HW 07 – Services PDF 1

By :- Faraz Ahmed

Contents

1. RockyDBServer: Configure database operations

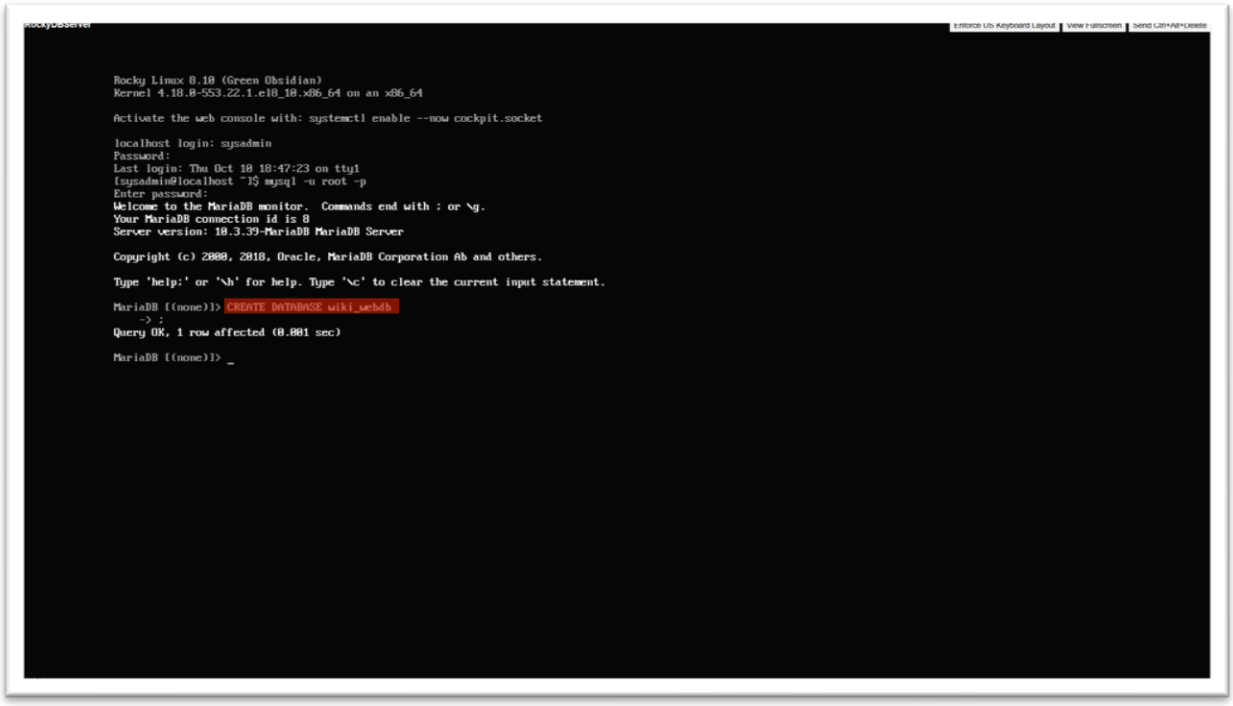

Figure 1: Screenshot of creating new database by entering "CREATE DATABASE wiki_webdb;".

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

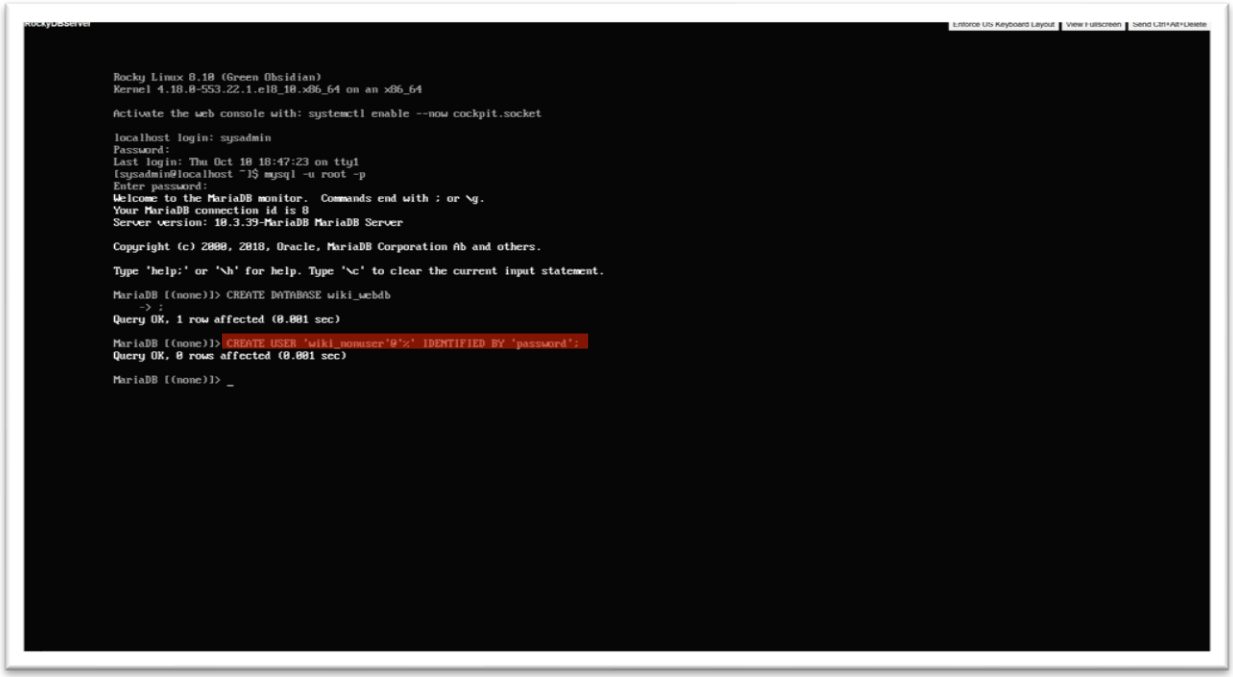

Figure 2: Screenshot of creating new non-root user by entering "CREATE USER 'wiki_nonuser'@'%' IDENTIFIED BY 'password';".

- 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.

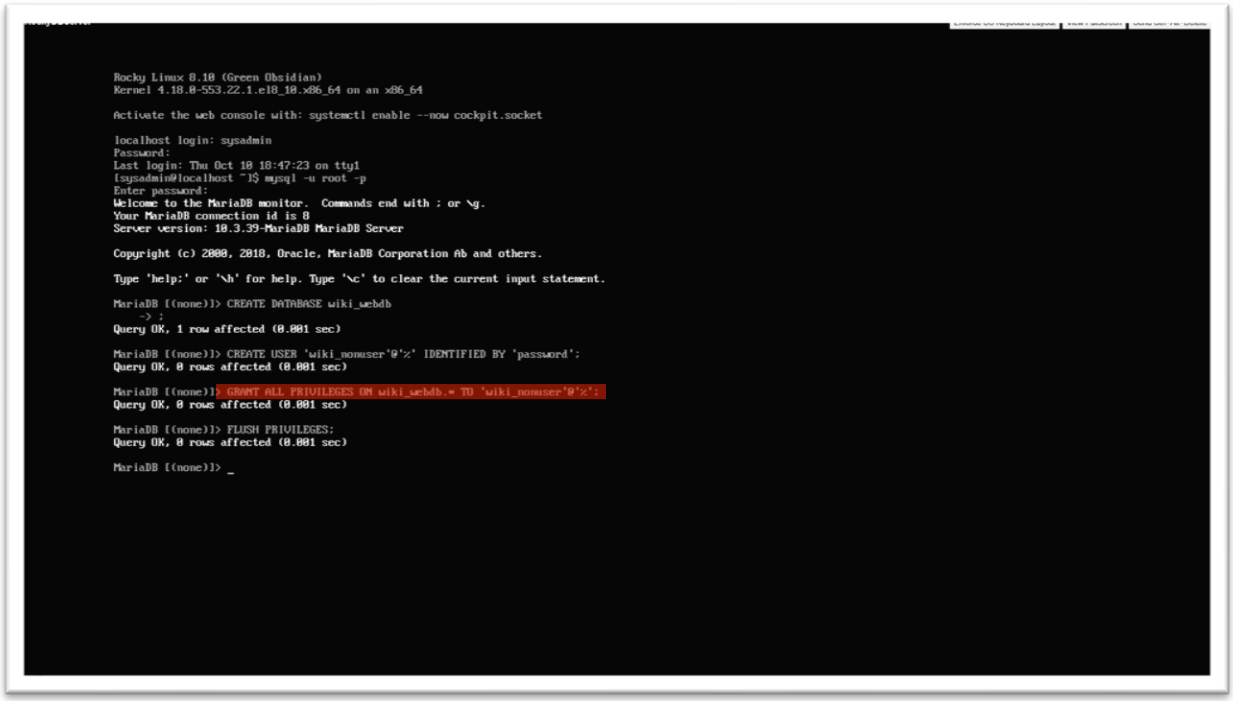

Figure 3: Screenshot of grant new user privileges by entering "GRANT ALL PRIVILEGES ON wiki_webdb.* TO 'wiki_nonuser'@'%';".

