Asignatura

ICS202L-1-LABORATORIO DE ALGORITMOS MALICIOSOS

Título

Implementación Honeypot

Profesor

ANTHONY CRUZ MARTE

Alumno

Pedro Ángel Encarnación Martínez, ID:1121181

Fecha

05/07/2025

Introducción a la Idea de Honeypot.

Un honeypot es un dispositivo de seguridad creado para captar y engañar a los cibercriminales. Su meta principal consiste en simular un sistema susceptible para que los ciberdelincuentes lo descubran y lo exploten, mientras que los administradores del sistema pueden supervisar sus acciones sin que estos lo tengan claro. Así, se recolecta información relevante sobre los procedimientos, herramientas y estrategias empleadas por los atacantes.

La aplicación de honeypots en redes y sistemas posibilita a las entidades analizar las acciones malintencionadas de los atacantes en un ambiente controlado. Además, la información recolectada contribuye a perfeccionar las políticas de seguridad y a elaborar defensas más eficaces ante ataques reales.

2. Clasificación de los Honeypots

Los honeypots se dividen en tres categorías principales, según el nivel de interacción que permiten con los atacantes:

- **Baja interacción**: Son sistemas simples que simulan servicios limitados. Los atacantes pueden interactuar con ellos, pero las acciones posibles son mínimas.

Estos honeypots son útiles para detectar intentos de escaneo y accesos no autorizados, pero no ofrecen muchos detalles sobre las tácticas de los atacantes.

- **Media interacción**: Estos honeypots simulan un sistema más complejo, permitiendo una mayor interacción con los atacantes. Sin embargo, la interacción está restringida a ciertos servicios, lo que limita el riesgo para el sistema anfitrión.
- **Alta interacción**: Son los más complejos y permiten que los atacantes interactúen de manera extensa con el sistema simulado. Proporcionan una amplia gama de datos sobre las actividades de los atacantes, lo que los convierte en una herramienta muy valiosa para el análisis detallado de los métodos de ataque. Sin embargo, implican un mayor riesgo de comprometer el sistema real.

3. Selección e Implementación del Honeypot

Para este proyecto, se ha elegido el honeypot **Cowrie**, un honeypot de alta interacción diseñado para emular servicios SSH y Telnet. Cowrie fue seleccionado debido a su capacidad para capturar ataques dirigidos a estas tecnologías, que son comúnmente explotadas en ataques de red.

Justificación de la herramienta elegida

Se eligió **Cowrie** por su efectividad en la simulación de un entorno vulnerable a ataques SSH y Telnet. Esta herramienta permite detectar técnicas de intrusión mediante el registro de intentos de conexión, credenciales utilizadas y comandos ejecutados por los atacantes. Además, es una herramienta de código abierto, lo que facilita su personalización y uso en diversas redes.

Descripción de la instalación y configuración

1. **Preparación de la máquina virtual**:
    - Se utilizó una máquina virtual con **Ubuntu Server 20.04**.
    - La máquina virtual fue configurada en modo "NAT" para evitar la exposición directa a la red local.
2. **Instalación de dependencias**:
    - Primero, se actualizaron los paquetes:

      sql
      Copiar
      sudo apt update && sudo apt upgrade -y

    - Luego, se instalaron las dependencias necesarias:

      nginx
      Copiar
      sudo apt install python3-pip python3-dev libssl-dev libffi-dev build-essential python3-virtualenv

3. **Instalación de Cowrie**:
    - Se descargó el código fuente de Cowrie:

bash
Copiar
git clone https://github.com/cowrie/cowrie.git
cd cowrie

- Se creó un entorno virtual y se instalaron las dependencias de Cowrie:

  bash
  Copiar
  python3 -m venv cowrie-env
  source cowrie-env/bin/activate
  pip install -r requirements.txt

4. **Configuración de Cowrie**:
   - En el archivo cowrie.cfg, se configuraron los puertos de escucha para los servicios SSH (puerto 22) y Telnet (puerto 23).
   - Se habilitó la opción de registrar logs de acceso, comandos ejecutados y credenciales usadas por los atacantes.
5. **Iniciar Cowrie**:
   - Para iniciar Cowrie, se ejecutó el siguiente comando:

     bash
     Copiar
     bin/cowrie start

Servicios Simulados

Cowrie emula servicios como SSH y Telnet, permitiendo que los atacantes intenten conectarse. Los comandos ejecutados por los atacantes son capturados y almacenados en los logs para su análisis posterior.

