Redes y Sistemas Distribuidos

Laboratorio 5: scapy

Descargo de Responsabilidad

En este laboratorio se va a demostrar cómo capturar contenido enviado por red a la computadora en la que estamos trabajando. Si se conservan copias de archivos cuyo contenido está protegido por derecho de autor se esta cometiendo un delito a menos que uno cuente con la autorización del tenedor de estos derechos.

Asimismo, si se captura y observa información dirigida a *otra* computadora (lo cual no se explica en este laboratorio) también se corre el riesgo de estar incurriendo en un delito puesto que, por ejemplo, leer correspondencia ajena esta penado con entre 15 días y 6 meses de prisión en Argentina.

Ni la cátedra de Redes y Sistemas Distribuidos, ni FaMAF, ni la UNC se hacen responsables del uso inapropiado del material dado para estas clases.

Contenido de la Presentación

- Características del proyecto
- Motivación
- ¿Qué es un sniffer?
  - tcpdump
- ¿Qué es Scapy?
  - Instalación de Scapy
- Tarea a realizar

Características del Proyecto

- Aprender a usar una herramienta de análisis de red.
- Entender mínimamente el funcionamiento de TCP.
- Poder capturar información enviada en una conexión TCP.

Motivación

- Muchas aplicaciones webs esconden su funcionamiento interno (por ejemplo, last.fm o el viejo reproductor de grooveshark).
- Estas aplicaciones reciben datos desde un servidor y funcionan como intermediario para que nosotros accedamos a estos.
- El mecanismo utilizado esta oculto a nosotros (por ejemplo, para reforzar la protección de derechos de autor). No tenemos acceso directo a los datos transportados.
- Como la mayoría de estos datos viajan a través del protocolo TCP (pues la mayoría de las aplicaciones web están construidas sobre este), estos datos pueden ser capturados con opción a conservarlos (siempre y cuando no se estén violando derechos de autor o ningún otro tipo de licencias de uso)

¿Qué es un sniffer?

- Al desarrollar un protocolo de red nuevo, o para buscar fallas de seguridad en un protocolo existente, resulta de enorme utilidad poder observar la información que intercambia el protocolo a través de la red.
- Las herramientas que permiten observar este tipo de actividad se llaman herramientas de análisis de red.
- De entre estas la herramienta más básica es el **sniffer**.
- Un sniffer es un programa que permite capturar los paquetes de datos tal cual son enviados por la placa de red.

¿Qué es un sniffer?: tcpdump

- **tcpdump** es un sniffer clásico en varios sistemas *nix.
- Este sniffer lee paquetes de una interfaz de red y los puede presentarlos al usuario de varias formas.
- Lo que captura tcpdump puede ser volcado en ciertos archivos de extensión *.pcap (utilizando la opción -w, referirse al enunciado para ejemplos)
- Pcap es un formato, más o menos estándar, para almacenar paquetes capturados.
- Estos archivos pueden ser analizados con la herramienta **Scapy**.

¿Qué es Scapy?

- **Scapy** es una biblioteca de python que, entre otras cosas, permite manipular paquetes capturados en archivos *.pcap.
- Se puede conseguir de manera gratuita en la página oficial:
  - http://www.secdev.org/projects/scapy/
- De la página oficial también se puede conseguir documentación (ya que la librería analiza distintos tipos de paquetes, no únicamente paquetes *.pcap)
- Además existe una lista de correo y un trac (todos accedibles desde la página oficial)

¿Qué es Scapy?: Instalación de Scapy

- Necesitan Python 2.5 para versiones mayores o iguales a la 2.x de Scapy. Y Python 2.4 para versiones menores a la 2.x.
- En la página oficial, en la sección download, pueden descargar la última versión de Scapy en .zip o .tar.gz (Scapy's latest release).
- También existen un .deb (para instalar en sistemas basados en Debian, como Ubuntu) y un .rpm (para instalar en sistemas basados en RedHat, como Fedora). Aunque no es seguro que tengan la última versión.
- Habiendo descargado .zip o el .tar.gz, una vez extraído en algun directorio (usualmente de nombre scapy-x.y.z), ejecutan desde una consola:

```
$ sudo python setup.py install
```

Tarea a realizar

- Se proveerá un programa (kickstart) de manera que pueda **capturar** archivos enviados por protocolo HTTP sobre TCP/IP
- El programa implementado tiene que (al menos) poder guardar archivos de tipo .jpg, .ogg (audio) y .ogg (video), enviados en una conexión HTTP sobre TCP, a partir de paquetes capturados en un archivo .pcap proveído por la cátedra.
- Se debería ser capaz de capturar cualquier archivo que tenga un tipo MIME declarado en el encabezado del pedido HTTP a partir de cualquier archivo de captura.

Tarea a realizar

- Esto se logrará completando:
  - La clase Connection_status desde el esqueleto (docstring y prototipos en el kickstart) para que pueda identificar cuando una conexión fue establecida y cuando una conexión se cerró.
  - La clase Reassembler desde el esqueleto (docstring y prototipos en el kickstart) para que pueda reconstruir el stream de datos enviado en una conexión TCP.
- Reensamblar TCP/HTTP correctamente todas las veces es una tarea compleja, por lo que para este laboratorio **no** es necesario que su programa pueda capturar todos los archivos todas las veces. Es suficiente con que su programa se comporte bien en las situaciones más comunes (este no es software crítico).

¿Preguntas?