Honeypot

Índex

Cicle formatiu: CFGS Administració de sistemes informàtics en xarxa (ASIX)

Mòdul: 0378 - Seguretat i alta disponibilitat

Sistema operatiu: Ubuntu Server 26.04 LTS

---

1. Què és un Honeypot?

Un **honeypot** és un sistema o servei informàtic dissenyat expressament per semblar vulnerable o atractiu per a un atacant, però que en realitat està aïllat i monitorat. La seva funció no és prestar cap servei real, sinó **detectar, desviar i estudiar** intents d'intrusió.

Es tracta d'un "esquer": un sistema que simula tenir serveis oberts, vulnerabilitats o dades interessants, però que no té cap ús legítim. Per tant, **qualsevol interacció amb ell és sospitosa per definició** --- cap usuari legítim hauria de connectar-s'hi mai.

Figura 1: Honeypot

1.1. Per a què serveix

- **Detecció precoç**: qualsevol connexió és una alerta d'activitat maliciosa, sense falsos positius habituals.
- **Estudi d'atacs**: permet observar tècniques, eines i comportament dels atacants (per exemple, quines credencials proven, quines comandes executen).
- **Desviació**: allunya l'atacant dels sistemes reals, fent-li perdre temps en un entorn fals.
- **Intel·ligència d'amenaces**: recull IPs, patrons, malware descarregat, etc.

1.2. Tipus

- **Baixa interacció**: només simula serveis (ports oberts, banners), risc mínim, informació limitada.
- **Alta interacció**: sistema real o molt realista (com **Cowrie**, que simula un servidor SSH/Telnet complet), permet capturar molta més informació, però requereix més aïllament i control.

2. Crea un Honeypot SSH amb Cowrie

Desplega un honeypot **Cowrie** en una VM aïllada del laboratori, captura les credencials i comandes que introdueixin els atacants, i envia aquesta informació a **Wazuh** per correlacionar-la amb la resta d'alertes de seguretat.

AVÍS

Un honeypot mal aïllat és un risc, no una eina de seguretat. Abans de connectar-lo a Internet, revisa la secció: “Aïllament de xarxa”.

2.1. Arquitectura

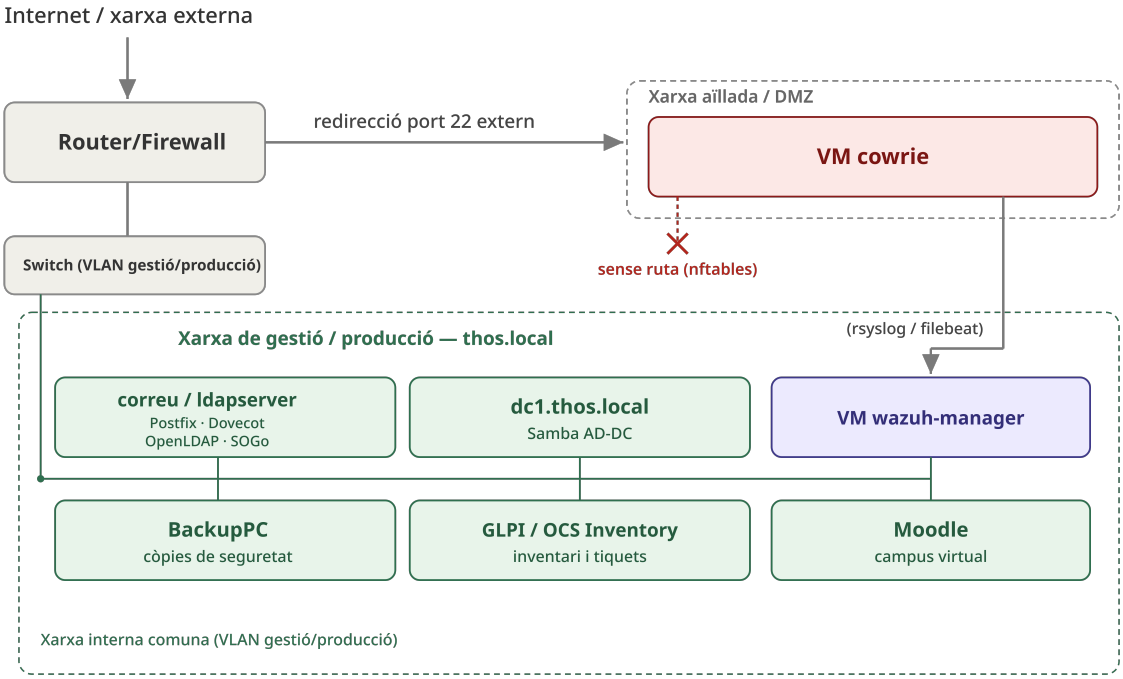

Figura 2: Arquitectura

La VM `cowrie` **no** ha de tenir accés a la resta de la xarxa `thos.local` (ni al domini Samba AD-DC, ni a BackupPC, ni a Moodle). Se li permet únicament sortida cap al gestor de Wazuh pel port 1514/1515.

2.2. Preparació de la VM

VM nova a VirtualBox, adaptador de xarxa en mode **intern** o en una xarxa VLAN separada.

