Department of Information Engineering and Computer Science

NETWORK SECURITY
LABORATORY REPORT

LAB 10
HONEYPOT

Da Rold Giovanni

224291

Meschini Marcello

220222

Academic year 2020/2021

Contents

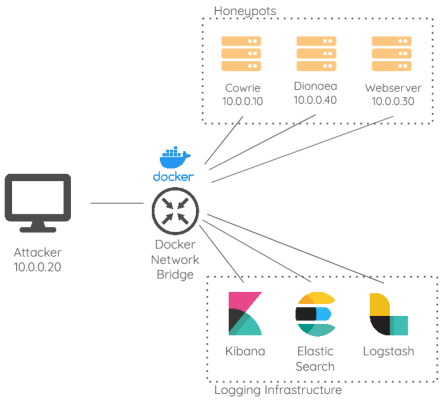
Info about the lab

Requirements

- Docker Engine version 17.05 or newer
- Docker Compose version 1.20.0 or newer
- 2 GB of RAM
- At least 20 GB of disk

Network Topology

To create a network for the laboratory we used Docker Compose and created the following topology:



The docker-compose.yaml file containing the definition for the containers can be found in the netsec-honeypot-lab folder on the desktop or in our Github public repository:

`https://github.com/Marcy-P/netsec-honeypot-lab`

The repository README also contains additional info for accessing the containers and the references to some Docker images we used.

Starting up the lab

To start the laboratory login into the VM with the credentials: username: *netsec* and password: *password*. Then open a terminal in the folder \netsec-honeypot-lab on the desktop and type the following command:

```
$ ./start.sh
```

Shutting down the lab

To shut down the docker-compose network type:

```
$ docker-compose down
```

To also clean the persistent data present in Elasticsearch type:

```
$ docker-compose down -v
```

1 What is a honeypot?

"A honeypot is a network-attached system set up as a decoy to lure cyber attackers and detect, deflect and study hacking attempts to gain unauthorized access to information systems"[1]. So it is a system that is unprotected and serves no business purpose but sits in the network waiting to be contacted. Every interaction with a honeypot is suspicious because no legitimate user should utilize it.

2 Characteristics of a honeypot

Honeypots have four main characteristics; they have to be: **Deceptive** , **Discoverable**, **Interactive**, **Monitored**.

2.1 Deception

Deception can be defined as an advantageous distortion of an adversary's perceptions of reality. Honeypots heavily use this concept as a tool because they appear as real systems, but they do not serve any functional purpose for a business.

There are various frameworks that try to classify various deception strategies, and we considered the taxonomy proposed by Bell and Whaley [2]. According to this model deception consists in two main parts: hiding the real (dissimulation) and showing the false (simulation).

Honeypots are meant to be reachable, so you do not hide it entirely, but you often have to hide specific features of them to make them less suspicious. The techniques to hide the real are:

- **Masking**: hide the real by making relevant objects be undetectable or blend into the background;
- **Repackaging**: hide the real by making it appear like it is something different;
- **Dazzling**: hide the real by altering an object to confuse the adversary;

The techniques to show the false are:

- **Mimicking**: show the false by using characteristics present in the actual real object;
- **Inventing**: show the false by giving the perception that a relevant object exist while it does not;
- **Decoying**: show the false by misdirecting and attracting the attacker attention away from real objects.

Note that not every honeypot has to strictly follow these techniques. It is just a theoretical framework that you may want to use when creating and deploying a honeypot.

2.2 Discoverability

Honeypots are not meant to be accessed by legitimate users but just the attacker. So when designing your honeypot you have to consider the point of view of the attacker. In particular, you can ask yourself the following questions:

- Where is the attacker more likely to enter your network?

  - User workstations
  - Vulnerable services exposed to the internet
  - Stolen VPN credentials

- What tool will they use to discover your asset?
- What assets will they be interested in?

To increase discoverability, you might also place **breadcrumbs** in systems that might be compromised. They are data that will lead the attacker to your honeypot while the attacker is gathering information needed to do lateral movement in the network. An example of breadcrumbs is a clear text document containing an URL or IP of a honeypot and some credential or SSH private keys.

2.3 Interactivity

Technically a single attacker interaction is enough to produce an alert. But is better if the honeypot responds back to the attacker for two main reasons:

- Each interaction that attacker does with the honeypot could provide you important information. For example, if they stole some credential, you may be able to know the account that was compromised. You may be able to find out what tool they use to enter your network
- Make the attacker waste time. Every second that the attacker spends interacting with the honeypot is a second you can invest into finding out which systems were compromised and isolate the attacker

3 Honeypots classification

4 Advantages of honeypots

5 Cowrie

6 Dionaea

7 Honeytokens

8 Lab monitoring infrastructure

9 Honeypots for IoT

Bibliography

[1] What is a honeypot. https://searchsecurity.techtarget.com/definition/honey-pot.

[2] Barton Whaley. Toward a general theory of deception. *Journal of Strategic Studies*, 5(1):178–192, 1982.