Providing a familiar interface for network engineers to harness (some of the) power of OpenBSD
Intro

• I’m a net admin not a developer.
• I am a maintainer for the NSH port/package for OpenBSD.
• I assist the author and founder of the NSH project Chris Cappuccio.
• I don’t speak for the OpenBSD project, I’m a user.
• Trampling over the shoulders of giants
  • Chris Cappuccio, Stefan Sperling, Stuart Henderson, Kurt Moisjeuk Henning Brauer, Peter Hessler, Claudio Jecker, Job Snijders, Sebastian Benoit, mpi, Alexander Bluhm, Theo de Raadt, Chris Cappuccio.
  • + many many more people
• Their Time and generosity has made my working life so much better
• Allows us continue bringing high quality high-speed internet to rural parts of Ireland, and quality network services to businesses all over Ireland.
NSH Team

• Chris Cappuccio – Founder, Author -has remarkable combination of Programming ability and Networking knowledge.
• Stefan Sperling - programming legend and has made many optimisations, and feature additions to nsh
• Alan McGrath & Scott McDonnell (2022 contribution)  Software engineers from my alma mater Dublin City University.
• Tom Smyth – port maintainer of nsh for OpenBSD – Documentation and Wishlist curation 😊
• You ? do you want to help us ?
Why OpenBSD

• Written by people with an attention to detail
  • Design
  • Implementation
  • Documentation (Thanks Ingo )

• Syntax of pf, relayd, bgpd and network interface configuration is brief but profound.
Why NSH???

• NSH is a CLI intended for management of OpenBSD-based network appliances.
• It replaces
  • ifconfig, sysctl and route with its own simple command language, and encapsulates configuration for other daemons into one place, effectively replacing /etc/netstart and parts of /etc/rc for appliance-style usage.
How does one configure other network Devices?

• Lets take a look ...
Lets take a Simple Example

- Create a vlan 101 on a switch
- make one of the ports of the switch a layer 2 tagged port allowing passing vlan 101 tagged traffic on that port
- Disable Spanning tree and any other discovery protocols etc.
Standardised configuration approach - Brocade VDX 6720-24

ntp server 5.134.89.123
ntp server 5.134.90.123
chassis-name VDX6720-24
  host-name DFZ-10G-SW2-40B
interface Vlan 101
  description eir-wc-Roscrea

interface TenGigabitEthernet 1/0/10
  mtu 9216
  no fabric isl enable
  no fabric trunk enable
  switchport
  switchport mode trunk
  switchport trunk allowed vlan add 101
  switchport trunk tag native-vlan
  bpdu-drop enable
  spanning-tree shutdown
  lldp disable
  no shutdown
Standardised configuration approach- Arista 7050-QX32

hostname 40G-SW2-U34B
!
ntp server 5.134.89.123 prefer
ntp server 5.134.90.123
Vlan 101
description eir-wc-Roscrea
!
interface Ethernet9/2
  description Core1G-SW2-52
  mtu 9214
  switchport trunk allowed vlan 101
  switchport mode trunk
  spanning-tree bpdufilter enable
  no lldp transmit
  no lldp receive
Standardised configuration approach- Cisco 2960s-24-dt-1

hostname DFZ-1G-SW1-U37
ntp peer 5.134.90.123
ntp peer 5.134.89.123 prefer
!
vlan 101
description Interface GigabitEthernet1/0/6
    switchport trunk allowed vlan 101
    switchport mode trunk
    no keepalive
    no cdp enable
    no lldp transmit
    no lldp receive
    spanning-tree portfast
    spanning-tree bpduguard enable
hostname "MGMT-1G-SW4-U34"

vlan 101
name "internet101"
system-time ntp UTC 5.134.89.123 5.134.90.123 1
system-time dst predefined Europe
interface gigabitEthernet 1/0/41
  description "wc-bgp1-eth5"
  jumbo
  switchport mode trunk
  switchport trunk allowed vlan 101
  spanning-tree bpdufilter
  no lldp receive
  no lldp transmit
Standardised configuration approach- Ubnt- Edge switch

hostname "DUFIFI1-SW01"
no snntp server "1.ubnt.pool.ntp.org"
no snntp server "2.ubnt.pool.ntp.org"
sntp server "5.134.89.123"
sntp server "5.134.90.123"!
vlan database
101,192-207,406
vlan name 101 "Internet101"

interface 0/6
description 'P2P-Link'
spanning-tree bpdubilter
no spanning-tree port mode
mtu 9216vlan
pvid 406
vlan participation exclude 192-207
vlan participation include 406,101
Vlan tagging 101,406
switchport protected 0
poe opmode shutdown
no flowcontrol
Exit
Did anyone notice a pattern (ish) Emerging?
Standard OpenBSD configuration tools to implement previous config