- 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

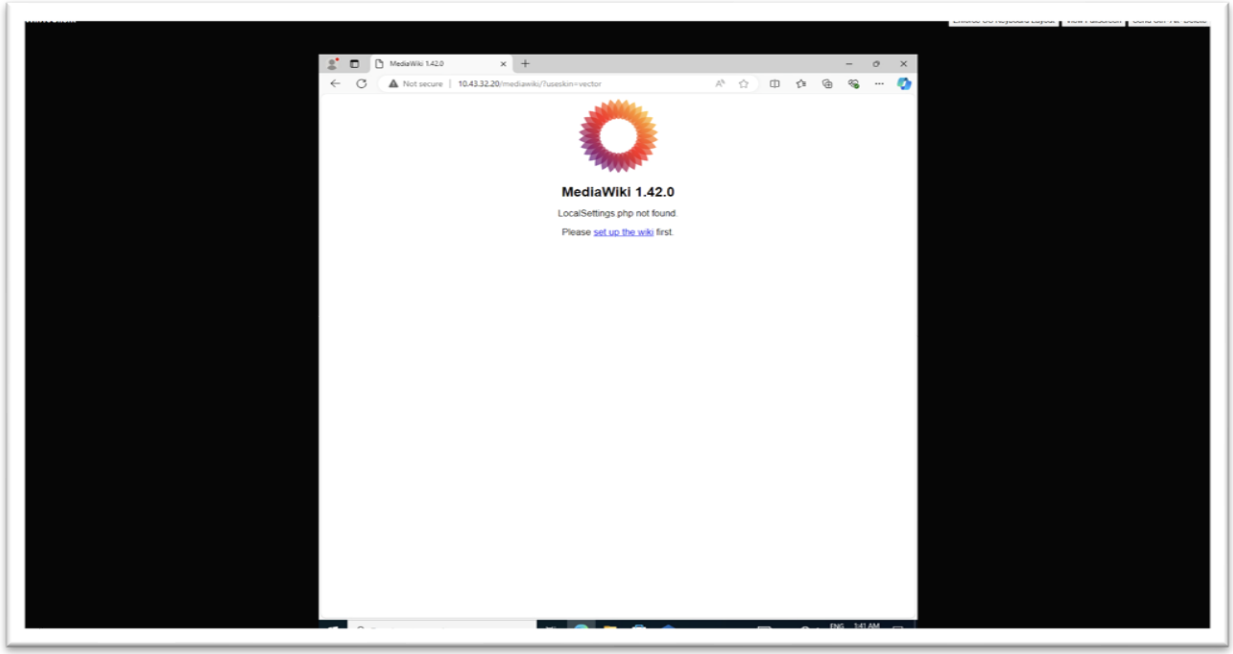

Figure 4: Screenshot of "default home page of MediaWiki".

- 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.

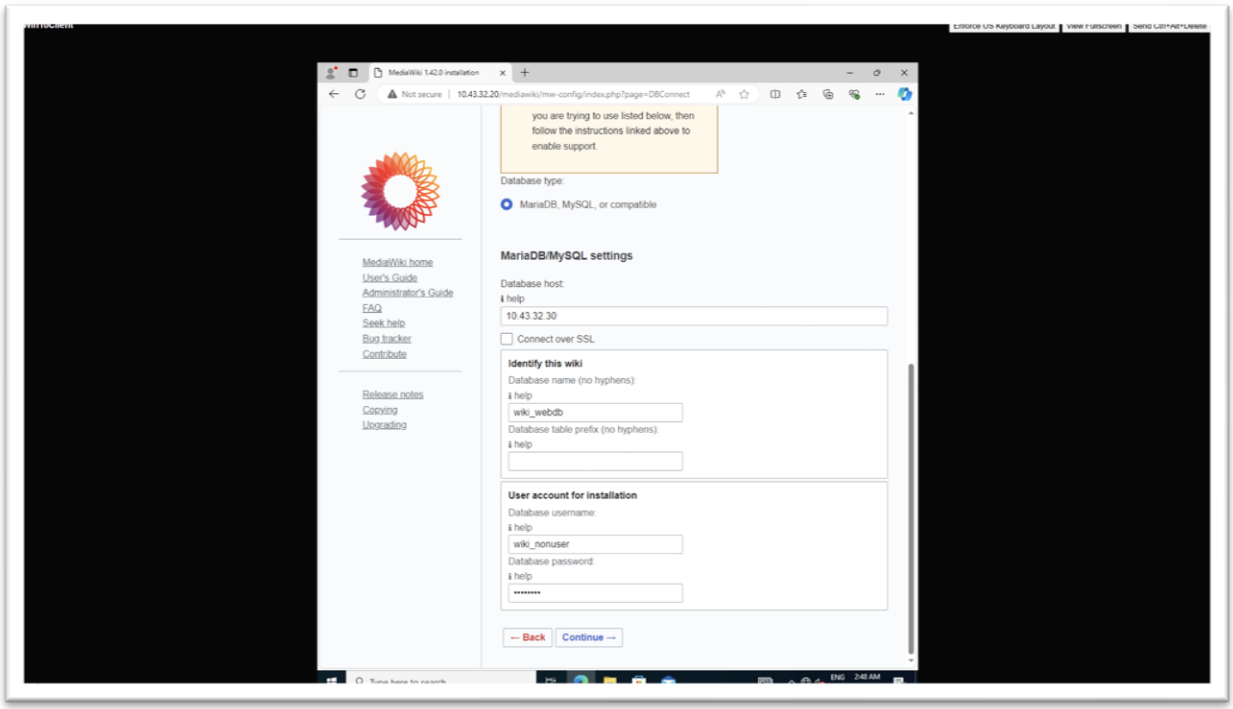

Figure 5: Screenshot of filling details in "MariaDB/MySQL settings".

- 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.

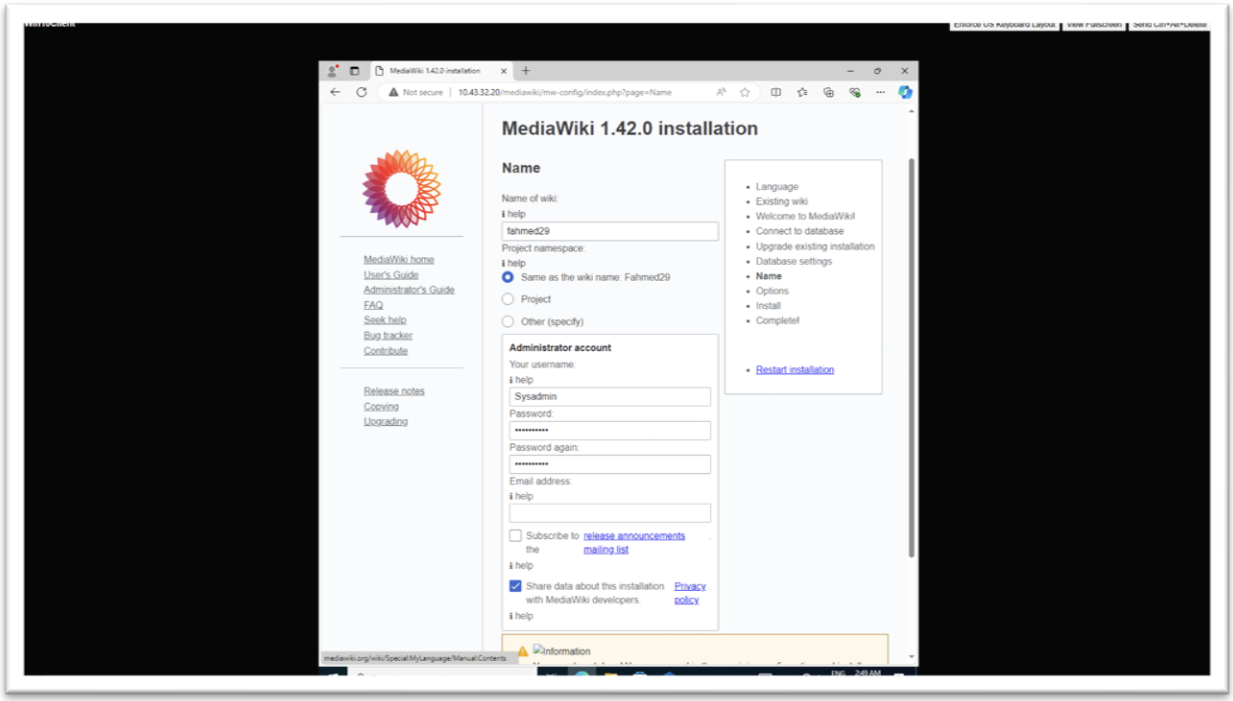

Figure 6: Screenshot of entering details of "Administrative MediaWiki user".

