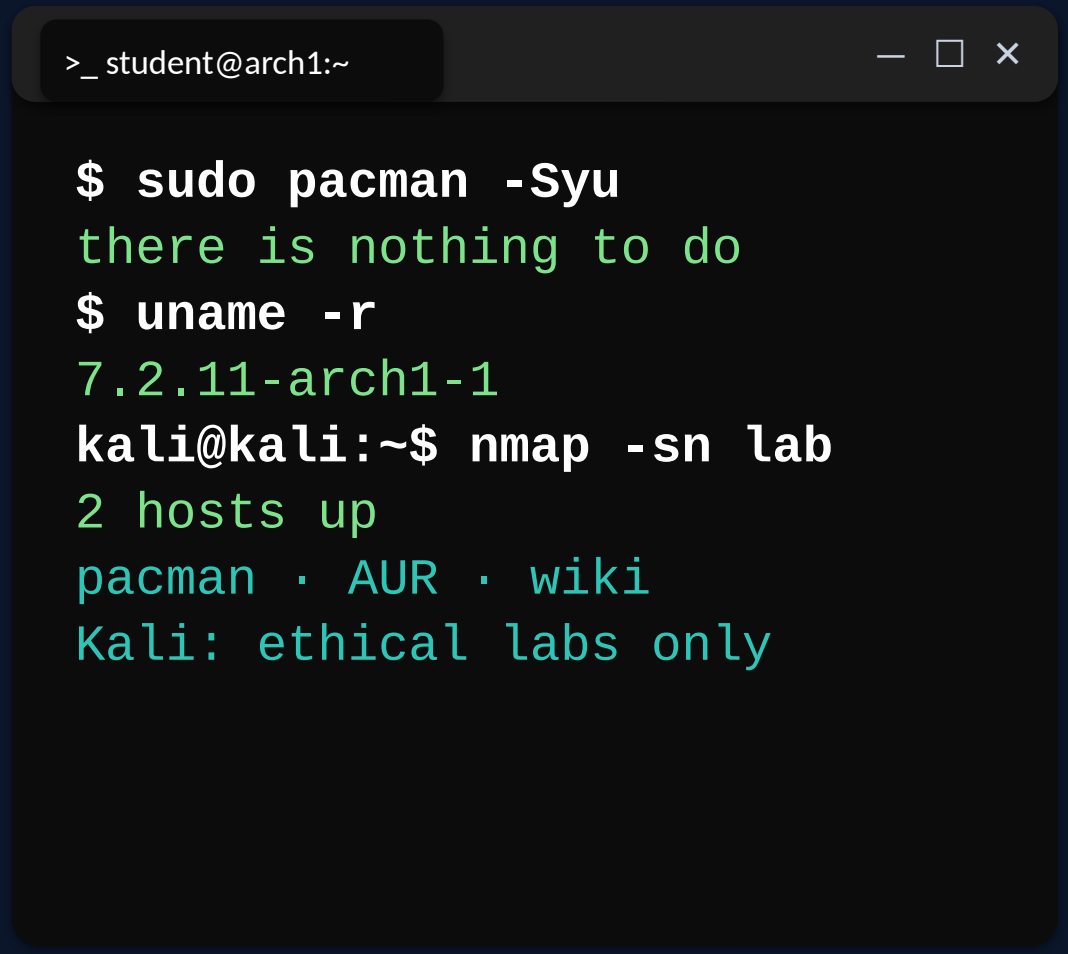


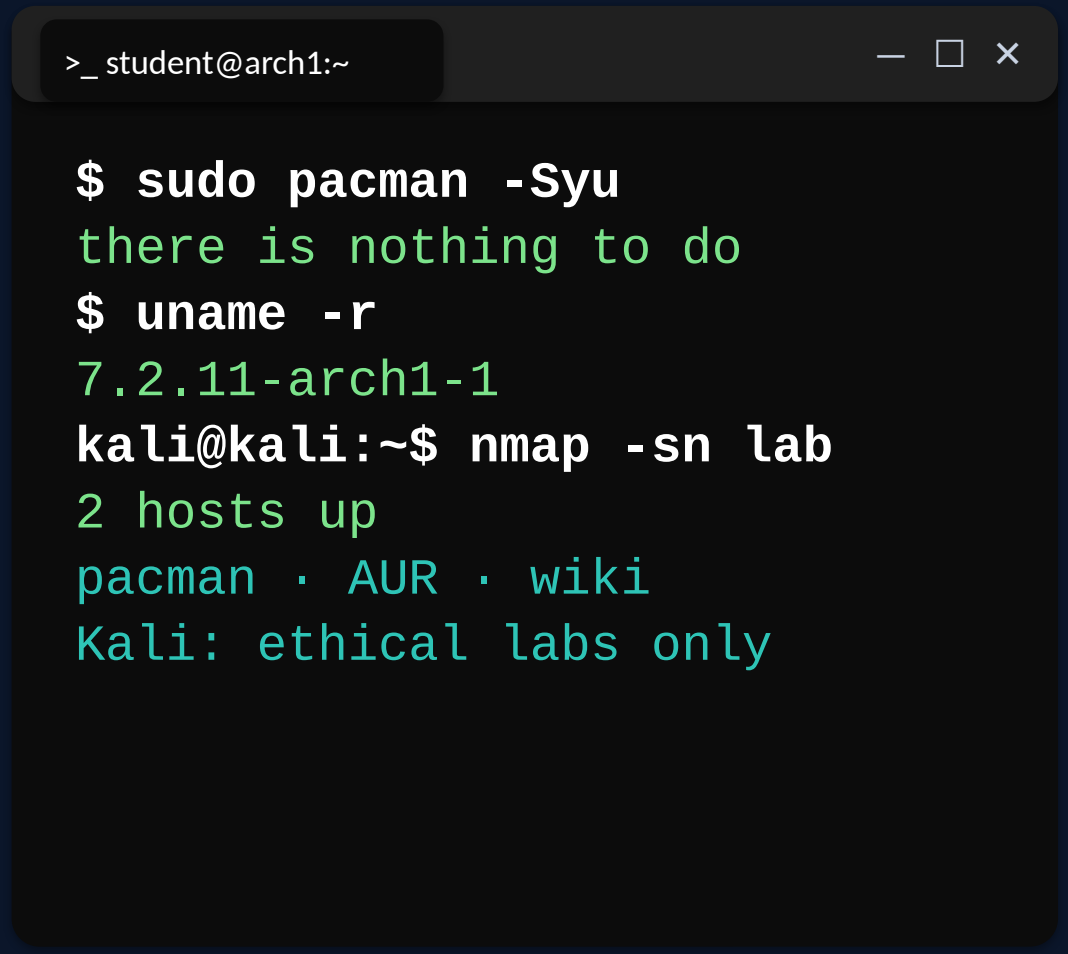
FULL COURSE · 12 LESSONS

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

archinstall, pacman, AUR, Kali tools & safe labs

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)



