

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----



## **Triển Khai OWASP Honeypot**

### **Báo Cáo Cuối Kỳ An Toàn Mạng**

**Nhóm 20:**

**N18DCAT058: Hồ Minh Phong**

**N18DCAT100: Trần Quốc Trọng**

**N18DCAT102: Huỳnh Tiến Vĩ**

**TP.HCM-2021**

## Mục Lục

### Contents

Chương 1. Cơ sở lý thuyết.....	3
1.1 Các Khái niệm .....	3
1.2 Chức năng .....	3
1.3 Giải thích thành phần và hoạt động .....	4
Chương 2. Triển khai.....	5
2.1 Mô hình.....	5
2.2 Cài đặt và cấu hình .....	6
2.2.1 Thiết lập Elasticsearch và API trên cùng 1 server .....	6
2.2.2 Thiết lập máy Honeypot (modules) .....	10
2.3 Demo hoạt động.....	12
2.3.1 Demo module ssh/weak_password.....	12
2.3.2 Demo module ssh/strong_password .....	16
2.4 Môi liên hệ giữa các container .....	17

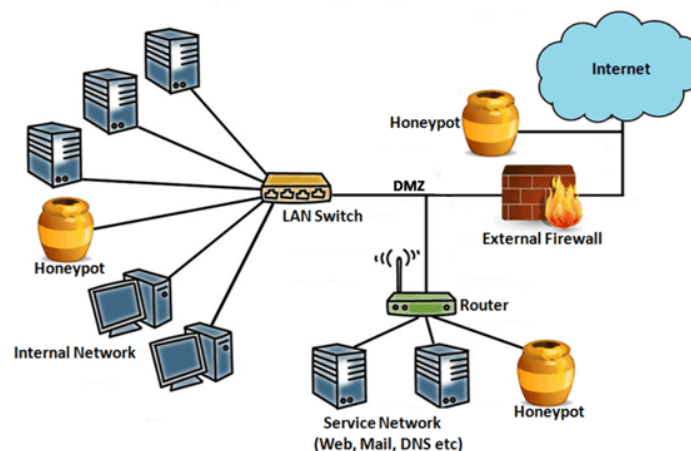
# Chương 1. Cơ sở lý thuyết

## 1.1 Các Khái niệm

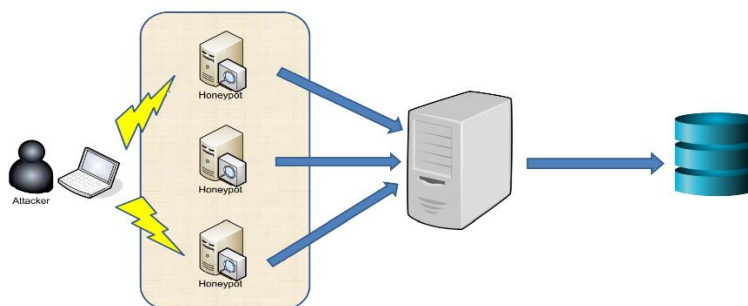
- OWASP Honeypot là một phần mềm mã nguồn mở viết bằng ngôn ngữ Python, được thiết kế để tạo honeypot bằng cách sử dụng docker container.
- Honeypot là một hệ thống gắn liền với mạng được thiết kế giả dạng các tài sản có giá trị cao như servers hoặc một lỗ hổng mạng.
- Docker Container là một gói phần mềm độc lập, có thể thực thi, bao gồm các ứng dụng và phụ thuộc của chúng.

## 1.2 Chức năng

- OWASP Honeypot thu hút, đánh lạc hướng tấn công của tin tặc, ngăn không cho chúng tấn công hệ thống thật.



- Nó phát hiện, giám sát, thu thập dữ liệu hoạt động của tin tặc để người quản trị phân tích những mối đe dọa và giải quyết mọi điểm yếu của hệ thống.



### 1.3 Giải thích thành phần và hoạt động

OWASP Honeypot có ba phần ElasticSearch server, API server và các module.

- **ElasticSearch server:** database nơi lưu trữ dữ liệu hoạt động của tin tặc hoặc thu được trong mạng.
- **API server:** cung cấp API giúp truy xuất, khai thác, phân tích những dữ liệu từ database ElasticSearch, hiển thị lên giao diện WebUI trên <https://localhost:5000>.
- **Modules:** thiết lập các honeypot và gửi dữ liệu thu thập được từ tin tặc đến ElasticSearch server.

Sau khi chạy Honeypot, sẽ tạo ra 2 cơ sở dữ liệu trên ElasticSearch là: **ohp\_event**(lưu trữ dữ liệu từ các loại event) và **ohp\_file\_archive**(lưu trữ các file bắt được trên network)

#### OHP Events

- + Honeypot Events: lưu trữ tất cả các sự kiện honeypot
- + Network Events: lưu trữ data tách biệt khỏi honeypot events, không gây hại cho server đang chạy.
- + Credential Events: lưu trữ thông tin đăng nhập từ các module `strong_password`
- + File Change Events: theo dõi đường dẫn tệp bị tin tặc thay đổi trên Honeypot
- + Data Events: lưu trữ dữ liệu thu thập được từ các module như `smtp`

Hoạt động của Honeypot qua các module có sẵn:

