

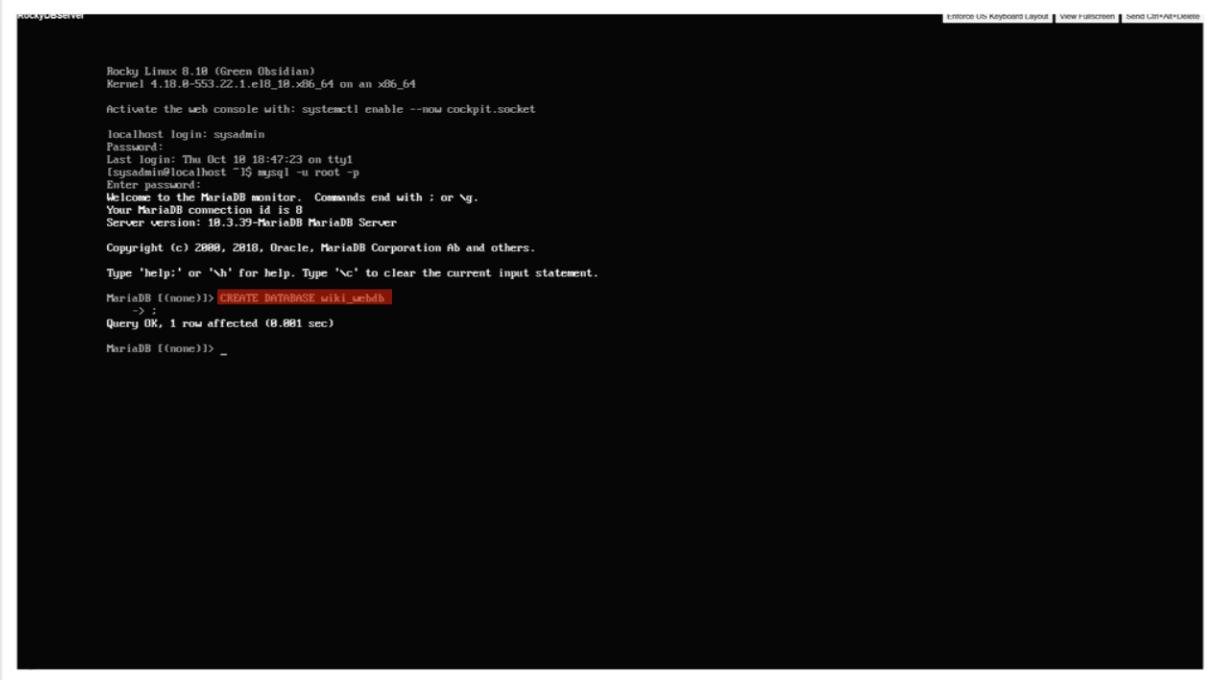
HW 07 – Services PDF 1

By :- Faraz Ahmed

Contents

1.	RockyDBServer: Configure database operations.....	3
2.	UbuntuWebServer: Configure MediaWiki.....	5
3.	Proof of Completion	7
4.	EAS 595 Additional Tasks.....	12
5.	Update Topology	16

1. RockyDBServer: Configure database operations



```
Rocky Linux 8.10 (Green Obsidian)
Kernel 4.18.0-553.22.1.el8_10.x86_64 on an x86_64
activate the web console with: systemctl enable --now cockpit.socket
localhost login: sysadmin
Password:
Last login: Thu Oct 18 18:47:23 on ttym1
[sysadmin@localhost ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.39-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

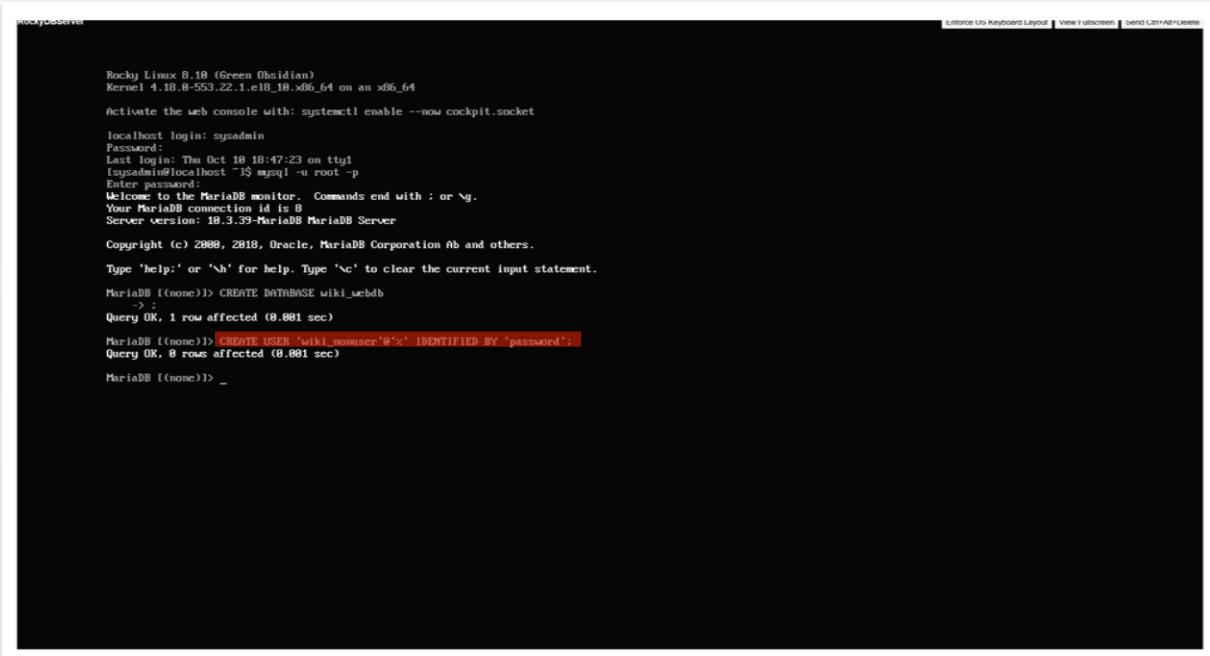
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wiki_webdb
-> ;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> _
```

Figure 1: Screenshot of creating new database by entering “CREATE DATABASE wiki_webdb;”.

- We can create a new database to support a Wiki website using command “CREATE DATABASE wiki_webdb;” as highlighted in Figure 1.



```
Rocky Linux 8.10 (Green Obsidian)
Kernel 4.18.0-553.22.1.el8_10.x86_64 on an x86_64
activate the web console with: systemctl enable --now cockpit.socket
localhost login: sysadmin
Password:
Last login: Thu Oct 18 18:47:23 on ttym1
[sysadmin@localhost ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.39-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wiki_webdb
-> ;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'wiki_nonuser'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> _
```

Figure 2: Screenshot of creating new non-root user by entering “CREATE USER ‘wiki_nonuser’@’%’ IDENTIFIED BY ‘password’;”.

