Teoría de las Comunicaciones

Segundo Cuatrimestre de 2012

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Taller de Capa de Transporte

Taller N°3

Integrante	LU	Correo electrónico
Mancuso, Emiliano	597/07	emiliano.mancuso@gmail.com
Mataloni, Alejandro	706/07	amataloni@gmail.com
Gonzalez, Matias	453/07	curtu_infinito73@hotmail.com

Reservado para la catedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. Primera consigna

En cuanto a la complejidad de los algoritmos, el *Connect scan* realiza todo el 3-way handshake mientras que *SYN scan* no termina la conexion, por lo tanto genera un paquete menos en la red, por lo que deberia ser mas rapido. Sin embargo en nuestra implementacion, y a pesar de lo expuesto anteriormente, el *Connect scan* funciona mas rapido. Esto lo adjudicamos a que los sockets estan mejor implementados por lo que hace que el scan tarde menos aunque manejando una cantidad mayor de paquetes.

Como una primer método de ataque y por si solo Port Scannin parece un método bastante inocuo. De hecho también puede ser utilizado para conocer el estado de los puertos en un sistema propio. Una primera reflexión puede hacer pensar que solo se trata de la detección de los puertos y sus estados.

Pero una mirada mas profunda y un análisis mas objetivo revela que esto no es cierto, este método puede resultar de gran impacto contra un sistema, pues da un gran panorama del estado del sistema y sus posibles vulnerabilidades. Teniendo en cuenta esto Port Scaning se puede pensar por ejemplo como la base de un ataque de mayor escala. Una vez obtenida la información de los distintos puertos podría llegar a inferirse el tipo de sistema operativo utilizado, usarse otras técnicas para obtener mayor información sobre el sistema en si o las aplicaciones que tiene, incluso la función e información que maneja el host.

La detección de puertos es reconocible, por supuesto dependiendo de la variante o la clase de reconocimiento que se este realizando varía la dificultad de la detección. En base a lo investigado consideramos que la prevención contra este método es muy complicada, una posible manera es cerrar todos los puertos que no están siendo y no serán utilizados. Una mejor aproximación a lidiar con este problema es llevar y mantener un control sobre este tipo de ataques reportarlo en forma inmediata y analizar si existe un riesgo de un ataque de mayor complejidad.

Por lo general para detectar el uso de este método se utilizan herramientas especificas de sistemas de detección de intruso (IDS) para la red, control de los logs del sistema, configurar firewalls para la detección de este tipo de ataques.

En cuanto a la efectividad de este tipo de ataques creemos que depende de con cuanta conciencia se haga la configuración de un servidor, por ejemplo a un servidor bien preparado se le puede realizar el escaneo de algunos puertos pero el servidor al detectarla puede bloquear el origen o aumentar los controles y medidas de seguridad, al intentar obtener mayor información o un ataque de mayor escala se ven comprometidos. Y también como mencionamos anteriormente de las contra medidas que se utilicen.

2. Segunda Consigna

2.1. Banner Grabbing y servicios suceptibles:

Como se explico en clase Banner Grabbing es una tecnica en la cual se obtiene informacion a traves de las aplicaciones que corren en los distintos puertos de un host. La informacion obtenida puede ser versiones y nombre de las aplicaciones, esta inforamcion puede ser utilizada para detectar o inferir el sistema operativo o para explotar las vulnerabilidades de la aplicacion en si.

Debido a que la tecnica trabaja sobre aplicaciones, son varios los servicios que deberian ser vulnerables. Una de las primera ideas que tuvimos es que todos los puertos dedicados a aplicaciones podrian ser suceptibles. La realidad nos mostro que eso no es cierto, en la mayoria de los casos los host no implmentan todos, ni muchos, de los servicios disponibles.

Por esta razon es que investigamos un poco y descubrimos que los servicios suceptibles esta tecnica, mas comunes suelen ser:

- Web/Aplicaciones Web puerto 80
- FTP (transferencia de archivos) puerto 21
- SMTP (mail) puerto 25

Pero nuestro primer pensamiento no era del todo errado como Banner Grabbinng se utiliza para obtener informacion sobre servicios tambien existen otros que son suceptibles, pero la suceptibilidad depende varios factores como: uso del host, seguridad del mismo, etc. Los servicios, y la "suceptibildiad" de los mismos, no suelen ser tan comunes. Algunos de ellos son:

- DNS puerto 53
- MYSQL (base de datos) 3306
- POSTGRE (base de datos) 5432
- POP3 (mail) 110

Banner Grabbing se puede realizar de muchas maneras como utilizar alguna de las herramientas de testeo y analisis vista en clase, utilizando alguna aplicacion de conexion como telnet. Debido que al establecer la conexion las aplicaciones presentan inforacion sobre las ellas, entonces lo unico que se necesita es cambiar el puerto al que nos conectamos en el host para que los distintos servicios nos brinden informacion.