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

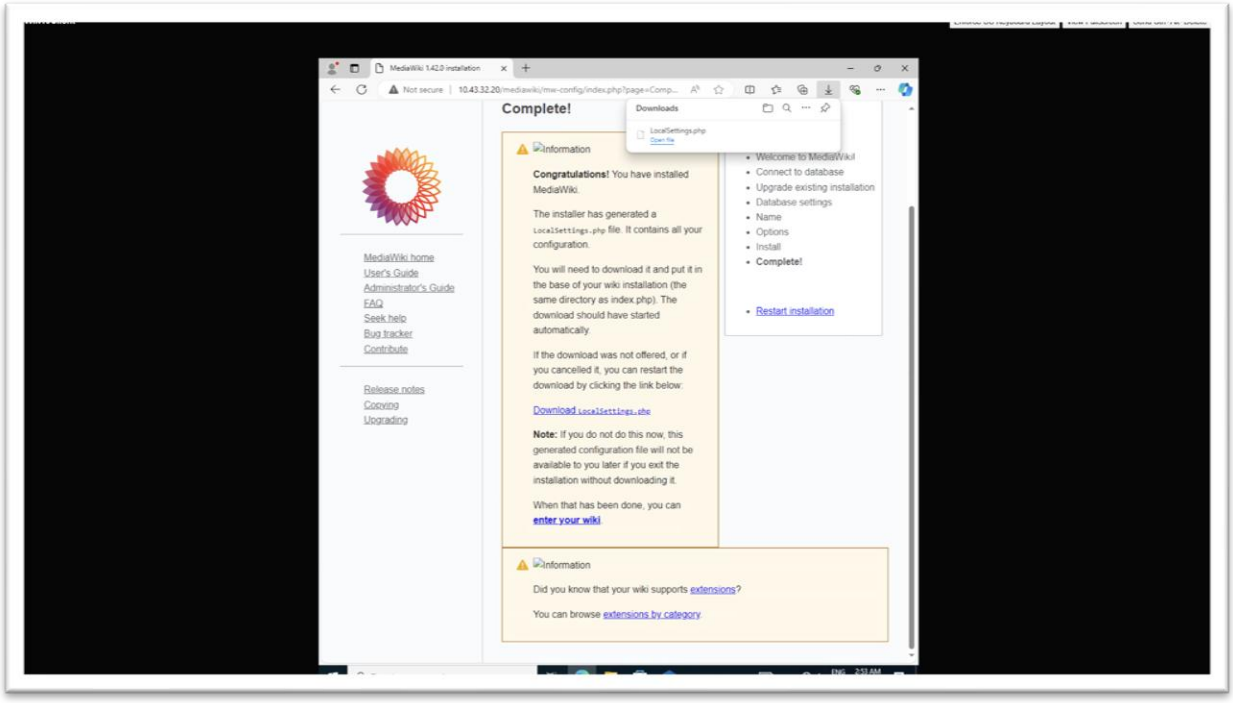

Figure 7: Screenshot of Installation complete and "LocalSettings.php" downloaded.

- 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

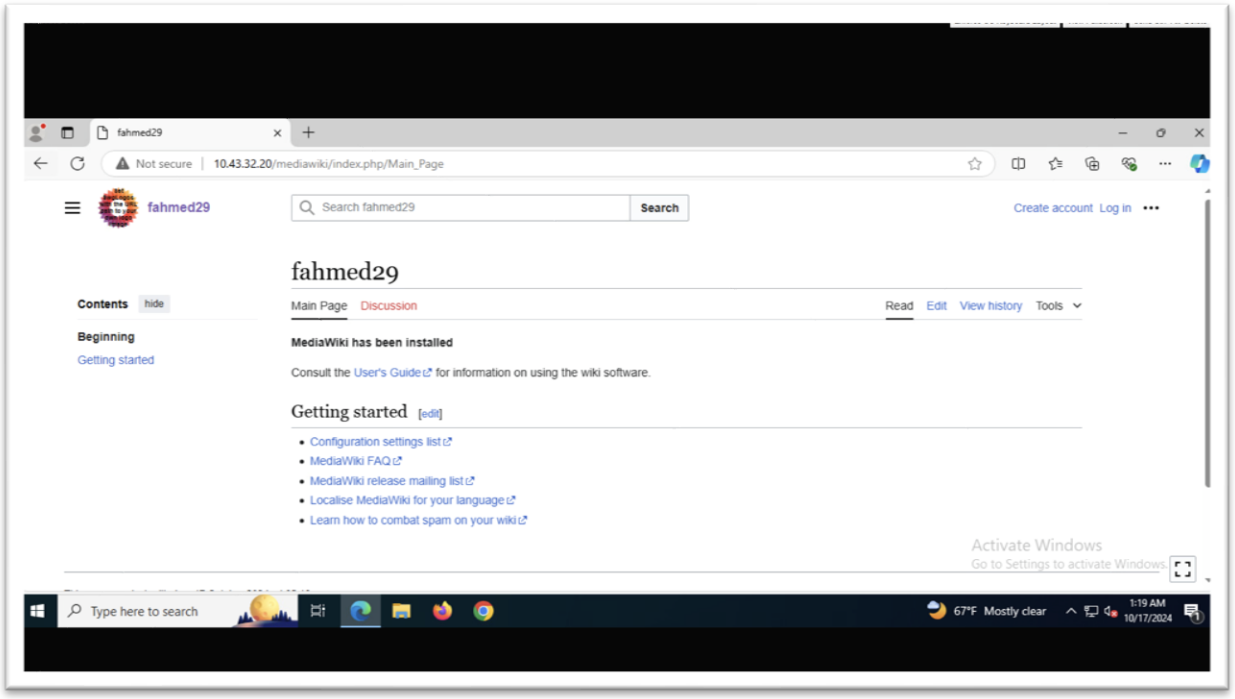

Figure 8: Screenshot of edit "Main Page into UBIT name".

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

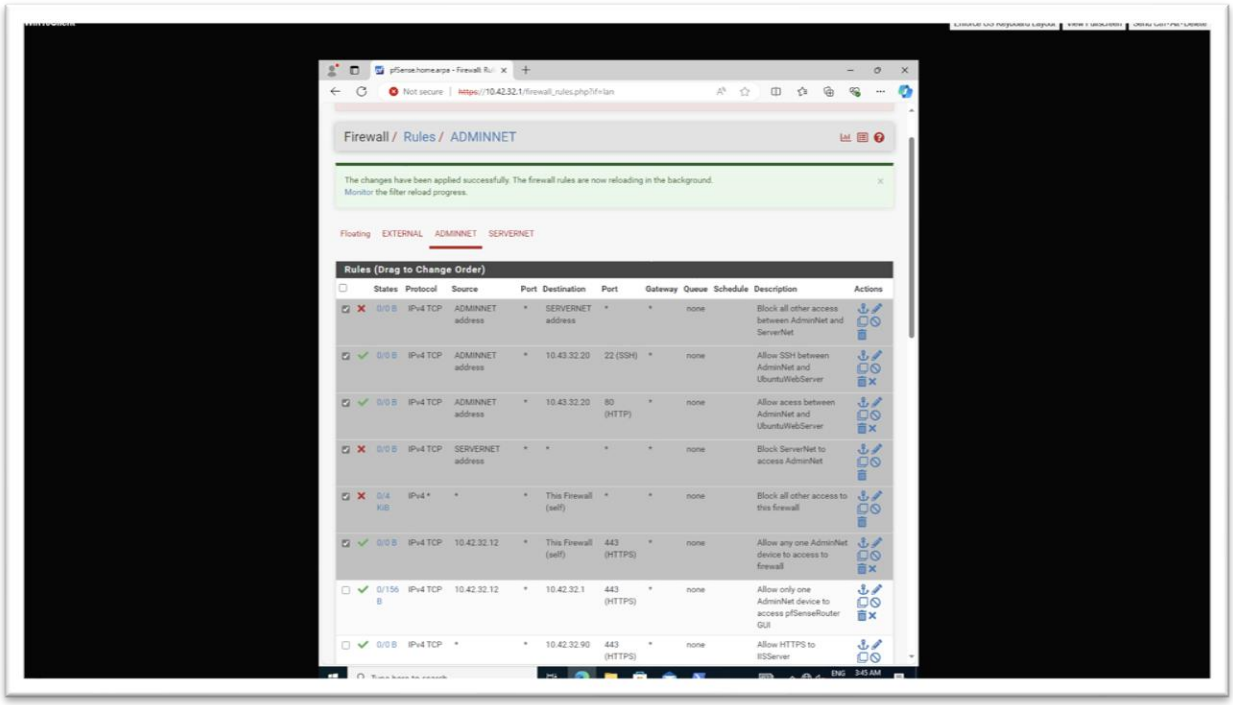

Figure 9: Screenshot of "all highlighted firewall rules using in this tasks".

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

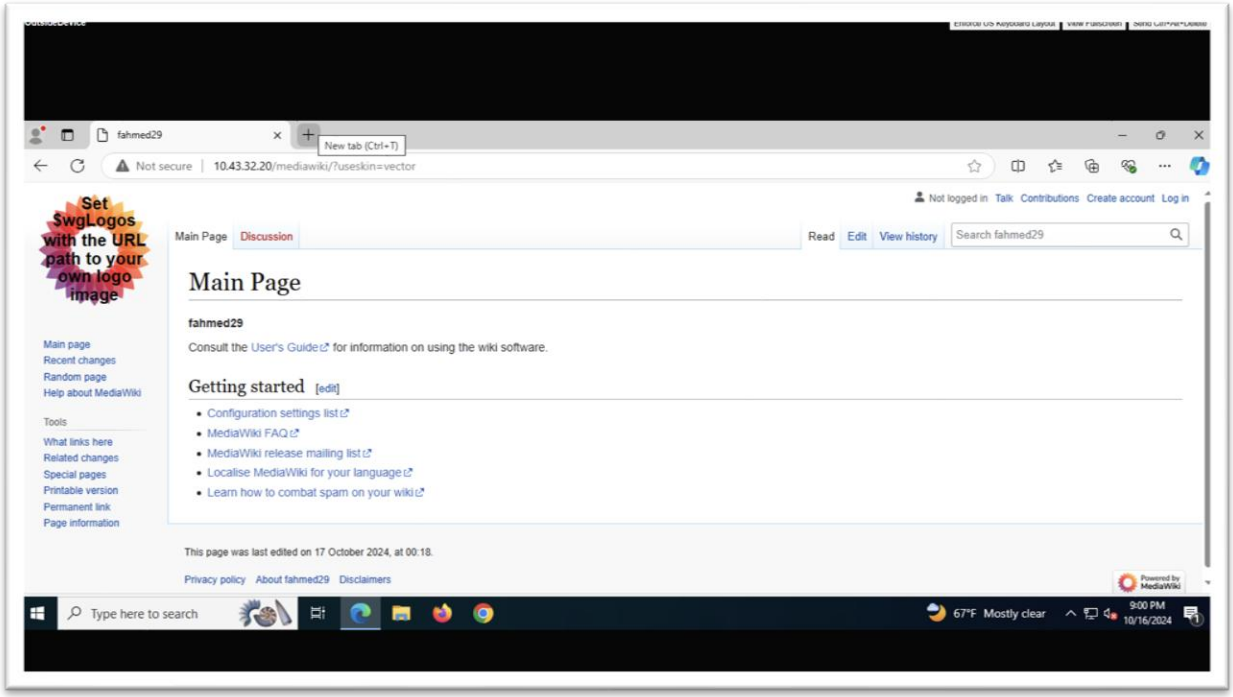

Figure 10: Screenshot of proof that "OutsideDevice can access MediaWiki".