- We can create a new non-root user for supporting a Wiki website which will access RockyDBServer (remotely) from UbuntuWebserver using command “CREATE USER ‘wiki_nonuser’@’%’ IDENTIFIED BY ‘password’;” as highlighted in Figure 2.

```
Rocky Linux 8.10 (Green Obsidian)
Kernel 4.18.0-553.22.1.el8_18.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket
localhost login: sysadmin
Password:
Last login: Thu Oct 18 10:47:23 on ttys1
[sysadmin@localhost ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.39-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wiki_webdb
->;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'wiki_nonuser'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wiki_webdb.* TO 'wiki_nonuser'@'%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Figure 3: Screenshot of grant new user privileges by entering “GRANT ALL PRIVILEGES ON wiki_webdb.* TO ‘wiki_nonuser’@‘%’;”.

- Now, we give or grant the new user remote or non-local privileges using “GRANT ALL PRIVILEGES ON wiki_webdb.* TO ‘wiki_nonuser’@‘%’;” to manipulate the database.

2. UbuntuWebServer: Configure MediaWiki

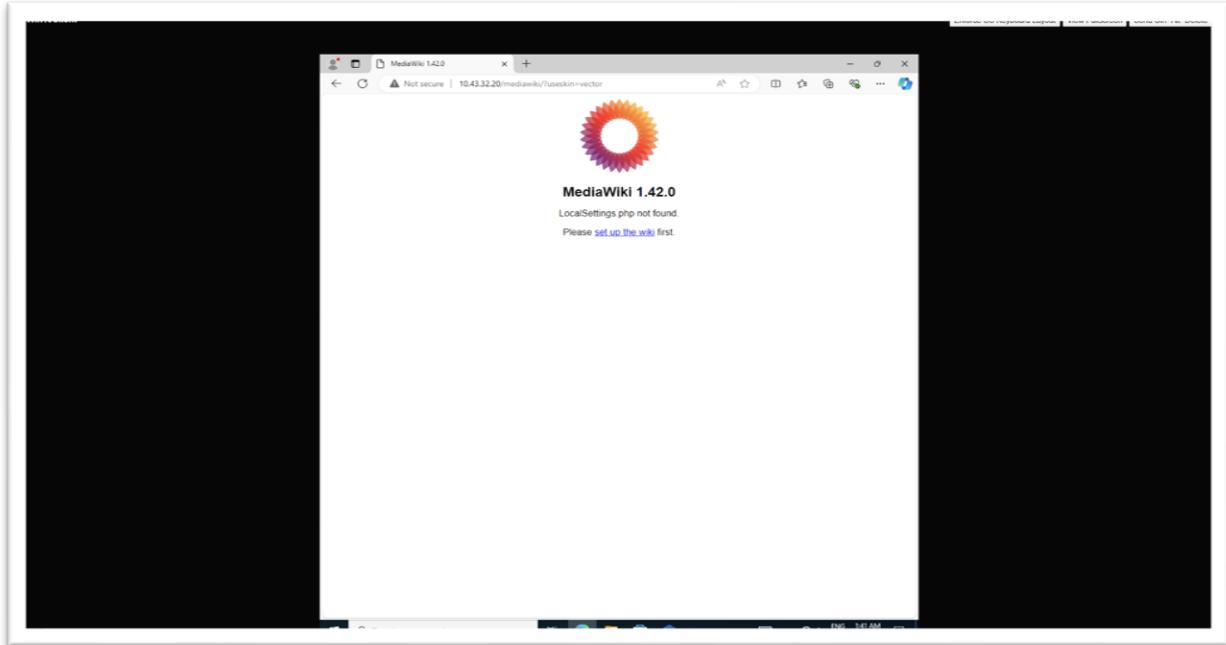


Figure 4: Screenshot of “default home page of MediaWiki”.

