Willie de Klerk

WIL620 E.L.O 2

Student Number: 20230254

Student Year: 2 (2024)

CTUTRAINING.AC.ZA | 0861 100 395 | ENQUIRY@CTUTRAINING.CO.ZA

CTU training solutions

Private
Tertiary
Education
Provider

Declaration of Authenticity

A critical aspect of any assignment is *authenticity*. Because you are completing much of the work for the assignments *unsupervised*, the examiner must be convinced that it is all your work. For this reason, you must complete the *Declaration of Authenticity* provided in the study guide and have it counter-signed by your manager, mentor, or lecturer.

NB	The declaration of authenticity is a legal document, and if found that you have made a false declaration, then not only will your results be declared null and void, but you could also have criminal charges brought against you. It is not worth taking the risk!

Please complete the declaration of authenticity below for all assignments:

DECLARATION OF AUTHENTICITY

I _____WILLIE DE KLERK___________________________________ hereby

declare that the contents of this assignment are entirely my work except for the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	Date
Exit Level Outcome 2	**2024/07/26**

Signature: ____________________ Willie ________________ Date: _________**2024/07/26**______

Company/Mentor Stamp

Contents

Declaration of Authenticity...1
Introduction.. **3**
What is Pi.Alert?..3
Business Justification...3
Debian instance Install Checklist...4
Pi.Alert installation Checklist..4
Debian Instance Configuration...**4**
1. Configure the bridge interface...4
2. Install Tools...**4**
3. Container directory Creation and Debian Instance download..............................5
4. Debian instance Initial Configuration Items...5
5. Ensure that the Debian Instance can boot when the edge node starts up...................6
Pi.Alert Installation...**7**
Automated Installation Script...7
Configuring login Credentials...8
Using the GUI to edit, backup, restore and maintain the database.........................8
Editing...8
Creating a database backup..9
Restoring a database backup.. **10**
Conclusion...**10**
Bibliography... **10**

Introduction

For the completion of exit level outcome 2) Design, use and maintain a database in a corporate environment I was tasked by my mentor (Ronald Bartels) to install Pi.Alert on the Fusion Broadband SD-WAN edge node. This task involved many configuration steps leading to me using and maintaining the database used by Pi.Alert in a corporate environment.

What is Pi.Alert?

At its core Pi.Alert is an intrusion detection system (ids) for local area networks (LANs) and wireless local area networks (WLANs). It offers the unique functionality of allowing administrators to scan connected devices and receive notifications for unknown device connections or known always connected devices that have disconnected.

Scanning methods

Pi.Alert makes use of various scanning techniques to identify the hosts on a network.

To find bad actors within a network Pi.Alert makes use of various scanning techniques such as scanning arp frames to search for devices on the network. Additional scanning for rogue DHCP servers. This is accomplished by sending out DHCP requests into the network and then ratting out the DHCP server to the system administrator.

If dnsmasq is being used for DHCP Pi.Alert can examine the DHCP leases to further discover hosts on the network that were not discovered by other methods. Additionally scanning and monitoring of device health is done via the ICMP ping command.

Business Justification

The Implementation of Pi.Alert within a production environment does not break business budgets since it is a free and open source tool that can be deployed on existing infrastructure or even a Raspberry Pi.

The setup of Pi.Alert does not require constant management and fiddling, nor does it require an entire team of IT staff to use and setup. It has a user-friendly web interface and a command line interface that can be used.

Pi.Alert integrates with other open source tools such as Nmap allowing the staff to scan for device vulnerabilities. Furthermore Pi.Alert can help staff identify faulty or misconfigured equipment.

Pi.Alert can potentially be used to keep time in a human resource manner, meaning when people turn their devices on to work, connecting to the network and when they turn them off, disconnecting from the network and leaving the office.

Version Information

With regards to the version of Pi.Alert, I will be installing the github fork made by leiweibau found here. All the information gathered by Pi.Alert is stored on a database found within the configuration known as pialert.db. With the built in web interface I am able to backup and restore the database.

To execute this task I have created a high level checklist. It serves as a guide for the installation of Pi.Alert.

Debian instance Install Checklist

- ☐ 1. Configure the bridge interface to allow the container to function within the edge node.
- ☐ 2. Install tools (systemd-container and debootstrap)
- ☐ 3. Make the required directory for the container and download the debian instance on which Pi.Alert will be running.
- ☐ 4. Perform Initial configuration items on the debian instance.
- ☐ 5. Ensure that the debian instance will start on system boot.

