

IS-4543: Cyber Attack and Defend

Project Milestone Report for Milestone 2

Name: Delton Robinson

Section: IS-4543-001

Date: 10/22/2024

Milestone Report

This report includes the approved project proposal (included for easy reference), a summary of what was done during this project milestone, a description of what I learned during this milestone, and documentation of the work completed during this milestone.

Project Proposal

Project Idea 3: Honeypot Deployment for Network Intrusion Detection

Description: This project focuses on setting up an internal honeypot within a virtualized environment to lure and log simulated internal attack attempts. Using tools like Cowrie, an SSH honeypot, I will create a fake, “enticing” system for simulated internal attacks. The goal is to analyze common attack vectors in a controlled environment, without exposing the honeypot to external networks.

Milestones:

1. Virtualized Environment Setup

- I’ll begin by setting up a completely isolated virtual environment using VMware. This environment will contain multiple virtual machines, including the honeypot (potentially running Cowrie on a Linux VM) and at least one attacker machine (Kali Linux). The network will be fully internal, ensuring no exposure to external traffic, and configured to simulate a typical corporate or home network.

2. Honeypot Configuration

- I’ll install and configure a honeypot on a Linux VM, setting it up as an SSH honeypot. The system will be designed to look like a valuable internal server, with fake services, decoy files, and user accounts to entice attackers. I’ll configure the system to log all actions in detail, including commands typed and files accessed.

3. Simulated Attack Scenarios

- Using the attacker VM (Kali Linux), I’ll simulate multiple types of attacks typically seen in an internal network scenario. This will include brute-force SSH login attempts, privilege escalation attempts, and file exploration. Each attack will be executed in a controlled manner, allowing the honeypot to capture detailed logs of the process.

4. Analyzing Logs and Attack Patterns

- After completing the attack simulations, I’ll dive into analyzing the logs captured by Cowrie. I’ll look for patterns that could indicate specific attacker behavior, such as

common commands used after gaining access or file manipulation strategies. I'll use these insights to build a report on the attack patterns observed.

5. Reporting and Future Implementations

- Finally, I'll create a detailed report summarizing the findings from the honeypot logs, describing the effectiveness of the honeypot in detecting and logging internal attacks. I'll also propose potential enhancements for internal network security, such as improved monitoring or stricter access control policies to mitigate the risks demonstrated during the simulation.

Summary of Activities

During this milestone, I successfully configured and verified the installation of several services within my Ubuntu honeypot in order to make the machine seem more realistic. Each service I installed was carefully selected by me to simulate a realistic and appealing target to bait (simulated) attackers within future milestones of the project. The activities include:

- **Cowrie SSH (Port 2222):** I verified that cowrie is running and accessible to simulate an SSH service and to capture login attempts as well as allowing the attacker to peruse a fake server. Cowrie is a low interaction SSH honeypot, so it was ideal for this scenario.
 - Cowrie is a **low-interaction SSH honeypot**, making it ideal for this scenario by logging actions without exposing the actual machine.
 - I confirmed Cowrie's functionality through **Nmap scans from the attacker VM**, ensuring that it responded on port 2222.
- **FTP Service (Port 21):** I installed and configured the vsftpd service with both authenticated and anonymous access to make the machine appear more vulnerable. I modified the configuration file to allow write FTP connections, and then I created a decoy ftpuser user for the service realism. Test FTP connections were performed from the kali machine to verify that the service worked correctly.
 - I installed and configured the **vsftpd service** with both authenticated and anonymous access to make the honeypot appear vulnerable.
 - I modified the **configuration file** to allow **write connections**, making it seem as if files could be uploaded.
 - A **decoy ftpuser account** was created to enhance the realism of the service.
 - I verified the service by performing test FTP connections from the **attacker VM** to confirm it was functional.
- **HTTP (Port 80):** I deployed Apache as a web server and created a simple (non-functional) login page when connecting to the honeypot via HTTP in order to simulate a corporate login portal, and therefore create a more enticing environment. I connected to the decoy login page by navigating to <http://10.0.0.190> within the kali machine.
 - I created the login page using a tutorial from W3 schools, due to me forgetting much of the syntax. After reading the documentation on HTML syntax, I created an html webpage within **/var/www/html/index.html**, replacing the default Apache webpage with a decoy corporate login page of my own creation.