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

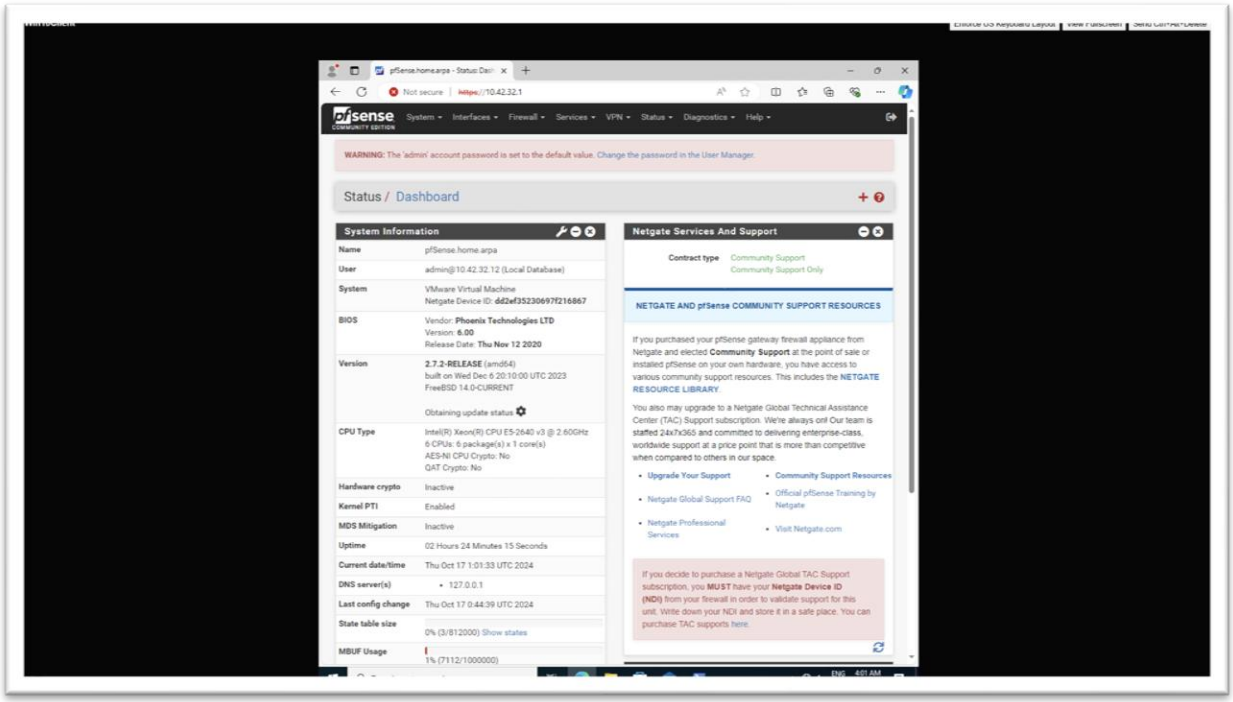

Figure 11: Screenshot of proof that "One AdminNet device (Win10Client) can access pfsense".

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

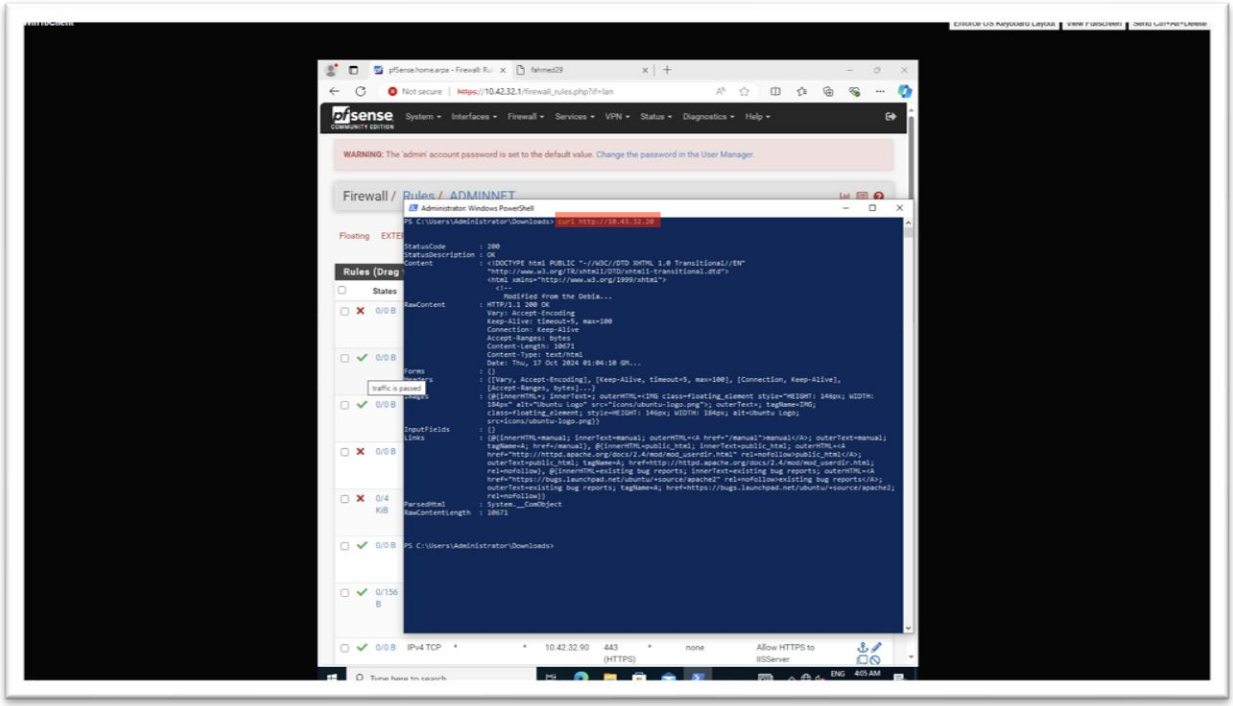

Figure 12: Screenshot of "AdminNet device accessing UbuntuWebServer using http".

- 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.

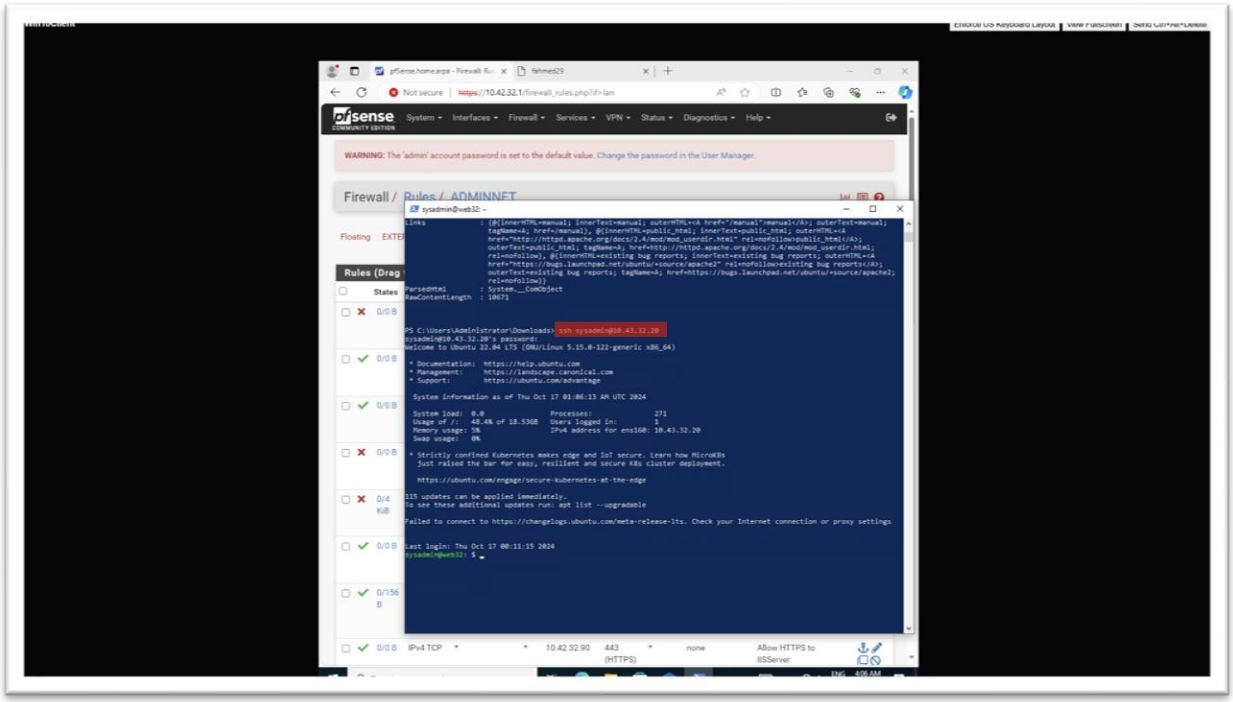

Figure 13: Screenshot of "AdminNet device accessing UbuntuWebServer using ssh".