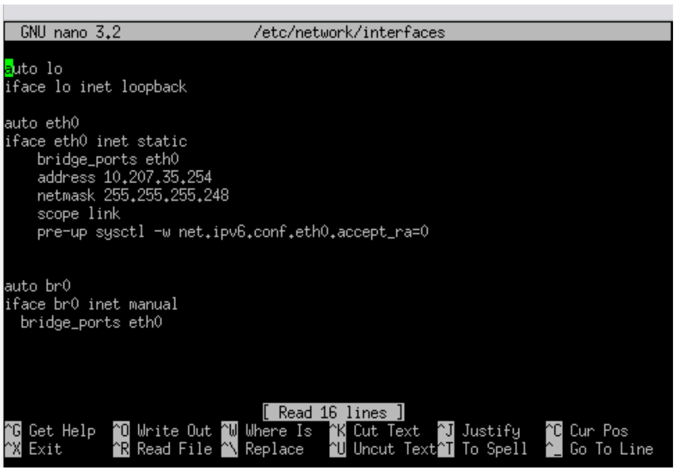
Pi.Alert installation Checklist

- ☐ 1. Make use of the automated installation script to install Pi.Alert
- ☐ 2. Configure login credentials.
- ☐ 3. Take steps through the graphical user interface to edit, backup and restore the database.

Debian Instance Configuration

1. Configure the bridge interface

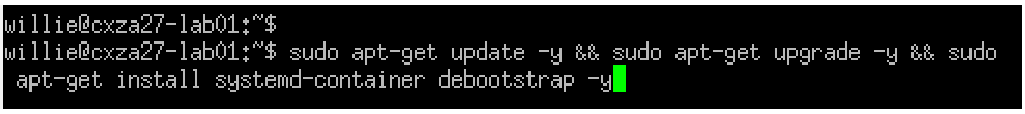
In order for the container to function and be able to reach devices it needs to be connected to a bridge interface, sed bridge interface should be connected "bridged" to the physical LAN interface which in this case would be eth0. The following configuration takes place on the host operating system (FB SD-WAN edge node). This configuration takes place in /etc/network/interfaces. Afterwards the networking service should be restarted (sudo systemctl restart networking.service)



```
  GNU nano 3.2                 /etc/network/interfaces

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    bridge_ports eth0
    address 10.207.35.254
    netmask 255.255.255.248
    scope link
    pre-up sysctl -w net.ipv6.conf.eth0.accept_ra=0


auto br0
iface br0 inet manual
  bridge_ports eth0




                              [ Read 16 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

2. Install Tools

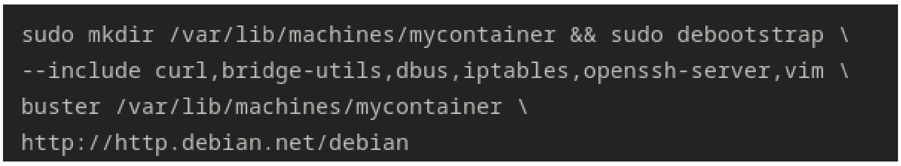
The following needs to be installed on the SD-WAN edge node: systemd-container and debootstrap.



```
willie@cxza27-lab01:~$
willie@cxza27-lab01:~$ sudo apt-get update -y && sudo apt-get upgrade -y && sudo
 apt-get install systemd-container debootstrap -y
```

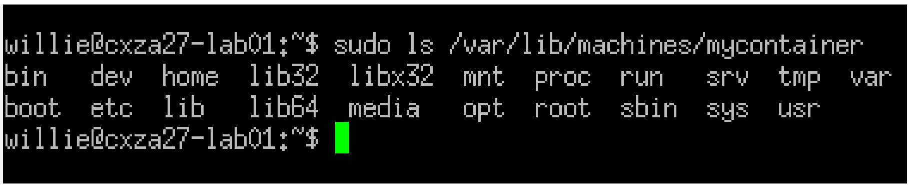
3. Container directory Creation and Debian Instance download.

For systemd-nspawn to function as intended it is important to install the container into a directory made within /var/lib/machines. The naming of the container directory is important as it should maintain clarity, consistency and it should be predictable where possible.



