Building a SD-WAN appliance suitable for an Australian Health Sector NFP/NGO

Jason Tubnor

ICT Manager

Latrobe
Community
Health Service

Introduction

- About Me
- Latrobe Community Health Service (LCHS)
- History of our SD-WAN journey
- Design Choices
- OpenBSD VPN and routing technologies
- Using Ansible for orchestration, deployment and management

About Me

- 31 years of IT experience
- Introduced to Open Source in the mid 90’s
- Discovered OpenBSD in 2000
- A user and advocate of OpenBSD and FreeBSD
- BSDNow Co-host
- Life outside of computers:
  - Ultra endurance bikepacking

Race from the Rocks – Sep 2022
Sydney Opera House

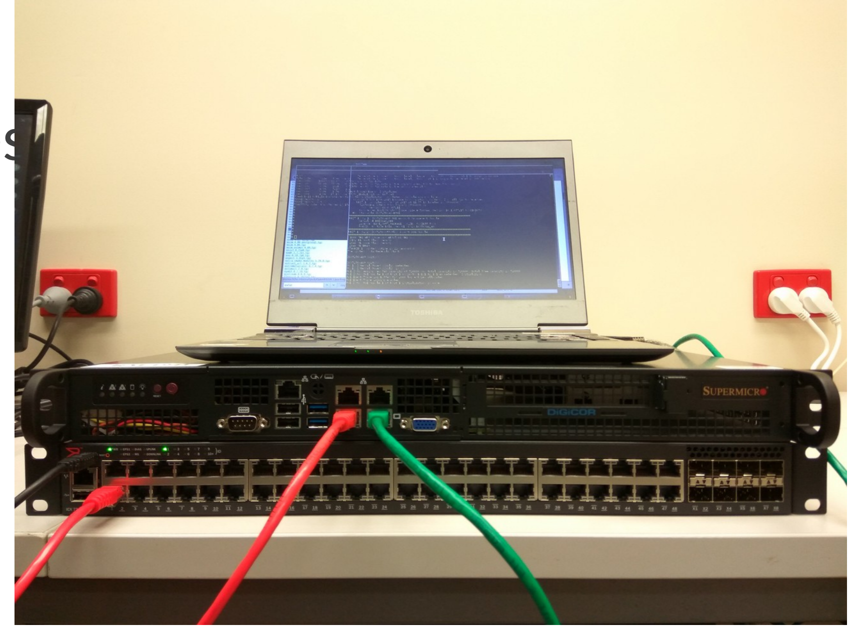
Latrobe Community Health Service (LCHS)

- Originally a Gippsland based NFP/NGO health service
- ICT manages 1500+ users
- Servicing 40 sites across Victoria, Australia (3 offices in Sydney)
- Covering ~230,000km^2
  - Roughly the size of Romania Europe, Laos Asia or Minnesota USA
- “Better health, Better lifestyles, Stronger communities”

Latrobe Community Health Service

History of our SD-WAN Journey

- We came up with a cost effective ‘Hub and Spoke’ des
- Used OpenBSD within a bhyve host (11.1 thru 14.0)
- Supermicro SuperServer 5019A-FTN4 and 5019S-ML
- OpenBSD technology such as:
  - PF, OpenIKED, RIPd and VXLAN
  - dhclient, dhcrelay
  - UDP syslog
- Grew to 25 sites



History of our SD-WAN Journey cont.

- Hub concentrated 11 machines (spokes) into a single connection
- There were 2 hubs nested within a FreeBSD/bhyve hypervisor

Latrobe Community Health Service

History of our SD-WAN Journey cont.

- Pros
  - Extremely stable and efficient
  - Low cost
  - Scaled exceptionally well
  - Network traffic path was easy to determine faults
  - Could run on cheap retail NBN TC-4 connections
  - Had the ability to secure traffic and control data quality

History of our SD-WAN Journey cont.

- Cons
  - Issues with SuperMicro support
    - Off-shore servicing (Taiwan)
  - Managing fleet was overwhelming
    - Hypervisor maintenance and upgrades
    - OpenBSD guest updates to track release
    - PF rule maintenance, especially block lists
  - IKEv2 configuration wasn’t resilient
  - Route table interruption causing havoc on UDP services

Evolution Design Considerations

- IKEv2 required a more robust configuration using lo1
- Reduced pseudo (VXLAN) interface complexity
- Move to dhcpleased
- Reassess the use of dhcrelay
- Change out UDP processes to use TCP where available
- Modernise the routing stack (Peter Hessler)

Evolution Design Considerations cont.