- 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.

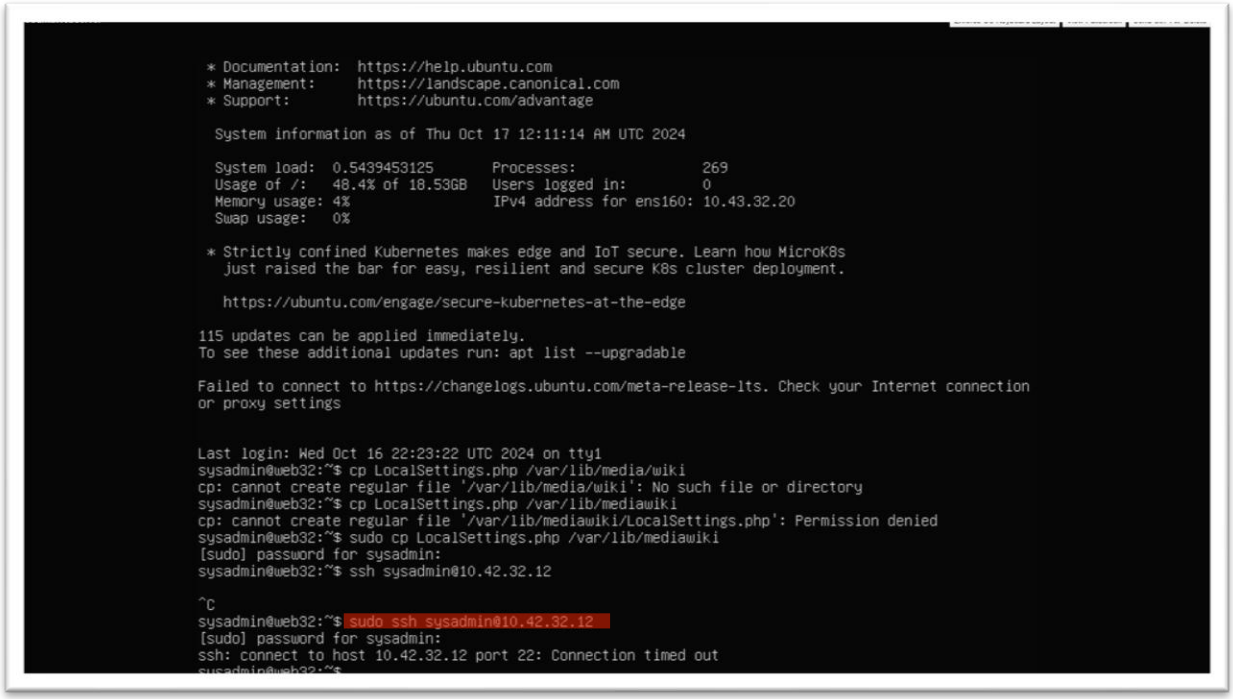

Figure 14: Screenshot of denial of access by entering "sudo ssh sysadmin@10.42.32.12".

- 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.

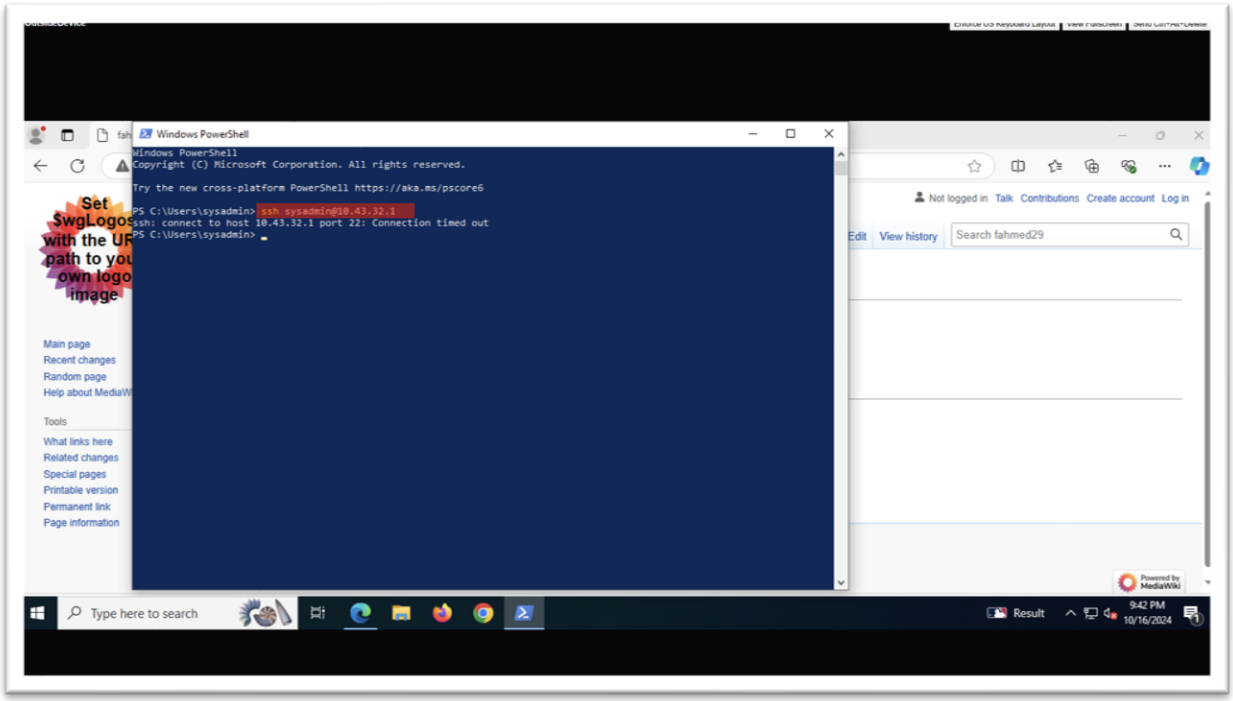

Figure 15: Screenshot of entering "ssh sysadmin@10.43.32.1" to check OutsideDevice access with ServerNet.

- 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.

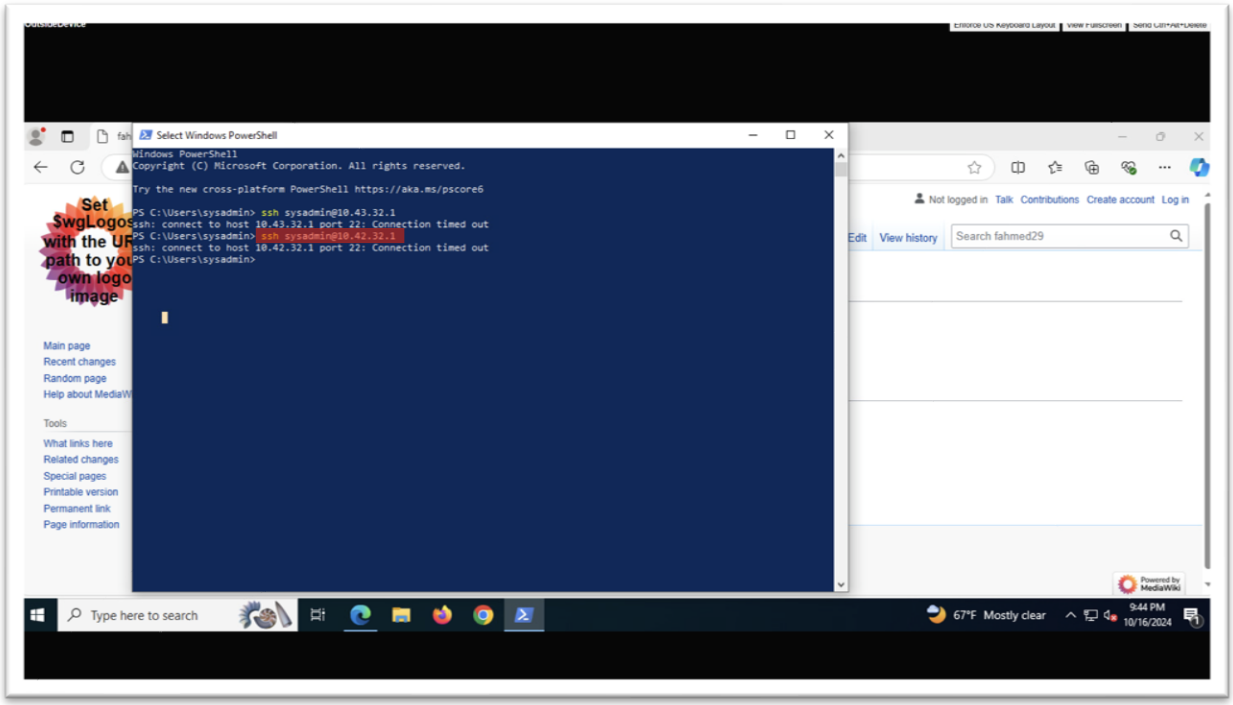

Figure 16: Screenshot of entering "ssh sysadmin@10.42.32.1" to check OutsideDevice access with AdminNet.

