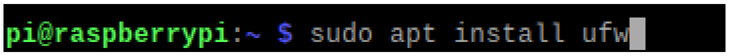
Matthew Townsend
4/13/23

Kali Defense

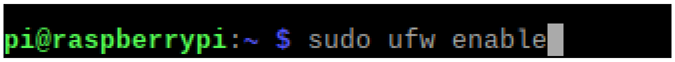
In this lab, the objective was to prevent the attack we carried out in the previous lab which involved reading packets that write data and then repackaging the data to write to the HMI activating the circuit without anything touching the buttons on the circuit. To do this we used UFW or Uncomplicated FireWall. This program allows for the creation of firewalls to protect the system. This method of risk mitigation was chosen as in theory an industrial network is air-gapped meaning there is no outside network but even on one if the machines do not need to communicate they should be blocked.

To begin we install ufw on the pi.



```
pi@raspberrypi:~ $ sudo apt install ufw
```

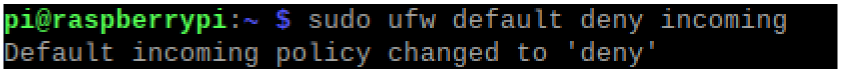
Press y if prompted then the service is installed and now needs to be enabled and given rules.



```
pi@raspberrypi:~ $ sudo ufw enable
```

To enable the firewall or turn it on the firewall will start following its rules. If it is not enabled no rules added will be followed.

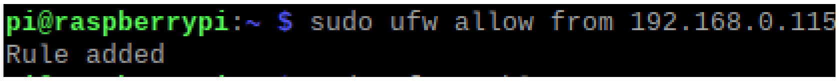
Rule #1



```
pi@raspberrypi:~ $ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
```

The default deny incoming rule changes the default mode of the firewall to deny all incoming traffic. It does not allow any incoming packets to reach their destination ports even legitimate packets from the HMI.

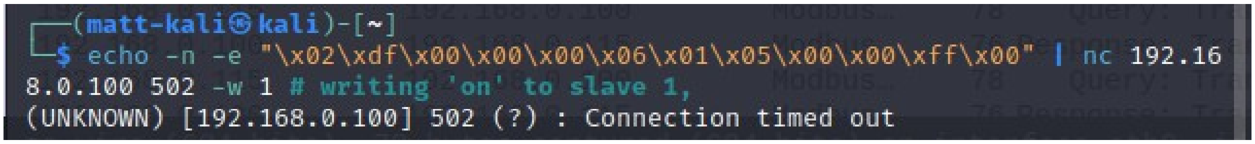
Rule #2



```
pi@raspberrypi:~ $ sudo ufw allow from 192.168.0.115
Rule added
```

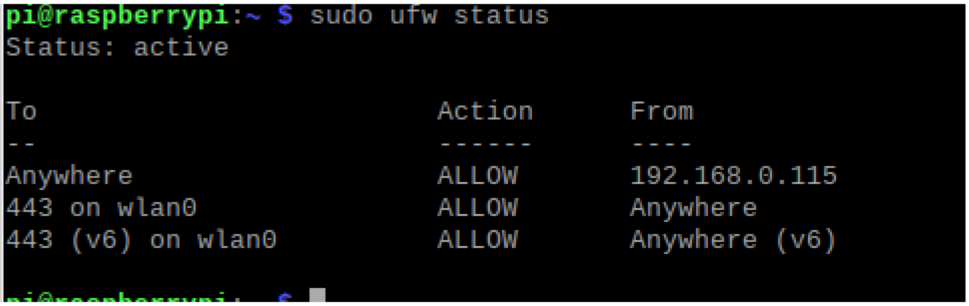
The allow from IP rule does as it states allow traffic from the IP to the pi. So in this case it is allowing traffic from the HMI to reach its destination on the PLC.

With these rules, we can successfully block the previous attack. Running the attack on the Kali VM resulted in this.



```
┌──(matt-kali㉿kali)-[~]
└─$ echo -n -e "\x02\xdf\x00\x00\x00\x06\x01\x05\x00\x00\xff\x00" | nc 192.16
8.0.100 502 -w 1 # writing 'on' to slave 1,
(UNKNOWN) [192.168.0.100] 502 (?) : Connection timed out
```

The false packet could not connect to the PLC and reach its destination. The firewall successfully blocked the traffic while allowing the HMI to function
The status on the ufw.



```
pi@raspberrypi:~ $ sudo ufw status
Status: active

To                           Action      From
--                           ------      ----
Anywhere                     ALLOW       192.168.0.115
443 on wlan0                 ALLOW       Anywhere
443 (v6) on wlan0            ALLOW       Anywhere (v6)

pi@raspberrypi:~ $
```

Allow all traffic from 192.168.0.115, the HMI, and all traffic on port 443 or HTTPS, this was added so I can save and retrieve the images through the web instead of physical storage.

sudo ufw allow in on wlan0 to any port 443.

This is a basic level of defense on the device level for an infrastructure environment, placing a firewall to reject incoming packets from unknown or unwanted devices.