APZU Network Management

Technical Description

### **Introduction**

The internal network(s) of APZU are classified by

* a slow, asymmetrical sat link (eLeFaNt network)
* a big local Wi-Fi coverage (multiple hops, high internal latency)
* a big user group (>75)
* an intentional 'fair use' policy and (almost) free for all access
* VIP users that \*need\* to have connectivity

### **Components**

The APZU Network management system is mainly based on the open source router and firewall distribution pfSense. pfSense already provides numerous components and was extended by APZU to provide the basics for the APZU networks. pfSense relies on FreeBSD as a foundation and was initially forked from the firewall m0n0wall.

However it lacks some features required by the typical network environments of APZU and also needs to be configured in the right way to fulfill the needs. All this is done as part of the APZU Network Management system. In a nutshell these main components are utilized:

* Inner and Outer firewall
* Captive Portal
* RADIUS server
* DHCP
* Local DNS
* Transparent Proxy
* Traffic Management
* Custom pages and configs for APZU
* Low-cost network devices

### **Self-sign up**

Network access to the APZU network is free for all. However systems need to be registered once before they can access the network. For every new system (read not been connected to the network before) the user needs to his name and email address. Based on this and some additional technical data elements a user account is created. Every user account can be configured with individual access permissions and speed levels.

Complete list of data elements for user accounts:

* Name (provided by user)
* Email (provided by user)
* Ownership of the system (MoH/government, APZU, Private) (provided by user)
* MAC address (automatically determined)
* Initial IP (automatically determined)
* DNS name if available (automatically determined)
* Netbios name if available (automatically determined)
* Date of registration (automatically determined)

For a detailed technical description of the self-sign up process please refer to the appendix.

### **User management**

*Technical implementation*

*Groups of users*

*Restricting access*

*Up- and Downgrading of users*

### **Traffic limits & shaping**

*Technical implementation*

*Special purpose shaping*

Like Video calls, high priority sites

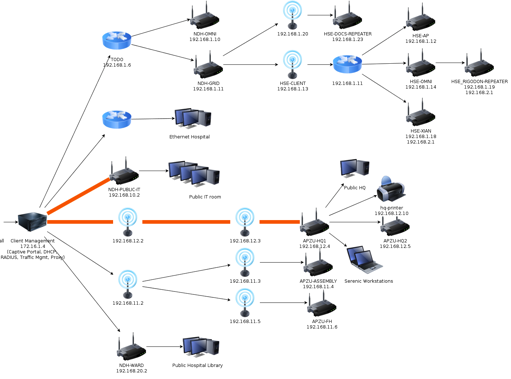
### **Network topology**

In a nutshell the network is divided into a server segment and multiple segments (both wired and Wi-Fi) where client can connect. The pfSense system has (for now dedicated physical) links to every segment and thus creating a star topology.

MAC addresses for the connected computers need to be available and visible at the network interfaces of pfSense. Therefore the topology needs to preserve the MAC addresses from the end point where clients are connected to up to the physical network interface of pfSense. For wired switched environments this is simple, but in traditional Wi-Fi networks the MAC address of the access point will be used at the outer interface. Here access points and wireless backbone links needs to support WDS (Wireless Distribution System) to enable the visibility of the MAC addresses through all network segments.

In Neno the core backbone links are mostly wireless as well. All of them are ‘terminated’ by either a simple Wi-Fi access point or link another more complex network segment. Under no circumstances a client system is allowed to connect directly to a backbone link.

The following diagram illustrates the layout of the segments clients can freely connect to.



(Note: The first network segment will be re-arranged.)

VLANs could be deployed to reduce physical network infrastructure and to some extend increase security, but for the purpose of APZU and with the chosen layout this isn’t necessary as it also increases complexity; especially for Wi-Fi networks.

Fur further information about optimizing Wi-Fi networks (both client and backbone links), check the Wi-Fi performance FAQ.

### **Traffic accounting**

RADIUS accounting

### **Realtime observations**

nTop

RRD graphs

Traffic graphs

2-hours Squid reports

DHCP leases

Captive portal sessions

### **Monitoring**

As a slightly complex, but flexible monitoring solution Zabbix is used. For the internal network infrastructure every segment with its network devices is monitored. As the subnets behind the pfSense are divided into independent IP subnets, a simple ping to check the availability from outside of the pfSense is not possible. Either a zabbix-proxy needs to be installed on the pfSense or a simple port forward to the HTTP management interface of every inner network device needs to be configured. The latter one is also helpful to have remote (outside) access to the various (inner) network devices for configuration purposes. A port-forward and a matching firewall rule to grant access also allows Zabbix to perform HTTP check about the availability. This way Zabbix can monitor downtimes of individual devices as well as downtimes of the whole subnet (if the last devices, e.g. end-user access point is monitored). Notifications (mail, SMS) can be sent out if needed.