- First, we use any VM (in my case I used Win10Client) and enter URL- “<http://<UbuntuWebServer IP>/mediawiki?useskin=vector>” and we will get homepage as shown in Figure 4.

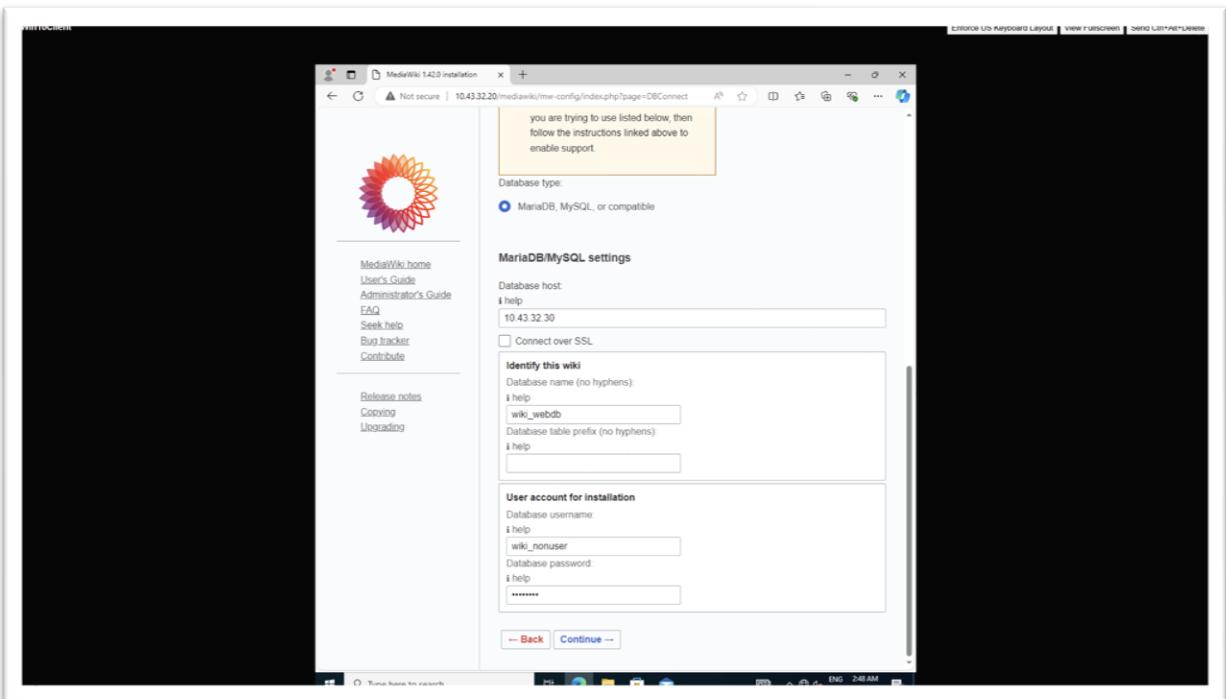


Figure 5: Screenshot of filling details in “MariaDB/MySQL settings”.

- Now, we enter ip of “RockyDBServer- 10.43.32.30” then enter the newly created database from Figure 1 and newly created username and password from Figure 2 as shown above in Figure 5.

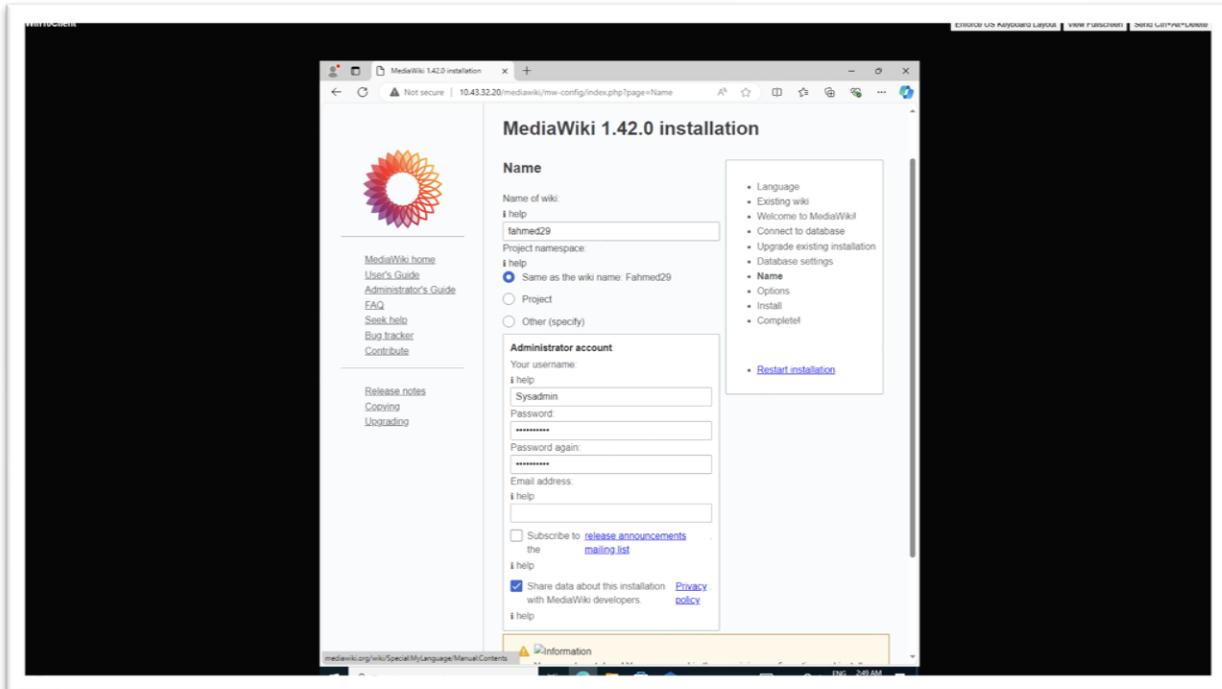


Figure 6: Screenshot of entering details of “Administrative MediaWiki user”.

- So, we enter our “UBIT name” in Name of wiki and enter “username- sysadmin and password- Change.me!” as shown in Figure 6.

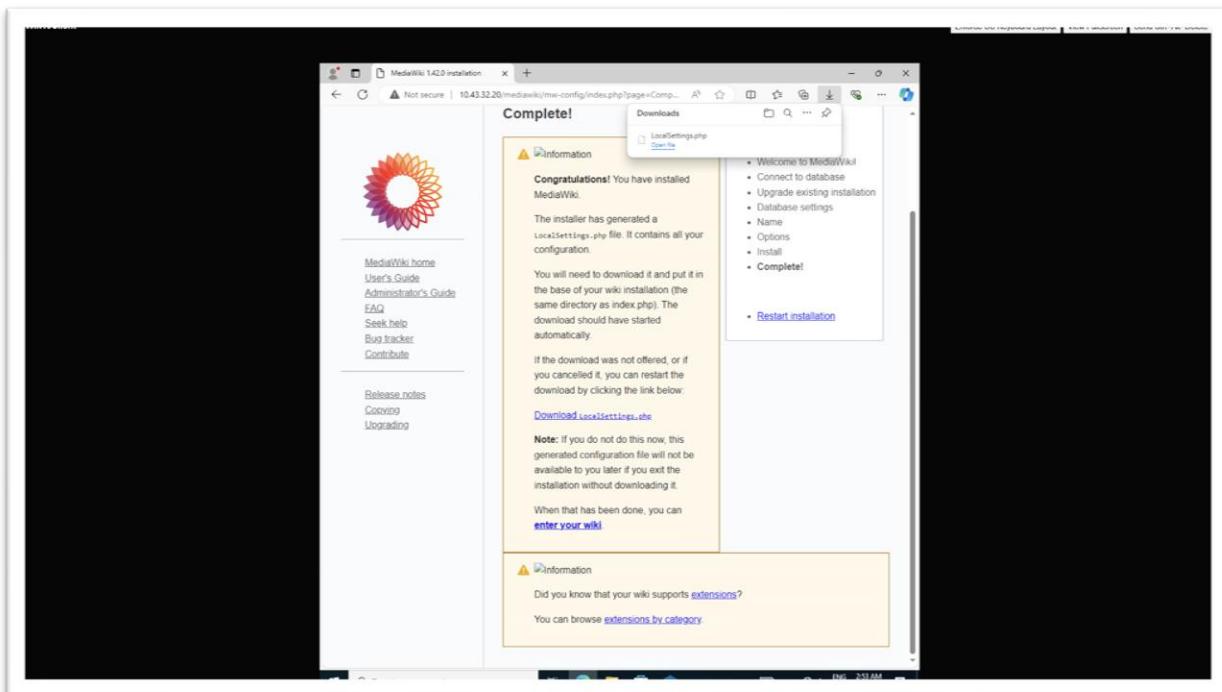


Figure 7: Screenshot of Installation complete and “LocalSettings.php” downloaded.

- After we complete the installation, we get a congratulation message to complete installation and then click to download file “LocalSettings.php”.

3. Proof of Completion

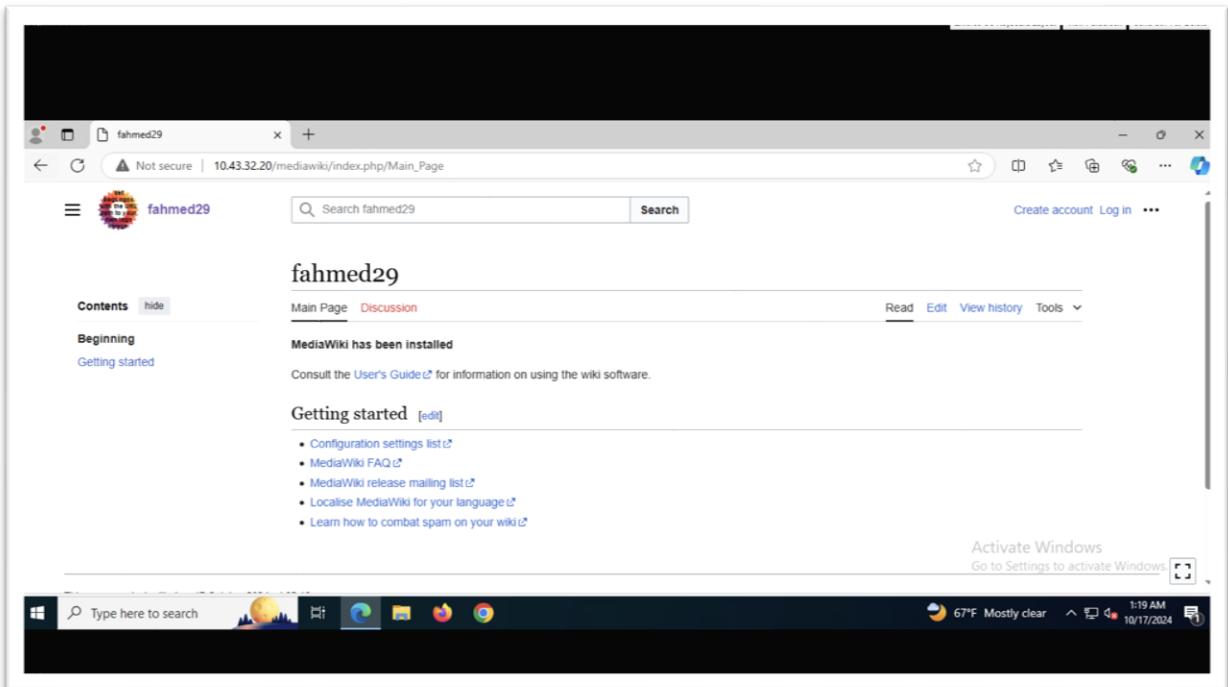


Figure 8: Screenshot of edit “Main Page into UBIT name”.

- So, after that we can edit out wiki’s “Main Page into fahmed29” which is my UBIT name as shown above.

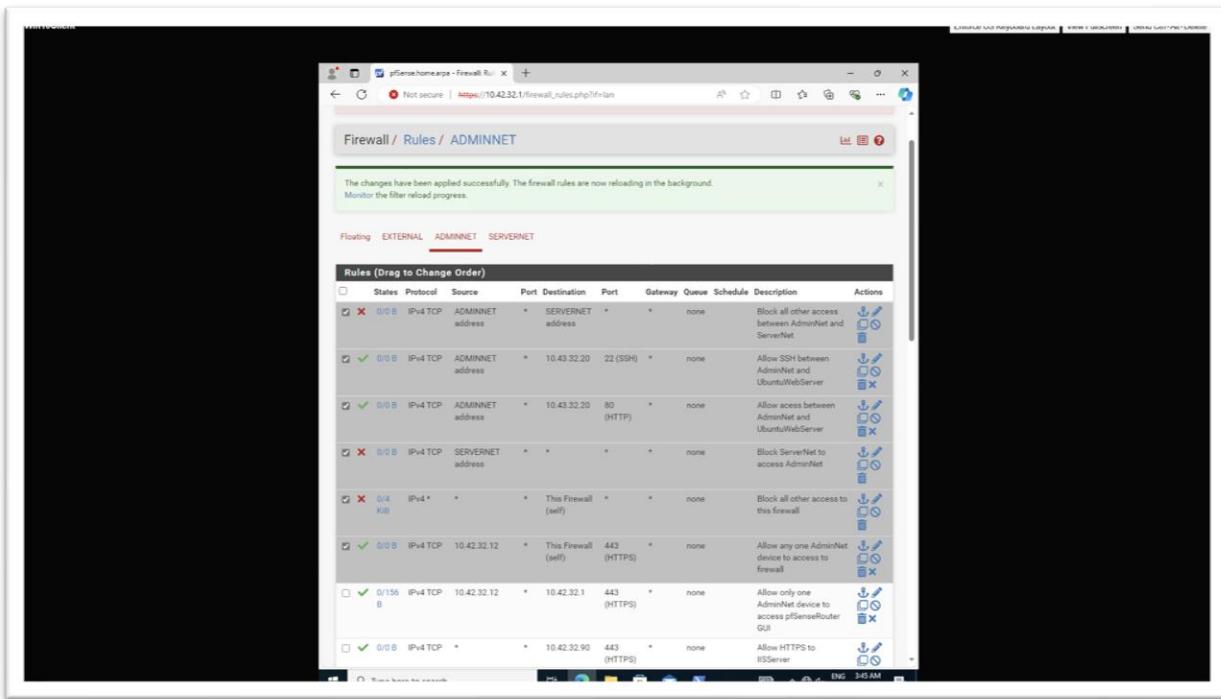