- Commodity hardware
  - Serial port, quad port ethernet
- Device should be ephemeral, disposable
- Automation
  - Build, upgrades and maintenance
  - No logging into devices
  - Zero touch installs
- Terminating spokes into OpenBSD hubs on VMWare vSAN

Enter sec(4)

- Written by David Gwynne at the University of Queensland (dlg@)
- A pseudo interface added to OpenBSD
  - *The sec driver provides point-to-point tunnel interfaces for IPv4 and IPv6 protected by the ipsec(4) Encapsulating Security Payload (ESP) protocol.*
- Tightly integrated into OpenIKED

Enter sec(4) cont.

- Fully featured interface:

  -
    ```
    /etc/hostname.sec0
    inet 192.168.4.0 255.255.255.254 192.168.4.1
    up
    ```

  -
    ```
    sec0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1280
            index 11 priority 0 llprio 3
            groups: sec
            inet 192.168.4.0 --> 192.168.4.1 netmask 0xfffffffe
    ```
- Simple OpenIKED configuration:

  -
    ```
    Server:
    ikev2 passive from em0.dc.example.com to em0.site.example.com \
    srcid em0.dc.example.com iface sec0
    ```

  -
    ```
    Client:
    ikev2 active from em0.site.example.com to em0.dc.example.com \
    srcid em0.site.example.com iface sec0
    ```

Hardware

- Lenovo ThinkCentre M70s Gen 5 Small Form Factor
  - Core i5 CPU
  - PCIe 3.0 x16 low-profile (Intel Quad Port)
  - 256GB SSD
  - 16GB RAM
- Cheap and available in Australia
- Have additional units on the shelf
- We did investigate ARM and cheaper x86
  - International slavery laws

Hardware Issues

- Machines were shipped with Ubuntu 22.04 LTS and were booting from factory
- OpenBSD bootloader would come up
- Kernel feature would display but system would hang on ACPI initialisation
- This pointed to a firmware issue within the Lenovo device
- New firmware 3 months later fixed the issue
  - This firmware was pinned to be used across the upcoming fleet

Network Stack

- The use of an additional loopback (lo1) interface
  - Source for IKEv2
  - Router interface for OSPF loopback
  - Source interface for iBGP
- Team is familiar with this routing configuration, used within ISPs
- Removed overlays operating over IKEv2

Network Stack cont.

- IKEv2 using loopback as source allows:
  - NAT to (egress)
  - WAN can be swapped out on faults or maintenance
- dhcpleased easily detects changes in link state and refreshes assigned address
  - ISP maintenance
  - No need for custom validation and prime scripts

Services Stack

- Moved syslog output from UDP to TCP
  - This is required for our SIEM
  - Being TCP allows for log spooling to resume when route table appears again
- dhcrelay duties were moved into the downstream L3 switch
- A simplified PF that was more generic

Universal Configuration

- Planning to use automation framework to configure and manage devices
  - Simplifying the configuration means fewer moving parts
- OpenBSD has an ‘include’ feature
  - Generic *.conf files could be distributed and modified as needed
  - A singular /etc/conf.local configuration file was conceived to hold unique system configuration

Universal Configuration cont.

- /etc/conf.local
  0600 root:wheel

```
#Interfaces
vlan10_if="em0“
sec_if="sec10“
ext_if="em1“
egress_bw="100M“

# Host IP Addresses
terminator=“1.2.2.3“
terminator_name=“terminator01.vic.example.com“
lo1="10.8.0.13“
lo1_name="lo1.site01.188ffee00.example.com“
```

Universal Configuration cont.

- /etc/conf.local

```
#Services
sec_addr="10.1.2.25“
snmp_listenaddr=$sec_addr
snmp_contact="LCHS ICT Administrators (ournoc@lchs.com.au)“
snmp_description="NDIA Ballarat, Victoria“
snmpv3_user=“snmpuser“
snmpv3_authkey=“SecretAuth“
snmpv3_enckey=“SecretEnc“
ospf_id=$sec_addr
ospf_area="0.0.0.1“
bgp_asn="65001“
bgp_routerid=$sec_addr
bgp_mynetworks="172.16.1.8/30“
bgp_localneighbor="10.9.103.0"
```

Include Examples

- /etc/bgpd.conf

```
include "/etc/conf.local“

AS $bgp_asn
router-id $bgp_routerid
nexthop qualify via bgp

prefix-set mynetworks {
    $bgp_mynetworks
}

network prefix-set mynetworks set large-community $bgp_asn:1:1

group "ibgprr" {
    remote-as $bgp_asn
    local-address $bgp_routerid
    neighbor 10.9.3.7
    neighbor 10.9.4.7
    neighbor $bgp_localneighbor {
        route-reflector
        }
    }
```

Include Examples

- /etc/ospfd.conf

```
include "/etc/conf.local“

router-id $ospf_id

area $ospf_area {
    interface $sec_if {
        type p2p
    }
interface $vlan10_if {
        type p2p
    }
}
```

