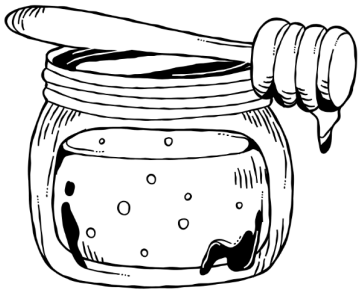
Pestaña 1

Cowrie Honey-Pot

Laboratorio de Sistemas Operativos y Redes

UNQ - 1c2025



Bottiggi Ornella Victoria

Ghioni Ian Malcolm

Marzaroli Candela

Introducción y explicación

- ¿Qué es un HoneyPot?

Un **honeypot** es una herramienta de seguridad informática que se instala en una red/dispositivo diseñado para proteger de un ataque informático. El uso principal que tienen es el de distraer posibles atacantes de información y máquinas importantes de la red. Un atacante, al intentar ingresar a una red o a una máquina preparada con un Honeypot, cuando lo intente podrá creer que ingresó efectivamente al sistema y empezar su ataque, pero en realidad va a estar interactuando con un sistema falso y todas sus acciones quedan registradas para ser estudiadas.

Un sistema de red con múltiples Honeypots en funcionamiento se conoce como *Honeynet*.

Existen 4 tipos de honeypots:

- *Honeypots puros*: Es un sistema de producción completo. Son sistemas completos y operativos, que se comportan como sistemas de producción reales, incluyendo datos y servicios simulados, pero que en realidad son un señuelo para el atacante.
- *Honeypots de alta interacción*: Simula un sistema de producción completo. Imitan las actividades de servicios de un sistema real, pero en realidad no es nada más que una simulación con fuertes herramientas de monitoreo.
- *Honeypots de baja interacción*: Simula solo un servicio en específico y con respuestas muy limitadas. Son simples y fáciles de desplegar. El atacante no puede ejecutar comandos reales ni obtener acceso real.
- *Honeypots de media interacción:* Simulan un servicio como los de baja interacción, pero son menos limitados. Permiten comandos y dar la ilusión de estar dentro de un sistema, pero no tiene un sistema operativo real detrás del mismo que se vea afectado por los comandos. En esta categoría entra **Cowrie**.

- ¿Qué es Cowrie?

Cowrie es un honeypot de media interacción que simula ser un servicio SSH/Telnet diseñado con la intención de recopilar información de hackers/atacantes. Es un software de código abierto desarrollado con python que sirve principalmente para registrar los ataques que se realicen en el sistema en que Cowrie esté corriendo.

Cowrie permite configurarse para simular ser un sistema de archivos realista. Pueden configurarse los archivos que aparecen en el sistema, las credenciales de acceso, los comandos que pueden ejecutar y el output de los mismos.

Los atacantes se conectan a través de SSH/Telnet, e ingresan al Honeypot que simula ser un sistema operativo. Los atacantes pueden ejecutar comandos, crear carpetas, archivos, cargar y ejecutar malware, pero ninguna de esas acciones tendrán un efecto real, y todas las acciones quedan guardadas en un documento.

Las acciones de los atacantes se guardan con la ip y puerto del atacante, el número de sesión, la hora, el evento y además información relacionada al evento.
Los eventos que Cowrie registra son:

- **session.connect:** Un intento de conexión.
- **login.success:** Un atacante ingresó efectivamente.
- **login.failed:** Un atacante falló al intentar ingresar.
- **command.input:** Se registra el comando que un atacante ejecuto en el sistema
- **session.file_upload:** Se registra la subida de un archivo.
- **session.closed:** Cuando un atacante finaliza su sesión con nuestro sistema.