LESSON 1 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

Arch & Kali: Rolling Releases, Philosophy & When to Use Each

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 1

- Rolling release vs fixed release
- Arch Linux: the Arch Way, simplicity and user control
- Kali Linux: a security toolkit based on Debian testing
- Where each one fits — and where it does not
- The ethics and law of security testing
- Course plan and lab

Rolling vs Fixed Releases

	Rolling (Arch, Kali)	Fixed (Debian, Ubuntu LTS, RHEL)
Versions	No big versions; packages update continuously	Major releases every 2–3 years
Newest software	Days or weeks after upstream	Frozen at release, security fixes only
Upgrades	Small and often (pacman -Syu / apt full-upgrade)	Big release upgrades
Risk	Occasional breakage; read the news	Very stable, older software
Install media	Monthly ISO snapshot (Arch), quarterly (Kali)	Point release images
Best for	Enthusiasts, developers, labs, special toolkits	Servers, enterprise, long-term stability

Arch Linux at a Glance

Item	Arch Linux
Started	2002, Judd Vinet; community-run, elected project leader
The Arch Way	Simplicity, modernity, pragmatism, user centrality, versatility
Current ISO	2026.11.01 — Linux 7.2.11, systemd 262.1, Archinstall 4.5
Packages	pacman + official repos (core, extra, multilib) + the AUR
Docs	The Arch Wiki — useful for every Linux distribution
Architectures	x86-64 officially (ARM via the separate Arch Linux ARM project)

Kali Linux at a Glance

Item	Kali Linux
Made by	OffSec (Offensive Security); successor of BackTrack (2013)
Base	Debian testing → 'kali-rolling' branch
Current	Kali 2026.3 (Oct 2026): Linux 7.1, Xfce default, KDE Plasma 6.7, GNOME
Purpose	Penetration testing, security research, forensics, reverse engineering
Tools	600+ packaged tools in metapackages (kali-linux-default, -large, -everything)
Runs on	VMs, bare metal, live USB, WSL, cloud, ARM, Android (NetHunter)

Where Each One Fits

Use	Arch	Kali
Daily desktop / developer laptop	Yes — if you enjoy control	No — not a general desktop
Learning how Linux really works	Excellent	Some
Production servers	Rarely (prefer fixed releases)	No
Security labs, CTFs, certifications	Possible (BlackArch repo)	Yes — the standard
Forensics / incident response	—	Yes (forensic boot mode)
Gaming	Yes (Steam, SteamOS is Arch-based)	No

Ethics and Law: Rules for the Kali Half

Rule	Why
Only test systems you own or have WRITTEN permission to test	Unauthorized access is a crime in most countries
Stay inside the agreed scope and time window	Out-of-scope testing is still unauthorized
Use your own isolated lab for practice	Intentionally vulnerable VMs, no real users
Never test your school, employer, ISP or neighbors' Wi-Fi without authorization	Good intentions are not a legal defense
Protect any data you see; report findings responsibly	Confidentiality and responsible disclosure
Learn to defend, not only to attack	The goal is safer systems for everyone

The Course Plan

Lessons	Arch half	Lessons	Kali half
1	Rolling releases & philosophy	7	Purpose, ethics, law, editions
2	Installing Arch (archinstall)	8	Installing Kali: VM, USB, WSL
3	pacman, repos, mirrors, keys	9	kali-rolling, metapackages, tweaks
4	AUR, PKGBUILD, helpers	10	Tools tour (in your lab)
5	Running Arch well	11	Safe lab & Kali Purple
6	Arch family & troubleshooting	12	NetHunter, reporting, careers

The Course Lab

Host	System	Network	Used for
arch1	Arch Linux (VM, UEFI)	NAT, 10.0.0.51	Lessons 2–6
kali	Kali Linux 2026.3 (VM)	NAT + host-only 192.168.56.10	Lessons 7–12
target1	Intentionally vulnerable VM (e.g. Metasploitable 2)	host-only 192.168.56.20	Practice target — never on the internet
target2	DVWA / OWASP Juice Shop in a container	host-only 192.168.56.30	Web practice
Host	VirtualBox, KVM/virt-manager or VMware	8 GB+ RAM, 80 GB disk	Snapshots before changes
Users	student (Arch), kali (Kali default user)		sudo

Summary and Homework

- Rolling = continuous updates, newest software, read the news
- Arch: simple, user-controlled, pacman + AUR, the Arch Wiki
- Kali: Debian testing-based security toolkit by OffSec; 2026.3 current
- Arch for desktops/learning; Kali for authorized security work and labs
- Ethics: own systems or written permission, scope, isolated lab
- Homework: download the Arch ISO and Kali VM image; set up the host-only network

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

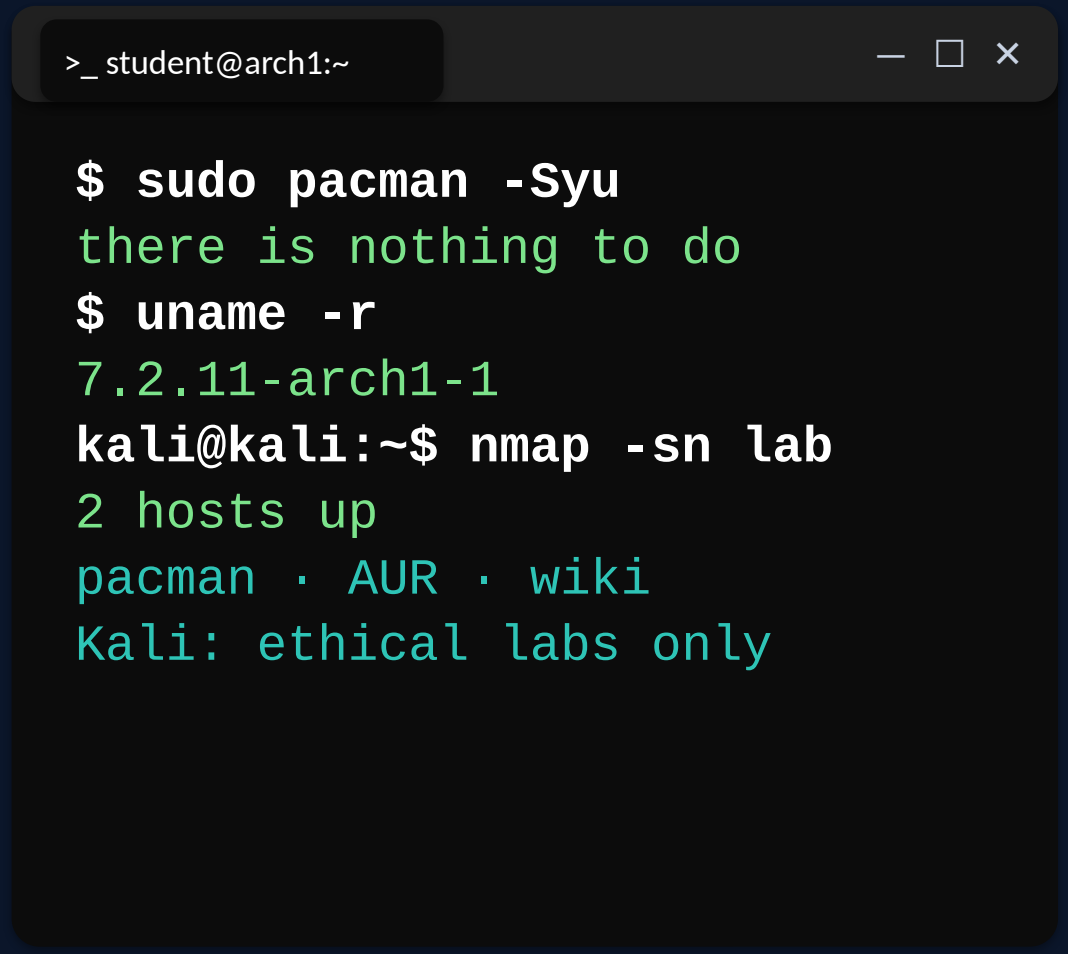
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 2 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