```
sudo mkdir /var/lib/machines/mycontainer && sudo debootstrap \
--include curl,bridge-utils,dbus,iptables,openssh-server,vim \
buster /var/lib/machines/mycontainer \
http://http.debian.net/debian
```

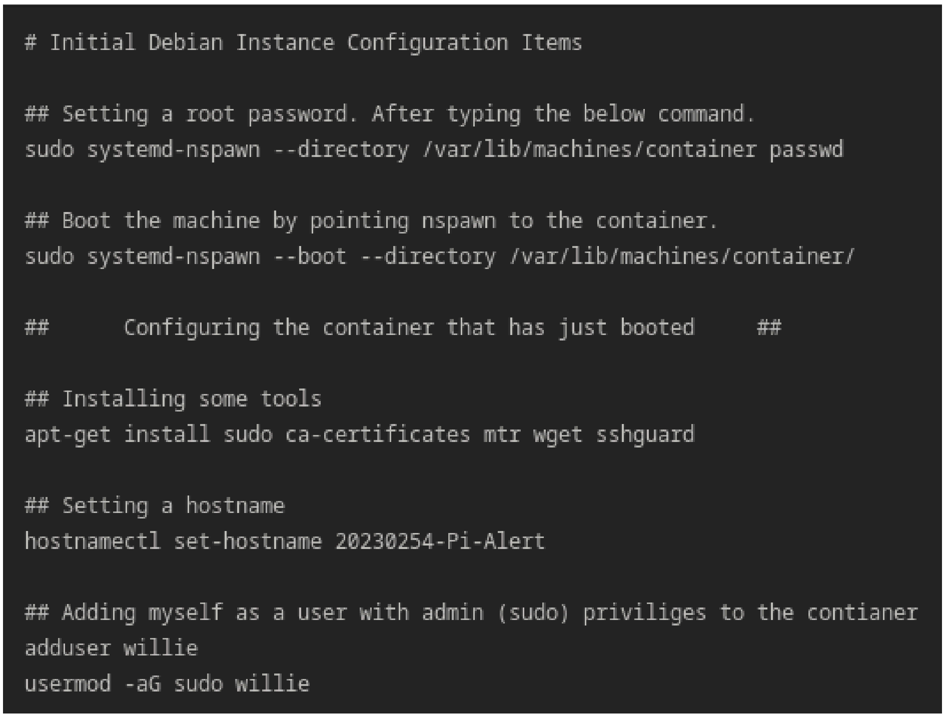
To verify that the container folder has been created and the instance image has been downloaded to the correct directory:



```
willie@cxza27-lab01:~$ sudo ls /var/lib/machines/mycontainer
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr
willie@cxza27-lab01:~$ 
```

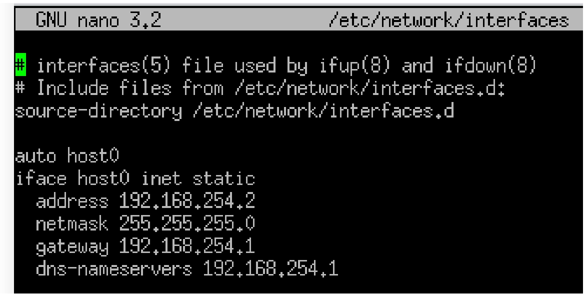
4. Debian instance Initial Configuration Items

For simplicity I added the initial configuration steps to a text file and took a screenshot, taking screenshots in between each step would not result in much more useful information.



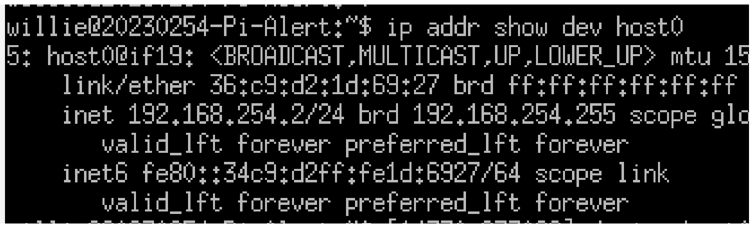
```
# Initial Debian Instance Configuration Items

## Setting a root password. After typing the below command.
sudo systemd-nspawn --directory /var/lib/machines/container passwd

## Boot the machine by pointing nspawn to the container.
sudo systemd-nspawn --boot --directory /var/lib/machines/container/

##      Configuring the container that has just booted     ##

## Installing some tools
apt-get install sudo ca-certificates mtr wget sshguard

## Setting a hostname
hostnamectl set-hostname 20230254-Pi-Alert

## Adding myself as a user with admin (sudo) priviliges to the contianer
adduser willie
usermod -aG sudo willie
```

For addressing I opted to create static configuration within the container instead of setting it via dhcp. Alternatively I could have set this within openwrt to a specific MAC address to make a static binding.



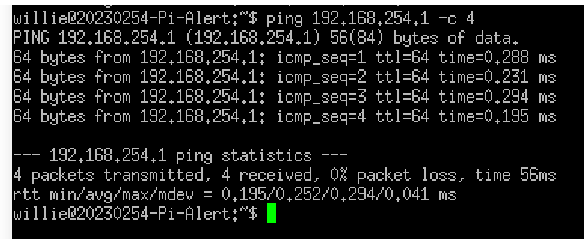
```
  GNU nano 3.2                        /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto host0
iface host0 inet static
  address 192.168.254.2
  netmask 255.255.255.0
  gateway 192.168.254.1
  dns-nameservers 192.168.254.1
```

After saving these changes and reloading networking service with systemctl, I verified that the changes have been loaded by issuing the following command.