Instalación paso a paso

1) Ya que Cowrie fue desarrollado con Python, se deben instalar las dependencias correspondientes
```
> sudo apt-get install -y python3-venv python3-dev
libssl-dev libffi-dev build-essential
```

2) Dejar andando el entorno virtual de Python para Cowrie. Esto es para que las dependencias queden aisladas y evitar conflictos.
```
> python3 -m venv cowrie-env
```

3) Activar el entorno de Cowrie.
```
> source cowrie_env/bin/activate
```

4) Clonar el repo de git de Cowrie.
```
> git clone https://github.com/cowrie/cowrie.git
```

5) Usando el Instalador de paquetes de Python (pip), instalar los paquetes de Cowrie necesarios indicados por el .txt.
```
>cd cowrie
>pip install -r requirements.txt
```

6) Copiar el archivo de configuración de ejemplo y pegar en un nuevo archivo. De esta manera, Cowrie queda configurado con su configuración defecto
```
> cp etc/cowrie.cfg.dist etc/cowrie.cfg
```

7) Copiar el archivo de ejemplo de usuarios en un archivo usersdb.txt, que Cowrie va a utilizar. En este archivo se encuentran los nombres de usuario y contraseñas que Cowrie deja ingresar o no al sistema.
```
> cd etc
> cp userdb.example userdb.txt
```

8) Finalmente, con este comando iniciamos Cowrie
```
> cd project/cowrie
> bin/cowrie start
```

- Vamos a simular que somos un atacante

Desde un usuario dentro de la misma red:

1) `> ssh -p 2222 root@ipDelAtacado`
   Donde **root** sería el usuario falso. La ip del dispositivo que esta ejecutando Cowrie y que se quiere atacar, se puede obtener con el comando ifconfig.
   Necesitamos indicar el puerto con el comando **-p** ya que por defecto ssh escucha en el puerto 22 y Cowrie escucha en el puerto 2222.
   En caso de que se quiera que Cowrie escuche el puerto 22, se puede configurar para que cualquier tráfico que reciba el puerto 22 se redireccione al puerto 2222 con el siguiente comando
   `> sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222`

2) Te va a pedir una contraseña, no se debe poner **root** ni **123456** ya que por defecto se excluyeron en el archivo de usersdb.txt

3) Una vez ingresada la contraseña, ya estás dentro del sistema falso.

Si vamos al directorio *cowrie/var/log/cowrie,* ahí es donde quedarán registrados todo lo que los atacantes hagan y todas las interacciones que tengan con nuestro equipo. Y nuestro equipo estará completamente seguro y protegido de los intentos de ataque que sufra.

Programa de alertas por Telegram

Para que, en caso de que se quiera dar aviso de que un atacante logró ingresar a nuestro sistema, podemos hacerlo con este simple script, que requiere de un token (Este se obtiene al crear un bot de telegram a través de BotFather) y un id de chat, que viene a ser el usuario de telegram destino de ese mensaje. De esta manera se nos notifica cuando un atacante ingresó a nuestro honeypot por medio de un mensaje de Telegram.

- Es necesario para el programa instalar **jq** con el comando:
  `> sudo apt install jq`

Jq es una herramienta que sirve para leer archivos de tipo *JSON* y filtrar palabras. En este caso lo necesitamos para leer el archivo **cowrie.json**, donde se registran todos los eventos.
El siguiente script debe guardarse como alertaTelegram.sh en la raíz del proyecto **cowrie/**:

```
#!/bin/bash

TOKEN="insertar_token"
CHAT_ID="insertar_chat_id"
LOG_PATH="var/log/cowrie/cowrie.json"

tail -F "$LOG_PATH" | while read line; do
    eventid=$(echo "$line" | jq -r '.eventid')

    if [[ "$eventid" == "cowrie.login.success" ]]; then
        MENSAJE="Se_Produjo_Un_Login"
        curl -s -X POST
"https://api.telegram.org/bot$TOKEN/sendMessage?chat_id=$CHAT_ID&text=$ME
NSAJE"
    fi
done
```

Una vez hecho eso, le damos permisos de ejecución:

`> chmod +x alertaTelegram.sh`

Y luego lo ejecutamos con el entorno virtual de python ya activado (Ver paso 2 de la instalación paso a paso):

```
> ./alertaTelegram.sh
```

Con esto, el programa queda corriendo en primer plano en la consola, y cada vez que un atacante ingrese a nuestro honeypot, nos lo avisa a través del bot de telegram correspondiente.