Aunque dependiendo de el servicion del que se esta queriendo obtener informacion, tambien se pueden utilizar aplicaciones para el trabajo con esos protocolos, como puede ser:

- ftp por linea de comando en linux.
- Navegadores.
- clientes de mail desde consola.

En algunos casos esta opcion tambien puede ser mas divertida.

En otros casos como el DNS, que en realdiad no tiene un "banner"para agarrar, se debe usar la herramiento dig para obtener la version.

2.2. Implementaciones de Captura

Como se menciona antes la tecnica de Banner Grabbing se puede realizar de varias maneras con distintas herramientas, conectandose a un host y probando en los distintos puertos.

- Telnet: *telnet <host> <port>*
- Netcat: *nc <host> <port>*

Tambien se puede realizar con herramientas propias de cada aplicacion de la que querramos obtener la informacion.

- FTP:
  - *ftp <host>*
  - Comenzar la aplicacion ingresando en consola *ftp*, dentro de la consola ftp escribir *open <host>*
- SSH: *ssh <host>*, esta opcion no siempre es la mejor porque suele dejar huellas de los ingresos.
- DNS:
  - *dig -c CH -t txt version.bind @<host>*, obtener la version simple
  - *dig -t tct -c chaos version.bind @<host>*, cuando se quiere acceder a la version a traves de clase CHAOSNET.

3. Tercera consigna

Corrimos el comando nmap contra 3 hosts locales con diferentes sistemas operativos. La herramienta logro identificar satisfactoriamente a 2 de ellos (Apple Mac OS X 10.7.0 y Windows XP) pero no pudo reconocer a Windows 7. Suponemos que esto se debe a que al ser un sistema relativamente nuevo, todavia no se cuenta con mucha informacion para reconocerlo facilmente.

Capturamos el trafico con *Wireshark* y pudimos verificar las distintas pruebas que ejecuta nmap.

Para el proceso de escaneo de puertos utiliza la tecnica de syn scan. Con esto se obtienen los diferentes puertos en estado *open*, *close* o *f*, los cuales luego utiliza para realizar varios tests.

```
Host is up (0.061s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
88/tcp   open  kerberos-sec
445/tcp  open  microsoft-ds
548/tcp  open  afp
3689/tcp open  rendezvous                                    TCP (T2–T7)
MAC Address: C8:BC:C8:A6:16:1B (Apple)
Device type: general purpose
```

Figura 1: Resultado syn scan Mac OS X

```
Not shown: 998 filtered ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:1F:C6:C1:BF:15 (Asustek Computer)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
```

Figura 2: Resultado syn scan Windows XP

Podemos ver una gran diferencia entre los 2 sistemas. Aparte de los puertos en estado *open*, vemos la diferente categorizacion del resto de los puertos. Por un lado en OS X se los clasifica como *closed* mientras que en windows XP se los clasifica como *filtered*. Esto se debe a que OS X respondia los SYN con paquetes RST mientras que windows no enviaba paquetes de respuesta.

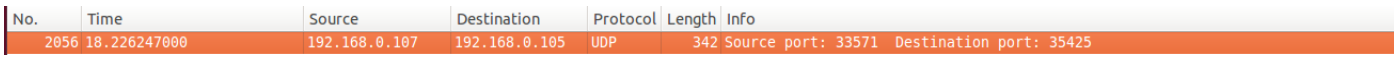
Las pruebas que pudimos observar:

- ICMP echo (ie): Se envian 2 paquetes ICMP con algunos parametros fijos. En el primero se setea el DF flag de la capa IP,el TOS (tipo de servicio) en 0, el codigo de ICMP es 9, el numero de secuencia es 295.
  En el segundo se setea el TOS en 4, el codigo en 0, y el ICMP requestId y numero de secuencia se incrementan en 1 de los valores del primer paquete.
- TCP explicit congestion notification (ECN): Se envia un paquete TCP con los siguientes flags a uno de los puertos en estado open: SYN, ECN, CWR.
- UDP (U1): Se envia un paquete UDP a un puerto en estado close. Se setea el IP ID en 0x1042 y se repite el caracter *C* en el campo de datos.

  Una prueba en comun para ambos hosts fue la de UDP (U1). La respuesta fue diferente. Por un lado el sistema OS X envio un paquete ICMP (Port unreachable), mientras que windows no respondio.

No.	Time	Source	Destination	Protocol	Length	Info
2475	26.623342000	192.168.0.107	192.168.0.100	UDP	342	Source port: 39884 Destination port: 33518
2476	26.624839000	192.168.0.100	192.168.0.107	ICMP	70	Destination unreachable (Port unreachable)

Figura 3: Prueba UDP Mac OS X



No.	Time	Source	Destination	Protocol	Length	Info
2056	18.226247000	192.168.0.107	192.168.0.105	UDP	342	Source port: 33571 Destination port: 35425

Figura 4: Prueba UDP Windows XP