```
willie@20230254-Pi-Alert:~$ ip addr show dev host0
5: host0@if19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 15
    link/ether 36:c9:d2:1d:69:27 brd ff:ff:ff:ff:ff:ff
    inet 192.168.254.2/24 brd 192.168.254.255 scope glo
       valid_lft forever preferred_lft forever
    inet6 fe80::34c9:d2ff:fe1d:6927/64 scope link
       valid_lft forever preferred_lft forever
```

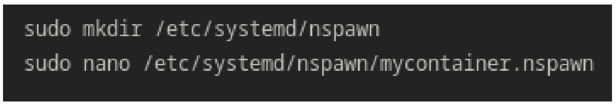
I verify that the default gateway is reachable.



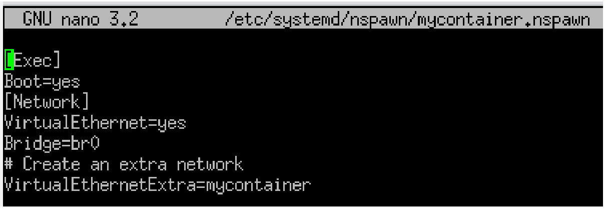
```
willie@20230254-Pi-Alert:~$ ping 192.168.254.1 -c 4
PING 192.168.254.1 (192.168.254.1) 56(84) bytes of data.
64 bytes from 192.168.254.1: icmp_seq=1 ttl=64 time=0.288 ms
64 bytes from 192.168.254.1: icmp_seq=2 ttl=64 time=0.231 ms
64 bytes from 192.168.254.1: icmp_seq=3 ttl=64 time=0.294 ms
64 bytes from 192.168.254.1: icmp_seq=4 ttl=64 time=0.195 ms

--- 192.168.254.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 56ms
rtt min/avg/max/mdev = 0.195/0.252/0.294/0.041 ms
willie@20230254-Pi-Alert:~$
```

5. Ensure that the Debian Instance can boot when the edge node starts up.

The following configuration has taken place on the edge node.

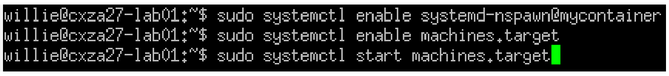


```
sudo mkdir /etc/systemd/nspawn
sudo nano /etc/systemd/nspawn/mycontainer.nspawn
```



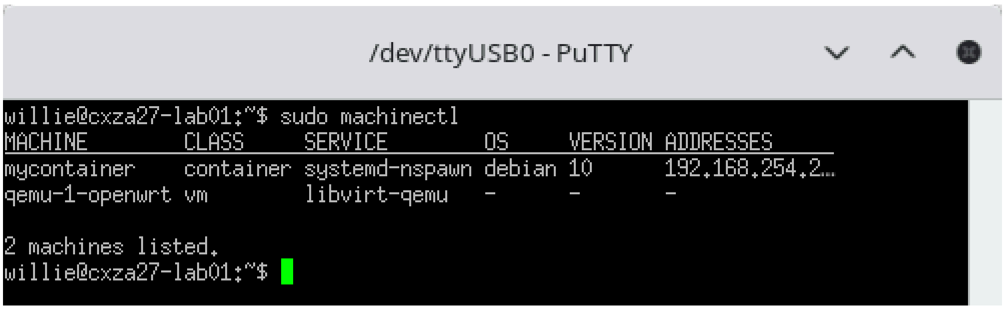
```
  GNU nano 3.2              /etc/systemd/nspawn/mycontainer.nspawn

[Exec]
Boot=yes
[Network]
VirtualEthernet=yes
Bridge=br0
# Create an extra network
VirtualEthernetExtra=mycontainer
```

Enable the container to run on startup.



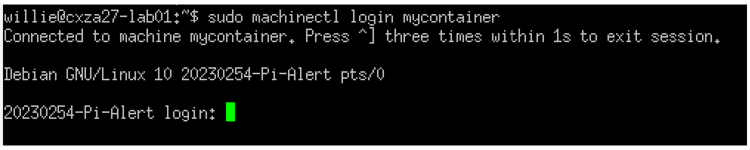
```
willie@cxza27-lab01:~$ sudo systemctl enable systemd-nspawn@mycontainer
willie@cxza27-lab01:~$ sudo systemctl enable machines.target
willie@cxza27-lab01:~$ sudo systemctl start machines.target
```

After rebooting the edge-node the container has started on boot along with the openwrt machine from a previous task.



```
willie@cxza27-lab01:~$ sudo machinectl
MACHINE        CLASS     SERVICE        OS     VERSION ADDRESSES
mycontainer    container systemd-nspawn debian 10      192.168.254.2…
qemu-1-openwrt vm        libvirt-qemu   -      -       -

2 machines listed.
willie@cxza27-lab01:~$
```

I am able to reach the login prompt and login with the credentials I created in the previous steps.

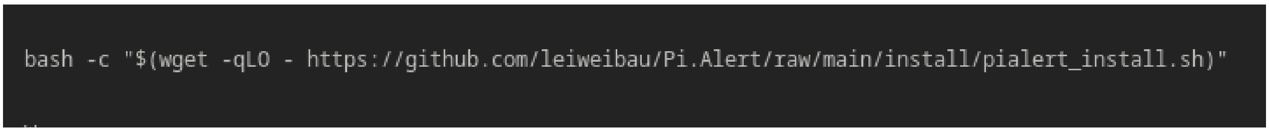


```
willie@cxza27-lab01:~$ sudo machinectl login mycontainer
Connected to machine mycontainer. Press ^] three times within 1s to exit session.

