto|filtrado	1024

  Puertos abiertos:

  - `80/tcp` abierto (SA)
  - `443/tcp` abierto (SA)

- **Moscu** (Manuel)

Protocolo	Respuesta	Cantidad
tcp	filtrado	962
tcp	filtrado (icmp t:3 c:10)	59
tcp	abierto (SA)	3
udp	abierto|filtrado	989
udp	filtrado (icmp t:3 c:10)	35

  Puertos abiertos:

  - `22/tcp` (SSH) abierto (SA)
  - `80/tcp` (HTTP) abierto (SA)
  - `443/tcp` (HTTPS) abierto (SA)

- **MIT** (Luciano)

Protocolo	Respuesta	Cantidad
tcp	filtrado	1022
tcp	abierto (SA)	2
udp	abierto|filtrado	1024

  Puertos abiertos:

  - `80/tcp` abierto (SA)

    - `443/tcp` abierto (SA)

Viendo estos datos, podemos responder algunas de las incógnitas planteadas.

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

  En todas aparecen los mismos dos puertos abiertos: `80/tcp` (HTTP) y `443/tcp` (HTTPS), y además en Moscú aparece el `22/tcp` (SSH/scp/sftp).

- **¿Cuántos puertos filtrados tenían los sitios web que se probaron?**

  En Osaka y MIT para TCP había 1022 puertos filtrados (todos menos el 80 y 443) y todos los de UDP fueron abierto|filtrado.

  En el caso de Moscú, para tcp hubieron 962 filtrados por falta de respuesta y 59 por ICMP *Destination Unreachable (type 3) / Host administratively prohibited (code 10)*. Para UDP, 989 abierto|filtrado por falta de respuesta y 35 por ICMP 3/10.

  En la documentación de `nmap` se menciona el uso de payloads específicos para intentar escanear los puertos UDP, ya que las aplicaciones que reciben paquetes vacíos suelen descartarlos. Suponemos que esta es la razón por la cual no obtuvimos ninguna respuesta UDP para estas universidades. Comprobamos que nuestra herramienta no detecta algunos puertos UDP que `nmap` sí, como por ejemplo el 53 (DNS) para la IP `8.8.8.8` (DNS de Google).

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

  Con la herramienta implementada no es posible determinar la presencia de un firewall para todos los casos. Esto es porque si un puerto está filtrado es imposible distinguir si no hubo respuesta del host o el paquete fue descartado (*drop*) por un firewall. Además, tampoco se puede en caso de recibir una respuesta, ya que ese puerto podría estar habilitado para ciertos protocolos por una regla. Por ejemplo, sería razonable que un servidor HTTP que aloja una página tenga habilitados solamente los puertos 80 (http) y 443 (https).

  En cambio, en los casos en los que se recibe ICMP 3/10 (*Host administratively prohibited*) se puede afirmar la presencia de un firewall.

- **¿Existen otros puertos bien conocidos que puedan estar abiertos en los hosts que se probaron?**

  El programa itera todos los puertos *well-known* (de 0 a 1024) y reporta el estado de todos ellos, pero podría suceder que un puerto esté abierto y que no sea detectado (como en el caso descrito arriba de UDP). Además existen los *registered ports* (de 1024 a 49151), que son de uso frecuente y en caso de estar abiertos nuestro programa no los detectaría.

DNS

En este trabajo implementamos, utilizando `scapy`, una versión simplificada del comando `dig` para resolución de nombres. Para ello, se extendió el código provisto por la cátedra para que a través de consultas sucesivas iterativas se obtenga el registro MX de un dominio dado. Este se ejecutó para las URLs de las mismas universidades mencionadas en la sección de Metodología.

Resultados

A continuación presentamos los resultados para cada universidad.

- **MIT**: `mit.edu` (Luciano)

  ```
  198.41.0.4 (a.root-servers.net)
    -> 192.33.14.30 (b.edu-servers.net.)
    -> 184.26.161.64 (usw2.akam.net.)

  Answer
  mit.edu. MX mit-edu.mail.protection.outlook.com.
  ```

  Que tiene las IPs `104.47.58.138` y `104.47.57.138`

- **Osaka University**: `osaka-u.ac.jp` (Elías)

  ```
  198.41.0.4 (a.root-servers.net)
    -> 203.119.1.1 (a.dns.jp.)
    -> 150.100.18.6 (dns-x.sinet.ad.jp.)

  Name Servers
  osaka-u.ac.jp. SOA -

  Found SOA, no MX record.
  ```

- **Moscow University**: `msu.ru` (Manuel)

  ```
  198.41.0.4 (a.root-servers.net)
    -> 193.232.128.6 (a.dns.ripn.net.)
    -> 93.180.0.1 (ns.msu.ru.)

  Answer
  msu.ru. MX mx.msu.ru.
  msu.ru. MX nss.msu.ru.
  ```

  Ambas con la IP `93.180.0.1`

Con ellos podemos resolver algunas cuestiones:

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

  Para todos los casos se recorrieron 3 niveles de servidores DNS.

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

  Si, ya que cada servidor nos indica mediante los Name Servers por qué servidores que contienen la información solicitada continuar buscando.

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

  Encontramos nombres de servidores de mail para Moscow (2 nombres de servidor) y MIT (1 nombre), todos con 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)**

  En el caso de Moscow se encontró que ambos mail servers tenían la misma IP, que se correspondían a un dispositivo prendido (que respondía `ping`). En cambio, el mail server de MIT tenía dos IPs, y ambas correspondían a dispositivos apagados.

- **¿Coinciden las IPs de los servidores de correo con las IPs de los servidores Web?**

  La de Moscow (www.msu.ru, `188.44.51.94`) no coincide y la de MIT (www.mit.edu, `104.87.47.78`) tampoco.

Conclusiones

Para todas las universidades analizadas los puertos que se encontraron abiertos fueron los usuales y esperados para una página web (80 y 443). Uno de los objetivos de este trabajo fue determinar la presencia de un firewall, sin embargo a fin de cuentas frente a la falta de respuesta de un puerto no podemos afirmar si fue por el descarte de un firewall o que no había ninguna aplicación escuchando en ese puerto. Creemos que sería razonable tanto que un servidor web no tenga aplicaciones corriendo en otros puertos, como que sí las tenga pero que estén bloqueadas por un firewall, ya que no es necesario acceder desde internet. En el caso de Moscú, se pudo determinar la presencia de uno.

Queda como trabajo a futuro implementar payloads específicos de UDP para cada *well-known port*, tal como se cuenta en la sección de resultados. Además, sería interesante realizar los mismos experimentos para puertos más allá del 1024, por ejemplo los *registered ports* y probar con URLs que no sean sitios web, para encontrar otra variedad de puertos abiertos, por ejemplo un DNS o mail server.

Si bien se logró imitar los resultados de `dig` para todos los hosts probados, funcionaba muchísimo más lento. Queda también como trabajo a futuro paralelizar las consultas, y estudiar si el programa utiliza alguna heurística para navegar el grafo de resolvedores de forma más eficiente.

Finalmente, concluimos que si bien Scapy es una herramienta poderosa que nos permite enviar y recibir paquetes de distinto tipo, tiene una gran falta de documentación y una de las dificultades más grandes del trabajo fue lograr entender cómo hacer las cosas.