Installing Arch Linux with archinstall (and What It Does)

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 2

- Download and verify the monthly ISO
- Boot the live environment: network, time, keyboard
- archinstall: disks, Btrfs, bootloader, users, profile
- Choosing kernels, audio, network and extra packages
- First boot: update, services, microcode, AUR prep
- The manual install: pacstrap, genfstab, arch-chroot
- When to choose manual vs archinstall

Download and Verify the ISO

```
# From archlinux.org/download (or a mirror close to you)
$ ls
archlinux-2026.11.01-x86_64.iso  archlinux-2026.11.01-x86_64.iso.sig  sha256sums.txt
$ sha256sum -c sha256sums.txt --ignore-missing
archlinux-2026.11.01-x86_64.iso: OK
$ gpg --auto-key-locate clear,wkd -v --locate-external-key pierre@archlinux.org
$ gpg --verify archlinux-2026.11.01-x86_64.iso.sig
gpg: Good signature from "Pierre Schmitz <pierre@archlinux.org>"
# On an existing Arch system: pacman-key -v archlinux-...iso.sig
# Write to USB: dd bs=4M if=archlinux-...iso of=/dev/sdX status=progress oflag=sync
```

The Live Environment

```
root@archiso ~ # ls /sys/firmware/efi/efivars >/dev/null && echo UEFI
UEFI
root@archiso ~ # loadkeys us                    # keyboard layout (e.g. de, fr)
root@archiso ~ # ip -br link                    # wired works automatically (DHCP)
root@archiso ~ # iwctl                          # Wi-Fi:
[iwd]# station wlan0 scan
[iwd]# station wlan0 connect "MyWiFi"
root@archiso ~ # ping -c3 archlinux.org
root@archiso ~ # timedatectl                     # NTP sync should be active
root@archiso ~ # archinstall
```

archinstall: Key Menu Choices

Menu item	Our choice for arch1
Archinstall language / Locales	English; keyboard us; locale en_US.UTF-8
Mirrors and repositories	Your country's mirrors; enable 'multilib' if you want Steam/Wine
Disk configuration	Best-effort layout on /dev/vda → Btrfs with subvolumes + compression
Disk encryption	Optional: LUKS with a passphrase (recommended on laptops)
Swap	zram (compressed swap in RAM) — default
Bootloader	systemd-boot (UEFI) — or GRUB if you need BIOS/advanced features
Hostname	arch1
Root password / User account	Leave root locked; user 'student' as superuser (sudo)

archinstall: Profile, Kernels and More

Menu item	Options
Profile	Desktop (GNOME, KDE Plasma, Xfce, Hyprland, COSMIC ...), Server, or Minimal
Graphics driver	Open-source (Intel/AMD/nouveau) or NVIDIA open kernel modules
Audio	PipeWire (recommended)
Kernels	linux (latest) + linux-lts as a fallback
Network configuration	NetworkManager (desktop) or systemd-networkd (server)
Additional packages	vim git base-devel openssh man-db
Timezone / NTP	Your timezone; automatic time sync on
Save configuration	Saves user_configuration.json for repeatable installs

First Boot

```
[student@arch1 ~]$ sudo pacman -Syu                # full system upgrade
[student@arch1 ~]$ systemctl --failed
[student@arch1 ~]$ sudo systemctl enable --now sshd   # Arch names it sshd
[student@arch1 ~]$ lscpu | grep 'Vendor ID'
[student@arch1 ~]$ pacman -Q intel-ucode amd-ucode 2>/dev/null   # CPU microcode
[student@arch1 ~]$ sudo pacman -S --needed base-devel git       # for the AUR (lesson 4)
[student@arch1 ~]$ sudo pacman -S pacman-contrib reflector man-db
[student@arch1 ~]$ sudo systemctl enable --now paccache.timer   # clean old packages
[student@arch1 ~]$ cat /etc/os-release | head -3
```

Underneath: The Manual Install

```
# What archinstall does, step by step (simplified)
# fdisk/parted: 1 GB EFI (vfat) + root
mkfs.fat -F32 /dev/vda1 ; mkfs.btrfs /dev/vda2
mount /dev/vda2 /mnt ; mount --mkdir /dev/vda1 /mnt/boot
pacstrap -K /mnt base linux linux-firmware intel-ucode vim networkmanager sudo
genfstab -U /mnt >> /mnt/etc/fstab
arch-chroot /mnt
  ln -sf /usr/share/zoneinfo/Asia/Kabul /etc/localtime ; hwclock --systohc
  sed -i 's/^#en_US.UTF-8/en_US.UTF-8/' /etc/locale.gen ; locale-gen
  echo arch1 > /etc/hostname ; mkinitcpio -P
  bootctl install    # + loader entries ;  useradd -m -G wheel student ; passwd student
  EDITOR=vim visudo  # uncomment: %wheel ALL=(ALL:ALL) ALL ; systemctl enable NetworkManager
```

Manual or archinstall?

Choose	When
archinstall	You want a working, standard Arch in 15 minutes; repeatable JSON configs
Manual	You want to learn every step, or need unusual layouts (custom LVM/RAID, Secure Boot keys)
Both	Same result: plain Arch, no installer leftovers — you own the system
Tip	Do the manual install once in a VM to learn, then use archinstall day to day
Reference	wiki.archlinux.org/title/Installation_guide

Summary and Homework

- Verify: sha256sum + gpg --verify (or pacman-key -v)
- Live ISO: loadkeys, iwctl for Wi-Fi, ping, timedatectl
- archinstall: Btrfs, systemd-boot, user with sudo, profile, linux + linux-lts
- First boot: pacman -Syu, sshd, microcode, base-devel, paccache.timer
- Manual: pacstrap → genfstab → arch-chroot → mkinitcpio → bootloader
- Homework: install arch1 with archinstall; save the JSON; try a manual install in a 2nd VM