- **SSH (Secure Socket Shell)**
  - Module này có 2 loại `ssh/weak_password` và `ssh/strong_password`, được triển khai trong hai docker container.
  - **ssh/weak\_password:** có mật khẩu dễ đoán (123456), được sử dụng để giám sát các hoạt động của tin tặc như những lệnh mà chúng thực hiện hoặc loại file chúng tải lên sau khi đăng nhập vào hệ thống. Dữ liệu được gửi về ElasticSearch dưới dạng File Change Events.
  - **ssh/strong\_password:** không thể đăng nhập dễ dàng, dùng để lấy thông tin xác thực của tin tặc cố gắng brute force. Mỗi lần cố gắng đăng nhập SSH, username và password tin tặc sử dụng được gửi và lưu trữ trong ElasticSearch dưới dạng bản ghi Credential Events.

- FTP (File Transfer Protocol)
  - Có hai loại ftp/weak\_password và ftp/strong\_password
  - **ftp/weak\_password** được sử dụng để theo dõi loại tệp/phần mềm độc hại nào mà tin tặc tải lên sau khi đăng nhập dễ dàng vào hệ thống. Dữ liệu được gửi về Elasticsearch dưới dạng File Change Events.
  - **ftp/strong\_password** để lấy thông tin xác thực của tin tặc
- HTTP (Hyper Text Transfer Protocol Secure)
  - Module có hai loại basic\_auth\_weak\_password và basic\_auth\_strong\_password
  - **basic\_auth\_weak\_password** sử dụng để theo dõi các request tới server và nếu hacker đang cố gắng tấn công Dos thì module này có thể dễ dàng phát hiện ra.
  - **basic\_auth\_strong\_password** để theo dõi username và password mà kẻ tấn công sử dụng để xâm nhập vào hệ thống.
- SMTP (Simple Mail Transfer Protocol)
  - SMTP có một phiên bản strong\_password, ghi lại tất cả thông tin xác thực (username, password) mà kẻ tấn công sử dụng để xâm nhập vào hệ thống.

## Chương 2. Triển khai

### 2.1 Mô hình

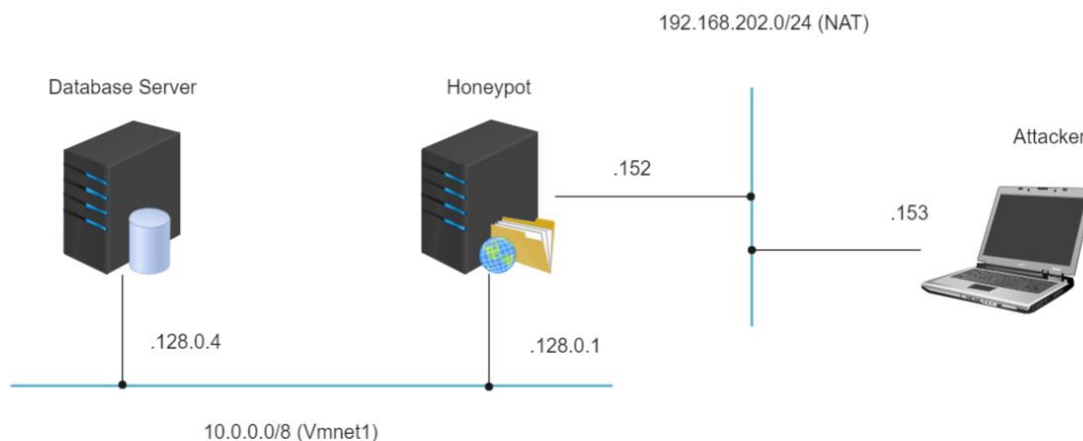


Figure 1. Mô hình demo hoạt động OWASP Honeypot

Trong đó:

- Máy Database: ElasticSearch + API server (Vmware Ubuntu 20.04)
- Máy Honeypot: Modules (Vmware Ubuntu 20.04)
- Máy Attacker: máy thật

Thông tin card mạng:

STT	Name		Interface 1	Interface 2
1	Database	inet	10.128.0.4	
		netmask	255.0.0.0	
2	Honeypot	inet	10.128.0.1	192.168.202.152
		netmask	255.0.0.0	255.255.255.0
3	Attacker	inet	192.168.202.153	
		netmask	255.255.255.0	

## 2.2 Cài đặt và cấu hình

### 2.2.1 Thiết lập ElasticSearch và API trên cùng 1 server

- **Bước 1: Cài đặt Docker**

- Chuẩn bị hệ thống

```
$ sudo apt-get update  
$ sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release
```

- Thêm Docker's official GPG key vào hệ thống

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/usr/share/keyrings/docker-archive-keyring.gpg
```

- Thiết lập Docker repository

