Ingeniería de Servidores

Seguridad en servidores

Explotación de vulnerabilidades

Introducción

Sobre qué vamos a hablar

Gran parte de la población necesita aplicaciones informáticas en su vida personal y laboral.

- Según el Instituto Nacional de Estadística usó Internet el 76% de la población española durante 2014.

La pregunta es, ¿están utilizando las aplicaciones que creen que están usando? La respuesta es claramente **NO**, y esto es debido a las vulnerabilidades que presentan.

Algunas de estas vulnerabilidades son el *Buffer Overflow*, el *SQL Injection*, el *Format String Bugs* y el *Cross-Site-Scripting (XSS)*.

Introducción 2

Por qué es importante

- Casi todas las empresas están informatizadas en la actualidad.
- Una pérdida de datos influye a todos los agentes relacionados con la propia empresa (confidencialidad de datos).
- Clientes que han pagado por un servicio necesitan que esté operativo siempre (disponibilidad).
- Si un servicio no funciona como debería: pérdida de clientes, pérdidas económicas y mala reputación.

Algunos casos históricos recientes son:

- Ataque a Sony Pictures, 2014.
- Ataque a la AppStore, 2015.

Exploits

Descripción y tipos

- **Exploit**

  Un exploit es una secuencia de acciones que tiene el fin de aprovechar una vulnerabilidad. Por medio de un error se busca obtener algún beneficio del sistema que no estaba considerado. Se utiliza un **payload**: es la parte del exploit que contiene el código que desea ejecutarse en la máquina vulnerada mediante el exploit. El objetivo puede obtener algún tipo de privilegio en la máquina objetivo.

- Exploits conocidos

  Todos los días aparecen nuevos exploits, es enorme la cantidad que se conoce. Hay repositorios que se encargan de recopiarlos y clasificarlos, como: `https://www.exploit-db.com/`

Exploits

0-day y frameworks

Aún más peligroso que un exploit es un **0-day**. Se trata de un exploit que aún no es conocido por los desarrolladores del software afectado y, por lo tanto, no existe ningún parche ni medida para afrontarlo.

Hay diversos frameworks que facilitan el trabajo de explotación por medio de los exploits. Estos frameworks incorporan herramientas para facilitar las auditorias a sistemas y servidores.

- Metasploit Framework
- Core Impact
- Canvas

Metasploit Framework

Qué es Metasploit Framework

Se trata de un proyecto *Open Source* compuesto por módulos con diferentes funcionalidades relacionadas con la seguridad.

Algunas de sus características más llamativas son:

- Está disponible en modo consola (msfconsole) y modo gráfico (Armitage, web UI).
- Contiene plugins adicionales: msfpayload, msfencode, etc.
- Su núcleo está formado por bibliotecas de las cuales dependen el resto de funcionalidades.

Metasploitable2

Ejemplo 1: entorno de explotación y búsqueda

En este ejemplo se usará un laboratorio virtual llamado **Metasploitable2**.

Comenzamos analizando la máquina objetivo mediante Nmap con el objetivo de buscar una vía de ataque.

```
Kali:~$ sudo nmap -sV -O -p "*" 192.168.56.33
```

Figure: Lanzando Nmap (muestra versiones, SO, para todos los puertos del objetivo)

```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
```

Figure: Seleccionando una vía de ataque

Metasploitable2

Ejemplo 2: explotación

Estudiaremos si hay alguna vulnerabilidadn conocida para esa versión concreta de **vsftpd** y lanzaremos el ataque. Para ello abrimos la *msfconsole* y hacemos lo siguiente:

- **search vsftpd**: Buscamos en los exploits disponibles.
- **use exploit “nombre_exploit"**: usamos el exploit.
- **set RHOST ip_destino**: indicamos máquina destino.
- **show payloads**: vemos los payloads disponibles para el exploit.
- **set PAYLOAD “payload”**: indicamos el payload a utilizar.
- **exploit**: Lanzamos el exploit.