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

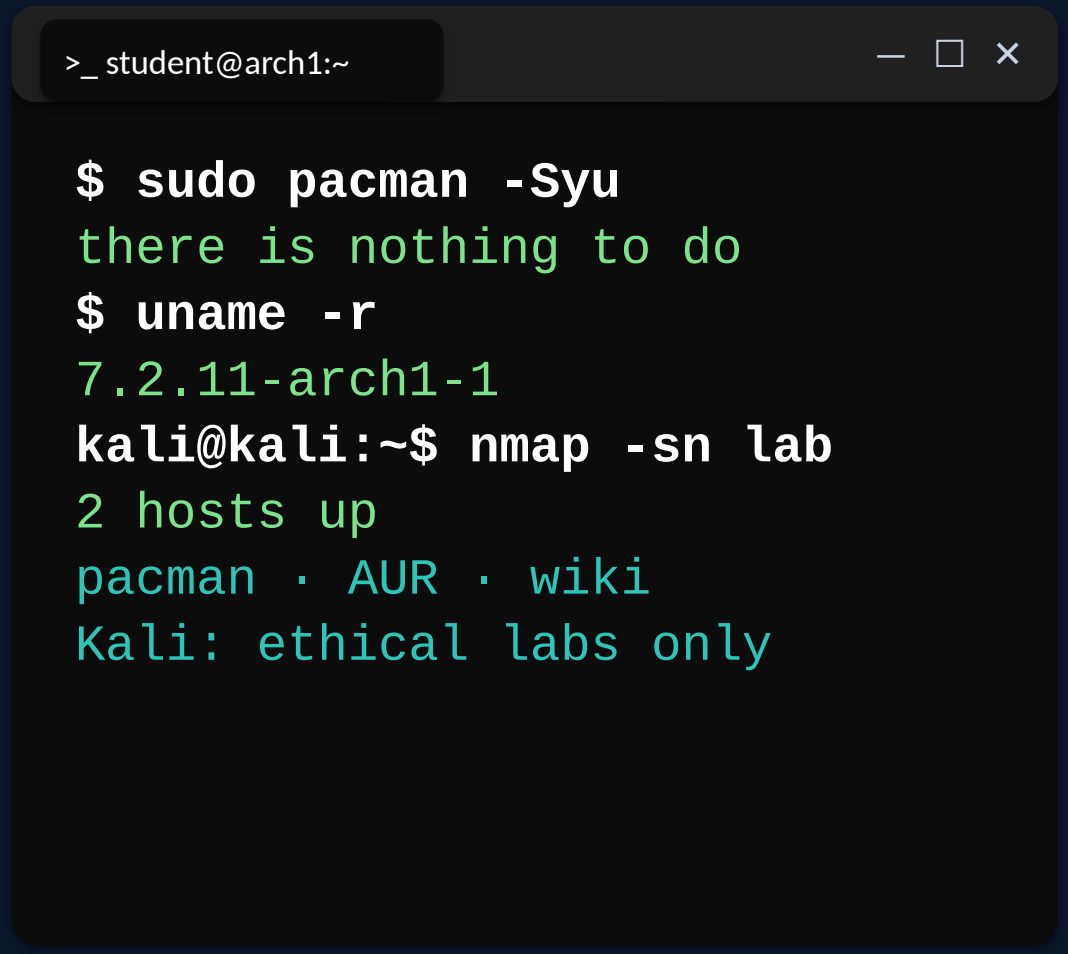
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 3 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

pacman: Packages, Repositories, Mirrors & Keys

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 3

- pacman operations: -S, -R, -Q, -F, -U and their flags
- The one rule: full upgrades only (pacman -Syu)
- Official repos: core, extra, multilib (and testing)
- Mirrors with reflector
- Package signing, archlinux-keyring and pacman-key
- .pacnew / .pacsave files and pacdiff
- The cache, downgrades and the Arch Linux Archive

pacman Operations

Command	Meaning
pacman -Syu	Sync databases (y) and upgrade everything (u) — the daily command
pacman -S nginx	Install from the repos (S = sync)
pacman -Ss / -Si nginx	Search repos / show repo package info
pacman -Rns nginx	Remove + unneeded dependencies (s) + config backups (n)
pacman -Q / -Qi / -Ql nginx	Query installed: list / info / files
pacman -Qo /usr/bin/ssh	Which package owns this file?
pacman -F bin/dig	Search files in packages not installed (after pacman -Fy)
pacman -U ./pkg.pkg.tar.zst	Install a local or downloaded package file

Everyday pacman

```
[student@arch1 ~]$ sudo pacman -Syu
:: Synchronizing package databases...
 core / extra / multilib is up to date
:: Starting full system upgrade...
Packages (14) linux-7.2.11.arch1-1 mesa-1:26.2.6-1 systemd-262.1-1 ...
:: Proceed with installation? [Y/n] y
[student@arch1 ~]$ sudo pacman -S --needed htop btop fastfetch
[student@arch1 ~]$ pacman -Qe | wc -l           # explicitly installed
[student@arch1 ~]$ pacman -Qdtq                 # orphans (unneeded deps)
[student@arch1 ~]$ sudo pacman -Rns $(pacman -Qdtq)
[student@arch1 ~]$ pacman -Qkk openssh 2>&1 | tail -1   # verify files
```

The One Rule: No Partial Upgrades

Do	Don't
sudo pacman -Syu (every few days)	sudo pacman -Sy then pacman -S somepkg
sudo pacman -Syu somepkg (upgrade + install)	Hold back core libraries for months
Read archlinux.org/news before big updates	Upgrade blindly during an important deadline
Reboot after kernel/systemd/glibc updates	Mix packages from different snapshots
Why	Arch supports only fully updated systems — partial upgrades break library links

Repositories and pacman.conf

```
[student@arch1 ~]$ grep -v '^#' /etc/pacman.conf | grep -v '^$'
[options]
HoldPkg     = pacman glibc
Architecture = auto
Color
ParallelDownloads = 5
SigLevel    = Required DatabaseOptional
[core]
Include = /etc/pacman.d/mirrorlist
[extra]
Include = /etc/pacman.d/mirrorlist
[multilib]                    # 32-bit libs: Steam, Wine (uncomment to enable)
Include = /etc/pacman.d/mirrorlist
# core-testing / extra-testing: for testers only
```

Fast Mirrors with reflector

```
[student@arch1 ~]$ sudo pacman -S reflector
[student@arch1 ~]$ sudo cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak
[student@arch1 ~]$ sudo reflector --country 'United States' --protocol https \
      --latest 20 --sort rate --save /etc/pacman.d/mirrorlist
[student@arch1 ~]$ head -5 /etc/pacman.d/mirrorlist
Server = https://mirror.example.edu/archlinux/$repo/os/$arch
[student@arch1 ~]$ sudo vim /etc/xdg/reflector/reflector.conf
[student@arch1 ~]$ sudo systemctl enable --now reflector.timer   # weekly refresh
[student@arch1 ~]$ sudo pacman -Syyu    # force-refresh databases after changing mirrors
```

Package Signing and Keys

```
[student@arch1 ~]$ pacman -Qi archlinux-keyring | grep Version
[student@arch1 ~]$ sudo pacman-key --list-keys | grep -c pub
# Error: 'signature from ... is unknown trust' / 'invalid or corrupted package'
[student@arch1 ~]$ sudo pacman -Sy archlinux-keyring && sudo pacman -Su
# Long-offline machine? Update the keyring first, then the rest.
[student@arch1 ~]$ systemctl list-timers | grep archlinux-keyring   # automatic refresh
# Last resort — reinitialize:
[student@arch1 ~]$ sudo rm -r /etc/pacman.d/gnupg && sudo pacman-key --init \
      && sudo pacman-key --populate archlinux
```