Actualitza la llista de paquets:

```
sudo apt update
```

Actualitza els paquets:

```
sudo apt upgrade
```

Instal·la els paquets:

```
sudo apt install git python3-venv python3-dev python3-pip libssl-dev \
  libffi-dev build-essential libpython3-dev authbind
```

Crea un usuari sense privilegis per executar Cowrie (mai com a root):

```
sudo adduser --disabled-password cowrie
sudo su - cowrie
```

2.3. Instal·lació de Cowrie

```
git clone http://github.com/cowrie/cowrie
cd cowrie
python3 -m venv cowrie-env
source cowrie-env/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

2.4. Configuració bàsica

```
cp etc/cowrie.cfg.dist etc/cowrie.cfg
nano etc/cowrie.cfg
```

Paràmetres clau a `cowrie.cfg`:

```
[honeypot]
hostname = srv-intranet01
listen_endpoints = tcp:2222:interface=0.0.0.0

[ssh]
enabled = true
listen_endpoints = tcp:2222:interface=0.0.0.0
```

```
[telnet]
enabled = true
listen_endpoints = tcp:2223:interface=0.0.0.0

[output_jsonlog]
enabled = true
logfile = var/log/cowrie/cowrie.json
```

Cowrie no pot escoltar directament al port 22 (privilegiat) sense privilegis. Fes servir `authbind` o una redirecció amb `iptables`/`nftables` (recomanat: iptables des de l'usuari root, Cowrie continua sense privilegis):

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT
 ↪  --to-port 2222
sudo iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT
 ↪  --to-port 2223
```

NOTA

Si a la mateixa VM necessites accedir per SSH real d'administració, mou el servidor SSH legítim a un altre port (per exemple 2200) al `/etc/ssh/sshd_config` **abans** d'aplicar la redirecció anterior.

2.5. Personalització del sistema de fitxers fals

Cowrie inclou un filesystem fals (`share/cowrie/fs.pickle`) i permet personalitzar:

- `honeyfs/etc/passwd`, `honeyfs/etc/hostname` --- per fer-lo creïble
- `txtcmds/` --- sortides de comandes com `uname -a`, `lsb_release -a`
- Usuaris i contrasenyes vàlides a `etc/userdb.txt` (per exemple, permetre sempre `root:root` o `admin:admin` per facilitar l'accés a l'atacant)

```
echo "root:x:!root" >> etc/userdb.txt
echo "admin:x:admin" >> etc/userdb.txt
```

2.6. Aïllament de xarxa (imprescindible)

Regles mínimes amb `nftables` a la VM `cowrie` perquè només pugui parlar amb el gestor Wazuh:

```
sudo nft add rule inet filter output ip daddr <IP_WAZUH_MANAGER>
 ↪  accept
sudo nft add rule inet filter output drop
```

No li donis mai ruta cap a `dc1.thos.local`, ni cap al segment de gestió (BackupPC, GLPI, OCS, LDAP).

2.7. Executar Cowrie com a servei (systemd)

```
sudo tee /etc/systemd/system/cowrie.service > /dev/null <<'EOF'
[Unit]
Description=Cowrie SSH/Telnet Honeypot
After=network.target

[Service]
Type=forking
User=cowrie
WorkingDirectory=/home/cowrie/cowrie
ExecStart=/home/cowrie/cowrie/bin/cowrie start
ExecStop=/home/cowrie/cowrie/bin/cowrie stop
Restart=on-failure

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl daemon-reload
sudo systemctl enable --now cowrie
sudo systemctl status cowrie
```

2.8. Comprovació

Des d'una altra màquina de la xarxa interna:

```
ssh admin@<IP_COWRIE>
```

Revisa els logs generats:

```
tail -f /home/cowrie/cowrie/var/log/cowrie/cowrie.json
```

Cada línia és un event JSON (login, comanda executada, fitxer descarregat, etc.).

3. Integració amb Wazuh

Opció A: agent Wazuh + monitoratge de fitxer de log

Instal·la l'agent Wazuh a la VM `cowrie` i afegeix al `ossec.conf` de l'agent:

```
<localfile>
  <log_format>json</log_format>
  <location>/home/cowrie/cowrie/var/log/cowrie/cowrie.json</location>
</localfile>
```

Opció B: Filebeat directe al manager

Alternativa sense agent complet, útil si vols mantenir la VM encara més minimalista.

Regles de detecció al Wazuh Manager

Crea una regla personalitzada a `/var/ossec/etc/rules/local_rules.xml`:

```
<group name="cowrie,honeypot,">
  <rule id="100100" level="10">
    <decoded_as>json</decoded_as>
    <field name="eventid">cowrie.login.success</field>
    <description>Cowrie: login reeixit al honeypot SSH</description>
  </rule>
  <rule id="100101" level="5">
    <decoded_as>json</decoded_as>
    <field name="eventid">cowrie.command.input</field>
    <description>Cowrie: comanda executada al honeypot</description>
  </rule>
</group>
```

Reinicia el manager:

```
sudo systemctl restart wazuh-manager
```

A partir d'aquí, qualsevol interacció amb el honeypot generarà una alerta a la interfície de Wazuh, ja classificada i correlacionable amb la resta de l'entorn `thos.local`.

4. Manteniment i bones pràctiques

- Revisa i arxiva `cowrie.json` periòdicament (pot créixer molt ràpidament si rep escombraries d'Internet).
- Actualitza Cowrie regularment (`git pull + pip install -r requirements.txt`).
- No reutilitzis mai contrasenyes reals de `thos.local` al honeypot.
- Considera exposar-lo només des d'una IP pública dedicada o mitjançant un DMZ real si vols dades d'atacants d'Internet.

Índex de figures

Versions d'aquest document

- HTML
- PDF
- ODT
- MD