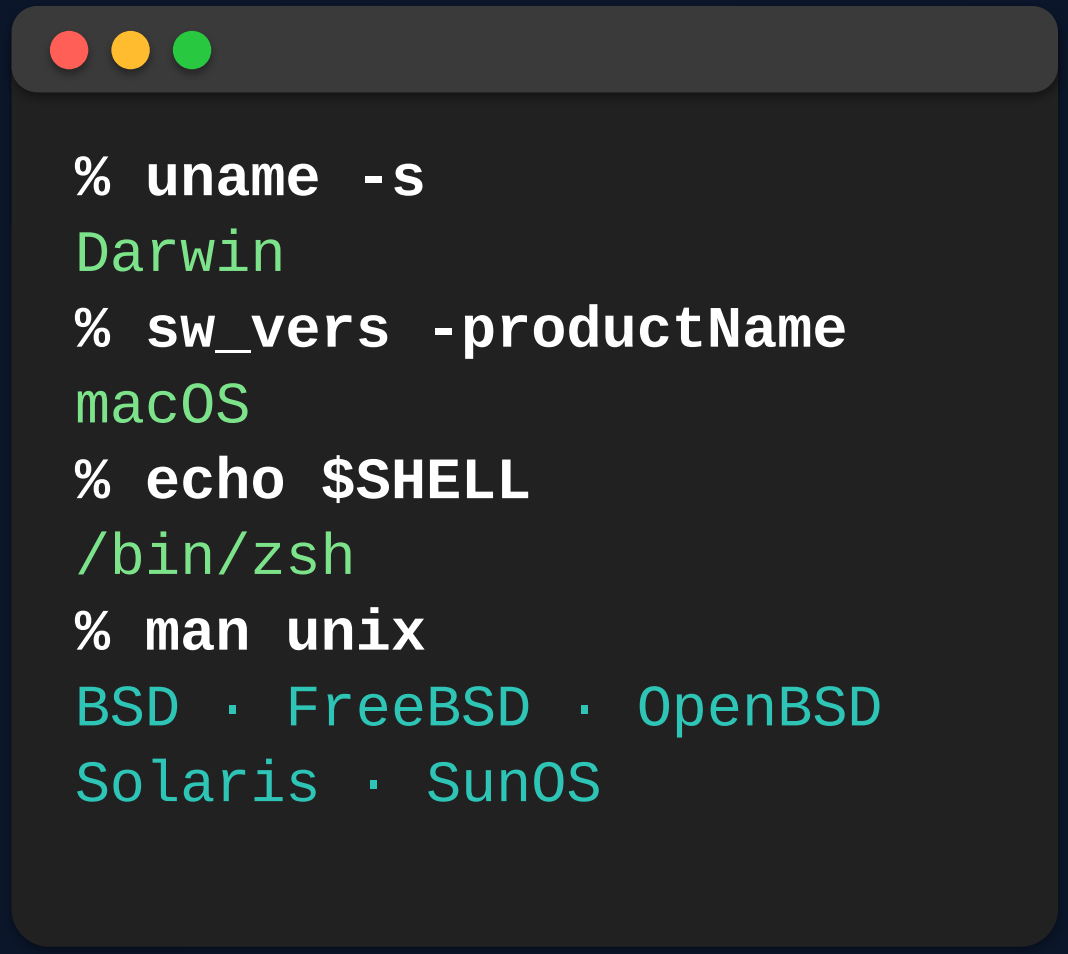


LESSON 10 OF 12

macOS UNIX

Darwin & BSD

Network Configuration, Firewall & SSH

Full Course · macOS 27 Golden Gate

What You Will Learn in Lesson 10

- Network services, interfaces and the Network settings panel
- networksetup: DHCP, static IP and DNS from Terminal
- ifconfig, ipconfig, netstat, route, scutil and hostnames
- Troubleshooting: ping, traceroute, dig, nc, curl, networkQuality
- The Application Firewall and the pf packet filter
- SSH: Remote Login, keys, config, hardening, scp and rsync
- Networking on FreeBSD, OpenBSD and Solaris