.pacnew Files, Cache and Downgrades

```
warning: /etc/pacman.conf installed as /etc/pacman.conf.pacnew
[student@arch1 ~]$ sudo DIFFPROG='vim -d' pacdiff        # merge new defaults

[student@arch1 ~]$ du -sh /var/cache/pacman/pkg
[student@arch1 ~]$ sudo paccache -rk2                    # keep 2 versions per package

# Downgrade a package that broke something
[student@arch1 ~]$ ls /var/cache/pacman/pkg/ | grep mesa
[student@arch1 ~]$ sudo pacman -U /var/cache/pacman/pkg/mesa-1:26.2.5-1-x86_64.pkg.tar.zst
# Not in cache? archive.archlinux.org/packages/m/mesa/
# Then report the bug and upgrade again when it is fixed
```

Summary and Homework

- -S install, -R remove, -Q query, -F files, -U local file
- Always full upgrades: pacman -Syu (never -Sy + -S)
- core, extra, multilib; pacman.conf: ParallelDownloads, Color
- reflector + reflector.timer for fast mirrors
- Keyring errors: pacman -Sy archlinux-keyring first
- Homework: enable multilib, run reflector, find orphans, merge a .pacnew with pacdiff

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

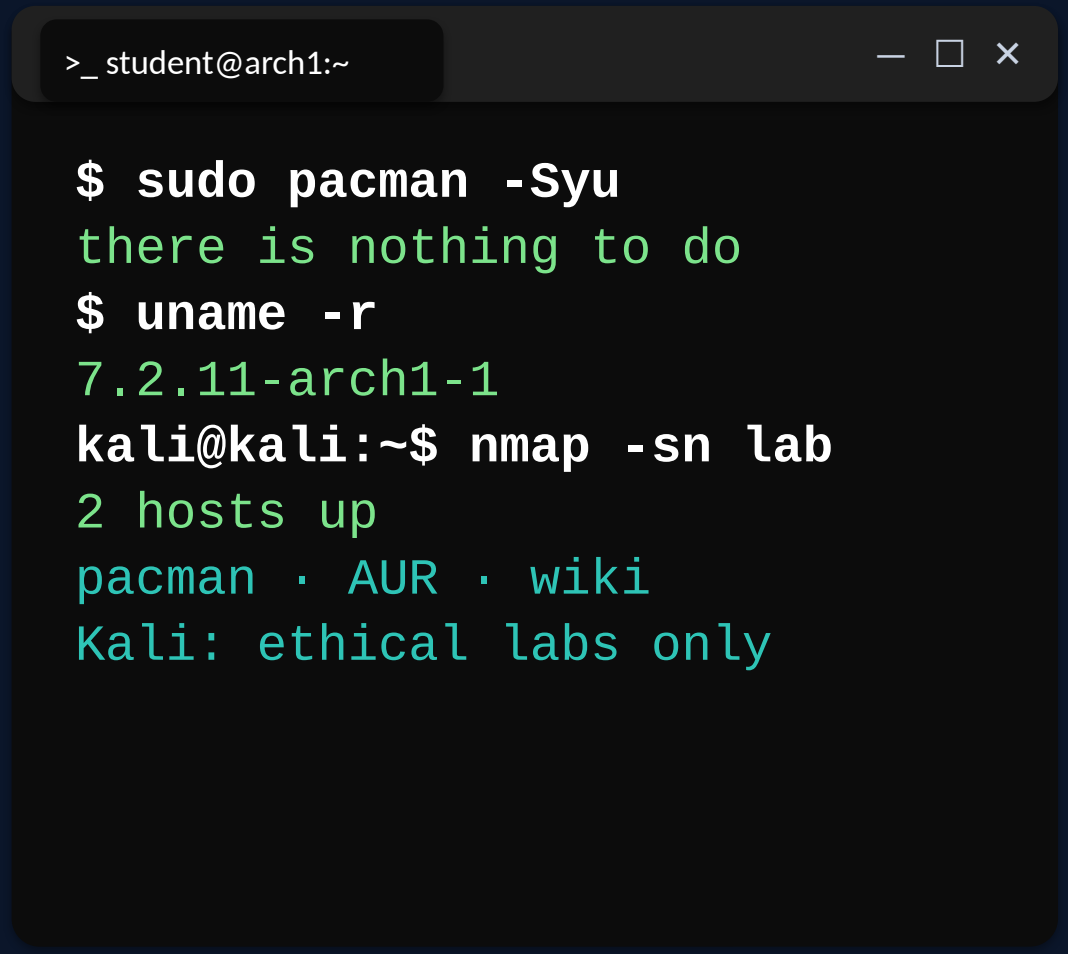
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 4 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

The AUR: PKGBUILD, makepkg, Helpers & Staying Safe

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 4

- What the AUR is — and what it is not
- Reading a PKGBUILD: what to check
- Building by hand with git + makepkg
- AUR helpers: yay and paru
- Updating AUR packages safely
- Real AUR malware incidents and how to protect yourself
- Writing your own PKGBUILD

The AUR at a Glance

Item	Details
What	Community-submitted PKGBUILDs (build recipes), not binary packages
Where	aur.archlinux.org — search, comments, votes, maintainer, last updated
Who	Any user can upload; Arch staff remove bad ones, but there is no pre-review
Support	Unsupported by Arch: you are responsible for what you build
Good signs	Many votes, active maintainer, recent update, source from the real upstream
Popular	Google Chrome, VS Code (binary), Spotify, many -git development versions

Reading a PKGBUILD

```
pkgname=hello-mwu
pkgver=1.0
pkgrel=1
pkgdesc='Friendly greeting from MyWebUniversity'
arch=('any')
url='https://www.mywebuniversity.com'
license=('MIT')
depends=('bash')
source=("https://example.com/hello-$pkgver.tar.gz")   # <- real upstream?
sha256sums=('<sha256-of-the-tarball>')                 # <- not SKIP for releases
package() {
  install -Dm755 "$srcdir/hello.sh" "$pkgdir/usr/bin/hello-mwu"
}
# CHECK: source URLs, checksums, curl/wget in functions, *.install scripts
```

Build by Hand: git + makepkg

```
[student@arch1 ~]$ sudo pacman -S --needed base-devel git
[student@arch1 ~]$ git clone https://aur.archlinux.org/yay-bin.git
[student@arch1 ~]$ cd yay-bin
[student@arch1 yay-bin]$ less PKGBUILD            # READ IT
[student@arch1 yay-bin]$ git log --oneline | head  # who changed what, recently?
[student@arch1 yay-bin]$ makepkg -si              # build, install deps (-s), install (-i)
==> Making package: yay-bin ...
==> Validating source files with sha256sums... Passed
==> Finished making: yay-bin ...
# Never run makepkg as root — it refuses to
```