```
$ echo \  
"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- Cài đặt Docker Engine

```
$ sudo apt-get update  
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

- **Bước 2: Cài đặt docker-compose**

- Tải Docker Compose

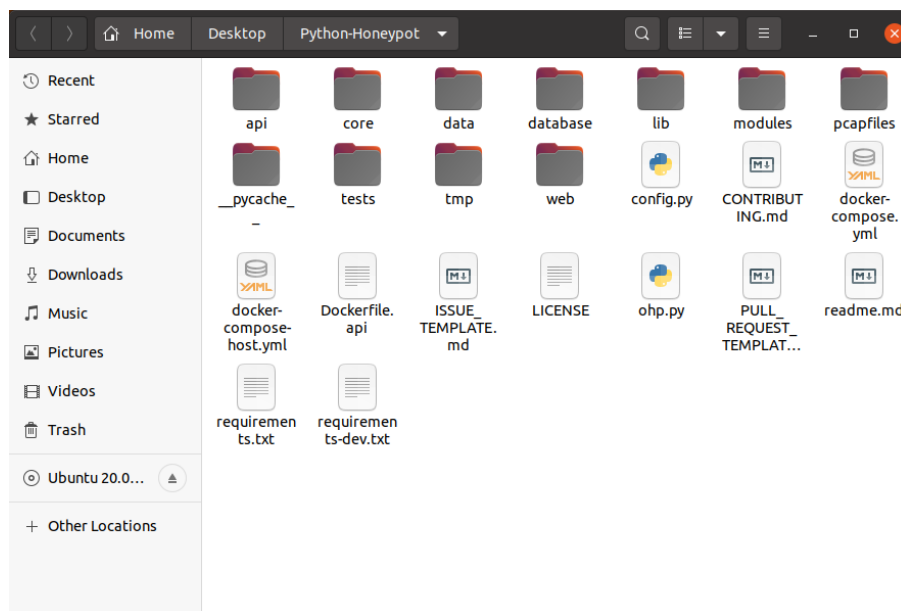
```
$ sudo curl -L  
"https://github.com/docker/compose/releases/download/1.29.2/docker-compose-  
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

- Thiết lập quyền thực thi cho docker-compose và tạo symlink

```
$ sudo chmod +x /usr/local/bin/docker-compose  
$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

- **Bước 3: Tải xuống Python-honeypot từ github và thiết lập quyền**

```
$ cd Desktop  
$ git clone https://github.com/OWASP/Python-HoneyPot.git  
$ sudo chmod 777 -R Python-HoneyPot/  
$ sudo chown -R 1000:1000 Python-HoneyPot/
```



- **Bước 4: Sửa đổi file docker-compose.yml**

- Để có thể chạy ElasticSearch 7.13.3 sửa đổi environmet:

- "ES\_JAVA\_OPTS=-Xms512m -Xmx512m"

- **Bước 5: Triển khai các container ElasticSearch, API Server bằng docker-compose**

```
$ sudo docker-compose up
```

Quá trình khởi động:

```
trantruong@trgpc:~/Desktop$ cd Python-Honeypot/
trantruong@trgpc:~/Desktop/Python-Honeypot$ sudo docker-compose up
Creating network "python-honeypot_default" with the default driver
Pulling elasticsearch (docker.elastic.co/elasticsearch/elasticsearch:7
7.13.3: Pulling from elasticsearch/elasticsearch
ddf49b9115d7: Pull complete
733fde5445ab: Pull complete
b52b722b5d76: Pull complete
a4a4d38c41c2: Pull complete
```

Figure 2. Docker tạo network cho các container

```
trantruong@trgpc:~/Desktop$ cd Python-Honeypot/
trantruong@trgpc:~/Desktop/Python-Honeypot$ sudo docker-compose up
Creating network "python-honeypot_default" with the default driver
Pulling elasticsearch (docker.elastic.co/elasticsearch/elasticsearch:7.13.3)...
7.13.3: Pulling from elasticsearch/elasticsearch
ddf49b9115d7: Pull complete
733fde5445ab: Pull complete
b52b722b5d76: Pull complete
a4a4d38c41c2: Pull complete
0270221fc6d4: Pull complete
3a92ffa864ad: Pull complete
38d800bfbaa0: Pull complete
Digest: sha256:930cdb7e960c842f89b063226bdb9374cb3a080372564b36a1c66931d4d80e09
Status: Downloaded newer image for docker.elastic.co/elasticsearch/elasticsearch:7.13.3
Building ohp
Sending build context to Docker daemon 14.52MB
Step 1/7 : FROM ubuntu:20.04
20.04: Pulling from library/ubuntu
```

Figure 3. Docker tải xuống các container từ dockerhub

```
Starting elasticsearch ... done
Starting python-honeypot_ohp_1 ... done
Starting grafana ... done
Attaching to elasticsearch, python-honeypot_ohp_1, grafana
grafana      | t=2021-10-01T07:56:23+0000 lvl=inf
grafana      | t=2021-10-01T07:56:23+0000 lvl=inf
grafana      | t=2021-10-01T07:56:23+0000 lvl=inf
```

Figure 4. Docker khởi động, attach và kết nối các container



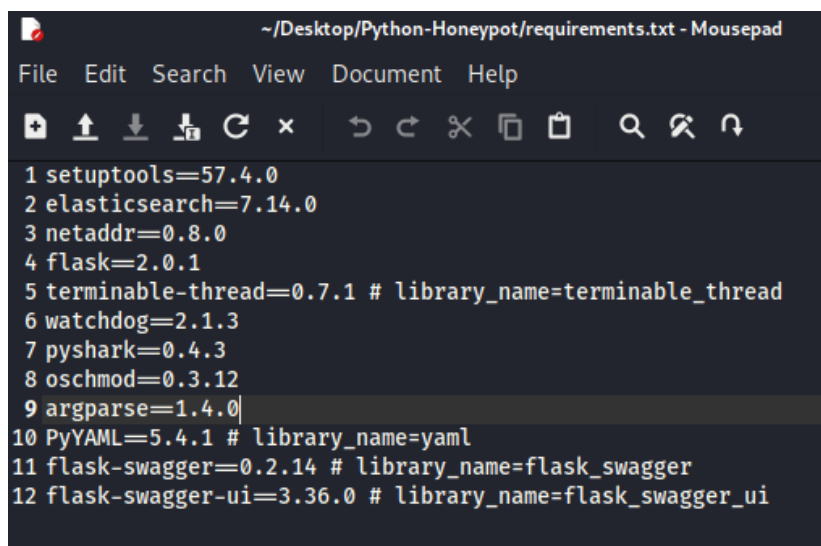
```
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
ohp_1  
  