Interfaces and Network Services

```
% networksetup -listallhardwareports
Hardware Port: Ethernet      Device: en0
Hardware Port: Wi-Fi         Device: en1
% networksetup -listallnetworkservices
% networksetup -getinfo "Ethernet"       # IP, mask, router, DHCP or manual

% ifconfig en0 | grep -E 'ether|inet '     # MAC and IP addresses
% ipconfig getifaddr en0                   # just the IPv4 address
% route -n get default | grep gateway      # default router
```

Example: device names vary (a UTM VM usually shows Ethernet en0).

Network Settings in the GUI

Where	What you set
System Settings → Network → (service) → Details	TCP/IP: DHCP or manual, router, IPv6
… → Details → DNS	DNS servers and search domains
… → Details → Hardware	MAC address, speed, MTU
Network → ⋯ → Set Service Order	Which connection is used first
Network → Firewall	Application Firewall on/off, options
Wi-Fi → Details	Private Wi-Fi address, Low Data Mode, forget network

networksetup: Configure From Terminal

```
% sudo networksetup -setmanual "Ethernet" 192.168.64.50 255.255.255.0 192.168.64.1
% sudo networksetup -setdnsservers "Ethernet" 1.1.1.1 9.9.9.9
% networksetup -getdnsservers "Ethernet"

% sudo networksetup -setdhcp "Ethernet"           # back to automatic
% sudo networksetup -setdnsservers "Ethernet" Empty

% networksetup -setairportpower en1 off           # Wi-Fi off / on
% sudo networksetup -setv6automatic "Wi-Fi"
```

Use addresses from your own network. In UTM's default shared network, DHCP is easiest.

Names: Hostnames and DNS

```
% scutil --get ComputerName;  scutil --get LocalHostName;  hostname
% sudo scutil --set ComputerName "MWU Lab Mac"     # shown in Finder, sharing
% sudo scutil --set LocalHostName mwu-lab          # Bonjour: mwu-lab.local
% sudo scutil --set HostName mwu-lab.example.com   # UNIX hostname

% scutil --dns | grep nameserver | sort -u         # resolvers in use
% dig +short www.MyWebUniversity.com
% dscacheutil -q host -a name www.apple.com        # lookup the macOS way
% sudo dscacheutil -flushcache; sudo killall -HUP mDNSResponder   # flush DNS
```

Troubleshooting From the Bottom Up

```
% ifconfig en0 | grep 'status\|inet '   # 1. link up? have an IP?
% ping -c 3 $(route -n get default | awk '/gateway/{print $2}')  # 2. router
% ping -c 3 1.1.1.1                     # 3. the internet by number
% dig +short apple.com                  # 4. DNS works?
% nc -vz github.com 443                 # 5. is the port open?
% curl -I https://www.MyWebUniversity.com # 6. does the service answer?
% traceroute -n 1.1.1.1                 # where does it stop?
% networkQuality                        # speed and responsiveness
```

Ports and Connections: netstat and lsof

```
% netstat -rn -f inet | head              # routing table
% netstat -an -p tcp | grep LISTEN       # listening TCP ports
% lsof -iTCP -sTCP:LISTEN -n -P           # ...with program names
% lsof -i -n -P | grep ESTABLISHED | head # current connections
% arp -a                                  # neighbors on the LAN
% sudo tcpdump -i en0 -n port 53 -c 10    # watch DNS packets
```