Debian GNU/Linux 10 20230254-Pi-Alert pts/0

20230254-Pi-Alert login:
```

Pi.Alert Installation

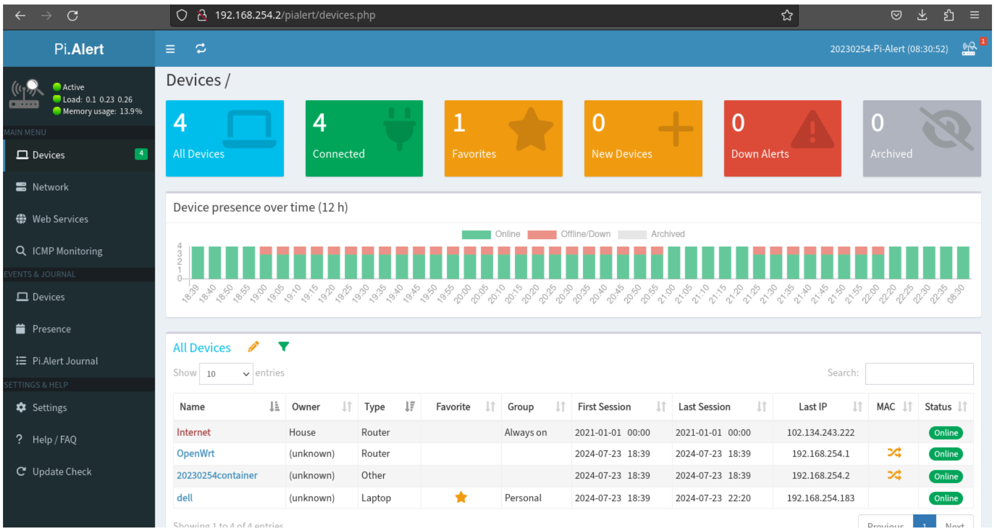
Automated Installation Script

For the installation of Pi.Alert I made use of the automated installation script that can be found here.



```
bash -c "$(wget -qLO - https://github.com/leiweibau/Pi.Alert/raw/main/install/pialert_install.sh)"
```

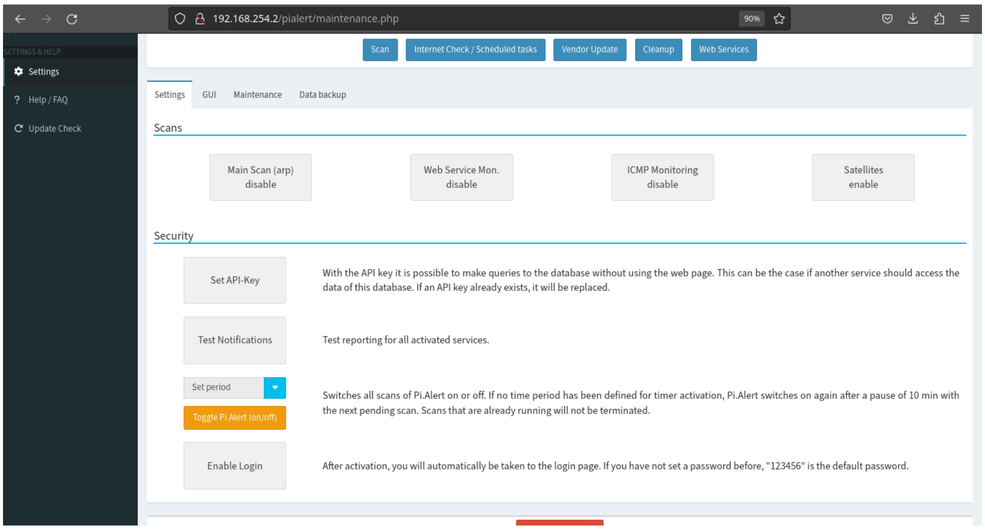
After installation I am greeted with this dashboard when I visit http://192.168.254.2



Name	Owner	Type	Favorite	Group	First Session	Last Session	Last IP	MAC	Status
Internet	House	Router		Always on	2021-01-01 00:00	2021-01-01 00:00	102.134.243.222		Online
OpenWrt	(unknown)	Router			2024-07-23 18:39	2024-07-23 18:39	192.168.254.1	🔀	Online
20230254container	(unknown)	Other			2024-07-23 18:39	2024-07-23 18:39	192.168.254.2	🔀	Online
dell	(unknown)	Laptop	★	Personal	2024-07-23 18:39	2024-07-23 22:20	192.168.254.183		Online

Configuring login Credentials

To configure login I first had to navigate to the settings blade and the settings tab. Under the security section I located the button to switch to a mode that requires login via password. After that I set a new login password.

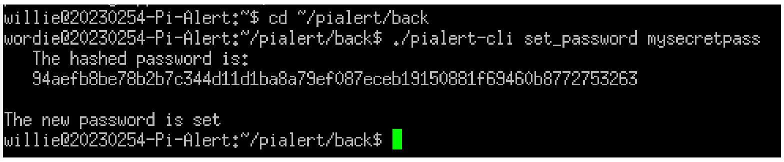


With the API key it is possible to make queries to the database without using the web page. This can be the case if another service should access the data of this database. If an API key already exists, it will be replaced.

Test reporting for all activated services.

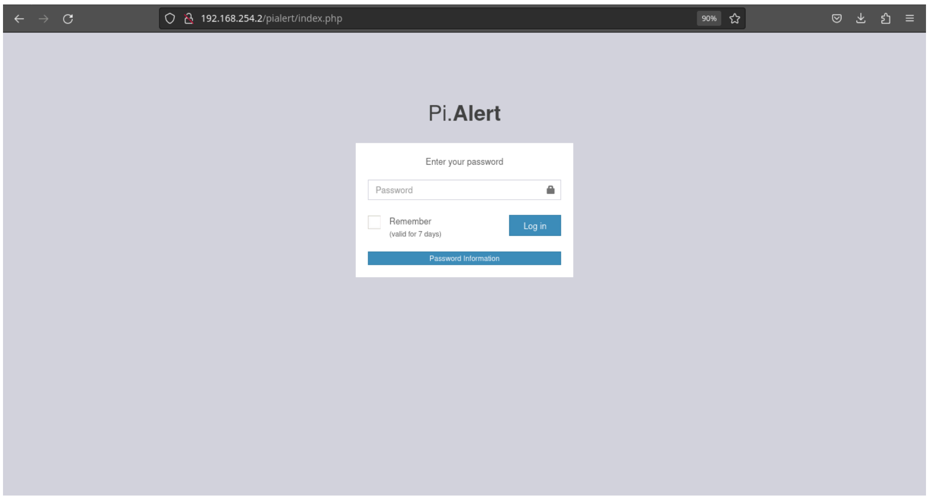
Switches all scans of Pi.Alert on or off. If no time period has been defined for timer activation, Pi.Alert switches on again after a pause of 10 min with the next pending scan. Scans that are already running will not be terminated.

After activation, you will automatically be taken to the login page. If you have not set a password before, "123456" is the default password.