• mg /etc/myname

  #Networkdevicename.domainsuffix
  Vlan-agg-node-01.wirelessconnect.ie

• mg /etc/hostname.ix0

  mtu 9216
  up

• mg /etc/hostname.ix1

  mtu 9216
  up
Standard OpenBSD configuration tools to implement previous

- mg /etc/hostname.vlan101
  
  ```
  parent ix1
  vnetid 101
  mtu 9216
  up
  ```

- mg /etc/hostname.bridge101
  
  ```
  maxaddr 16384 timeout 300
  up
  add ix0 -stp ix0
  add vlan101 protected vlan101 1 -stp
  ```
Standard OpenBSD configuration tools to implement previous

- `mg /etc/ntpd.conf`
  ```
  server 5.134.89.123
  server 5.134.90.123
  ```
- Restart services to apply changes
  ```
  rcct1 restart ntpd
  ```
- Restart the network to apply changes
  ```
  sh /etc/netstart
  ```
Interactive or rc.local script

ifconfig ix0 mtu 9216 up
ifconfig ix1 mtu 9216 up
ifconfig create vlan101 vnetid 101 mtu 9216 parent ix1 up
ifconfig create bridge101 up maxaddr 16384 timeout 300
ifconfig bridge101 add vlan101 -stp vlan101 protected vlan101 1
ifconfig bridge101 add ix0 -stp ix0
echo server 5.134.89.123 >/etc/ntpd.conf
echo server 5.134.90.123 >>/etc/ntpd.conf
rcctl restart ntpd.conf
echo vlan-agg-node-01.wirelessconnect.ie>/etc/myname
Why nsh?

- Interface names can confuse the non OpenBSD User
- ifconfig is awesome but....
- Persistent ip configuration
  - /etc/hostname.vio*
  - /etc/hostname.em*
  - /etc/hostname.ix*
  - /etc/hostname.ixl*
- sh /etc/netstart [vio1]
- What is a sysctl why is it not routing / forwarding?
why nsh?

• put the configuration of a network appliance in one easy to find location viewable with the command*

  show running-configuration
nsh configuration

hostname Vlan-agg-node-01.wirelessconnect.ie
interface ix0
  mtu 9216
  no shutdown

interface ix1
  mtu 9216
  no shutdown

interface vlan101
  vnetid 101
  parent ix1
  group vlan
  mtu 9216
  no shutdown

bridge bridge101
  group bridge
  maxaddr 16384
  timeout 300
  member vlan101 em0
  protect em0 1
  protect vlan101 1
  no shutdown

ntp rules
  server 5.134.89.123
  server 5.134.90.123
!
ntp enable
New to nsh? – use it to read running config rather than write with it

• use your favourite OpenBSD Standard tools to write settings to the kernel and network configuration
• Run nsh as a viewer of the system configuration you have made which in turn *should* show you how to implement the same change in nsh
• We welcome gap analysis and feedback on this
Recent nsh improvements

• Command line completion vastly improved
• more permissive forgiving command input (strict config reading and writing)

```
nsh(config-p)/
nsh(config-p)/interface vlan100
nsh(interface-vlan100)/exit
nsh(config-p)/interface vlan 100
% Interface name is vlan100 not "vlan 100"
nsh(interface-vlan100)/
```

• Improved support for pppoe(4), dhcpleased(8) and resolvd(8) features
Recent nsh improvements

- command line
- show active interface configuration

```
nsh(interface-em0)/description main uplink
nsh(interface-em0)/mtu 900
nsh(interface-em0)/mtu 9000
nsh(interface-em0)/description main uplink
nsh(interface-em0)/show active-config
interface em0
  description main uplink
group egress
  mtu 9000
autoconf4
```
improved command help

• command ? gives brief command help

```
nsh(config-p)/pf ?
% Arguments may be abbreviated
   enable       enable pf firewall
   disable      disable pf firewall
   edit         edit, test and stage firewall rules
   check-config test and display staged firewall rules
   reload       test and apply staged firewall rules
nsh(config-p)/
```

• command <Tab><Tab> displays a horizontal list of options

```
nsh(config-p)/pf
check-config disable edit enable reload
```
nsh double tab context help

```
nsh(interface-em0)/ip
ip <address>/<prefixlen>    # IP address parameter
ip <address>/<netmask>      # IP address parameter
ip autoconf                 # Use automatic configuration
ip dhcp                     # Use DHCP
```

wirelessconnect
nsh man integration

nsh(bridge-bridge101)/manual bridge

- improved command help with manual [feature]
check-config command

• Check Config user can check staged configuration

```
$ nsh(config-p)/pf check-config
Loaded 714 passive OS fingerprints
set skip on { lo }
/var/run/pf.conf.0:8: syntax error
nsh(config-p)/
```
added config mode