- I deployed **Apache** as a web server and created a **simple decoy login page** to simulate a corporate portal, adding realism.
- I accessed the login page by navigating to `http://10.0.0.190` from the attacker VM to ensure the service was available.
- This login page serves as a **non-functional bait** to entice attackers into trying potential login attempts or exploits.
- **SMB Service (Ports 139/445):** Installed Samba to simulate file-sharing services. This makes the honeypot appear more realistic to an Nmap scan. While I did not perform extensive testing on the SMB connection, the presence of the service within an Nmap scan contributes to the realism of the honeypot as an attack target.
 - I installed **Samba** to simulate file-sharing services and make the honeypot appear more realistic during **Nmap scans**.
 - While I did not perform extensive testing of SMB connections, I ensured the service appeared correctly in the **Nmap scan output**, contributing to the overall realism of the honeypot.
 - The inclusion of both **NetBIOS (port 139)** and **SMB (port 445)** aligns with real-world network setups (to my knowledge), making the environment more attractive to attackers.
- **MySQL Service (Port 3306):** I configured a MySQL database service as well as creating a simple decoy payroll table to represent employee payroll data. The presence of MySQL is mainly to serve the same purpose as the SMB service. However, due to me setting up a decoy table, it would allow this service to appear more enticing to an attacker due to there being “valuable info” present. After setting up the service, I referenced documentation provided by W3 Schools in order to learn more about MySQL syntax to perform this milestone.
 - As part of this milestone, I set up the MySQL service and created a **decoy employee payroll table** to simulate sensitive data. I used the CREATE TABLE command to define the table structure, with an id column as a unique identifier that auto-increments, along with name and salary fields. Finally, I inserted sample employee data into the table using the INSERT INTO command.
 - I referenced **W3Schools documentation** to familiarize myself with MySQL syntax and successfully complete this setup.
 - MySQL serves as both a **functional service** and an attractive target, as the presence of simulated sensitive data increases the honeypot’s appeal.