```
willie@20230254-Pi-Alert:~$ cd ~/pialert/back
wordie@20230254-Pi-Alert:~/pialert/back$ ./pialert-cli set_password mysecretpass
   The hashed password is:
   94aefb8be78b2b7c344d11d1ba8a79ef087eceb19150881f69460b8772753263

The new password is set
willie@20230254-Pi-Alert:~/pialert/back$ 
```

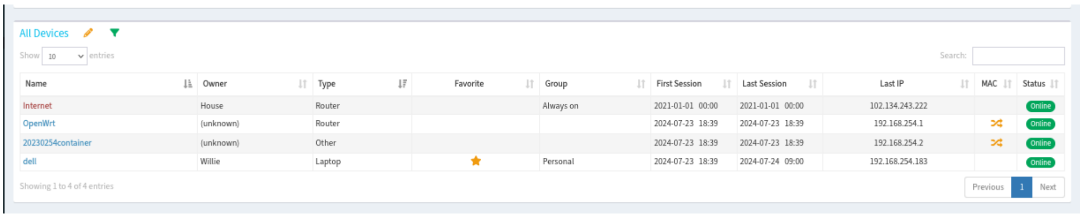
I am now greeted by a login screen when I visit http://192.168.254.2



Using the GUI to edit, backup, restore and maintain the database.

Editing

To edit values within the database I navigated to the All devices section accessible through the devices blade/tab. From there I clicked on the name for a device such as my personal computer.



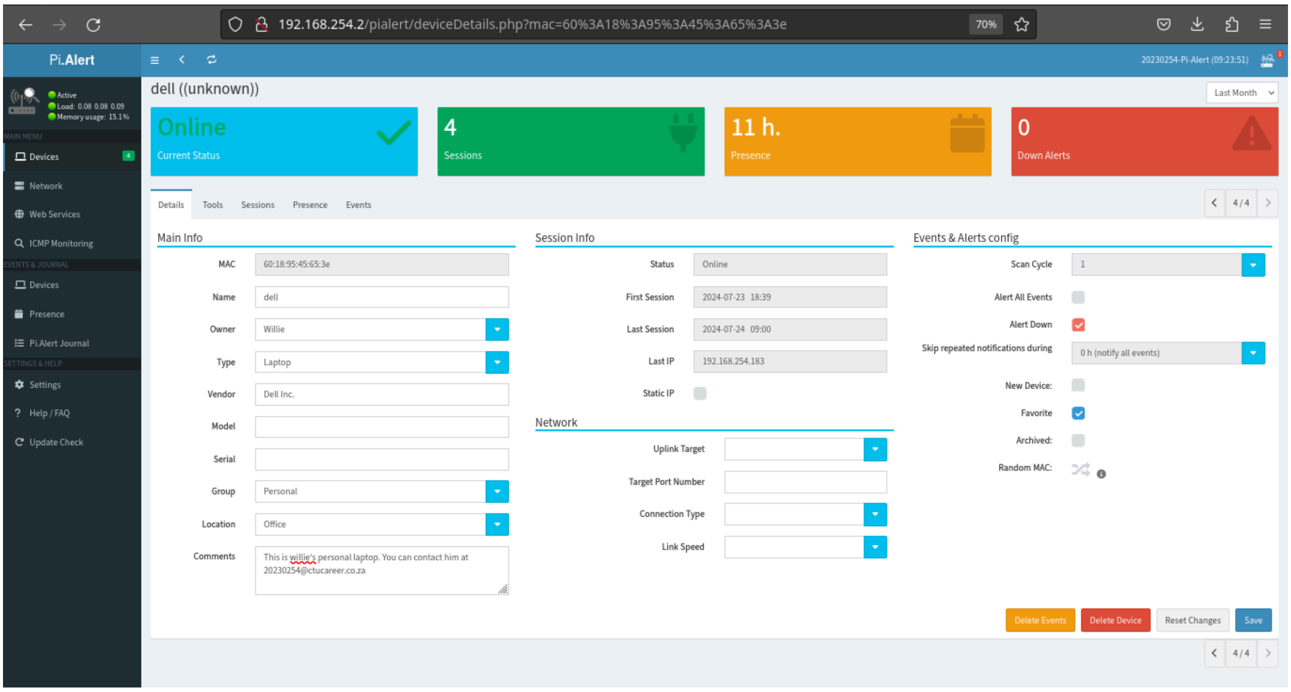
All Devices

Show 10 entries   Search:

Name	Owner	Type	Favorite	Group	First Session	Last Session	Last IP	MAC	Status
Internet	House	Router		Always on	2021-01-01 00:00	2021-01-01 00:00	102.134.243.222		Online
OpenWrt	(unknown)	Router			2024-07-23 18:39	2024-07-23 18:39	192.168.254.1	🔀	Online
20230254container	(unknown)	Other			2024-07-23 18:39	2024-07-23 18:39	192.168.254.2	🔀	Online
dell	Willie	Laptop	★	Personal	2024-07-23 18:39	2024-07-24 09:00	192.168.254.183		Online

Showing 1 to 4 of 4 entries

I then edited the database entry for my device, adding some details and specifying that alerts should be given when the device is down.



192.168.254.2/pialert/deviceDetails.php?mac=60%3A18%3A95%3A45%3A65%3A3e

dell ((unknown))

Online Current Status | 4 Sessions | 11 h. Presence | 0 Down Alerts

Main Info

MAC	60:18:95:45:65:3e
Name	dell
Owner	Willie
Type	Laptop
