Taller de Capa de Transporte

Teoría de las Comunicaciones
Departamento de Computación
FCEN - UBA

30.10.2012

1. Introducción

Continuando con los lineamientos que nos propusimos, en esta ocasión utilizaremos los servicios de la capa de transporte para extender el conocimiento de la red en la que nos encontramos. En los talleres anteriores hemos investigado algunas técnicas que pueden ser útiles para determinar cuáles son los hosts conectados en nuestro segmento de red y en una red IP. En esta oportunidad, en cambio, nos abocaremos a descubrir qué tiene para ofrecernos un host en particular, analizando el estado de los puertos, los servicios disponibles y el sistema operativo subyacente.

2. Normativa

- Fecha de entrega: martes 27 de noviembre de 2012
- El código deberá haber sido enviado por correo para esa fecha con el siguiente formato:
  to: tdc-doc at dc uba ar
  subject: debe tener el prefijo [tdc-transporte]
  body: nombres de los integrantes y las respectivas direcciones de correo electrónico
  attachment: el código fuente desarrollado.
- Se deberá entregar el informe impreso y abrochado con la lista de integrantes y los respectivos correos electrónicos (los mismos que fueran enviados por mail).

3. Enunciado

A partir de los conceptos explicados durante la clase de taller[1], cada grupo deberá realizar las consignas detalladas más abajo. Como es habitual, recomendamos fuertemente realizar las implementaciones requeridas en Scapy [1].

3.1. Consignas

3.1.1. Primera parte: port scanning

- Implementar SYN scan y Connect scan utilizando lo explicado en clase.
- ¿Cuál es la complejidad algorítmica de estos métodos? Comparar tiempo contra cantidad de operaciones.
- Discutir sobre cuán inocuos resultan estos métodos.
- ¿Es posible tomar conciencia de que una detección de puertos está siendo realizada?
- ¿En qué escenarios son efectivos? ¿En cuáles carecen de utilidad?

[1] Para más información, recurrir a las diapositivas publicadas en la web de la materia.

- Considerar el siguiente escenario: se desconoce la existencia de un firewall intermedio. ¿Qué se podría determinar con esta técnica?

3.1.2. Segunda parte: detección de versión

La técnica de *banner grabbing* consiste, como hemos explicado, en realizar una interacción preestablecida con un servicio en donde éste envía información sobre su implementación (i.e, número de versión, sistema operativo, etc.).

Documentar al menos tres servicios susceptibles a la técnica de banner grabbing y dar una implementación para la captura de dicho banner.

3.1.3. Tercera parte: OS fingerprinting vía Nmap

En esta última consigna pondremos en práctica el uso de Nmap [2]. El objetivo será explorar la técnica de detección de sistemas operativos provista por dicha herramienta.

- Correr Nmap especificando la opción `-O` (OS fingerprinting) contra al menos dos hosts conocidos con distinto sistema operativo (e.g., Windows y Linux). Analizar la precisión de los resultados arrojados por Nmap y sacar conclusiones.
  Nota: sugerimos también activar la opción `-v` para aumentar el nivel de verbosidad.
- Repetir la actividad anterior capturando el tráfico generado con Wireshark. A partir de lo observado, contestar las siguientes preguntas:
  - ¿Puede identificarse un proceso de escaneo de puertos en las capturas? Si es así, indicar qué algoritmo se utiliza y qué resultados se obtuvieron.
  - Identificar alguna de las pruebas de Nmap mencionadas en [3]. ¿Qué tipo de paquetes se envían? ¿Qué características tienen? (i.e., flags, opciones, etc.).
  - En caso de haber una prueba en común para los hosts analizados, estudiar si las respuestas generadas contienen diferencias.

Referencias

[1] Scapy (Web)
`http://www.secdev.org/projects/scapy/`

[2] Nmap (Web)
`http://nmap.org/`

[3] TCP/IP Fingerprinting Methods Supported by Nmap
`http://nmap.org/book/osdetect-methods.html`