OWASP  
HoneyPot
```

```
* API access key: NOT REQUIRED!  
* Serving Flask app 'api.server' (lazy loading)  
* Environment: development  
* Debug mode: off  
* Running on all addresses.  
WARNING: This is a development server. Do not use it in a production deployment.  
* Running on http://172.18.0.3:5000/ (Press CTRL+C to quit)
```

### 2.2.2 Thiết lập máy Honeypot (modules)

- **Bước 1: Cài đặt docker (đã trình bày ở trên)**
- **Bước 2: Tải xuống Python-honeypot từ github (đã trình bày ở trên)**
- **Bước 3: Cài đặt các gói yêu cầu/phụ thuộc**  
Để có thể chạy các modules cần cài đặt các phần phụ thuộc:

```
$ sudo pip3 install -r requirements.txt
```



The screenshot shows a text editor window titled '~/.Desktop/Python-Honeypot/requirements.txt - Mousepad'. The editor contains a list of Python dependencies with their versions and optional library names in comments. The dependencies are: setuptools=57.4.0, elasticsearch=7.14.0, netaddr=0.8.0, flask=2.0.1, terminable-thread=0.7.1 (comment: library\_name=terminable\_thread), watchdog=2.1.3, pyshark=0.4.3, oschmod=0.3.12, argparse=1.4.0, PyYAML=5.4.1 (comment: library\_name=yaml), flask-swagger=0.2.14 (comment: library\_name=flask\_swagger), and flask-swagger-ui=3.36.0 (comment: library\_name=flask\_swagger\_ui).

- **Bước 4: Đặt lại địa chỉ của ElasticSearch và API trong file config.py để Honeypot có thể kết nối và gửi dữ liệu về Server.**

```
10
19 def api_configuration():
20     """
21     API Config (could be modify by user)
22
23     Returns:
24         a JSON with API configuration
25     """
26     # DOCKER_ENV variable is set in the docker-compose file.
27     if os.environ.get('ELASTICSEARCH_DOCKER_ENV') == "true":
28         db_url = "elasticsearch:9200"
29     else:
30         db_url = "10.128.0.4:9200"
31
32     return { # OWASP Honeypot API Default Configuration
33         "api_host": "10.128.0.4",
34         "api_port": 5000,
35         "api_debug_mode": False,
36         "api_access_without_key": True,
37         "api_access_key": generate_token(), # or any string, or None
38         "api_client_white_list": {
39             "enabled": False,
40             "ips": [
```

Figure 8. config.py

- **Bước 5: Khởi động một module để kiểm tra**

Kiểm tra file cấu hình ssh/strong\_password

```
1#!/usr/bin/env python
2# -*- coding: utf-8 -*-
3
4
5def category_configuration():
6    """
7    category configuration
8
9    Returns:
10        JSON/Dict category configuration
11    """
12    return {
13        "virtual_machine_name": "ohp_sshserver",
14        "virtual_machine_port_number": 22,
15        "virtual_machine_internet_access": True,
16        "real_machine_port_number": 22|
17    }
```

Figure 9 \_\_init\_\_.py

Chạy module ssh/strong\_password