- 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

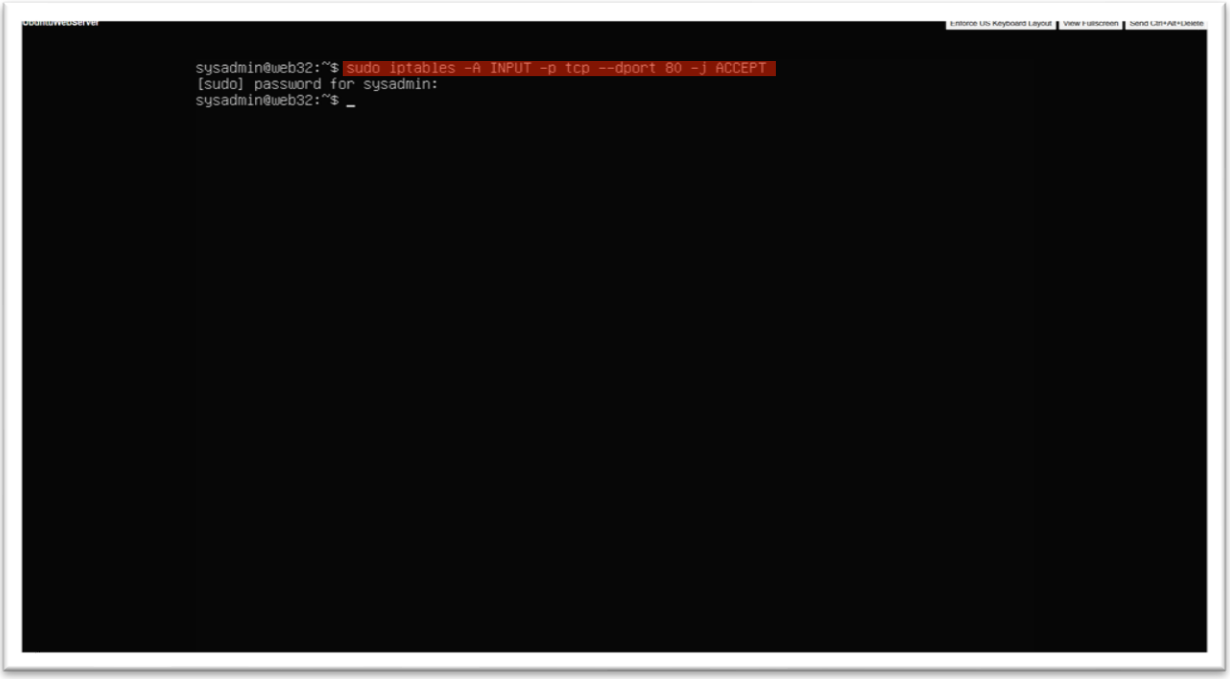

Figure 17: Screenshot of entering "sudo iptables -A INPUT -p tcp –dport 80 -j ACCEPT" to allow for package updates.

- 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.

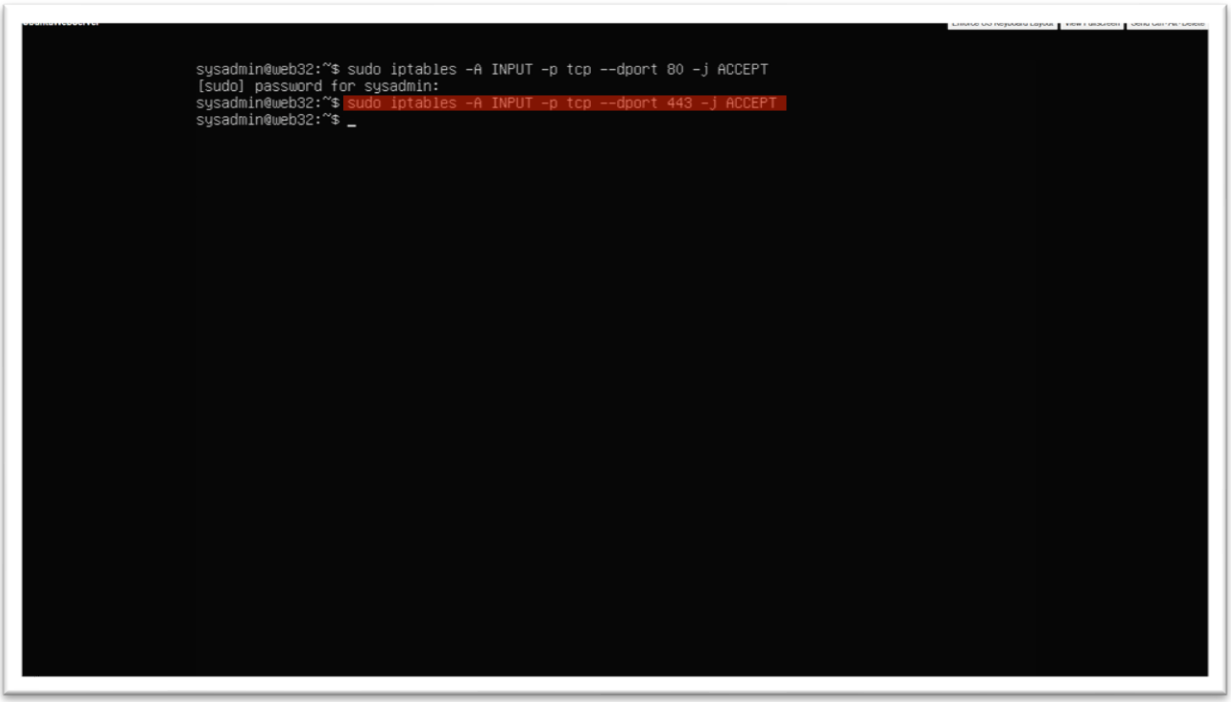

Figure 18: Screenshot of entering "sudo iptables -A INPUT -p tcp –dport 443 -j ACCEPT" to allow for package updates.

- 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.

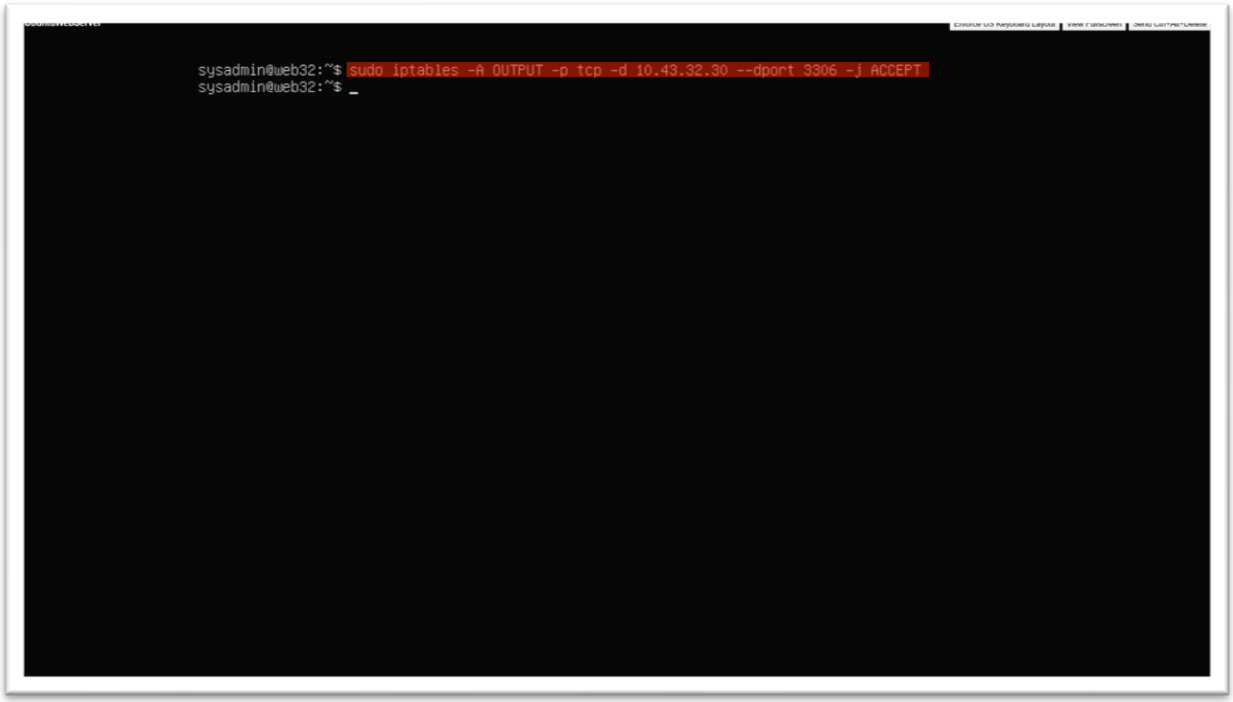

Figure 19: Screenshot of entering "sudo iptables -A OUTPUT -p tcp -d 10.43.32.30 --dport 3306 -j ACCEPT" to allow all necessary connections to RockyDBServer.

- 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.