The availability of the Internet access (satellite link) is best monitored from the outside. Various free services like pingdom.com, mon.itor.us, with notifications are available.

### **Physical layout, hardware, and power**

pfSense runs on the typical server hardware designed and customized by Baobab Health. With this all investments and solution around stability and power can be reused (mainly network design and backup power). This standard server hardware for Malawi is extended by PCI network cards and USB-to-Ethernet adapters to increase the number of physical networks interfaces.

For wireless backbone links network devices from Ubiquiti have proven to be reliable and cost effective. Mikrotik (Routerboards) systems can be used as well. They are extremely flexible both for hardware deployment and software configuration, but due to this are also more complex to configure.

End points of the network where clients can connect to are mostly simple consumer-level Access Points on which an alternate firmware (dd-wrt) is installed. dd-wrt provides more advanced features (like repeating Wi-Fi signals) and most important unifies the configuration across different types of Access Points (in Neno various low-end Access Points from Linksys, Netgear, and ASUS are in use). Devices from Meraki could not successfully be integrated and attempts to install dd-wrt on them failed.

In places where it is hard to run a dedicated power line (outdoor access points) or where the devices need to be connected to the backup power, Power-over-Ethernet is used. With simple, cheap, passive and generic PoE injectors/splitters, almost any network devices that operates on DC can be connected.

### **Limitations**

*Plain HTTP request required for Captive Portal Session initialization*

The Captive Portal requires one initial HTTP request (no HTTPS, no SMTP, …) to establish an active session. This happens transparently in the background (the user doesn’t notice anything), but if this HTTP request doesn’t take lace, access to the network in granted. If an old session is expired and e.g. only https://gmail.com is opened in the browser, than no authentication can take place. For users with more privileges the session timeout can be increased to avoid these frequent required HTTP requests. However an increased timeout can have an impact on the timeliness and accuracy of the accounting.

*Accounting doesn’t distinguish between local and Internet traffic*

Due to the physical network setup all traffic (both for the local network and Internet traffic go through the same interfaces. For pfSense it is not possible to detect different destinations and therefore simply count all traffic. E.g. copying something from the fileserver or using the local EMR installation will be counted as well and this limiting the use of the accounting.

*MAC Spoofing*

Access to the network is mainly granted based on the MAC address. As MAC addresses can be altered or spoofed, this access control can be considered as weak. Additional host finger printing of the systems might overcome this issue, but as APZU implements a free-for-all policy without a payment component, this risk is low.

*Advanced tunneling techniques*

The content filter and to some extend the shaping component) might be bypassed by advanced encrypted tunneling techniques (not further explained here). This requires a deeper understanding of the APZU network layout and required tools; low risk.

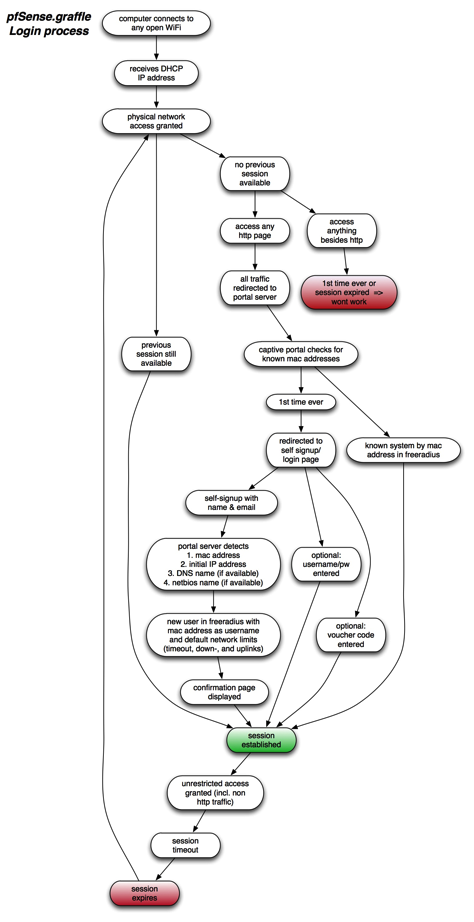
*Internal admin attacks*

Both software and hardware is only secured by passwords. Someone with access to these passwords can hide activities or delete traces and therefore bypass every filter, shaping, and logging mechanisms. Also the network infrastructure is not secured against extending the network. E.g. a malicious network access point can be added without further authentication. Such a devices could traffic and user data.