Figure 9: Screenshot of “all highlighted firewall rules using in this tasks”.

- Figure 9 shows all the firewall rules which are highlighted or selected (grey out) to complete the tasks.

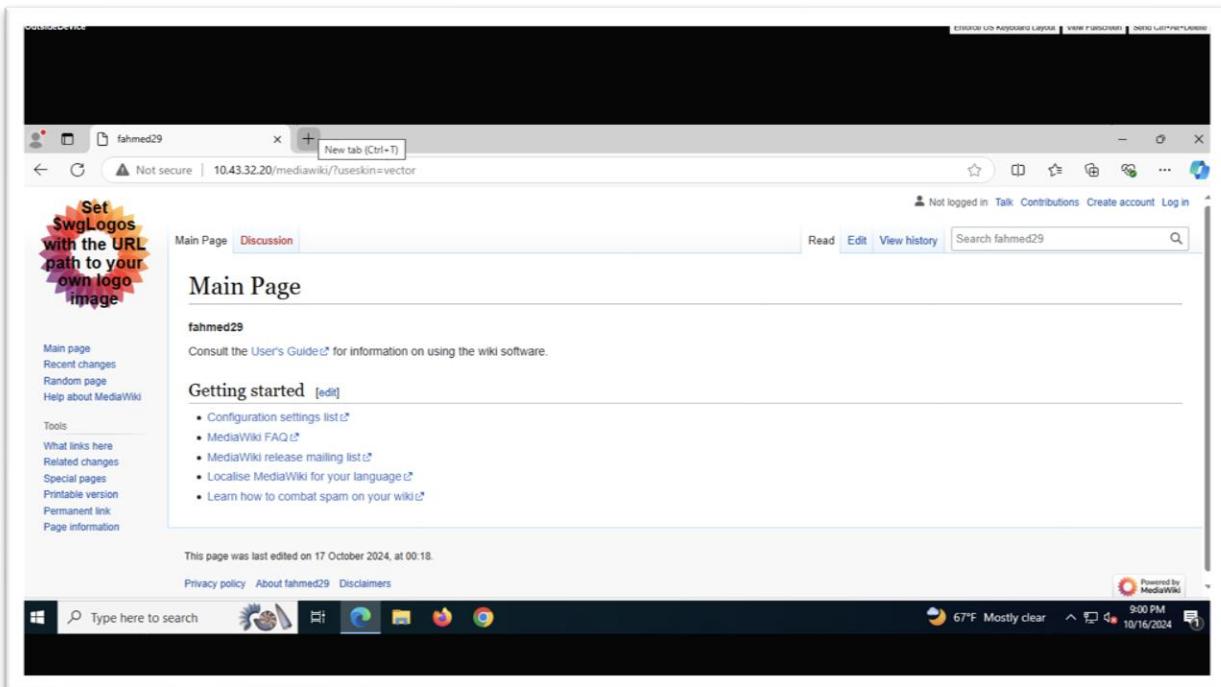


Figure 10: Screenshot of proof that “OutsideDevice can access MediaWiki”.

- We can access MediaWiki in OutsideDevice by entering URL- “<http://<UbuntuWebServer IP>/mediawiki?useskin=vector>” as shown in Figure 10.

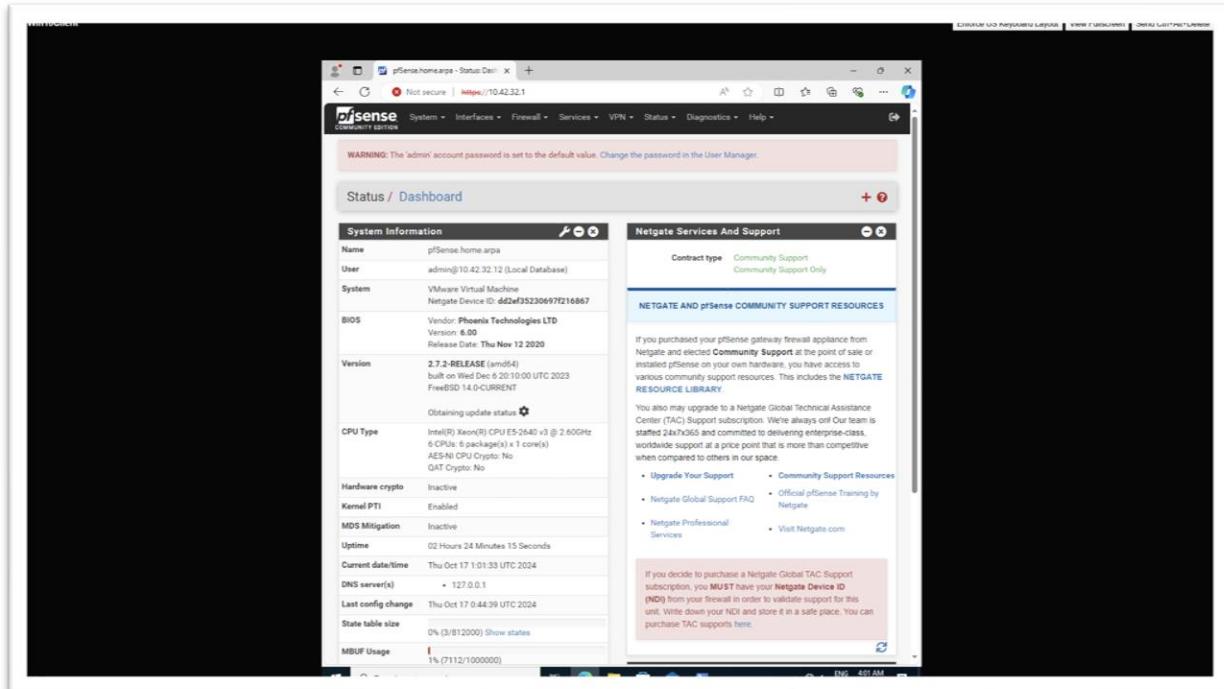


Figure 11: Screenshot of proof that “One AdminNet device (Win10Client) can access pfSense”.

- We can access “Pfsense webConfigurator GUI on any one AdminNet device (Win10Client)” as shown in Figure 11.

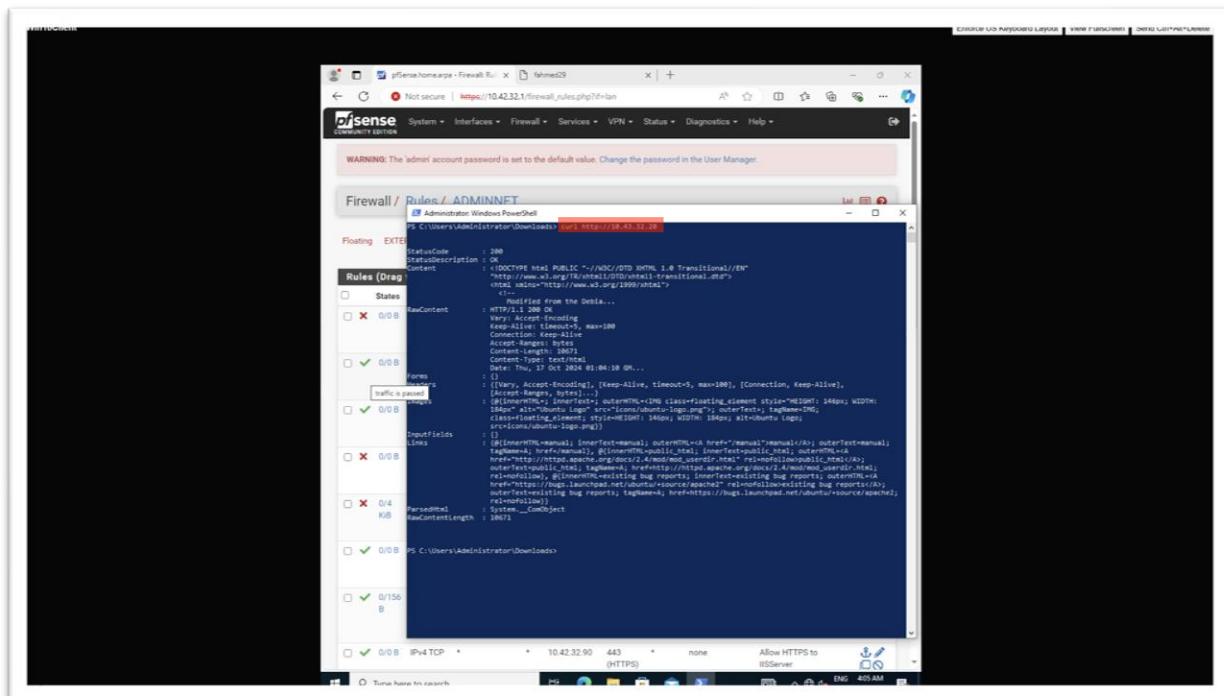


Figure 12: Screenshot of “AdminNet device accessing UbuntuWebServer using http”.

- Now, we enter “curl http://10.43.32.20” to check that AdminNet device can access UbuntuWebServer through http or not as highlighted in Figure 12.

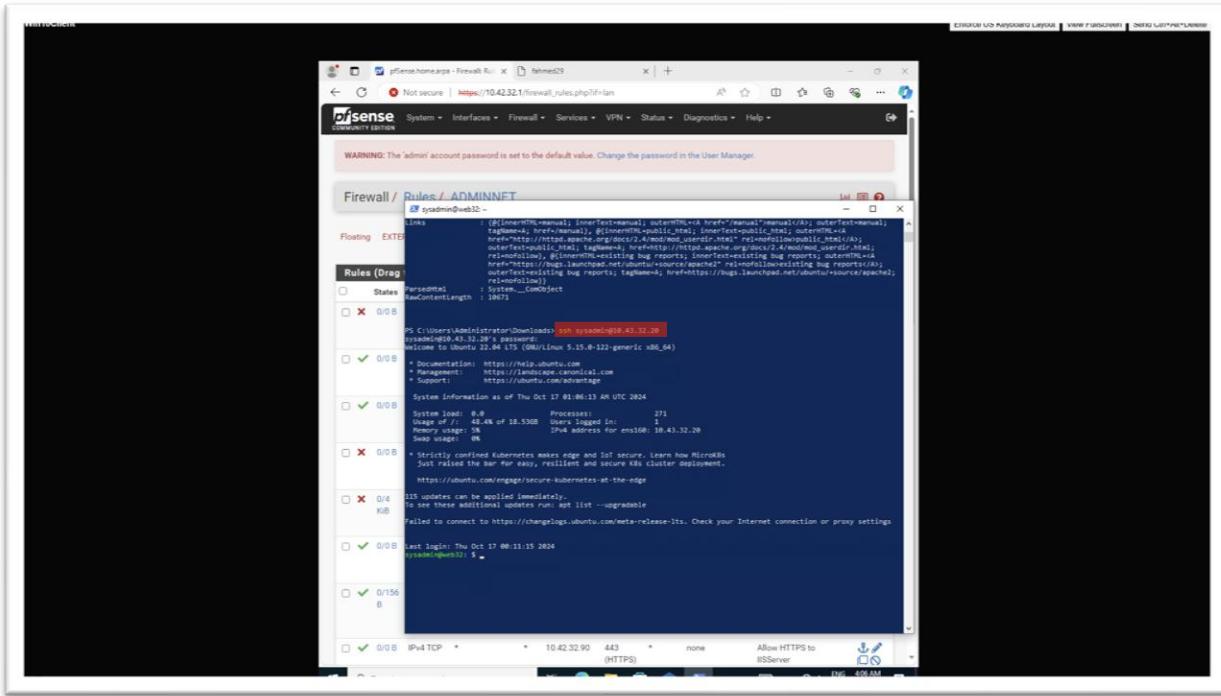


Figure 13: Screenshot of “AdminNet device accessing UbuntuWebServer using ssh”.

- Now, we enter “ssh sysadmin@10.43.32.20” to check that AdminNet device can access UbuntuWebServer through ssh or not as highlighted in Figure 13.