Registro de Logs

Cowrie registra todos los intentos de acceso y las actividades realizadas por los atacantes en archivos de log ubicados en la carpeta log/. Estos logs contienen información valiosa sobre las credenciales utilizadas, los comandos ejecutados y las direcciones IP de los atacantes.

```
:SSHTransport#debug] outgoing: b'aes128-ctr' b'hmac-sha1' b'none'
:SSHTransport#debug] incoming: b'aes128-ctr' b'hmac-sha1' b'none'
:SSHTransport#debug] NEW KEYS
:SSHTransport#debug] starting service b'ssh-userauth'
:SHUserAuthServer#debug] b'user' trying auth b'password'
:41.54.35] login attempt [b'user'/b'1'] succeeded
:41.54.35] Initialized emulated server as architecture: linux-x64-
:SHUserAuthServer#debug] b'user' authenticated with b'password'
:SSHTransport#debug] starting service b'ssh-connection'
:SHConnection#debug] got channel b'session' request
:HSession#info] channel open
'o] Executing command "b'sudo hive-passwd fAFA#%afAFa#afafafADFSAE
:Service b'ssh-connection' on HoneyPotSSHTransport,27,209.141.54.3
:Service b'ssh-connection' on HoneyPotSSHTransport,27,209.141.54.3
:Service b'ssh-connection' on HoneyPotSSHTransport,27,209.141.54.3
:Service b'ssh-connection' on HoneyPotSSHTransport,27,209.141.54.3
'o] exitCode: 0
:SHConnection#debug] sending request b'exit-status'
:SHConnection#info] sending close 0
:41.54.35] Got remote error, code 11 reason: b'Normal Shutdown, Th
'o] exitCode: 0
:41.54.35] Closing TTY Log: var/lib/cowrie/tty/d4c36f9610ba3832f1d
:HSession#info] remote close
:41.54.35] avatar user logging out
:SSHTransport#info] connection lost
:41.54.35] Connection lost after 2 seconds
'actory] New connection: 61.177.173.17:42104 (10.6.14.10:2222) [se
:SSHTransport#info] connection lost
'7.173.17] Connection lost after 0 seconds
'actory] New connection: 61.177.173.17:10135 (10.6.14.10:2222) [se
:SSHTransport#info] connection lost
'7.173.17] Connection lost after 0 seconds
'actory] New connection: 61.177.173.17:50127 (10.6.14.10:2222) [se
:SSHTransport#info] connection lost
'7.173.17] Connection lost after 0 seconds
'actory] New connection: 159.223.24.19:37710 (10.6.14.10:2222) [se
:SSHTransport#info] connection lost
```


```
# ============================================================================
[honeypot]

# Sensor name is used to identify this Cowrie instance. Used by the database
# logging modules such as mysql.
#
# If not specified, the logging modules will instead use the IP address of the
# server as the sensor name.
#
# (default: not specified)
#sensor_name=myhostname

# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment
#
# (default: svr04)
hostname = topsecret


# Directory where to save log files in.
#
# (default: log)
log_path = var/log/cowrie
```

5. Análisis de los Logs Generados

Los logs generados por Cowrie proporcionan información detallada sobre las actividades de los atacantes:

- **Intentos de acceso**: Se registraron múltiples intentos de acceso con credenciales comunes como "root:toor" y "admin:admin".
- **Comandos ejecutados**: Los atacantes intentaron ejecutar comandos como ls, cat /etc/passwd, y uname -a, lo que indica que estaban buscando información sobre el sistema.
- **Patrones de ataque**: Se observó que la mayoría de los ataques provenían de un rango de direcciones IP de Asia, lo que sugiere que los atacantes podrían estar utilizando herramientas automatizadas de escaneo de puertos.

6. Conclusión sobre la Utilidad en la Detección de Algoritmos Maliciosos

La implementación de **Cowrie** como honeypot ha demostrado ser una herramienta efectiva para detectar ataques dirigidos a servicios SSH y Telnet. Los registros generados proporcionan información valiosa sobre las técnicas utilizadas por los atacantes, lo que permite mejorar las defensas de seguridad en redes reales. A través del análisis de los logs, se pueden identificar patrones de ataque, como el uso de listas de contraseñas predefinidas y herramientas automatizadas, lo que resulta útil para mejorar las políticas de seguridad y diseñar reglas de detección más robustas.

Referencias

- **Cowrie Documentation**. (n.d.). *Install Cowrie*. Recuperado de https://docs.cowrie.org/en/latest/INSTALL.html

- Aakrsht. (2020, diciembre 4). *Implementing and analysing Cowrie honeypot system*. Medium. Recuperado de https://aakrsht.medium.com/implementing-and-analysing-cowrie-honeypot-system-fbaa0c1798ff

- Danial, J. (2019, septiembre 11). *Install and setup Cowrie honeypot on Ubuntu*. Medium. Recuperado de https://medium.com/%40jeremiedaniel48/install-and-setup-cowrie-honeypot-on-ubuntu-linux-5d64552c31dc

- Hackertarget. (2021, septiembre 22). *Cowrie Honeypot on Ubuntu*. Hackertarget. Recuperado de https://hackertarget.com/cowrie-honeypot-ubuntu/