```
$ sudo python3 ohp.py -m ssh/strong_password
```

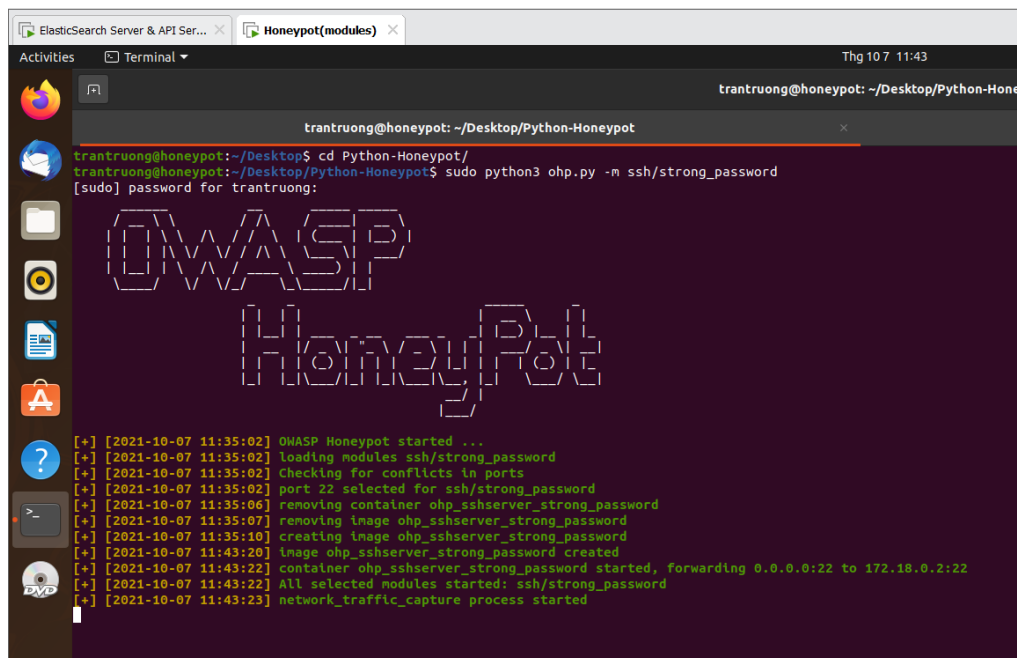


Figure 10. Cài đặt thành công

## 2.3 Demo hoạt động

### 2.3.1 Demo module ssh/weak\_password

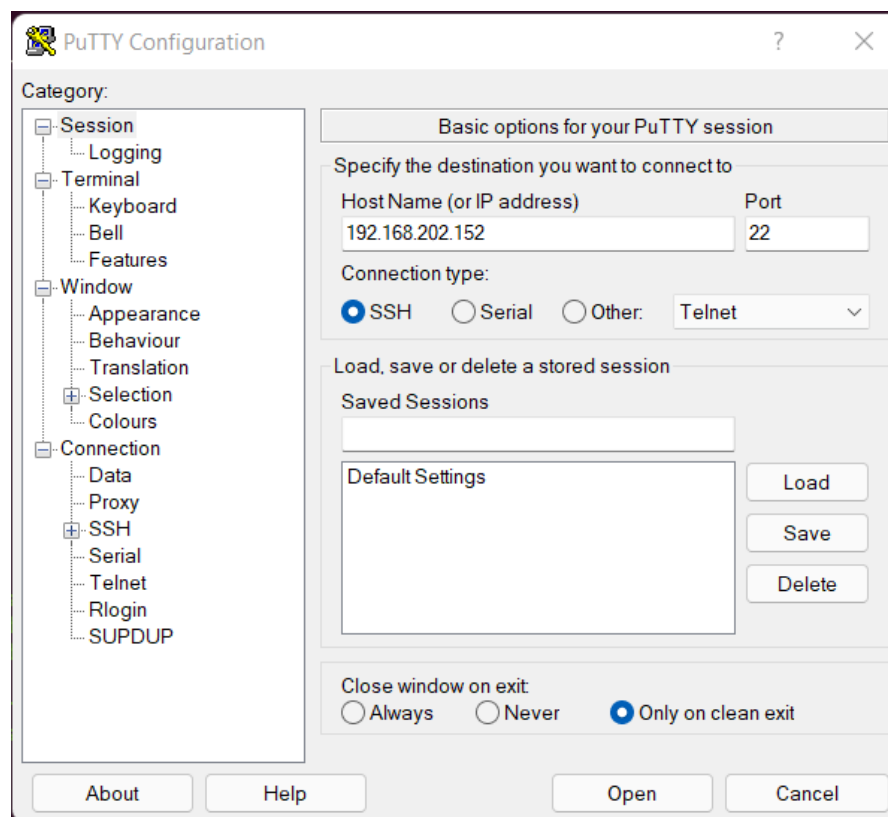
- Module này có mật khẩu yếu 123456 nên hacker có thể đoán dễ dàng.

```
trantruong@honeypot:~/Desktop/Python-HoneyPot$ sudo python3 ohp.py -m ssh/weak_password

OWASP
HoneyPot

[+] [2021-10-03 16:38:49] OWASP HoneyPot started ...
[+] [2021-10-03 16:38:49] Loading modules ssh/weak_password
[+] [2021-10-03 16:38:49] Checking for conflicts in ports
[+] [2021-10-03 16:38:49] port 22 selected for ssh/weak_password
[+] [2021-10-03 16:38:51] removing container ohp_sshserver_weak_password
[+] [2021-10-03 16:38:52] removing image ohp_sshserver_weak_password
[+] [2021-10-03 16:38:54] creating image ohp_sshserver_weak_password
[+] [2021-10-03 16:39:42] image ohp_sshserver_weak_password created
[+] [2021-10-03 16:39:43] container ohp_sshserver_weak_password started, forwarding 0.0.0.0:22 to 172.18.0.2:22
[+] [2021-10-03 16:39:43] All selected modules started: ssh/weak_password
[+] [2021-10-03 16:39:44] network_traffic_capture process started
```

- Giả sử hacker có được mật khẩu và đăng nhập được vào hệ thống



```
192.168.202.152 - PuTTY
login as:root
root@192.168.202.152's password:
root@f6283ee8edfc:~#
```

- Hacker sẽ thực hiện những lệnh sau trên hệ thống: pwd, cat /etc/passwd ...