```

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Thu Oct 17 12:11:14 AM UTC 2024

System load: 0.5439453125  Processes:      269
Usage of /: 48.4% of 18.53GB  Users logged in:      0
Memory usage: 4%           IPv4 address for ens160: 10.43.32.20
Swap usage: 0K

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

115 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Oct 16 22:23:22 UTC 2024 on tty1
sysadmin@web32:~$ cp LocalSettings.php /var/lib/media/wiki
cp: cannot create regular file '/var/lib/media/wiki': No such file or directory
sysadmin@web32:~$ cp LocalSettings.php /var/lib/mediawiki
cp: cannot create regular file '/var/lib/mediawiki/LocalSettings.php': Permission denied
sysadmin@web32:~$ sudo cp LocalSettings.php /var/lib/mediawiki
[sudo] password for sysadmin:
sysadmin@web32:~$ ssh sysadmin@10.42.32.12
^C
sysadmin@web32:~$ sudo ssh sysadmin@10.42.32.12
[sudo] password for sysadmin:
ssh: connect to host 10.42.32.12 port 22: Connection timed out

```

Figure 14: Screenshot of denial of access by entering “sudo ssh sysadmin@10.42.32.12”.

- After that, we enter “sudo ssh sysadmin@10.42.32.12” to check if ServerNet devices can access AdminNet devices is not and my access got denied as highlighted in Figure 14.

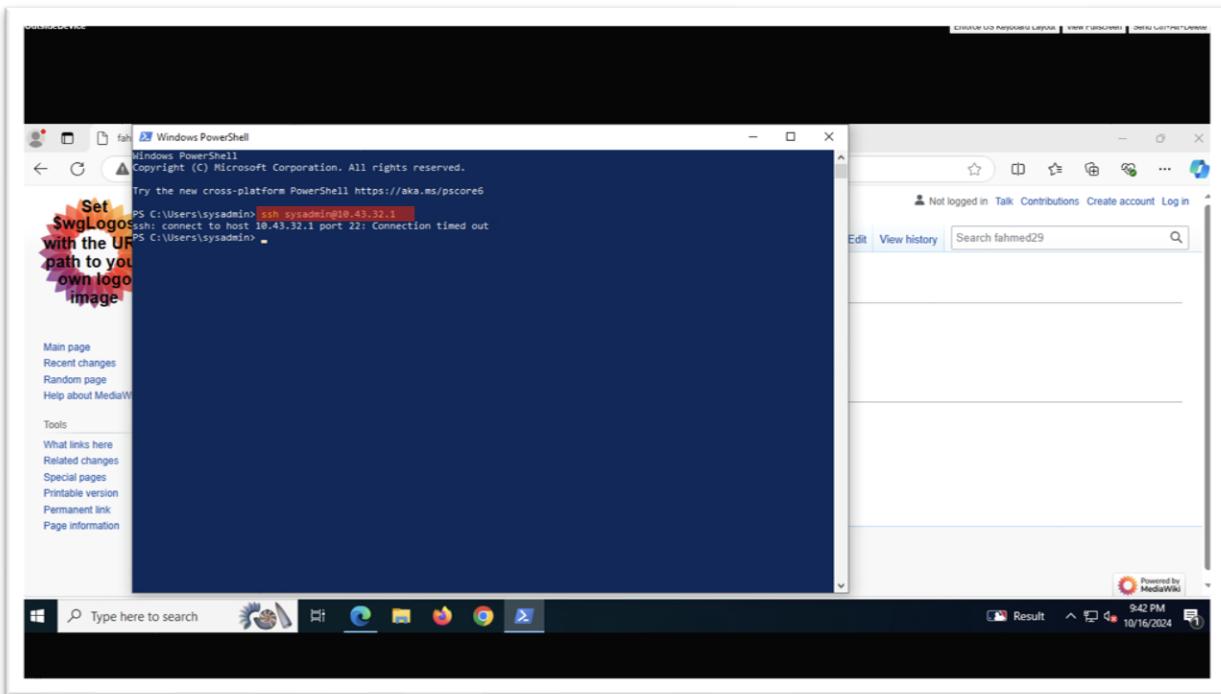


Figure 15: Screenshot of entering “ssh sysadmin@10.43.32.1” to check OutsideDevice access with ServerNet.

- By entering “ssh sysadmin@10.43.32.1” to check if OutsideDevice can access with ServerNet or not and from my output, my request was block as highlighted in Figure 15.

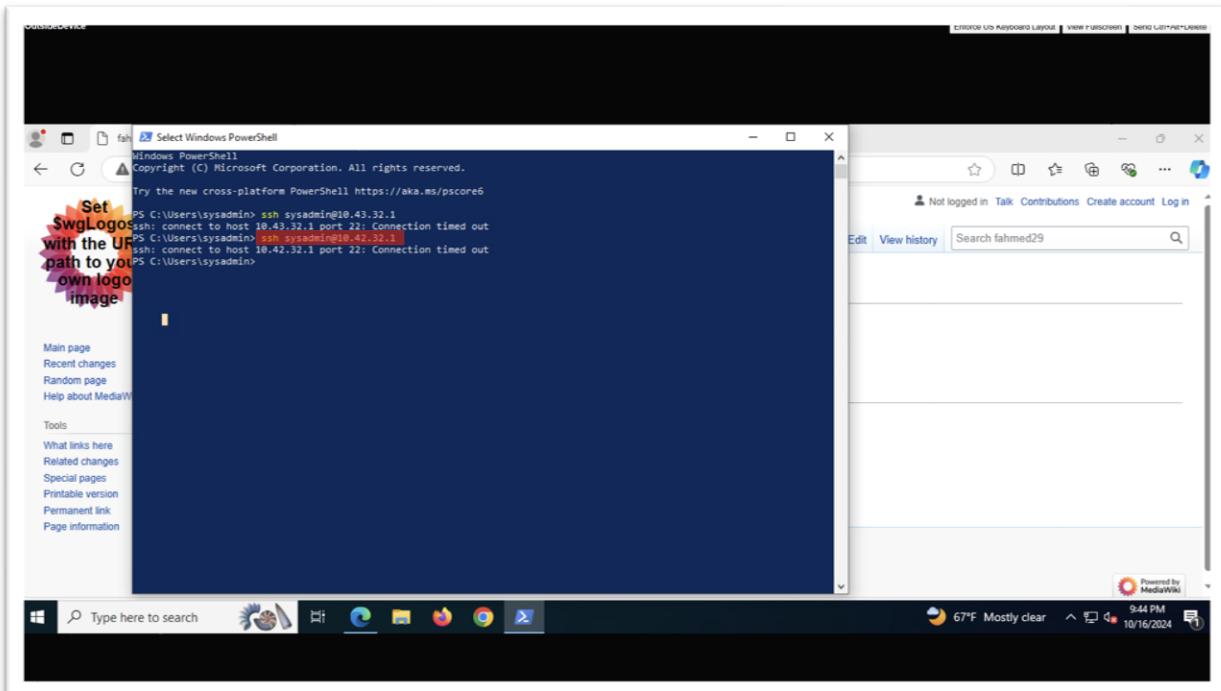
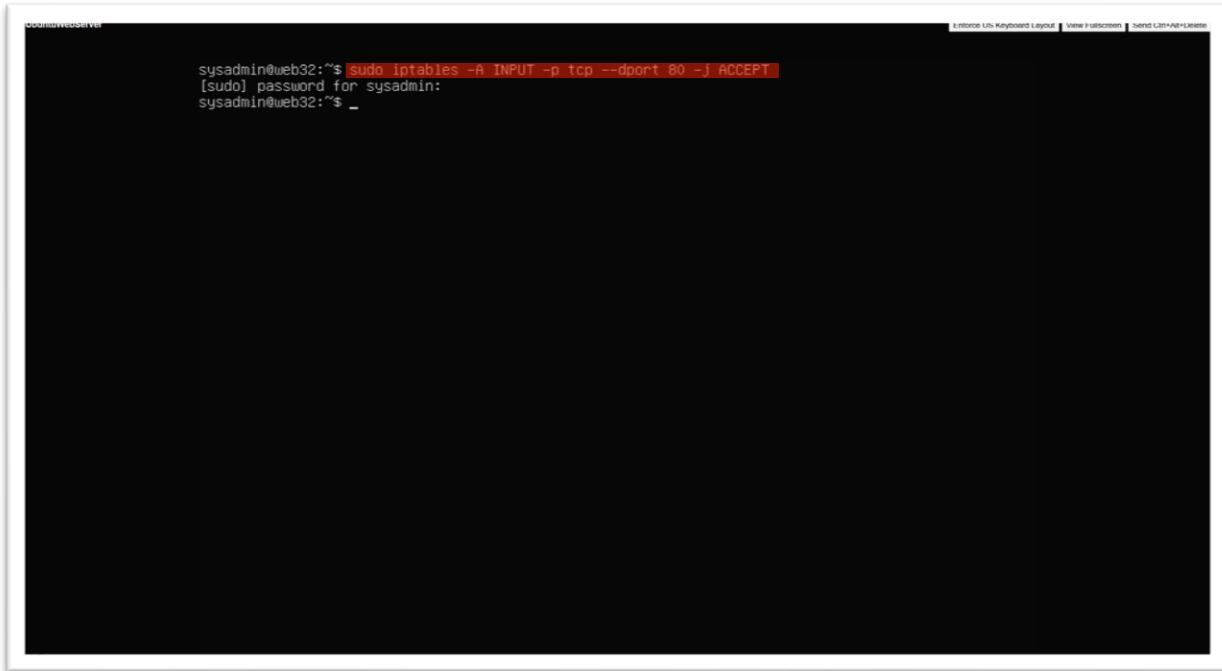


Figure 16: Screenshot of entering “ssh sysadmin@10.42.32.1” to check OutsideDevice access with AdminNet.

- By entering “ssh sysadmin@10.42.32.1” to check if OutsideDevice can access with AdminNet or not and from my output, my request was block as highlighted in Figure 16.

4. EAS 595 Additional Tasks

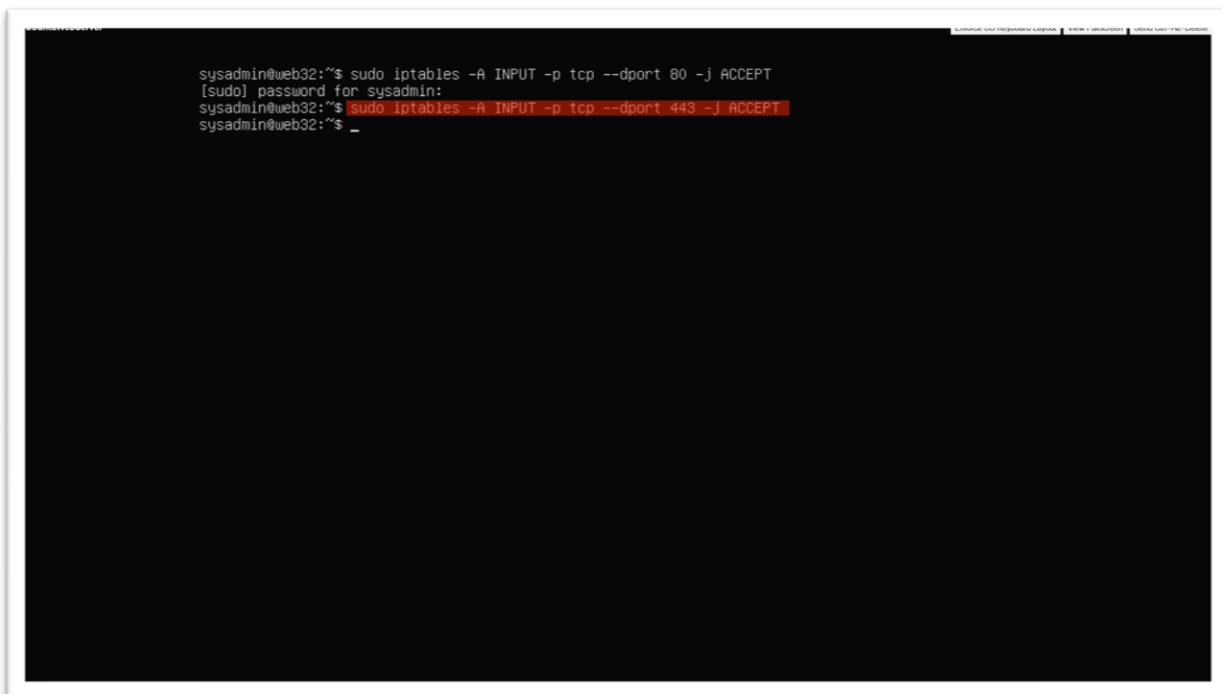