Include Examples

- /etc/pf.conf

```
#Version: 2024082200
include "/etc/conf.local"
```

Automatic Builds - OpenBSD

- Configure an answer script for the build
  - Including the public key for the ansible user as a root authorized_key
  - Don’t worry, PF protects SSH and can only be connected to from a couple of Ips
- Insert OpenBSD minirootXX.img USB and boot
  - Press A

Ansible – Priming OpenBSD

- Install pre-requisites on fresh build:
  - ansible-playbook –i newhost installpkgs.yml

```
---
# Bootstrap OpenBSD pkgs required for new build
   - hosts: all
     gather_facts: false
     remote_user: root
     vars_files:
       - vars.yml

     tasks:
       - name: Install required packages to bootstrap machine
       raw: 'pkg_add -I python3 gtar--'
```

Ansible – Copy in Unique Settings

```
---
  - hosts: all
    gather_facts: no
    remote_user: root
    vars_files:
      - vars.yml

    tasks:
    - name: Copy unique conf.local to inventory
      ansible.builtin.copy:
       src: "unique/{{inventory_hostname}}/etc/conf.local"
       dest: "/etc/conf.local“
       mode: '0600’
       owner: root
       group: wheel
```

Ansible – Extract IKEv2 Public Keys

```
tasks:
 - name: Fetch local.pub from OpenIKED instances on remote hosts
   fetch:
    src: /etc/iked/local.pub
    dest: ./iked
```

Ansible – Modify System Files

```
tasks:
 - name: Force fsck to check disks on each reboot
   ansible.builtin.lineinfile:
     path: /etc/rc
     regexp: 'fsck -p'
     line: '  fsck -y "$@"'

 - name: Enable Banner (issue) in sshd_config
   ansible.builtin.lineinfile:
     path: /etc/ssh/sshd_config
     regexp: 'Banner'
     line: Banner /etc/issue
   notify: Reload service sshd
```

Ansible – Modify System Files cont.

```
- name: Enable remote log host for local2 info
  ansible.builtin.lineinfile:
    path: /etc/syslog.conf
    regexp: 'local2.info'
    line: local2.info                          @tcp://loghost.internal.example.com:601
  notify: Restart service syslogd

handlers:
- name: Restart service syslogd
  ansible.builtin.service:
     name: syslogd
     state: restarted

- name: Reload service sshd
  ansible.builtin.service:
     name: sshd
     state: reloaded
```

Ansible – Update PF rules

```
---
- hosts: all
  gather_facts: yes
  remote_user: root
  vars_files:
    - vars.yml

  tasks:
  - name: Copy pf.hardblock to inventory
    ansible.builtin.copy:
     src: "p5root/etc/pf.hardblock"
     dest: "/etc/pf.hardblock"
     mode: '0600'
     owner: root
     group: wheel
    notify: Reload the pf.conf file if valid
```

Ansible – Update PF rules cont.

```
- name: Copy pf.conf to inventory
  ansible.builtin.copy:
   src: "p5root/etc/pf.conf"
   dest: "/etc/pf.conf"
   mode: '0600'
   owner: root
   group: wheel
   validate: pfctl -nf %s
  notify: Reload the pf.conf file if valid

handlers:
- name: Reload the pf.conf file if valid
  ansible.builtin.command:
    cmd: pfctl -f /etc/pf.conf
```

Ansible – Custom OpenBSD patches

```
---
# LCHS Custom Patch Servers

- hosts: all
  gather_facts: yes
  remote_user: root
  vars_files:
    - vars.yml

  tasks:
    - name: Apply all custom LCHS patches
      ansible.builtin.unarchive:
        src: https://mirror.internal.example.com/pub/patches/openbsd-lchs-
{{ ansible_distribution_version }}.tar
        dest: /
        remote_src: yes
      notify:
        - Reorder kernel
        - Wait until kernel reorder
        - Reboot after applying patches
```

Ansible – Custom OpenBSD patches cont.

```
handlers:
   - name: Reorder kernel
     ansible.builtin.shell:
       "/usr/libexec/reorder_kernel && touch /tmp/_rebootnow"

   - name: Wait until kernel reorder
     ansible.builtin.wait_for:
       path: /tmp/_rebootnow

   - name: Reboot after applying patches
     ansible.builtin.reboot:
```

Ansible – OpenBSD syspatch

```
---
# Syspatch Servers

- hosts: all
  remote_user: root
  vars_files:
    - vars.yml
  tasks:
    - name: Apply all patches and store result
      community.general.syspatch:
      register: syspatch

    - name: Reboot if patch requires it
      ansible.builtin.reboot:
      when: syspatch.reboot_needed
```