AUR Helpers: yay and paru

Task	yay	paru
Written in	Go	Rust
Search repos + AUR	yay -Ss name	paru -Ss name
Install	yay -S name	paru -S name
Upgrade all (repos + AUR)	yay (= yay -Syu)	paru (= paru -Syu)
Review PKGBUILD diffs	yay --diffmenu / editmenu	Shows diffs by default
Clean	yay -Yc	paru -c
Rule	Never use sudo with them; still READ every PKGBUILD and diff	

Updating AUR Packages Safely

```
[student@arch1 ~]$ paru                          # repos first, then AUR
:: Looking for AUR upgrades...
 aur/google-chrome  141.0-1 -> 142.0-1
:: Review changes? [Y/n] y                      # diff of PKGBUILD shown
[student@arch1 ~]$ pacman -Qm                    # list foreign (AUR) packages
[student@arch1 ~]$ paru -Gc google-chrome        # read the AUR comments
# After big repo updates (e.g. Python, ICU, Qt) rebuild affected AUR packages:
[student@arch1 ~]$ paru -S --rebuild <package>
# Orphaned/abandoned on the AUR? Consider an official alternative or Flatpak
```

AUR Malware: What Happened and How to Protect Yourself

Event / Rule	Details
July 2025	Three AUR packages with malware (remote-access trojan) found and removed
June 12, 2026	Arch news: active AUR malicious packages incident — review recent commits
Pattern	Fake '-bin' or popular-sounding names; source points to an unknown server
Protect	Read PKGBUILD + .install + git log; prefer official repos and Flatpak
Protect	Check votes, age and maintainer; be wary of brand-new packages
If infected	Remove the package, rotate passwords/keys, reinstall from trusted media

Write Your Own PKGBUILD

```
[student@arch1 ~]$ mkdir hello-mwu && cd hello-mwu
[student@arch1 hello-mwu]$ printf '#!/usr/bin/bash\necho Hello from MyWebUniversity\n' \
      > hello.sh
[student@arch1 hello-mwu]$ vim PKGBUILD   # source=('hello.sh'); package() {...}
[student@arch1 hello-mwu]$ updpkgsums      # fill in sha256sums (pacman-contrib)
[student@arch1 hello-mwu]$ namcap PKGBUILD # lint (pacman -S namcap)
[student@arch1 hello-mwu]$ makepkg -si
[student@arch1 hello-mwu]$ hello-mwu
Hello from MyWebUniversity
[student@arch1 hello-mwu]$ makepkg --printsrcinfo > .SRCINFO   # needed to publish
```

Summary and Homework

- AUR = community PKGBUILDs, unsupported, no pre-review
- Read PKGBUILD, .install and git log BEFORE building
- git clone + makepkg -si; never as root
- yay / paru: repo + AUR upgrades; review diffs; no sudo
- Malware incidents (2025, Jun 2026): prefer official repos/Flatpak, check votes
- Homework: build yay-bin or paru-bin by hand; write and build hello-mwu

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

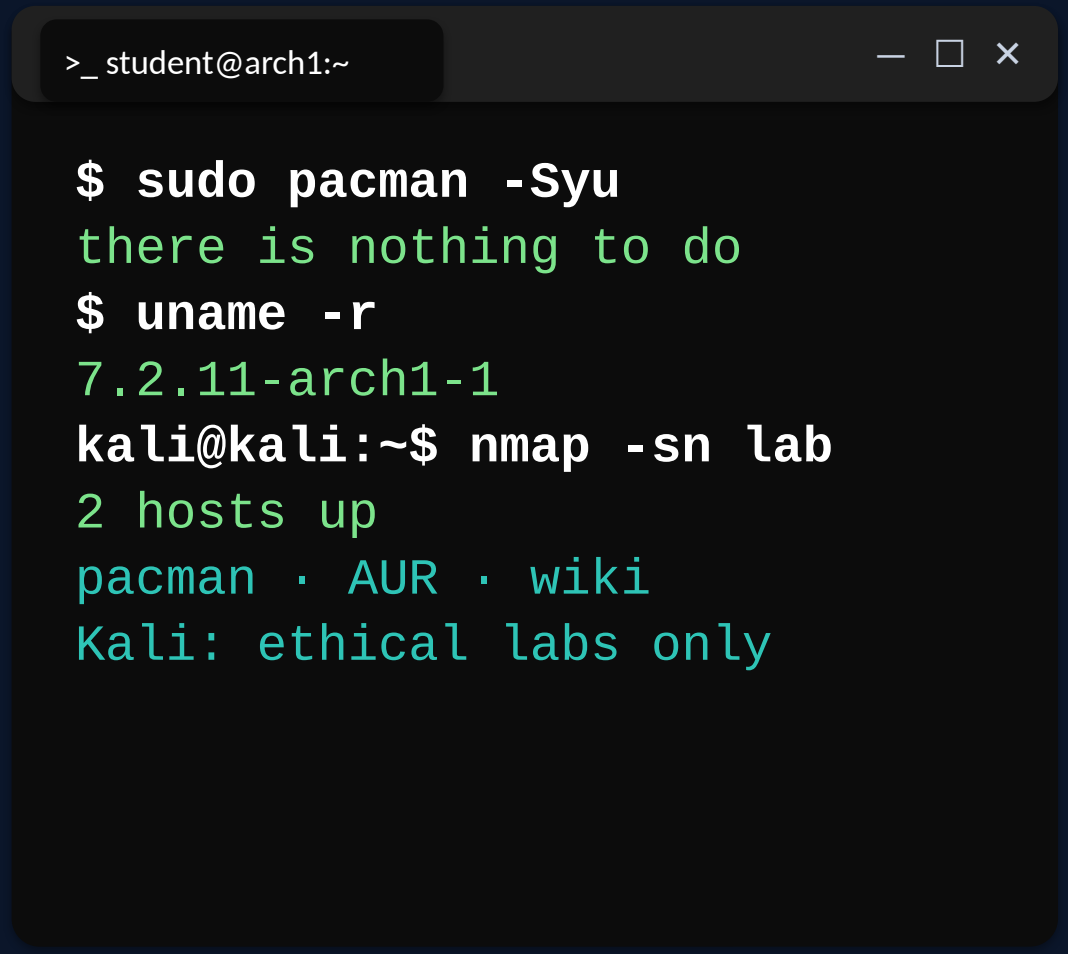
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 5 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

Running Arch Well: Updates, News, Snapshots & Desktops

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 5

- A weekly maintenance routine
- Arch news and manual interventions
- Automatic Btrfs snapshots with snapper + snap-pac
- Booting a snapshot and rolling back
- Desktops, display managers and Flatpak
- Security basics Arch leaves to you: firewall, updates, SSH
- Backups and keeping a package list

A Weekly Maintenance Routine

#	Step	Command
1	Read the news first	archlinux.org/news (or arch-announce mailing list / RSS)
2	Full upgrade	sudo pacman -Syu (or paru for repo + AUR)
3	Merge config changes	sudo pacdiff
4	Check for problems	systemctl --failed ; journalctl -b -p err
5	Clean up	paccache -rk2 ; pacman -Qdtq | sudo pacman -Rns -
6	Reboot after kernel/systemd/mesa	uname -r vs pacman -Q linux
Tip	Upgrade weekly, not monthly	Small steps are easier to fix than huge jumps