```
UbuntuWebServer
```

```
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
[sudo] password for sysadmin:
sysadmin@web32:~$ _
```

Figure 17: Screenshot of entering “sudo iptables -A INPUT -p tcp –dport 80 -j ACCEPT” to allow for package updates.

- We enter “sudo iptables -A INPUT -p tcp –dport 80 -j ACCEPT” to allow all “http” package updates in UbuntuWebServer as highlighted in Figure 17.



```
UbuntuWebServer
```

```
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
[sudo] password for sysadmin:
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
sysadmin@web32:~$ _
```

Figure 18: Screenshot of entering “sudo iptables -A INPUT -p tcp –dport 443 -j ACCEPT” to allow for package updates.

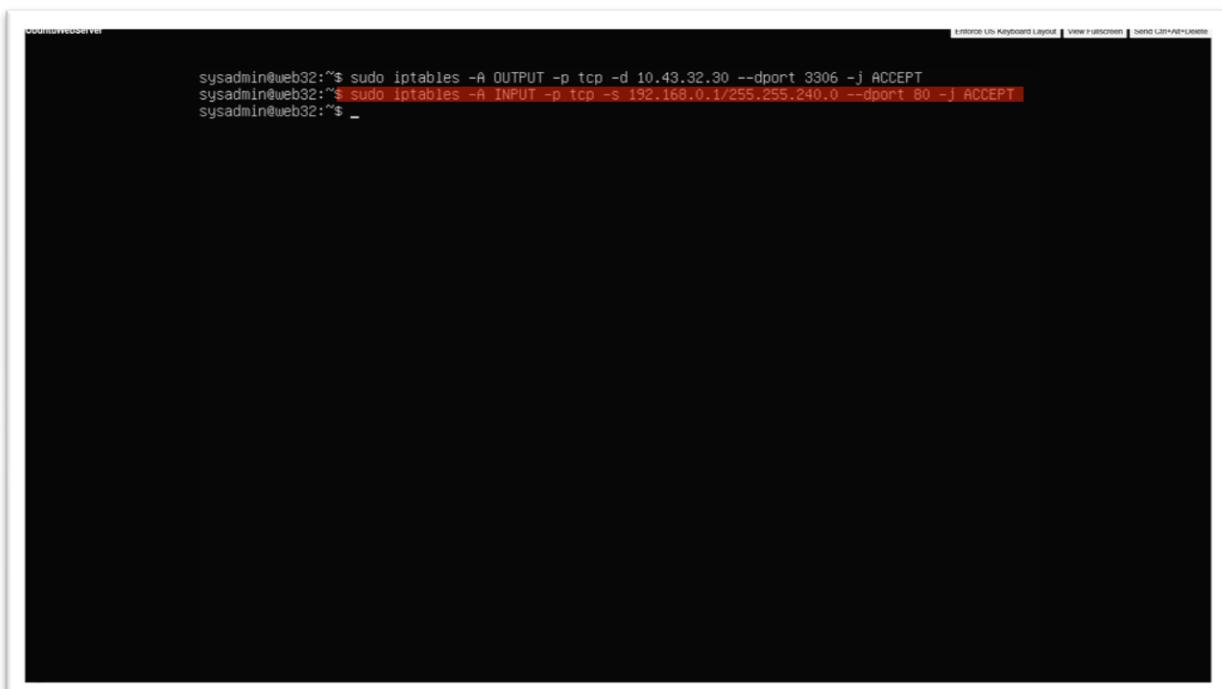
- We enter “sudo iptables -A INPUT -p tcp –dport 443 -j ACCEPT” to allow all “https” package updates in UbuntuWebServer as highlighted in Figure 18.



```
sysadmin@web32:~$ sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT
sysadmin@web32:~$
```

Figure 19: Screenshot of entering “sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT” to allow all necessary connections to RockyDBServer.

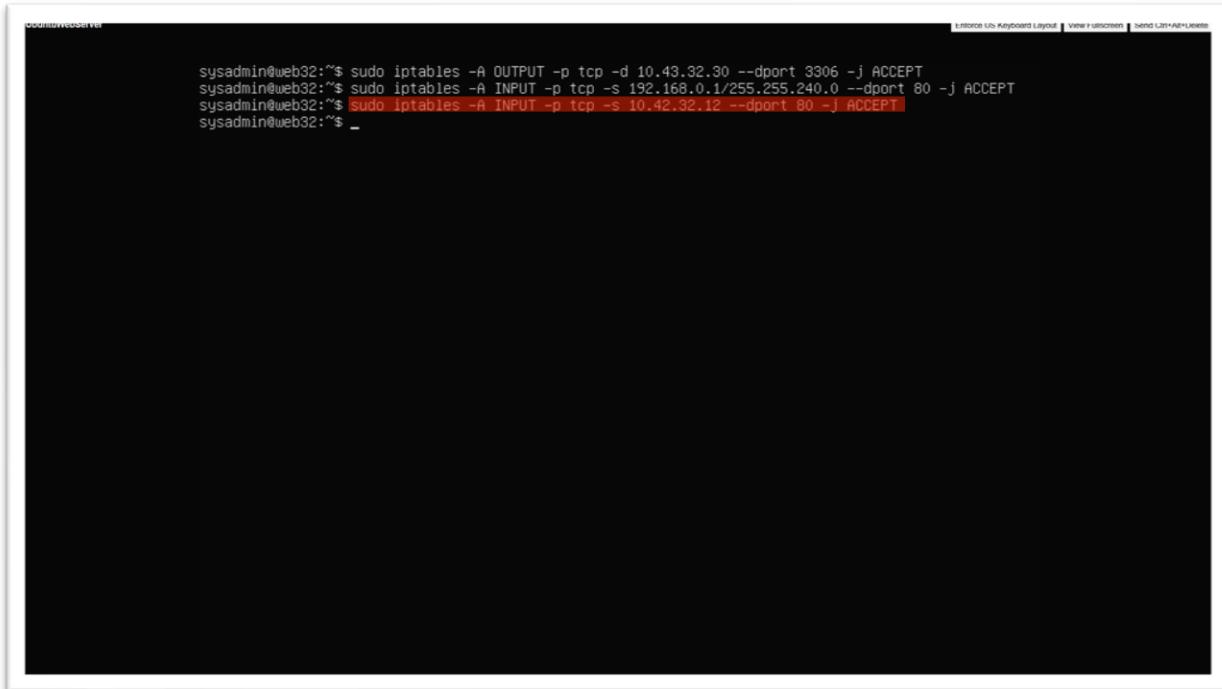
- Now, we enter “`sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT`” to allow all necessary connections possible to RockyDBServer as highlighted in Figure 19.