```
192.168.202.152 - PuTTY
login as:root
root@192.168.202.152's password:
root@f6283ee8edfc:~# pwd
/root
root@f6283ee8edfc:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

- Hacker upload file hello.py (giả sử là mã độc) lên hệ thống:

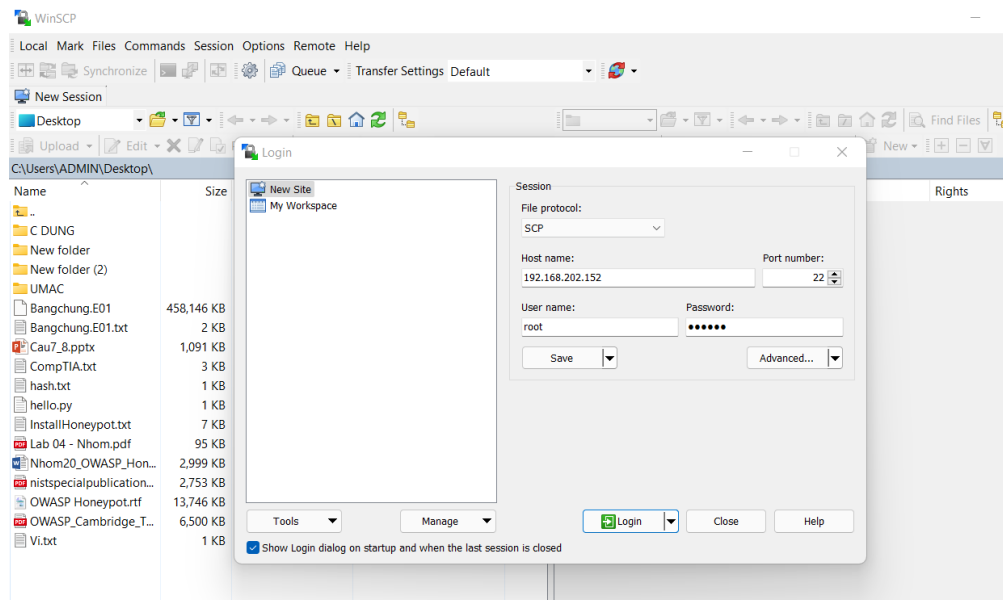


Figure 11. Hacker login SSH server

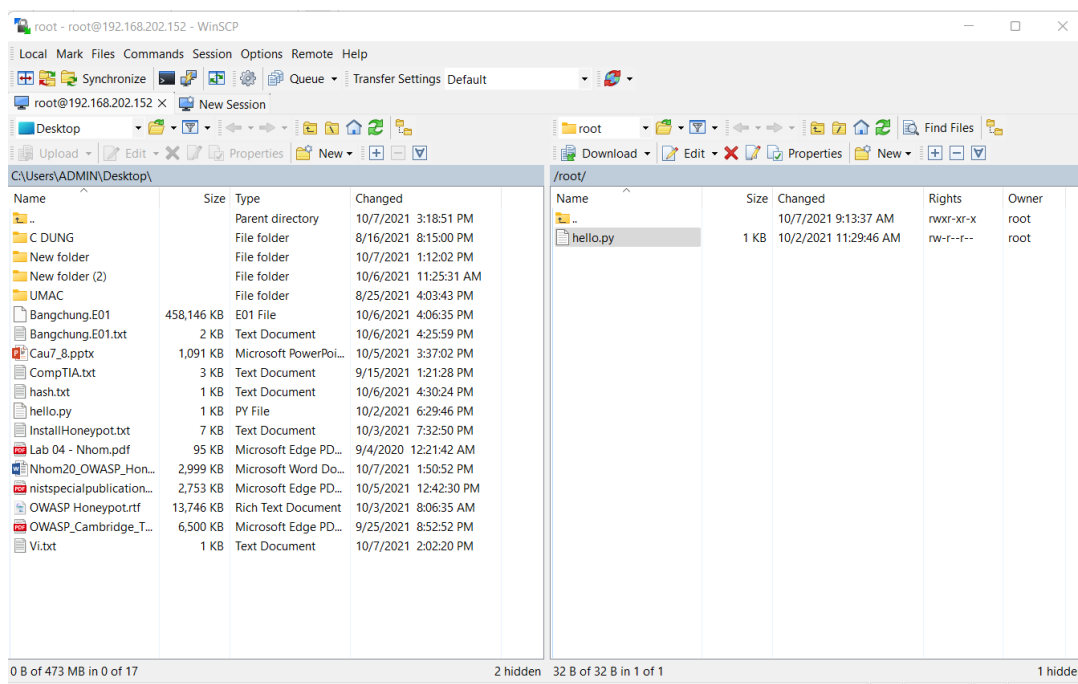


Figure 12. Upload file chứa mã độc

- Những lệnh, file hacker sử dụng sẽ được lưu lại trong Honeypot. Hoạt động của hacker sẽ được gửi tới Elasticsearch được lưu lại dưới dạng File Change Events. Người quản trị có thể theo dõi Event này thông qua Log Explorer trên giao diện web của máy API (<http://localhost:5000>):

Rows per page: 10

Search filter...