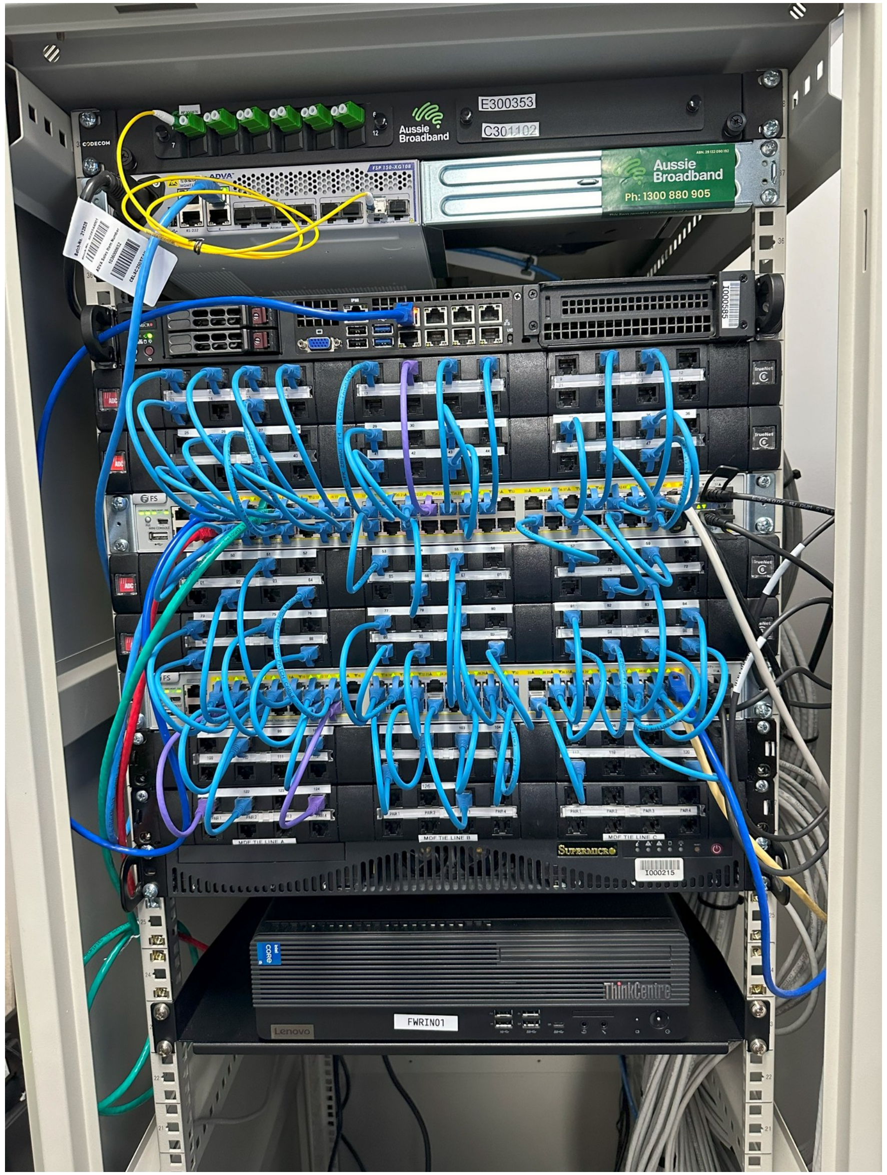
Ansible – OpenBSD sysupgrade

```
---
# Sysupgrade Servers

- hosts: all
  remote_user: root
  vars_files:
    - vars.yml

  tasks:
    - name: Sysupgrade host and store result
      community.general.sysupgrade:
      register: sysupgrade

    - name: Reboot system if needed
      ansible.builtin.reboot:
      when: sysupgrade.changed
```

Ansible – OpenBSD pkg upgrade

```
---
# Upgrade OpenBSD pkgs after sysupgrade

- hosts: all
  gather_facts: false
  remote_user: root
  vars_files:
    - vars.yml

  tasks:
    - name: Upgrade Ansible dependencies after sysupgrade
      raw: 'pkg_add -Uuv python3 gtar--'

    - name: Upgrade installed packages
      community.general.openbsd_pkg:
        name: '*'
        state: latest

    - name: Clean up orphaned packages
      raw: 'pkg_delete -a'
```

The End Product



Conclusion

- Building on past experiences, we were able to reiterate our device into an ephemeral appliance
- Leveraged newer and more supported OpenBSD technologies
- BSD continues to be a valuable asset to the organisation
- Indirectly, BSD has assisted in providing better services and outcomes for our clients and staff

A Special Thanks

- David Gwynne – OpenBSD
- OpenBSD Project
- ….. and all those that work tirelessly on open-source software

Donate

- You too can help:
  - OpenBSD Foundation http://www.openbsdfoundation.org/

Thank You

- Jason Tubnor
  - Email: jason.tubnor@lchs.com.au
  - Email: jason@tubnor.net
  - Mastodon: Tubsta@soc.feditime.com

Q & A