```
sysadmin@web32:~$ sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.255.240.0 --dport 80 -j ACCEPT
sysadmin@web32:~$
```

Figure 20: Screenshot of entering “sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.255.240.0 --dport 80 -j ACCEPT” to get requests from Gretzky's Core-Red subnet.

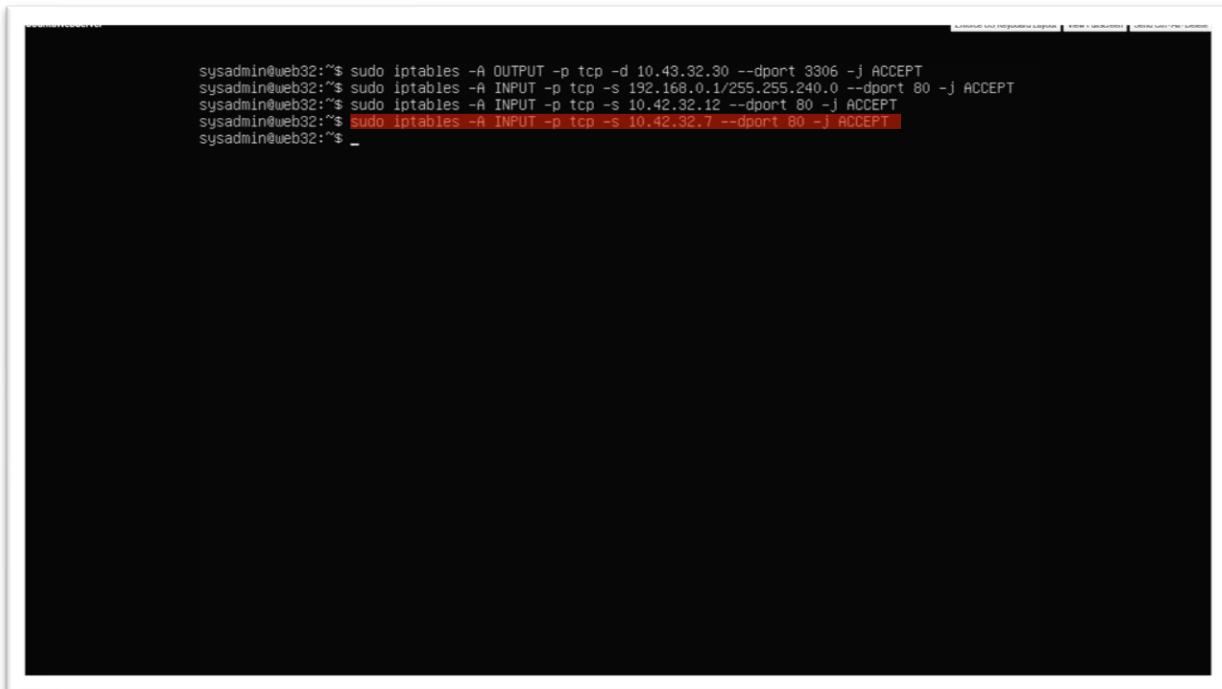
- So now, we enter “`sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.255.240.0 --dport 80 -j ACCEPT`” to get requests from Gretzky's Core-Red subnet as highlighted in Figure 20.



```
sysadmin@web32:~$ sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.255.240.0 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 10.42.32.12 --dport 80 -j ACCEPT
sysadmin@web32:~$
```

Figure 21: Screenshot of entering “sudo iptables -A INPUT -p tcp -s 10.42.32.12 --dport 80 -j ACCEPT” to get requests from Win10Client.

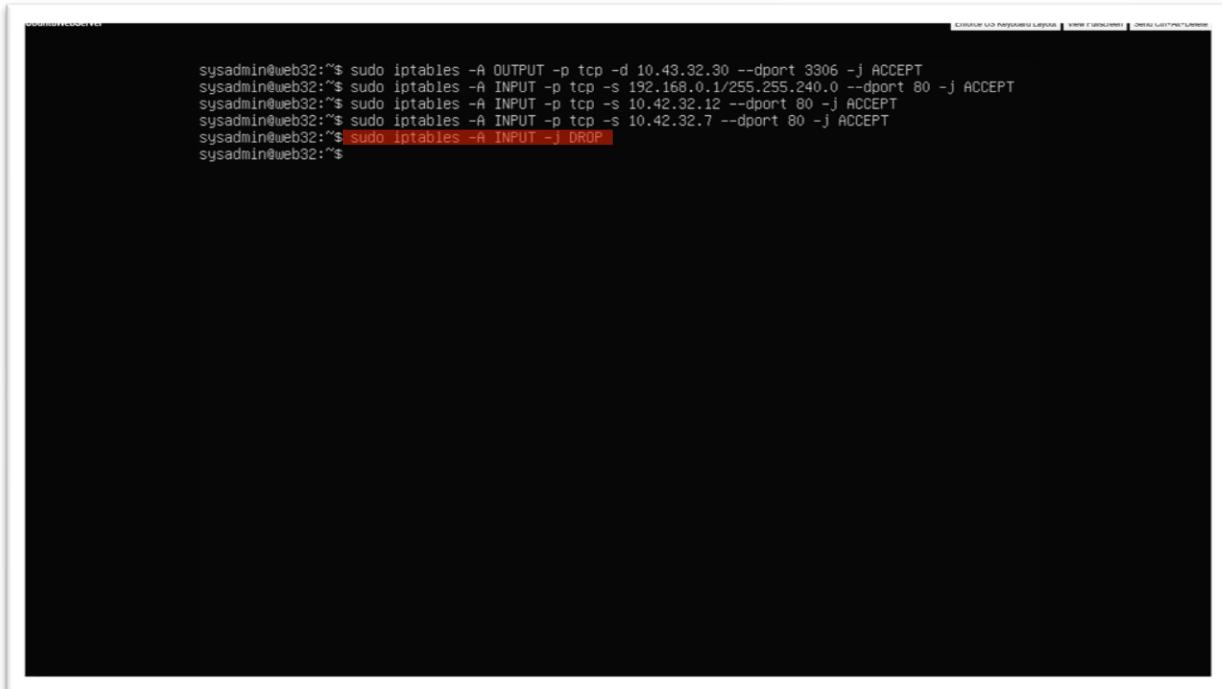
- Now enter “sudo iptables -A INPUT -p tcp -s 10.42.32.12 --dport 80 -j ACCEPT” to get requests from Win10Client as highlighted in Figure 21.



```
sysadmin@web32:~$ sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.255.240.0 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 10.42.32.12 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 10.42.32.7 --dport 80 -j ACCEPT
sysadmin@web32:~$
```

Figure 22: Screenshot of entering “sudo iptables -A INPUT -p tcp -s 10.42.32.7 --dport 80 -j ACCEPT” to get requests from UbuntuClient.

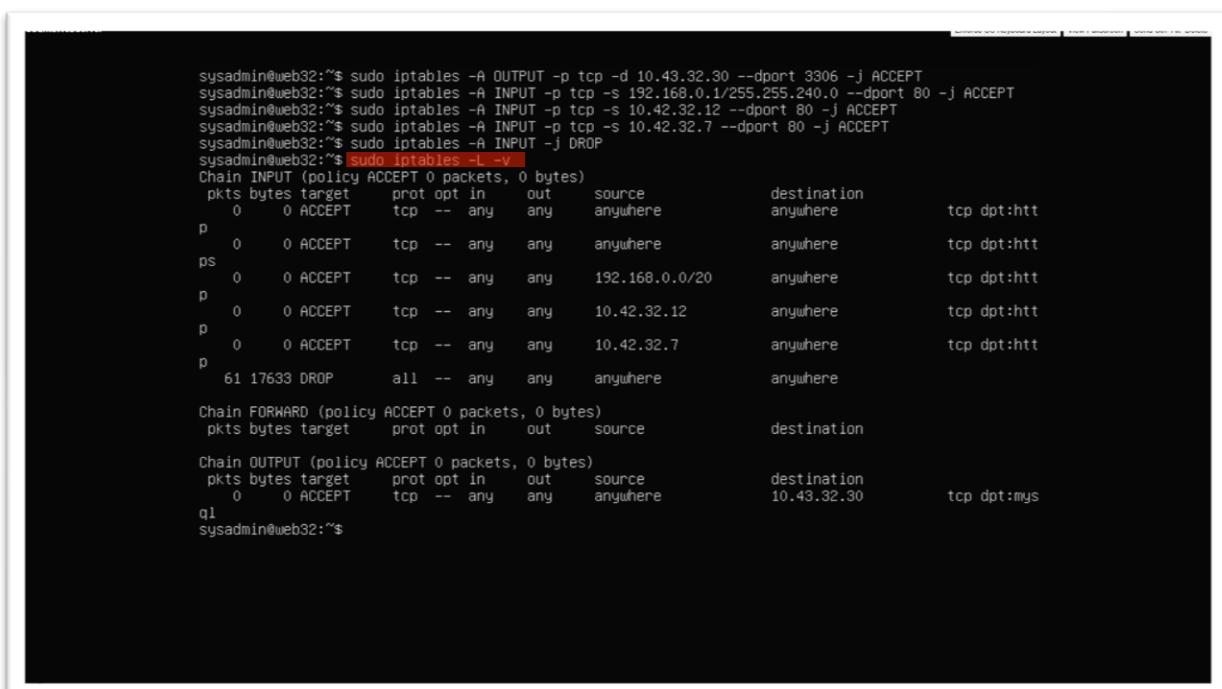
- Now enter “sudo iptables -A INPUT -p tcp -s 10.42.32.7 --dport 80 -j ACCEPT” to get requests from UbuntuClient as highlighted in Figure 22.