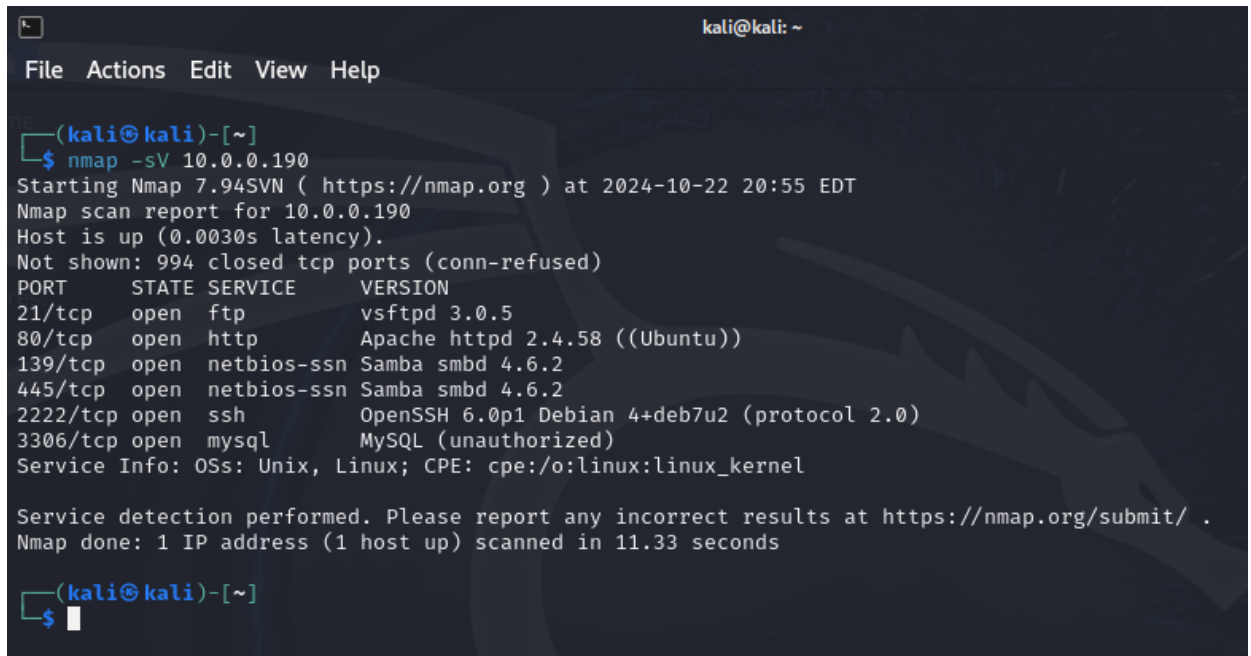
Description of Learning Completed

Throughout the completion of this milestone, I gained experience configuring core services within a machine that I have not set up before. I have also verified that the services are functional and display within a Nmap Scan, all within an isolated virtual environment.

- Multi-Service Management: In the process of installing, configuring, and managing multiple services within the honeypot, I gained a better understanding of how to install and manage the various services I worked with during this milestone.
- Services:
 - **Cowrie SSH (Port 2222):** I learned that Cowrie is extremely configurable, and if I desired, I could change the active port to 22 in order to make the service less recognizable as a common low-interaction honeypot.
 - **FTP (Port 21):** Configuring vsftpd allowed me to learn more about file transfer protocols, and primarily how to edit the configuration file of vsftpd. I enabled anonymous login, which may allow for another attack vector for future tests.
 - **HTTP (Port 80):** I learned how to install Apache and set up a basic login page using HTML. The last time I used HTML was in high school, making sample web pages, so using it again now was a nice callback to then, and required me to re-learn some of the basic HTML syntax in order to create the decoy login page.
 - **SMB (Ports 139/445):** Through setting up Samba, I learned that I can set a directory to be public, and create decoy data to live in these public directories to attract would be attackers. This is why I modified the configuration file to add a [PublicShare] portion, which sets the path to the public share as /srv/samba/public. Within this directory (/srv/), I ran the command `#sudo chmod 777 /samba/public`, in order to make this service clearly vulnerable.
 - **MySQL (Port 3306):** The timing of the completion of this milestone coincided with the completion of another project from my Forensics course. Setting up the MySQL service as well as creating the decoy employee payroll table required me to more deeply understand the MySQL commands, which allowed me to use commands such as `INSERT INTO` to add values into my decoy table. I was required to learn the syntax behind the `CREATE TABLE` command. For example, the command I used to create the table: `CREATE TABLE employees (id INT AUTO INCREMENT PRIMARY KEY, name VARCHAR(100), salary DECIMAL(10, 2));` utilizes `INT` `AUTO INCREMENT PRIMARY KEY`, to make a unique identifier for each employee within the list, that auto increments for each new employee added. The parameter `name VARCHAR(100)` means the name of the employee can be up to 100 characters long, and the parameter salary `DECIMAL(10, 2)` means that the employee's salary can be 10 digits in total, with 2 digits being reserved for decimal places. (e.g 12345678.90)

Documentation of Work Completed

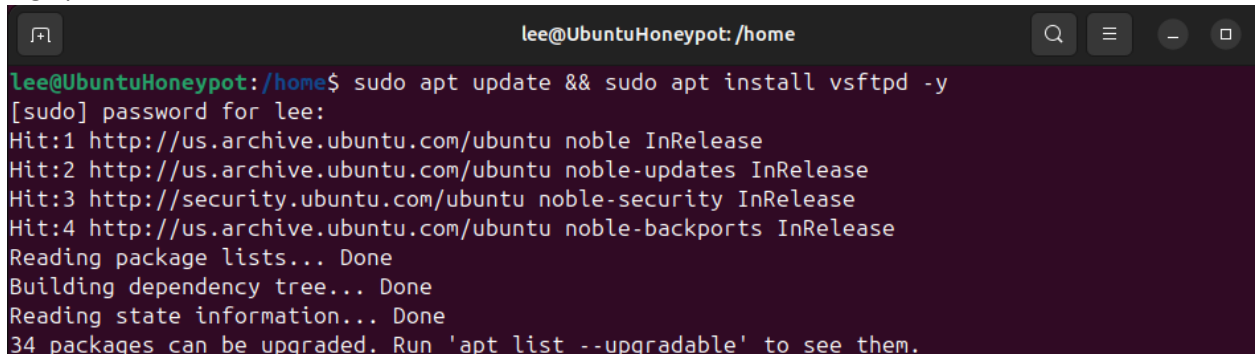
1. Full Nmap Scan verifying the successful installation & configuration of the services listed within this milestone.



```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ nmap -sV 10.0.0.190  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 20:55 EDT  
Nmap scan report for 10.0.0.190  
Host is up (0.0030s latency).  
Not shown: 994 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 3.0.5  
80/tcp    open  http         Apache httpd 2.4.58 ((Ubuntu))  
139/tcp   open  netbios-ssn  Samba smbd 4.6.2  
445/tcp   open  netbios-ssn  Samba smbd 4.6.2  
2222/tcp  open  ssh          OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)  
3306/tcp  open  mysql        MySQL (unauthorized)  
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 11.33 seconds  
  
(kali@kali)-[~]  
$
```