Vendor	Dell Inc.
Model	
Serial	
Group	Personal
Location	Office
Comments	This is willie's personal laptop. You can contact him at 20230254@ctucareer.co.za

Session Info

Status	Online
First Session	2024-07-23 18:39
Last Session	2024-07-24 09:00
Last IP	192.168.254.183
Static IP	□

Network

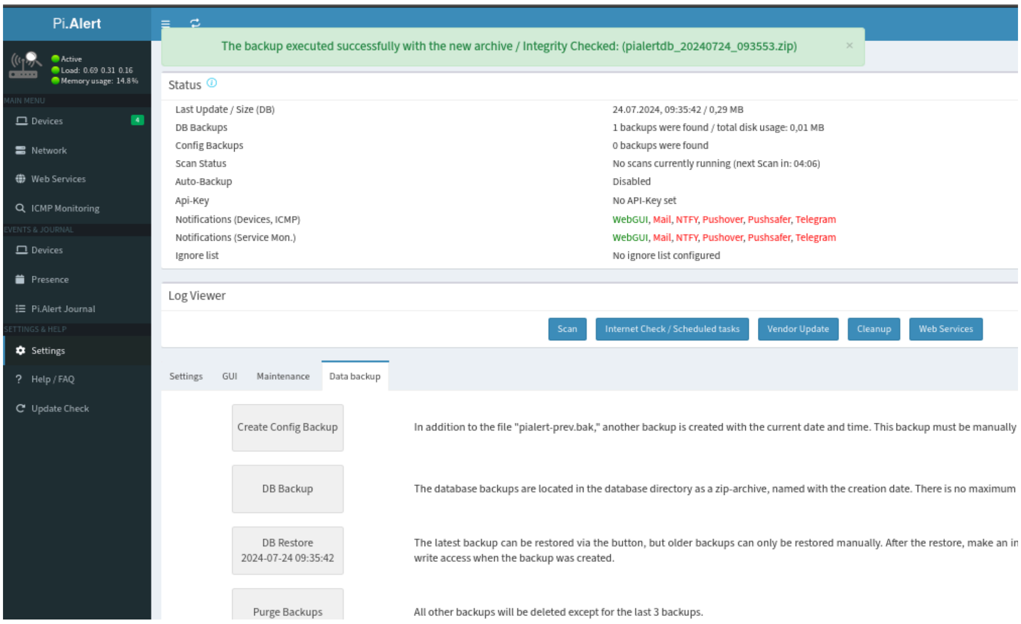
Uplink Target	
Target Port Number	
Connection Type	
Link Speed	

Events & Alerts config

Scan Cycle	1
Alert All Events	□
Alert Down	☑
Skip repeated notifications during	0 h (notify all events)
New Device:	□
Favorite	☑
Archived:	□
Random MAC:	🔀

Creating a database backup

Within the settings menu I find a tab for Data backup related settings. I then click create config backup.



The backup executed successfully with the new archive / Integrity Checked: (pialertdb_20240724_093553.zip)

Status

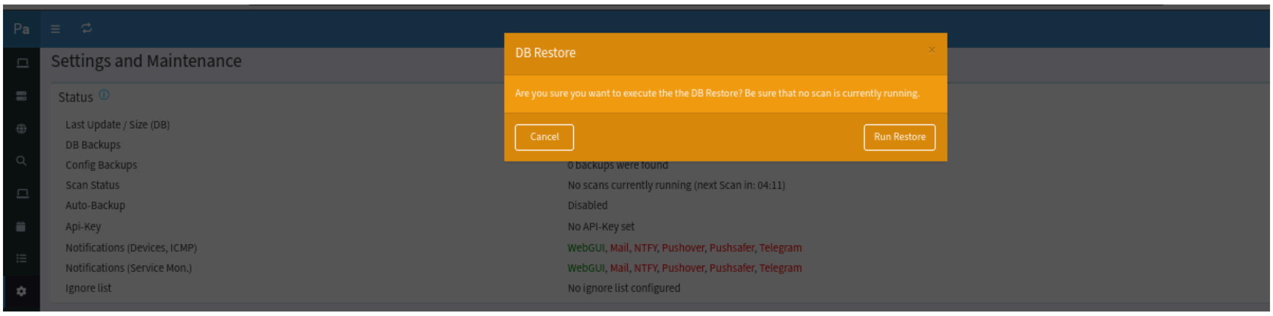
Last Update / Size (DB)	24.07.2024, 09:35:42 / 0,29 MB
DB Backups	1 backups were found / total disk usage: 0,01 MB
Config Backups	0 backups were found
Scan Status	No scans currently running (next Scan in: 04:06)
Auto-Backup	Disabled
Api-Key	No API-Key set
Notifications (Devices, ICMP)	WebGUI, Mail, NTFY, Pushover, Pushsafer, Telegram
Notifications (Service Mon.)	WebGUI, Mail, NTFY, Pushover, Pushsafer, Telegram
Ignore list	No ignore list configured

Log Viewer

Create Config Backup	In addition to the file "pialert-prev.bak," another backup is created with the current date and time. This backup must be manually
DB Backup	The database backups are located in the database directory as a zip-archive, named with the creation date. There is no maximum