Arch News: Manual Interventions (examples from 2026)

Date	News item	Lesson
2026-10-21	Update php ≥ 8.5.12 / php-legacy ≥ 8.3.36 to prevent remote code execution	Security: upgrade now
2026-10-21	Update ca-certificates-mozilla ≥ 3.131.1-1 (compromised CA removed)	Security: upgrade now
2026-07-21	virtualbox-ext-vnc ≥ 7.2.12-2 requires manual intervention	Follow the steps
2026-06-12	Active AUR malicious packages incident	Review AUR packages
2026-05-25	varnish renamed to vinyl-cache — breaking change	Config paths change
How	archlinux.org/news, RSS, arch-announce list; tools like informant	Before every -Syu

Automatic Snapshots: snapper + snap-pac

```
[student@arch1 ~]$ findmnt -no FSTYPE /          # needs Btrfs root
btrfs
[student@arch1 ~]$ sudo pacman -S snapper snap-pac
[student@arch1 ~]$ sudo snapper -c root create-config /
[student@arch1 ~]$ sudo snapper -c root set-config TIMELINE_LIMIT_DAILY=7 \
      NUMBER_LIMIT=20
[student@arch1 ~]$ sudo systemctl enable --now snapper-timeline.timer \
      snapper-cleanup.timer
[student@arch1 ~]$ sudo pacman -S htop            # snap-pac makes pre/post snapshots
[student@arch1 ~]$ sudo snapper -c root list | tail -3
  42 | pre  |    | ... | pacman -S htop
  43 | post | 42 | ... | htop
```

Boot a Snapshot and Roll Back

```
# GRUB: sudo pacman -S grub-btrfs ; sudo systemctl enable --now grub-btrfsd
#   → 'Arch Linux snapshots' submenu in GRUB lets you boot a snapshot read-only
# systemd-boot / Limine: use limine-snapper-sync or boot from the ISO

# Undo only the changes of one pacman transaction (pre 42 → post 43):
[student@arch1 ~]$ sudo snapper -c root undochange 42..43

# Full rollback of / to snapshot 42 (then reboot):
[student@arch1 ~]$ sudo snapper -c root rollback 42
[student@arch1 ~]$ sudo reboot
# Note: /home is a separate subvolume — your files are not rolled back
```

Desktops, Display Managers and Flatpak

```
[student@arch1 ~]$ sudo pacman -S plasma-meta sddm konsole dolphin   # KDE Plasma 6.8
[student@arch1 ~]$ sudo systemctl enable sddm
# or GNOME:  sudo pacman -S gnome gdm && sudo systemctl enable gdm
[student@arch1 ~]$ sudo pacman -S pipewire pipewire-pulse wireplumber
[student@arch1 ~]$ sudo pacman -S noto-fonts noto-fonts-cjk noto-fonts-emoji \
      vazirmatn-fonts                    # Persian/Dari font
[student@arch1 ~]$ sudo pacman -S flatpak
[student@arch1 ~]$ flatpak install flathub org.libreoffice.LibreOffice
[student@arch1 ~]$ sudo pacman -S power-profiles-daemon bluez bluez-utils
[student@arch1 ~]$ sudo systemctl enable --now bluetooth
```

Security Basics Arch Leaves to You

```
# No firewall is enabled by default on Arch
[student@arch1 ~]$ sudo pacman -S ufw && sudo ufw default deny incoming
[student@arch1 ~]$ sudo ufw allow ssh && sudo systemctl enable --now ufw && sudo ufw enable
# SSH hardening
[student@arch1 ~]$ printf 'PermitRootLogin no\nPasswordAuthentication no\n' | \
      sudo tee /etc/ssh/sshd_config.d/10-hardening.conf
[student@arch1 ~]$ sudo systemctl restart sshd
# Security advisories
[student@arch1 ~]$ sudo pacman -S arch-audit && arch-audit   # known CVEs installed
# No automatic updates: you upgrade (weekly!) — never auto-upgrade a rolling system blindly
```

Backups and a Package List

```
[student@arch1 ~]$ pacman -Qqen > ~/pkglist-repo.txt      # explicit, official repos
[student@arch1 ~]$ pacman -Qqem > ~/pkglist-aur.txt       # explicit, AUR/foreign
# Reinstall on a new machine:
[student@arch1 ~]$ sudo pacman -S --needed - < pkglist-repo.txt
[student@arch1 ~]$ paru -S --needed - < pkglist-aur.txt

# Snapshots are NOT backups — copy data to another disk or server:
[student@arch1 ~]$ rsync -aHAX --delete ~/ /mnt/backup/student/
[student@arch1 ~]$ sudo pacman -S borg     # deduplicated, encrypted backups
[student@arch1 ~]$ borg init -e repokey /mnt/backup/borg && borg create /mnt/backup/borg::{now} ~/
```

Summary and Homework

- Weekly: news → pacman -Syu → pacdiff → check → clean → reboot if needed
- Read archlinux.org/news before upgrading (manual interventions)
- snapper + snap-pac: pre/post snapshots on every pacman run
- Roll back: grub-btrfs boot, snapper undochange / rollback
- Arch enables nothing by default: desktop, firewall, Bluetooth are your choice
- Homework: set up snapper on arch1; break and roll back a package; save your package lists