- unprivileged mode
  - when you start nsh initially

- enable privileged mode
  - (read config including sensitive config) but config cannot be modified (safety)

- privileged config mode
  - (write config)
Improvements

• Implemented support for all OpenBSD Interfaces
  • umb (4G /5G radio interface)
  • wireguard
Improvements

- new integration.sh script to
- scripts/shell/extensive-nsh-openbsd-integration.sh
  - backup original /etc/ daemon and network configurations
  - import *Basic* configuration to nsh and save it as /etc/nshrc (startup configuration)
  - (on a newly installed OpenBSD box (to allow for nsh full control)
  - allow a user to select their preferred editor for complex daemon configuration editing
Improvements Roadmap

- Usability, Usability, Usability
- read only diagnostics
- documentation
- Idempotent configuration handling for automation
- ------------------------Proof of Concept complete------------------------
- Solicit funding for work privilege separated client vs server architecture (user experience shouldn’t change)
  - Rest config
  - Web interface
nsh – install

• simples 😊
• pkg_add nsh
• choose either static or dynamically linked flavour 😊
• in this state
  • you can use nsh to read the current kernel networking configuration* and make non persistent
making nsh and openBSD more get-at-able 😊

• The word is accessible Tom ... accessible !!!

• to use nsh as the primary configuration interface one has to do the following
  • backup existing configuration
  • copy relevant config files you want to import to a location where nsh will import them

```
KSH# cp /etc/bgpd.conf /var/run/bgpd.conf.0
KSH# mv /etc/bgpd.conf /var/nsh/backup
nsh(config-p)# bgp enable
```
ways to run nsh

• interactively nsh
  • #nsh

• update config
  • #nsh -c /home/config-script-to-update-config

• Initialise config (startup config)
  • #nsh -i /etc/nshrc
Roadmap

1. Usability improvements
   https://github.com/users/yellowman/projects/1/views/2
2. Read only diagnostics
   https://github.com/users/yellowman/projects/1/views/3
3. Documentation
   https://github.com/users/yellowman/projects/1/views/6
4. Security & Architecture
   https://github.com/users/yellowman/projects/1/views/4
5. Config-Functionality
   https://github.com/users/yellowman/projects/1/views/1
BSDCAN 2024 – First nsh tutorial

1. Daylong course, with labs with 2 virtual OpenBSD /NSH boxes for each student.
To Do High Priority

• get config syntax dialled in.
• modify nsh to utilise rcctl infrastructure

• Implement support for all OpenBSD base Daemons
  • nsd
  • unbound
  • bgplg

• allow configuration
• Improve testing with configuration test scripts / sample configurations (similar to freertr)
To do

• working support for additional network packages / ports
  • Openvpn
  • LLDPd
  • pmacct
  • nfdump
  • pftop
  • iftop
  • fastnetmon

• Review the advances in bgpctl and try to apply lessons learned to nsh
  • Configuration management, display of large data sets, syntax validation...

• review the syntax used in commands to ensure that they are succinct, and understandable

• investigate the possibility of improving of detection of running daemons and the config files used by them (to display in the running-config)
To Do

• [https://github.com/users/yellowman/projects/1/views/5](https://github.com/users/yellowman/projects/1/views/5)
• squash bugs
  • improve automated testing / regression testing
• improve the configuration import capability from other daemons
Other Software worth checking out and learning from...

- Openbgpd- bgpctl learn from the extensive functionality for querying status and changing configuration at scale and reliability. Thanks Claudio, Job, Theo and the other Theo 😊
- Bsdrp – BSD Router Project - [https://bsdrp.net/](https://bsdrp.net/)
- FRR – Free Range Routing - [https://frrouting.org/](https://frrouting.org/)
  - Examine the syntax and cli setup and compare with nsh
  - Java based router software with a phenomenal feature set, and excellent CI and regression test setup.
  - Runs on FreeBSD or Linux
NSH for FreeBSD *BSD

• NSH Project will try to facilitate
• Feedback welcome on how we can make NSH more conducive to
• supporting multiple Oses
Funding for NSH development

- Wireless connect has been funding development in 2022-2023
- Stefan Sperling - has provided excellent value for money in the development of the software.
- Looking for additional funding to support development of the project.
- If you want a feature that makes sense and are willing to fund its implementation we will match funding.
NSH - Network SHell for OpenBSD

Thanks for your patience

Questions?