The Application Firewall

```
% FW=/usr/libexec/ApplicationFirewall/socketfilterfw
% sudo $FW --getglobalstate
Firewall is disabled. (State = 0)
% sudo $FW --setglobalstate on
% sudo $FW --setstealthmode on            # don't answer pings/probes
% sudo $FW --listapps
% sudo $FW --add /opt/homebrew/bin/nginx  # allow an app
% sudo $FW --blockapp /Applications/Unwanted.app
```

pf: The Packet Filter From OpenBSD

```
% sudo pfctl -s info | head -3            # is pf enabled?
% grep anchor /etc/pf.conf                # macOS loads com.apple/* anchors

% echo 'block drop in quick proto tcp to any port 8080' | sudo tee /etc/pf.anchors/mwu
% sudo pfctl -a com.apple/mwu -f /etc/pf.anchors/mwu   # load our rules
% sudo pfctl -E                           # enable pf (with a reference)
% sudo pfctl -a com.apple/mwu -s rules     # show them

% sudo pfctl -a com.apple/mwu -F rules     # flush our rules when done
```

Test pf on your VM first: a wrong rule can cut off your SSH session.

SSH: Remote Login and Keys

```
# On the Mac you want to reach (or: General → Sharing → Remote Login):
% sudo systemsetup -setremotelogin on      # Terminal may need Full Disk Access
% sudo systemsetup -getremotelogin

# On your own computer:
% ssh-keygen -t ed25519 -C "wahid@mwu-lab"
% ssh-copy-id wahid@mwu-lab.local
% ssh wahid@mwu-lab.local
% ssh-add --apple-use-keychain ~/.ssh/id_ed25519   # passphrase in Keychain
```

SSH Config, Hardening and File Transfer

```
# ~/.ssh/config  (client)
Host lab
    HostName mwu-lab.local
    User wahid
    UseKeychain yes
    AddKeysToAgent yes
% ssh lab;   scp report.pdf lab:~/;   rsync -avz ~/site/ lab:~/site/

# Server: /etc/ssh/sshd_config.d/100-mwu.conf
PasswordAuthentication no
KbdInteractiveAuthentication no
PermitRootLogin no
% sudo launchctl kickstart -k system/com.openssh.sshd
```

Keep one SSH session open while you test a new server setting.

Networking on Other UNIX Systems

Task	FreeBSD	OpenBSD	Solaris 11.4
Set DHCP	sysrc ifconfig_em0=DHCP	echo 'inet autoconf' > /etc/hostname.em0	ipadm create-addr -T dhcp net0/v4
Apply	service netif restart	sh /etc/netstart em0	(immediate)
Show links	ifconfig	ifconfig	dladm show-phys; ipadm
Firewall	pf or ipfw (pf_enable=YES)	pf (on by default)	PF (svcadm enable firewall)
SSH server	sysrc sshd_enable=YES	rcctl enable sshd	svcadm enable ssh

Lab: Connect, Protect, Reach

```
% networksetup -listallhardwareports; ipconfig getifaddr en0
% scutil --dns | grep nameserver | sort -u;  dig +short apple.com
% sudo scutil --set LocalHostName mwu-lab
% sudo /usr/libexec/ApplicationFirewall/socketfilterfw --setglobalstate on
% sudo systemsetup -setremotelogin on
# From your host Mac (or another VM):
% ssh-keygen -t ed25519 && ssh-copy-id you@mwu-lab.local && ssh you@mwu-lab.local
% lsof -iTCP -sTCP:LISTEN -n -P | grep sshd
```

Summary & Homework

- Hardware ports (en0) vs network services; networksetup and scutil
- Troubleshoot bottom-up: link → router → IP → DNS → port → service
- Application Firewall (socketfilterfw) + pf anchors (pfctl)
- SSH: ed25519 keys, ssh-copy-id, ~/.ssh/config, no passwords, no root
- Homework: do the lab; block and unblock a port with pf
- Next: Lesson 11 — Storage, APFS, Security & Backup

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

Slides for this lesson: MyWebUniversity.com/presentations

YouTube: @MyWebUniversity · @MyWebUniversity-Dari

Wahidullah Lutfy — Founder and Director of IT of MyWebUniversity, and CS teacher

Book: The Direct Path to Linux Ubuntu (Amazon · Barnes & Noble)

Learn it, then pass it on.