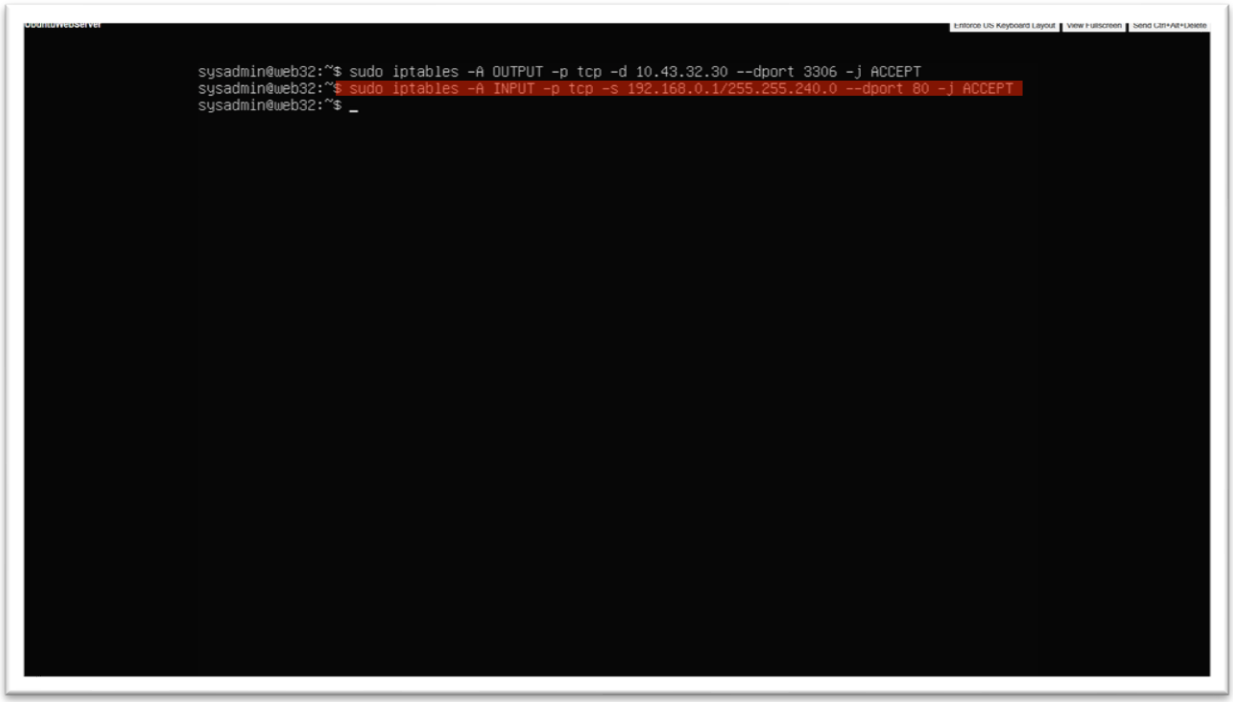

Figure 20: Screenshot of entering "sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.2555.240.0 --dport 80 -j ACCEPT" to get requests from Gretzky's Core-Red subnet.

- So now, we enter "sudo iptables -A INPUT -p tcp -s 192.168.0.1/255.2555.240.0 --dport 80 -j ACCEPT" to get requests from Gretzky's Core-Red subnet as highlighted in Figure 20.

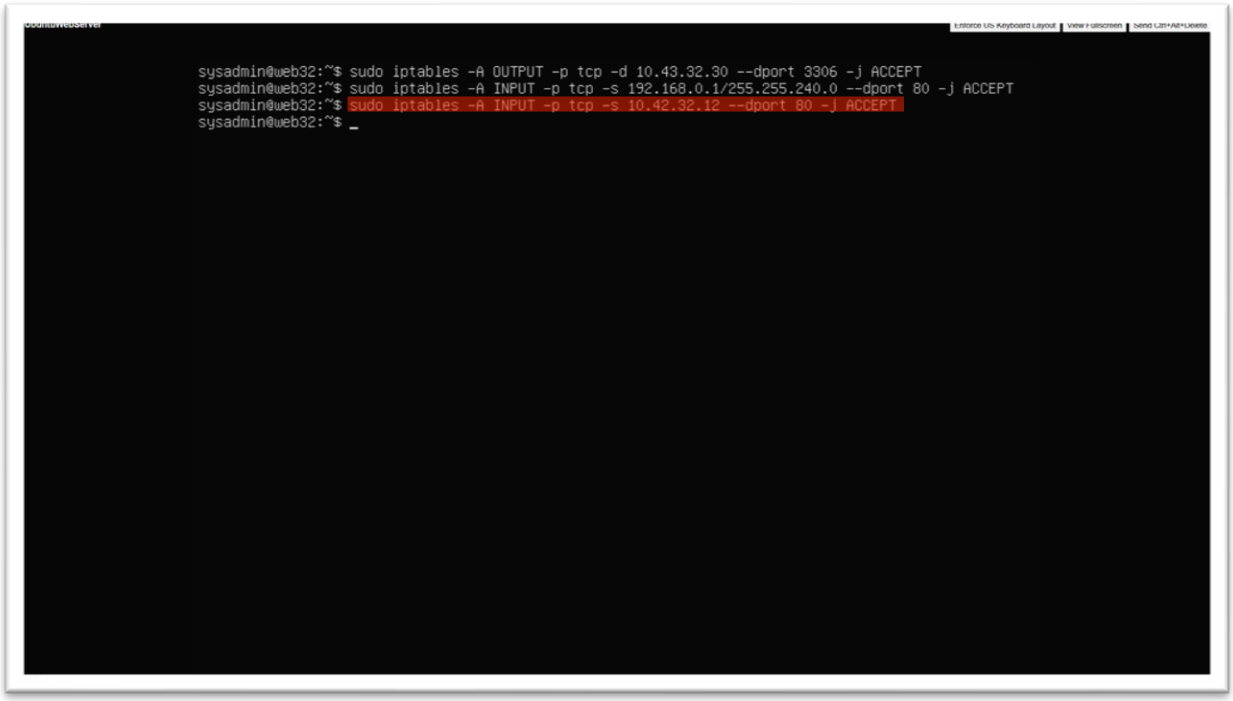

Figure 21: Screenshot of entering "sudo iptables -A INPUT -p tcp -s 10.42.32.12 --dport 80 -j ACCEPT" to get requests from Win10Client.

- 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.

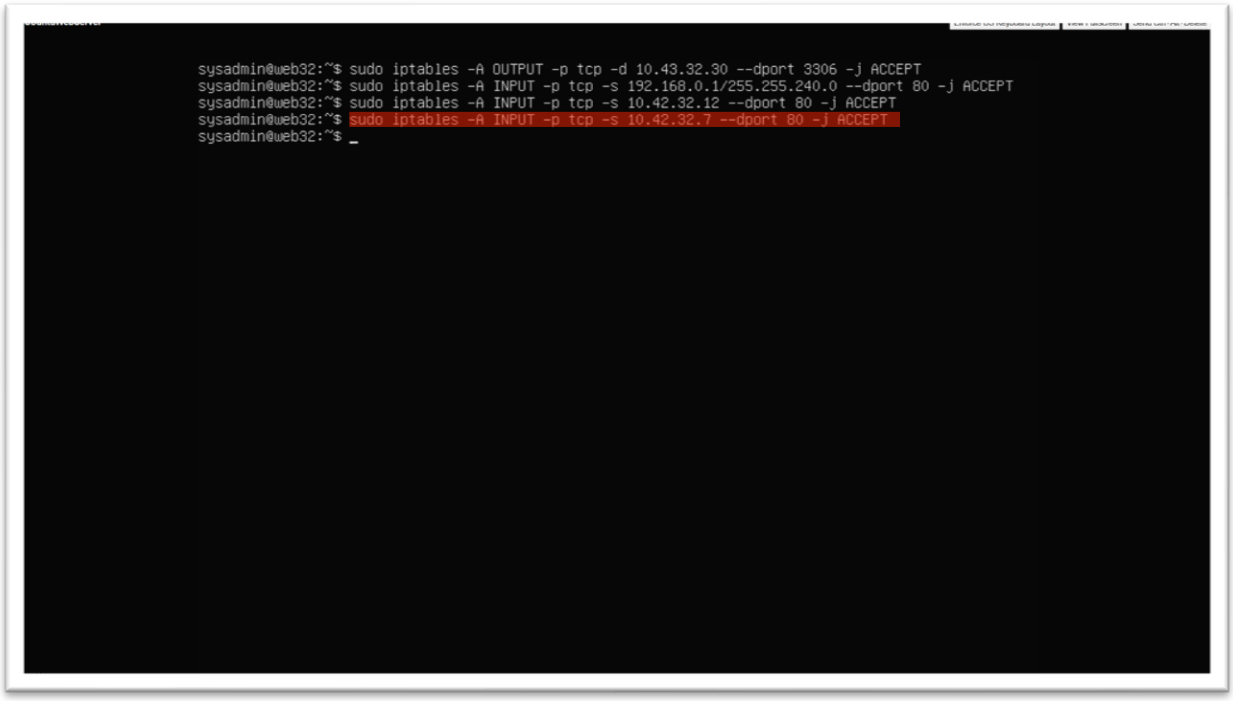

Figure 22: Screenshot of entering "sudo iptables -A INPUT -p tcp -s 10.42.32.7 --dport 80 -j ACCEPT" to get requests from UbuntuClient.

- 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.