DB Restore 2024-07-24 09:35:42	The latest backup can be restored via the button, but older backups can only be restored manually. After the restore, make an in write access when the backup was created.
Purge Backups	All other backups will be deleted except for the last 3 backups.

Restoring a database backup

Within the same menu there is a DB Restore button that can be used to restore the database. In my case the Database has been restored successfully.



Settings and Maintenance

DB Restore

Are you sure you want to execute the the DB Restore? Be sure that no scan is currently running.

Status

Last Update / Size (DB)	
DB Backups	
Config Backups	0 backups were found
Scan Status	No scans currently running (next Scan in: 04:11)
Auto-Backup	Disabled
Api-Key	No API-Key set
Notifications (Devices, ICMP)	WebGUI, Mail, NTFY, Pushover, Pushsafer, Telegram
Notifications (Service Mon.)	WebGUI, Mail, NTFY, Pushover, Pushsafer, Telegram
Ignore list	No ignore list configured

Conclusion

Pi.Alert is a powerful tool that can be employed in a production networking environment. It utilizes database technology to aid in achieving its goals, allowing users to create inventory of the devices on their networks.This includes creating, deleting and backing up inventory.

Bibliography

B, R. (2024, July 7). 🚢*Deploying Containers on Fusion's Edge using Nspawn*🐳. The Hub & Spoke | SD-WAN Blog. https://hubandspoke.amastelek.com/deploying-containers-on-fusions-edge-using-nspwan

leiweibau. (2024, July 24). *leiweibau/Pi.Alert*. GitHub. https://github.com/leiweibau/Pi.Alert