Metasploitable2

Ejemplo de acceso indebido ftp

Como esta versión de vsftpd instalada en la máquina es vulnerable, obtendremos un acceso a la máquina objetivo por medio del payload que ha inyectado el exploit, tal y como podemos ver en la siguiente imagen:

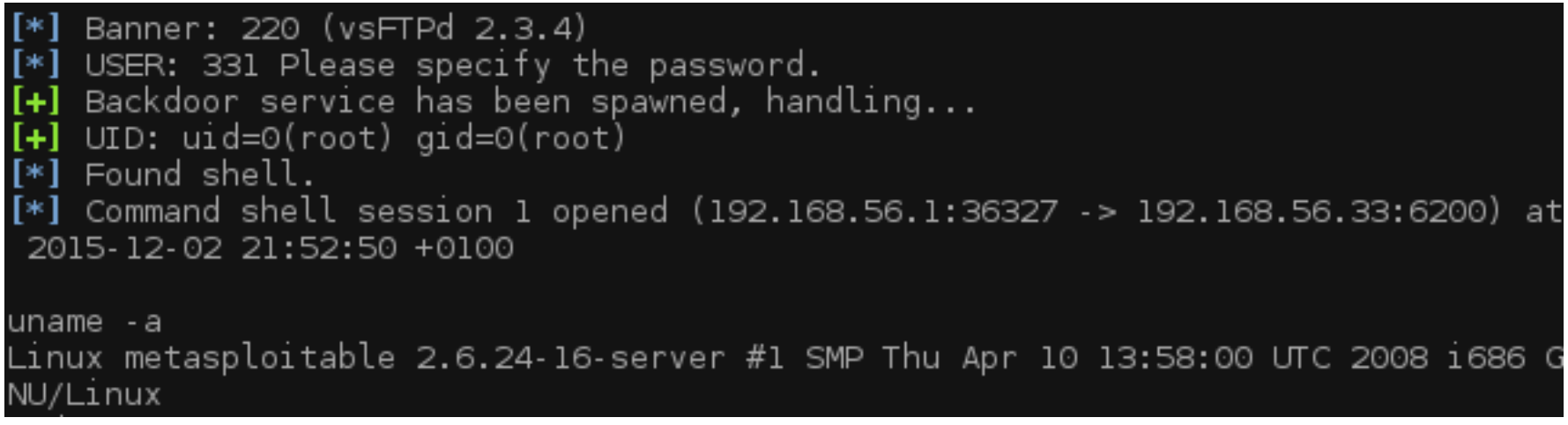

Figure: Hemos obtenido acceso privilegiado en la máquina vulnerable

Desarrollo de exploits

Cómo se desarrolla un exploit

Buscar un fallo en el software (bug) que permita inyectar y ejecutar código en la memoria a nuestro antojo.
Importante determinar el tipo de error: instrucción ilegal, violación de segmento,... En función del tipo de error podremos proceder de un modo u otro. Al igual que es importante saber si el error se produce por una entrada del usuario. Recomendable desensamblar y depurar.
Es complicado debido a que la memoria es difícil de precedecir. Hay que conseguir alojar código en memoria y saber la dirección exacta de esta (teniendo en cuenta los desplazamientos que sufre la memoria). Y posterioremente lograr que el software vulnerable pase a ejecutar esa dirección.

Prevención, detección y recuperación

Cómo estar a salvo

- **Prevención**: instalar los servicios fundamentales, mantenerlos actualizados y tener un firewall bien configurado.
- **Detección**: Sistemas de detección de intrusos (IDS); basados en red, basados en host e híbridos. Por ejemplo: CISCO, ISS, ISA SERVER, Snort, etc.
- **Recuperación**: Copias de seguridad. Algunos software especializados son Acronis, AOMEI Backupper Professional, etc.

Legalidad

Un terreno peligroso

- Es muy difícil determinar el límite de la legalidad.
  Muchas dudas y desconocimiento de leyes; varían en función del país en el que se realiza. Antes de realizar una auditoría es imprescindible marcar todos los límites y que todas las partes participantes estén de acuerdo y hayan firmado
- Grandes multas económicas, incluso años de cárcel.

Por ejemplo:

- Comprometer la seguridad de un sistema, aunque no se hayan obtenido datos: multa y 1 año de cárcel, hasta 10 años de cárcel si se repite.

Conclusión

Consideraciones finales

- Ningún sistema está exento de ataques informáticos en la actualidad. Estos ataques pueden provocar pérdidas y costes importantes. Por ello es importante estar prevenido.
- A menudo aparece nuevas vulnerabilidades en el software que afectan a multitud de sistemas. Cuando los desarrolladores lo descubren pasan de ser 0-days, a exploits, ya que estos tratarán de corregirlos y los usuarios sabrán a qué están expuestos. Por ello es importante mantener el software actualizado.
- Si un software sufre una vulnerabilidad, es bastante sencillo poder explotarla con herramientas como Metasploit Framework.