Date	File Path	Module Name	Is Directory	Machine Name
2021-10-07 16:15:12	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/hello.py	ssh/weak_password	false	honeypot
2021-10-07 16:15:12	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/hello.py	ssh/weak_password	false	honeypot
2021-10-07 16:15:12	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/hello.py	ssh/weak_password	false	honeypot
2021-10-07 16:15:12	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/hello.py	ssh/weak_password	false	honeypot

CSV EXCEL 1 - 4 of 4 Previous 1 Next

JSON

← → ↻ localhost:5000

OWASP Open Web Application Security Project

DASHBOARD LOG EXPLORER

Event Type: File Change Events Module Name: ssh/weak\_password

Start Date: 10 / 03 / 2021 End Date: 10 / 03 / 2021

SEARCH

Rows per page: 10

Search filter...

Date	File Path	Module Name	Is Directory	Machine
2021-10-03 16:41:43	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/bash_history	ssh/weak_password	false	h
2021-10-03 16:41:43	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/.bash_history	ssh/weak_password	false	h
2021-10-03 16:41:43	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/.bash_history	ssh/weak_password	false	h
2021-10-03 16:40:54	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/.bash_history	ssh/weak_password	false	h
2021-10-03 14:44:19	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/.bash_history	ssh/weak_password	false	s
2021-10-03 14:44:19	/home/trantruong/Desktop/Python-Honeypot/tmp/ohp_ssh_weak_container/.bash_history	ssh/weak_password	false	s

Figure 13. hình ảnh từ API server

```

trantruong@honeypot:~/Desktop$ sudo cat /home/trantruong/Desktop/Python-HoneyPot/tmp/ohp_ssh_weak_container/.bash_history
pwd
cat /etc/passwd
exit
pwd
cat /etc/passwd
exit

```

Figure 14. hình ảnh từ máy HoneyPot

### 2.3.2 Demo module ssh/strong\_password

```

trantruong@honeypot:~/Desktop/Python-HoneyPot$ sudo python3 ohp.py -m ssh/strong_password

OWASP
HoneyPot

[+] [2021-10-03 16:51:03] OWASP HoneyPot started ...
[+] [2021-10-03 16:51:03] loading modules ssh/strong_password
[+] [2021-10-03 16:51:03] Checking for conflicts in ports
[+] [2021-10-03 16:51:03] port 22 selected for ssh/strong_password
[+] [2021-10-03 16:51:03] removing container ohp_sshserver_strong_password
[+] [2021-10-03 16:51:03] removing image ohp_sshserver_strong_password
[+] [2021-10-03 16:51:04] creating image ohp_sshserver_strong_password
[+] [2021-10-03 16:59:23] image ohp_sshserver_strong_password created
[+] [2021-10-03 16:59:24] container ohp_sshserver_strong_password started, forwarding 0.0.0.0:22 to 172.18.0.2:22
[+] [2021-10-03 16:59:24] All selected modules started: ssh/strong_password
[+] [2021-10-03 16:59:25] network_traffic_capture process started

```

- Module này có mật khẩu mạnh nên hacker sẽ không đoán được
- Giả sử hacker thử đăng nhập vào hệ thống bằng tài khoản root và mật khẩu là 137950
- Mỗi lần hacker đăng nhập, thông tin xác thực hacker sử dụng sẽ được gửi tới database server. Người quản trị có thể theo dõi thông tin này thông qua Log Explorer trên giao diện web của máy API (<http://localhost:5000>)



**EXPLORER**

Event Type: Credential Events Module Name: All Modules

Start Date: 10 / 03 / 2021 End Date: 10 / 03 / 2021

**SEARCH**

Rows per page: 10 Search filter...

Country	Date	IP	Machine Name	Module Name	Password	Username
	2021-10-03 10:04:50		honeypot	ssh/strong_password	137950	root

CSV EXCEL 1 - 1 of 1 Previous 1 Next  
 JSON

- Dựa vào đó quản trị viên có thể phát hiện ra các cuộc tấn công brute-force

## 2.4 Môi liên hệ giữa các container

- Sau khi khởi động Database Server các container sẽ được kết nối vào lớp mạng: python-honeypot-default:

```

trantruong@trgpc:~/Desktop/Python-Honeypot$ sudo docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
cdb324ab4e83        bridge              bridge              local
af6257b00c47        host                host                local
e71b5281f0ef        none                null                local
76344d9a9992        python-honeypot_default bridge              local
trantruong@trgpc:~/Desktop/Python-Honeypot$

```

```

},
"ConfigOnly": false,
"Containers": {
  "07a5df36c0782ea6125d5235d4969c3e5a068ea24f21b0271db399ae88fbf283": {
    "Name": "elasticsearch",
    "EndpointID": "5c695ebe718402f43c90062369601c7930cde362ca90fb674b0b1a257b16ec39",
    "MacAddress": "02:42:ac:12:00:02",
    "IPv4Address": "172.18.0.2/16",
    "IPv6Address": ""
  },
  "9d2ce123b64abc0248ff7c0c5443100d7af492aad3fe5b59932790b841a5cf3": {
    "Name": "grafana",
    "EndpointID": "2427fc31adb98734b86406c3ab5685b8d9c05ac3455af3c43fb74a597a4226a9",
    "MacAddress": "02:42:ac:12:00:03",
    "IPv4Address": "172.18.0.3/16",
    "IPv6Address": ""
  },
  "f15b9b89f573e6ee1140341afebcb25f42af5b961972831cc96c752b595453c5": {
    "Name": "python-honeypot_ohp_1",
    "EndpointID": "b65b80ebdd694612085ed627af6ee7d3a74970a49d07d0d02bab687cce9a4822",
    "MacAddress": "02:42:ac:12:00:04",
    "IPv4Address": "172.18.0.4/16",
    "IPv6Address": ""
  }
},
"Options": {},
"Labels": {