a.

2. Setting up the FTP service.



```
lee@UbuntuHoneypot: /home  
lee@UbuntuHoneypot: /home$ sudo apt update && sudo apt install vsftpd -y  
[sudo] password for lee:  
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease  
Hit:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
34 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

a.

```
GNU nano 7.2 /etc/vsftpd.conf
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
```

b.

```
lee@UbuntuHoneypot: /home
lee@UbuntuHoneypot: /home$ sudo adduser ftpuser
info: Adding user `ftpuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `ftpuser' (1002) ...
info: Adding new user `ftpuser' (1002) with group `ftpuser (1002)' ...
warn: The home directory `/home/ftpuser' already exists. Not touching this directory.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
info: Adding new user `ftpuser' to supplemental / extra groups `users' ...
info: Adding user `ftpuser' to group `users' ...
lee@UbuntuHoneypot: /home$
```

c.

```
lee@UbuntuHoneypot: /home
lee@UbuntuHoneypot:/home$ sudo adduser ftpuser
info: Adding user `ftpuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `ftpuser' (1002) ...
info: Adding new user `ftpuser' (1002) with group `ftpuser (1002)' ...
warn: The home directory `/home/ftpuser' already exists. Not touching this directory.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
info: Adding new user `ftpuser' to supplemental / extra groups `users' ...
info: Adding user `ftpuser' to group `users' ...
lee@UbuntuHoneypot:/home$
```

d.

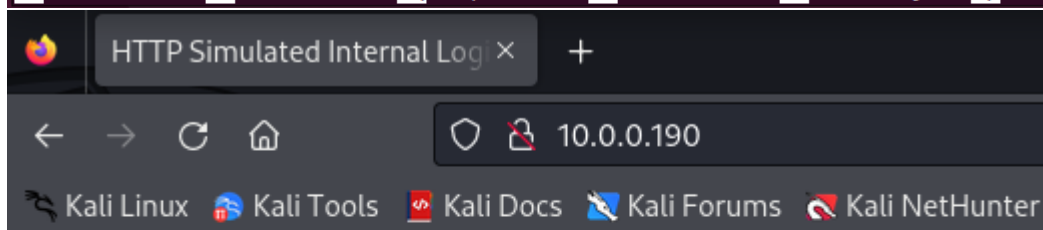
3. Setting up the HTTP service.

```
lee@UbuntuHoneypot: /home
lee@UbuntuHoneypot:/home$ sudo apt update && sudo apt install apache2 -y
[sudo] password for lee:
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
a. Fetched 126 kB in 1s (106 kB/s)
```

```
lee@UbuntuHoneypot: /var/www/html
GNU nano 7.2 index.html *
<html>
<head><title>HTTP Simulated Internal Login</title></head>
<body>
  <h1>Test Company Portal Login</h1>
  <form action="/login" method="POST">
    Username: <input type="text" name="username"><br>
    Password: <input type="password" name="password"><br>
    <button type="submit">Login</button>
  </form>
</body>
</html>
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

b.



Test Company Portal Login

Username:

Password:

c.