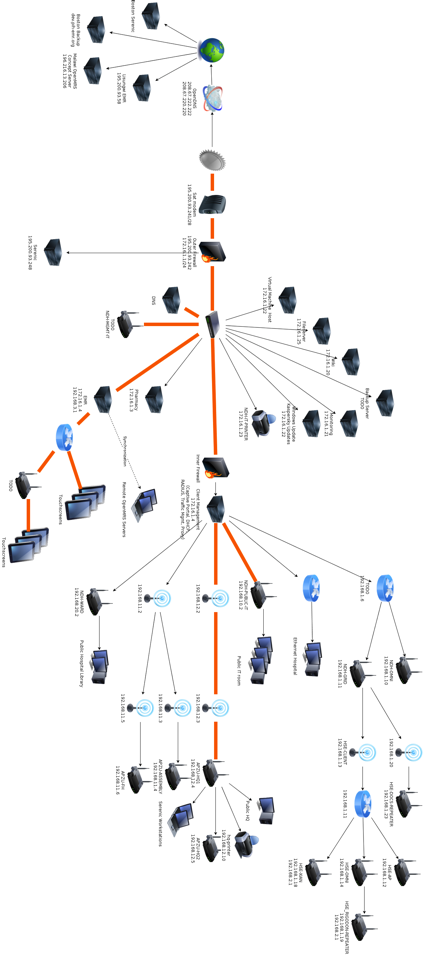
*(HTTP) session hijacking*

Currently all Access Points are open without encryption on the physical layer (no passwords required). While this eases the initial access and simplifies administration, it also opens the network for session hijackings attacks as everyone with the right tool might be able sniff data from other users.

### **Appendix**



*Self signup / login process*

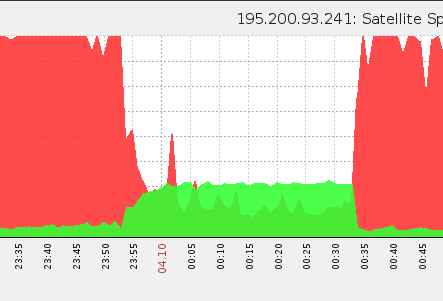
*Neno Network topology (Oct 2011)*

*Determining bandwidth limits*

*Neno Upper Bandwidth limits (Oct 2011)*

*Correlation of Up- and Downlink with Ariave links*

Traditionally the satellite links in Malawi experience a major drop of the maximum download rate whenever the uplink is peaking. A single system with a massive upload can therefore impact the whole network. In the picture below the downlink is shown in red; the uplink in green.



*pfSense Troubleshooting*

The FreeBSD subsystem is fully available for advanced configuration and especially troubleshooting.

web access

http://<lan ip>:8000 - portal page

https://<ip> - web admin

http://<ip>:3000 - ntop

Important dirs

/usr/local/bin/...

/usr/local/www

/var/log/

/usr/local/share/doc/freeradius

Log files

/var/log/radauth.log

/var/log/system.log

/var/log/apzu-portal.log

Debugging FREEradius

radiusd –X

radtest 002500486010 002500486010 192.168.10.1 0 pfSense

*pfSense stumbled upon additional resources*

Custom portal page example: <http://www.nwlab.net/tutorials/pfSense/Captive-Portal-Hotspot.html>

<http://blog.stefcho.eu/?tag=pfsense>

*Voucher roll example*

Voucher could be used to grant time-restricted access to the network. In Neno this feature is available, but not used a of now as the logistics around distributing the vouchers can be quite complex.

Nevertheless a voucher code like below could be used to gain access to the network.

# Voucher Tickets 1..10 for Roll 1

# Nr of Roll Bits 16

# Nr of Ticket Bits 10

# Nr of Checksum Bits 5

# magic initializer 681760529 (32 Bits used)

# Character Set used 2345678abcdefhijkmnpqrstuvwxyzABCDEFGHJKLMNPQRSTUVWXYZ

#

nVE3Qiu2L4k

2Y3y37GERsv

yN6Gn8EE8nR

in8KkWEzWic

Uc8vLnJCiKb

4b3etDPGRWJ

2sY62bPq46n

6Pq8iRQ4TsE

kdt7AqL2jCr

ATRi5m54TwY