```

Figure 15 inspect python-honeypot\_default

- Các container máy honeypot được kết nối vào mạng: ohp\_internet

```

trantruong@honeypot:/var$ sudo docker network ls
[sudo] password for trantruong:
NETWORK ID          NAME                DRIVER              SCOPE
cba5446d032d        bridge             bridge              local
fe4cdbf0367c        host               host                local
a0879ad89db3        none              null                local
c3926c0e8720        ohp_internet       bridge              local
a46cecb357f1        ohp_no_internet    bridge              local

```

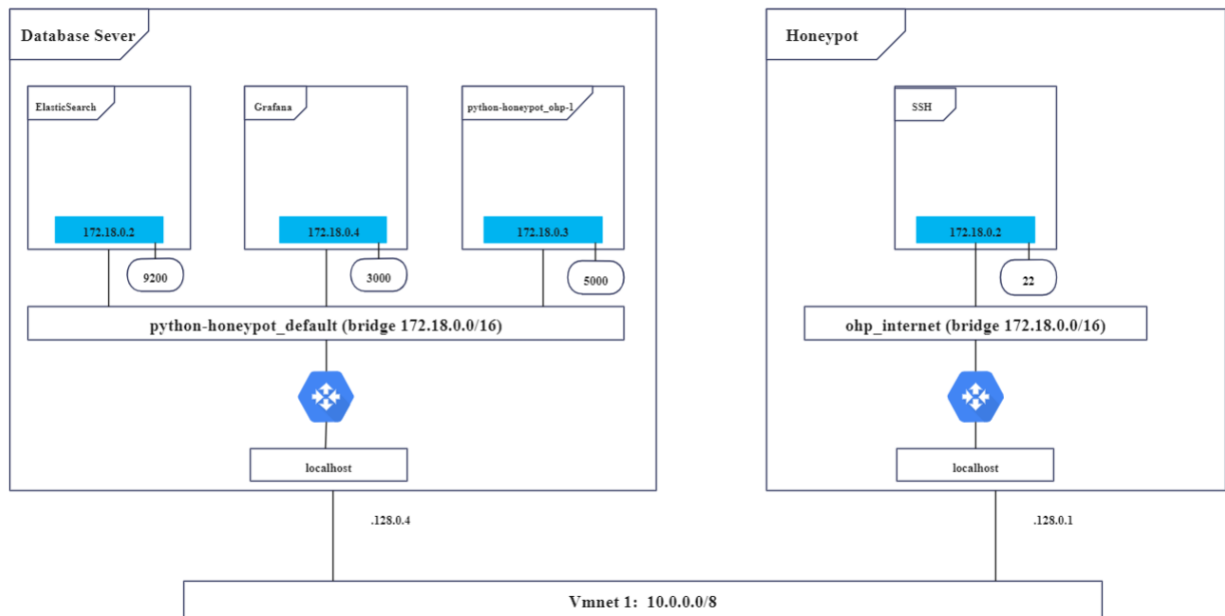
```

},
"ConfigOnly": false,
"Containers": {
  "d74b417949e731f8fe98ef0ff67ab608948e230a8d3752867e481175fdf26d7c": {
    "Name": "ohp_sshserver_weak_password",
    "EndpointID": "ea735f150194e05281b22cd90f38802d0c971fa3938bd8e67b1f3dedabbf6df0",
    "MacAddress": "02:42:ac:12:00:02",
    "IPv4Address": "172.18.0.2/16",
    "IPv6Address": ""
  }
},
"Options": {
  "com.docker.network.bridge.enable_icc": "true",
  "com.docker.network.bridge.enable_ip_masquerade": "true",
  "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
  "com.docker.network.driver.mtu": "1500"
},

```

Figure 16. Inspect ohp\_internet

- Sau khi triển khai, các container sẽ được kết nối với nhau như sơ đồ sau:



- Honeypot tổng hợp dữ liệu thu thập được từ hoạt động của hacker trong container SSH dưới dạng JSON
- Thiết lập kết nối TCP giữa 2 host Server và Honeypot
- Honeypot gửi dữ liệu JSON tới máy Server 10.128.0.4:9200 -> mapping qua 172.18.0.2:9200 tới container ElasticSearch để lưu trữ.
- Khi người dùng truy cập WebUI http://localhost:5000 và gửi request về container ohp-1(API), API tiếp nhận request và đọc dữ liệu từ container ElasticSearch rồi response dữ liệu JSON hiển thị lên giao diện web.

### Bảng phân chia công việc

Trần Quốc Trọng	33.33%
Huỳnh Tiến Vĩ	33.33%
Hồ Minh Phong	33.33%