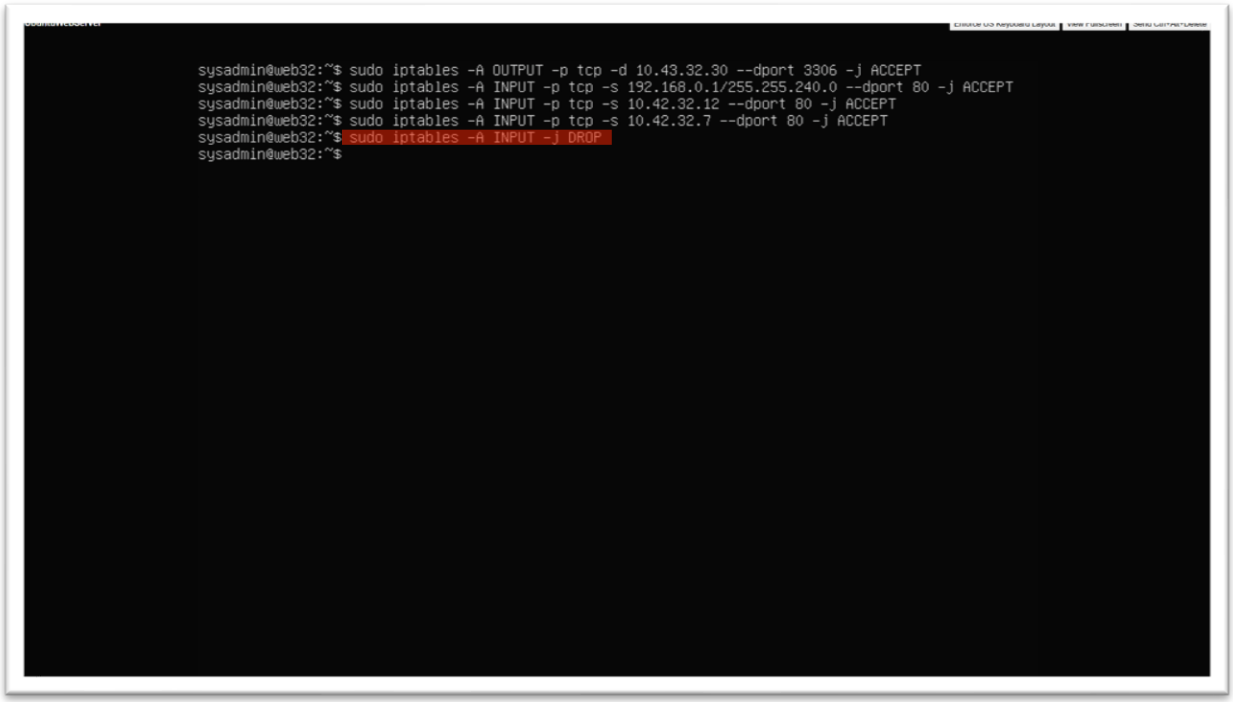

Figure 23: Screenshot of entering "sudo iptables -A INPUT -j DROP" to deny or drop all rules.

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

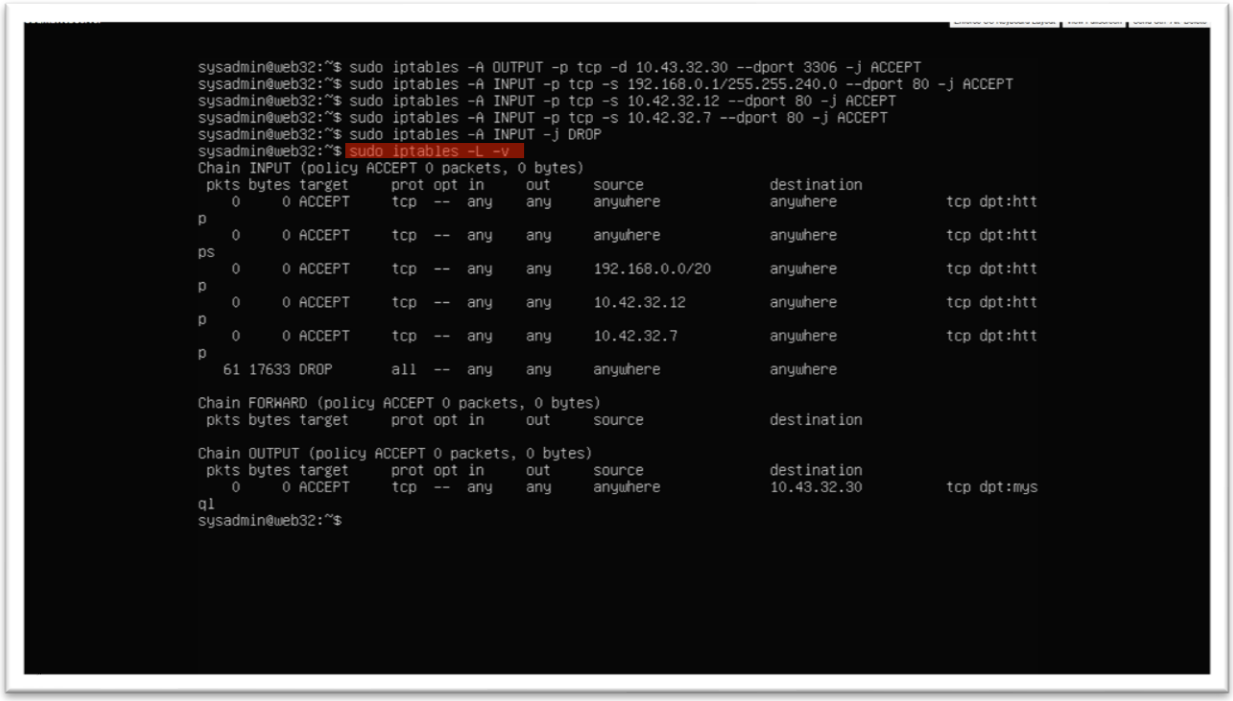

Figure 24: Screenshot of all enforced firewall rules is entering "sudo iptables -L -v".

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

5. Update Topology

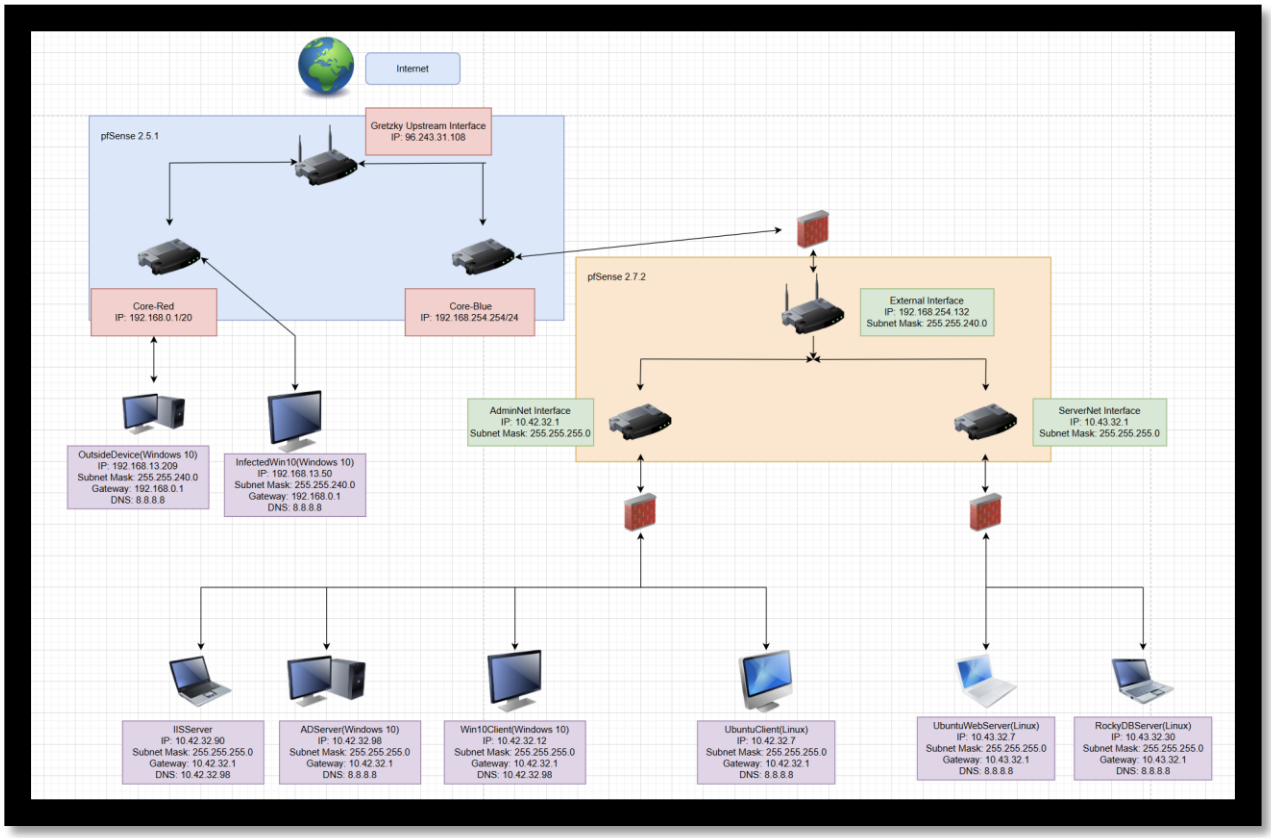

Figure 25: Screenshot of Updated Topology

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

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-

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