4. Setting up the SMB service.

a.

```
lee@UbuntuHoneypot: /var/www/html
lee@UbuntuHoneypot:/var/www/html$ sudo apt update && sudo apt install samba -y
[sudo] password for lee:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7,184 B]
```

b.

```
lee@UbuntuHoneypot: /etc/samba
GNU nano 7.2 smb.conf *
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[PublicShare]
path = /srv/samba/public
browseable = yes
read only = no
guest ok = yes
```

c.

```
lee@UbuntuHoneypot: /srv
lee@UbuntuHoneypot:/srv$ sudo mkdir -p /samba/public
lee@UbuntuHoneypot:/srv$ sudo chmod 777 /srv/samba/public
chmod: cannot access '/srv/samba/public': No such file or directory
lee@UbuntuHoneypot:/srv$ sudo chmod 777 /samba/public
lee@UbuntuHoneypot:/srv$ echo "Private Payroll information and Data" > /samba/public/payroll.txt
lee@UbuntuHoneypot:/srv$ cat /samba/public/payroll.txt
Private Payroll information and Data
lee@UbuntuHoneypot:/srv$
```

```
lee@UbuntuHoneypot: /srv
c/ payroll.txt
lee@UbuntuHoneypot: /srv$ cat /samba/public/payroll.txt
Private Payroll information and Data
lee@UbuntuHoneypot: /srv$ systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smbd.service; enabled; preset: enable>
   Active: active (running) since Tue 2024-10-22 18:49:12 CDT; 24min ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 25771 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 3 (limit: 4558)
    Memory: 21.9M (peak: 22.9M)
       CPU: 112ms
    CGroup: /system.slice/smbd.service
            └─25771 /usr/sbin/smbd --foreground --no-process-group
              └─25774 "smbd: notifyd" .
                └─25775 "smbd: cleanupd "
```

d.

5. Setting up the MySQL service.

```
lee@UbuntuHoneypot: /
lee@UbuntuHoneypot:/$ sudo apt update && sudo apt install mysql-server -y
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
a.
lee@UbuntuHoneypot:/$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabl>
   Active: active (running) since Tue 2024-10-22 19:15:31 CDT; 6min ago
 Process: 27381 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exi>
  Main PID: 27390 (mysqld)
    Status: "Server is operational"
     Tasks: 37 (limit: 4558)
    Memory: 365.5M (peak: 379.5M)
       CPU: 1.914s
    CGroup: /system.slice/mysql.service
            └─27390 /usr/sbin/mysqld
b.
```

```
lee@UbuntuHoneyPot: /

mysql> CREATE DATABASE payroll;
Query OK, 1 row affected (0.00 sec)

mysql> USE payroll;
Database changed
mysql> CREATE TABLE employees (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100)
, salary DECIMAL(10, 2));
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_payroll |
+-----+
| employees          |
+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO employees (name, salary) VALUES ('Alice', 74000.00), ('Brad', 67
000.00), ('Mallory', 55000.00), ('Evelyn', 80000.00), ('Yvette', 64000.00);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SHOW TABLES;
+-----+
| Tables_in_payroll |
+-----+
| employees          |
+-----+
1 row in set (0.01 sec)

mysql> SELECT * FROM employees;
+----+-----+-----+
| id | name   | salary |
+----+-----+-----+
| 1  | Alice  | 74000.00 |
| 2  | Brad   | 67000.00 |
| 3  | Mallory | 55000.00 |
| 4  | Evelyn | 80000.00 |
| 5  | Yvette | 64000.00 |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

c.

References

Canonical Ltd. (n.d.). *How to install and configure FTP server on Ubuntu Server*. Ubuntu Documentation. Retrieved October 21, 2024, from <https://documentation.ubuntu.com/server/how-to/networking/ftp/>

W3Schools. (n.d.). *HTML tutorial: What is HTML?* W3Schools. Retrieved October 21, 2024, from https://www.w3schools.com/whatis/whatis_html.asp

Red Hat. (n.d.). *Setting up the Apache HTTP server on RHEL 8*. Red Hat Documentation. Retrieved October 21, 2024, from https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/setting-apache-http-server_deploying-different-types-of-servers#apache-changes-to-rhel7_setting-apache-http-server

Red Hat. (n.d.). *Getting started with Samba*. Red Hat Blog. Retrieved October 21, 2024, from <https://www.redhat.com/en/blog/getting-started-samba>

W3Schools. (n.d.). *MySQL tutorial*. W3Schools. Retrieved October 21, 2024, from <https://www.w3schools.com/mysql/default.asp>