

Networks on Linux

Adam Boulton (www.bou.lt)

February 11, 2024

Contents

I	Networks and the Linux Kernel	2
II	Linux tools	3
1	inetutils	4
2	iputils including ping and tracepath (and traceroute)	5
3	iproute2 (and netstat)	6
4	tcpdump and pcap files	8
5	nmap suite including ncat (and netcat/nc)	9
III	Domain Name System (DNS)	10
6	BIND (including dig, nslookup and host)	11
7	whois	12
8	/etc/hosts and dnsmasq	13
IV	Other	14
9	git	15
10	email	17
11	rsync	18
12	Secure SHell (ssh)	19
13	Network File System (NFS)	20

<i>CONTENTS</i>	2
V Firewalls	21
14 iptables and ufw (uncomplicated firewall)	22
VI Virtual Private Networks (VPNs)	23
15 WireGuard and OpenVPN	24
VII Merkel Trees	25
16 Merkel trees	26

Part I

Networks and the Linux Kernel

Part II

Linux tools

Chapter 1

inetutils

1.1 Introduction

1.1.1 Introduction

telnet(d)

talk(d)

rsh(d)

rlogin(d)

ftp(d)

dnsdomainname

hostname

rcp

Chapter 2

iputils including ping and tracpath (and traceroute)

2.1 Introduction

2.1.1 Introduction

2.1.2 tracpath

By default does reverse DNS. n flag turns it off.

```
tracpath -n google.com
```

Can get reverse DNS and IP with -b flag

```
tracpath -b google.com
```

2.1.3 traceroute

Alternative to tracpath. Same behaviour. Different package.

send with ttl 1, 2, 3, 4 etc so get IPs of where ttl end uses dns stuff, so have after?

```
traceroute google.com
```

Chapter 3

iproute2 (and netstat)

3.1 Introduction

3.1.1 Introduction

not sure how to use netstat or ss. — grep 8080 isn't working for either when i have node server listening on 8080

3.1.2 ip link

information on network hardware on computer
replaces part of ifconfig

3.1.3 ip addr

information on connections, including local IP on network (eg 192.168.0.x)
ip -c addr (makes it easier to see colour)
replaces part of ifconfig

3.1.4 ip route

can show ip of router (default via...)
replaces route

3.1.5 ss (socket statistics)

3.1.6 Old: netstat

netstat replace with ss, ip route

netstat -inet -ap netstat -puna? + to see what processes using internet?

can see which programs using netstat netstat -program

Chapter 4

tcpdump and pcap files

4.1 Introduction

4.1.1 Introduction

can capture dns without extra stuff with

```
tcpdump -l -n port 53 | grep --line-buffered ' A? ' | awk -F ' A\\? ' '{ print $2 }' | awk -
```

can use -n flag. otherwise does reverse dns on everything, and this shows up in traffic.

can capture reverse dns results:

```
tcpdump -l not port 53 | grep --line-buffered -v " > arctop" | grep --line-buffered "arct
```

can save output to pcap file with -w flag can monitor traffic of whole network, but for wifi need card to be in monitor mode. can be done using airmon-ng from aircrack-ng

tcpdump port 53 (monitor dns requests)

tcpdump -D (shows interfaces possible to capture on) tcpdump -interface any (capture on all) (default behaviour)

can ignore specific domains:

```
tcpdump -f "not host ec2-44-228-235-78.us-west-2.compute.amazonaws.com"
```

```
tcpdump -f "not host ec2-44-228-235-78.us-west-2.compute.amazonaws.com or host "
```

Chapter 5

nmap suite including ncat (and netcat/nc)

5.1 Introduction

5.1.1 Introduction

5.1.2 nmap

eg nmap 192.168.0.0/24

can do nmap -sn to skip port scanning

5.1.3 ncat

5.1.4 netcat

netcat (nc) is an alternative to ncat. Not needed if have nmap suite.

used to connect to addresses using TCP or UDP can be used as back end by other programs

Part III

Domain Name System (DNS)

Chapter 6

BIND (including dig, nslookup and host)

6.1 Introduction

6.1.1 Introduction

dig google.com dig google.com +short

Can also do reverse DNS with dig

dig -x 1.2.3.4 dig -x 1.2.3.4 +short

host is another similar program

host google.com

You can do reverse DNS with host by just using the IP.

host 1.2.3.4

nslookup is like host and dig, but interactive.

nslookup

You can do the following anyway

nslookup google.com

nslookup 1.2.3.4

Chapter 7

whois

7.1 Introduction

7.1.1 Introduction

whois bou.lt

Chapter 8

/etc/hosts and dnsmasq

8.1 Introduction

8.1.1 /etc/hosts

```
/etc/hosts
127.0.0.1      localhost
#Can route an actual website
```

```
0.0.0.0       example.org
```

8.1.2 dnsmasq

The following in the following file will allow traffic to google.com and block everything else.

```
/etc/dnsmasq.conf
address=/google.com/
address=#!/0.0.0.0
```

Part IV

Other

Chapter 9

git

9.1 Introduction

9.1.1 Introduction

git init

HEAD,origin,remote,local,name of main branch (main aka master)

add; commit; stash; checkout

.git/ folder

single dev using production machine:

git commit: a flag adds files which have been already added but are modified.
doesn't add new files

reset hard

git status

git checkout specific branch, force?

".gitconfig" file

git: + return a file in git to last commit: git checkout - file.txt + HEAD is
most recent checkout + git checkout HEAD

git diff: look at changes that will be committed before doing a commit

9.1.2 Setting up a remote server

git fetch

pull

9.1.3 Managing multiple contributors

rebase

dealign with clashes

merge,

reset

fast forward (fast forward and non-fast forward are merge variants)

squash?

git blame

9.1.4 Managing multiple branches

git branch [branch name] (makes new branch)

9.1.5 reflogs

9.1.6 git-lfs

9.1.7 Git server

9.1.8 Hooks

+ ./git/hooks directory exists by default in project. anything in there without an extension is a hook. examples with extension are there by default. + types of hook: pre-commit; prepare-commit-msg; commit-msg; post-commit + page on server side git. server side git hooks.

Chapter 10

email

10.1 Introduction

10.1.1 Introduction

internet message access protocol (imap)

post office protocol (pop)

simple mail transfer protocol (smtp)

domain name system blacklist (dnsbl)

Chapter 11

rsync

11.1 Introduction

11.1.1 Introduction

sync between two directories. uses delta, and can use compression
can use multiple protocols?

Chapter 12

Secure SHell (ssh)

12.1 Introduction

12.1.1 Introduction

ssh over non standard ports ssh: + /.ssh/config file

Chapter 13

Network File System (NFS)

13.1 Introduction

13.1.1 Introduction

Part V

Firewalls

Chapter 14

iptables and ufw (uncomplicated firewall)

14.1 iptables

14.1.1 iptables

14.2 ufw

14.2.1 ufw

ufw built on iptables

deny on port

deny on ip

using iptables/ufw to block ports ufw status

```
ufw allow in on cni0 && sudo ufw allow out on cni0
```

```
ufw default allow routed
```

```
ufw allow 6443/tcp
```

```
ufw allow 443/tcp
```

14.3 Firewalling a network

14.3.1 Networks and router firewalls

Part VI

Virtual Private Networks (VPNs)

Chapter 15

WireGuard and OpenVPN

15.1 Introduction

15.1.1 WireGuard

Part VII

Merkel Trees

Chapter 16

Merkel trees

16.1 Introduction

16.1.1 Introduction