```
sysadmin@web32:~$ sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.255.240.0 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 10.42.32.12 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 10.42.32.7 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -j DROP
sysadmin@web32:~$
```

Figure 23: Screenshot of entering “sudo iptables -A INPUT -j DROP” to deny or drop all rules.

- Enter “sudo iptables -A INPUT -j DROP” to deny or drop all rules as highlighted in Figure 23.



```
sysadmin@web32:~$ sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.255.240.0 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 10.42.32.12 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -p tcp -s 10.42.32.7 --dport 80 -j ACCEPT
sysadmin@web32:~$ sudo iptables -A INPUT -j DROP
sysadmin@web32:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt In   out    source        destination
    0     0  ACCEPT     tcp  --  any    any    anywhere     anywhere      tcp dpt:htt
    p    0     0  ACCEPT     tcp  --  any    any    anywhere     anywhere      tcp dpt:htt
    ps   0     0  ACCEPT     tcp  --  any    any    192.168.0.0/20  anywhere      tcp dpt:htt
    p    0     0  ACCEPT     tcp  --  any    any    10.42.32.12   anywhere      tcp dpt:htt
    p    0     0  ACCEPT     tcp  --  any    any    10.42.32.7   anywhere      tcp dpt:htt
    p    61  17633  DROP     all  --  any    any    anywhere     anywhere
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt In   out    source        destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt In   out    source        destination
    0     0  ACCEPT     tcp  --  any    any    anywhere     10.43.32.30      tcp dpt:mys
    q1
```

Figure 24: Screenshot of all enforced firewall rules is entering “sudo iptables -L -v”.

- Now lastly, if we enter “sudo iptables -L -v” to check all enforced firewall rules as highlighted in Figure 24.

5. Update Topology

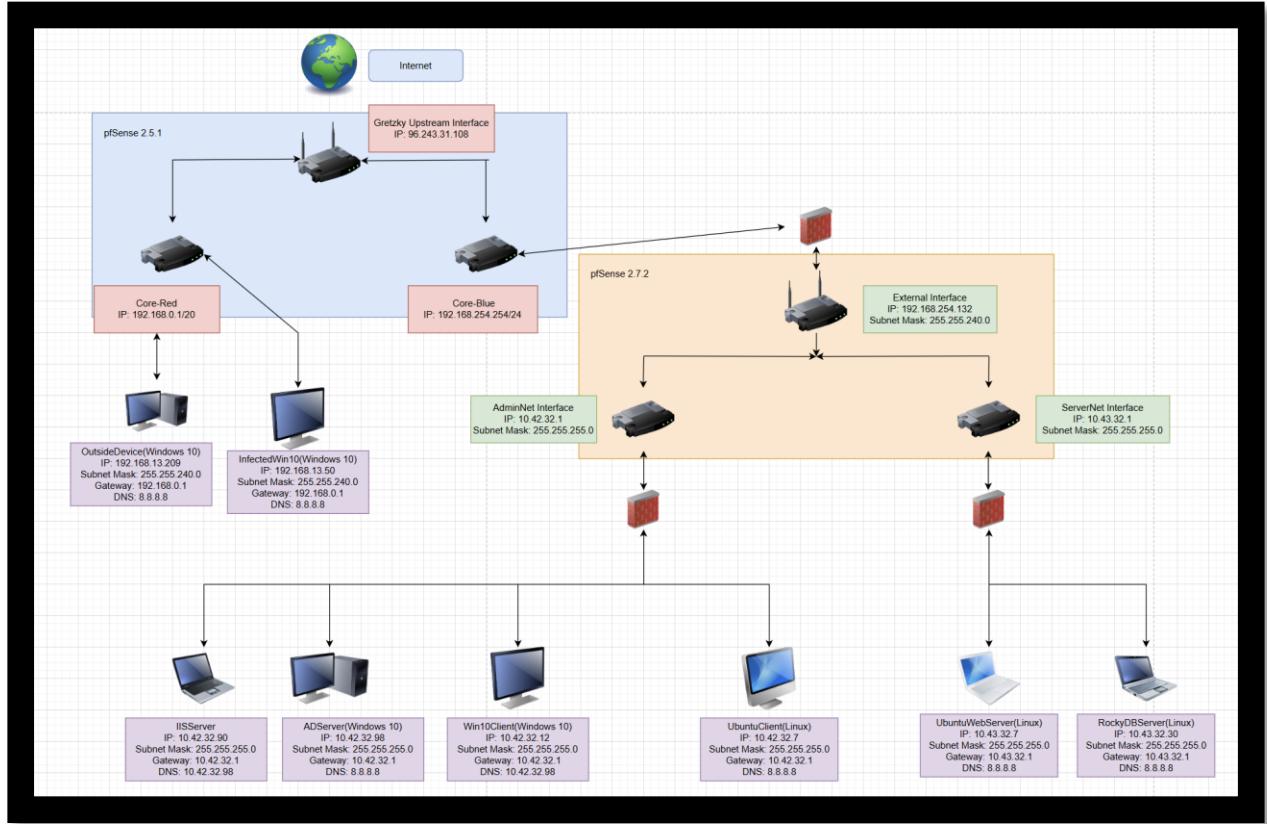


Figure 25: Screenshot of Updated Topology

HW 07 – Services PDF 2

By :- Faraz Ahmed

1. MEMO

To: David Murray, CEO, UBNetDef

From: Faraz Ahmed, Security Engineer, UBNetDef SysSec

Date: 16th October, 2024

Subject: Security Deficiency Remediation Report

Dear Mr. Murray,

Hey I am Faraz Ahmed and I am writing this report to detailing my research about the identification and rehabilitation pf the security issues within our MediaWiki network. Further into the report, we describe the steps taken by SysSec to prevent any vulnerabilities in the system and improve the system of the infrastructure. You can navigate through the table of content to locate that topics.

Table of Contents

1.	MEMO	2
2.	Executive Summary	4
3.	Technical Findings	5
4.	References	6

2. Executive Summary

UBNetDef SysSec finds that SSH services on the UbuntuWebServer are vulnerable to brute force attacks as recorded at 10:30am October 15th, 2024. So, because of this SysSec applied the “Fail2ban service” to block this vulnerability by limiting number of failed attempts to login and blocking all of the potential attackers from attacking the system. This is necessary to stop or mitigate SSH brute force attempts which could compromise all main systems leading to unauthorized access to our systems and modifying or deletion of sensitive or significant data. This could take a significant business impact like massive downtime, unavailability of services, modification or loss of sensitive data and getting access to all of the sensitive or important data about UBNetDef organization.

3. Technical Findings

The identified security issue which needs remediation is the vulnerability if SSH to brute force attacks. The organization can secure the systems from this SSH login by limiting login attempts done by the attackers to protect the sensitive data and also the integrity of the system and server which can be done by using “Fail2ban services”. Now, to install and configure Fail2ban on UbuntuWebServer, UBNetDef can follow following steps: -

- First update the package lists and then install Fail2ban in the system by using-
“sudo apt update
sudo apt install fail2ban”
- Then configure Fail2ban to protect SSH attacks by editing jail.local file-
“sudo nano /etc/fail2ban/jail.local”
Then add these commands to enable SSH protection-
“[sshd]
enabled = true
port = 22
logpath = /var/log/auth.log
maxretry = 5
bantime= 3600 # for 1 hour”
- After that, start and enable the Fail2ban service into the system-
“sudo systemctl start fail2ban
sudo systemctl enable fail2ban”
- Now check if the Fail2ban services or on or not-
“sudo fail2ban-client status sshd”
- This is the output we get-

```
status
|- Number of jail: 1
`- Jail list: sshd
```

Figure 1: Screenshot of result/output for installation of “Fail2ban”.

4. References

- [How To Protect SSH with Fail2Ban on Ubuntu 20.04 | DigitalOcean](#)
- [GitHub - fail2ban/fail2ban: Daemon to ban hosts that cause multiple authentication errors](#)
- [How to Install and configure Fail2ban | SecOps® Solution \(secopsolution.com\)](#)
- [Asked ChatGPT for steps to install Fail2ban](#)

Thanks for taking your time and reading the whole report properly to stop any further future vulnerabilities in the system.

Best Regards,

Faraz Ahmed

Security Engineer, UBNetDef SysSec