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

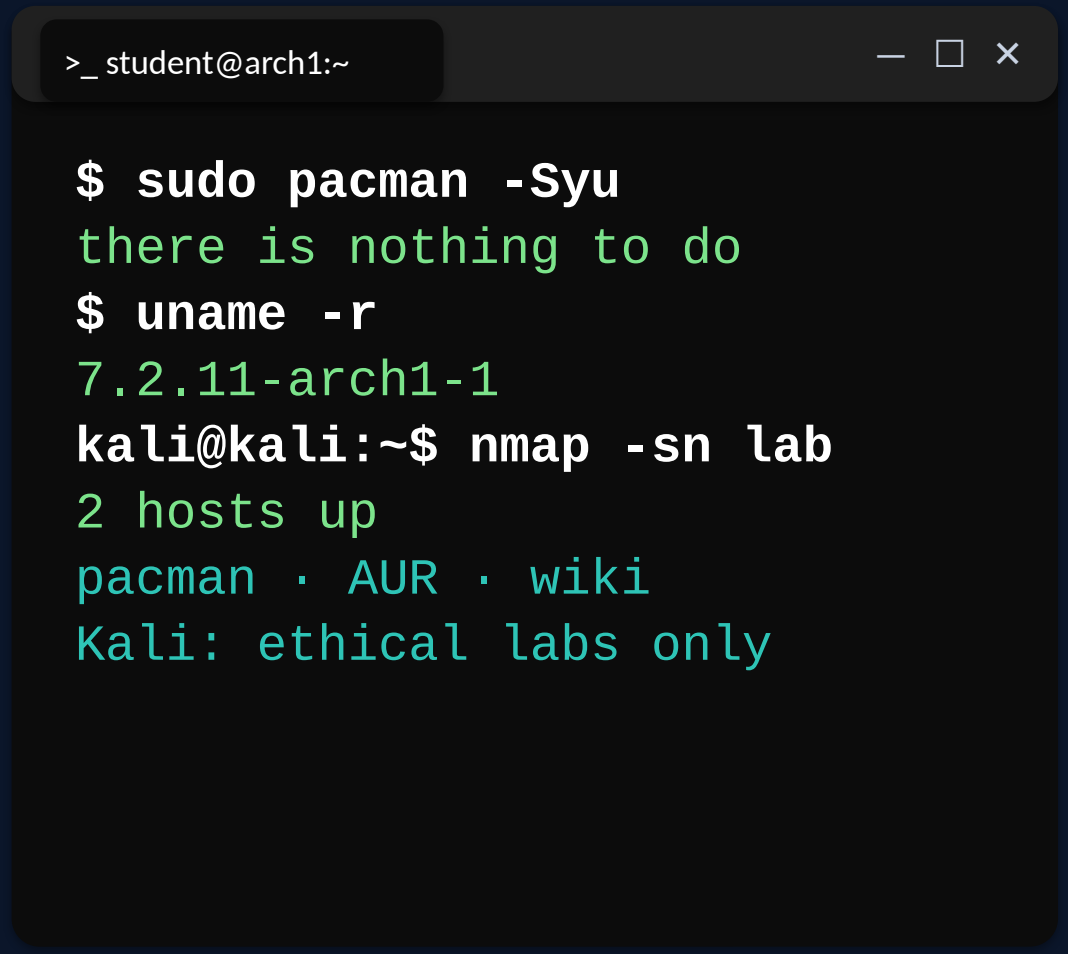
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 6 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

The Arch Family & Troubleshooting: EndeavourOS, CachyOS, SteamOS

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 6

- Arch-based distributions and how they differ
- EndeavourOS, CachyOS, Manjaro, SteamOS, Garuda
- Troubleshooting method for a rolling release
- Rescue with the ISO: mount, arch-chroot, fix
- Common breakages: kernel/initramfs, bootloader, keyring, GPU
- Getting help: Arch Wiki, forums, bug tracker
- Arch half review

The Arch Family

Distro	What it adds	Uses Arch repos?
EndeavourOS	Calamares installer, a few helpers; 'Arch with a friendly start'	Yes, directly (+ small own repo)
CachyOS	Performance: tuned kernels, x86-64-v3/v4 optimized packages	Rebuilt + own repos
Manjaro	Own delayed/tested repos, own tools (pamac, mhwd)	No — own branches
SteamOS 3	Valve's gaming OS for Steam Deck; immutable, A/B updates	Own snapshot of Arch
Garuda	Btrfs + snapshots by default, gaming tools, flashy desktops	Arch + Chaotic-AUR
BlackArch	2,800+ security tools as a repo on top of Arch	Add-on repository

A Troubleshooting Method for Rolling Releases

```
# 1. What changed? Look at the last pacman transactions
[student@arch1 ~]$ grep -E 'upgraded|installed' /var/log/pacman.log | tail -20
# 2. Is it already known? Check archlinux.org/news and the forum
# 3. Read the logs
[student@arch1 ~]$ journalctl -b -p err ; journalctl -b -1 -p err   # this / last boot
[student@arch1 ~]$ sudo dmesg -T --level=err,warn
# 4. Isolate: other kernel (linux-lts)? new user? snapshot from before?
# 5. Fix: downgrade from cache, roll back snapshot, or apply the news steps
[student@arch1 ~]$ sudo pacman -U /var/cache/pacman/pkg/<pkg-previous-version>.pkg.tar.zst
# 6. Report upstream / Arch bug tracker if it is a real bug
```

Rescue with the ISO and arch-chroot

```
# Boot the Arch ISO (same or newer than the installed system)
root@archiso ~ # lsblk -f
root@archiso ~ # cryptsetup open /dev/vda2 root      # only if LUKS
root@archiso ~ # mount -o subvol=@ /dev/vda2 /mnt     # Btrfs root subvolume
root@archiso ~ # mount /dev/vda1 /mnt/boot
root@archiso ~ # arch-chroot /mnt
[root@archiso /]# pacman -Syu                          # finish an interrupted upgrade
[root@archiso /]# mkinitcpio -P                        # rebuild initramfs
[root@archiso /]# bootctl update   (or grub-mkconfig -o /boot/grub/grub.cfg)
[root@archiso /]# exit ; umount -R /mnt ; reboot
```

Common Breakages and Fixes

Symptom	Likely cause	Fix
Kernel panic / no init after update	Interrupted upgrade, broken initramfs	ISO → arch-chroot → pacman -Syu → mkinitcpio -P
'unknown filesystem' / modules missing	Booted kernel ≠ installed modules	Reboot after kernel upgrade; keep linux-lts
No boot menu / bootloader gone	Firmware update reset EFI entries	arch-chroot → bootctl install / grub-install
Invalid or corrupted package (PGP)	Old keyring	pacman -Sy archlinux-keyring && pacman -Su
Black screen after GPU driver update	NVIDIA/Mesa change (e.g. 590 dropped Pascal)	TTY (Ctrl+Alt+F3), check news, nvidia-580xx (AUR) or downgrade
Disk full	Package cache, journals, snapshots	paccache -rk1; journalctl --vacuum-size; snapper delete

Getting Help the Arch Way

Where	How to ask well
Arch Wiki	Search first — most answers are already there (and work on other distros too)
Forums (bbs.archlinux.org)	Read the forum rules; show what you tried; paste outputs, not screenshots
Share logs	journalctl -b | curl -F 'file=@-' https://0x0.st (or paste.rs)
Bug tracker	gitlab.archlinux.org — packaging bugs; upstream for program bugs
Matrix / IRC	#archlinux on Libera.Chat for quick questions
Derivatives	EndeavourOS, CachyOS and Manjaro have their own forums — ask there first

Arch Half Review

Lesson	Key skills
1	Rolling vs fixed releases; the Arch Way; where Arch and Kali fit; ethics
2	Verify the ISO; archinstall (Btrfs, systemd-boot, profile); manual install steps
3	pacman -S/-R/-Q/-F/-U; full upgrades only; reflector; keyring; pacdiff; cache
4	AUR: read PKGBUILDs; makepkg -si; yay/paru; malware awareness
5	Weekly routine; Arch news; snapper + snap-pac; desktops; firewall; backups
6	Arch family; troubleshooting method; ISO rescue with arch-chroot; getting help

Summary and Homework

- EndeavourOS ≈ Arch + installer; CachyOS = performance; Manjaro = own repos
- SteamOS 3 is Arch-based; BlackArch adds security tools
- Method: pacman.log → news → logs → isolate → fix → report
- Rescue: ISO → mount → arch-chroot → pacman -Syu / mkinitcpio -P / bootloader
- Ask well: Wiki first, outputs not screenshots, derivative forums for derivatives
- Homework: break arch1's initramfs in a snapshot-protected VM and rescue it from the ISO

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

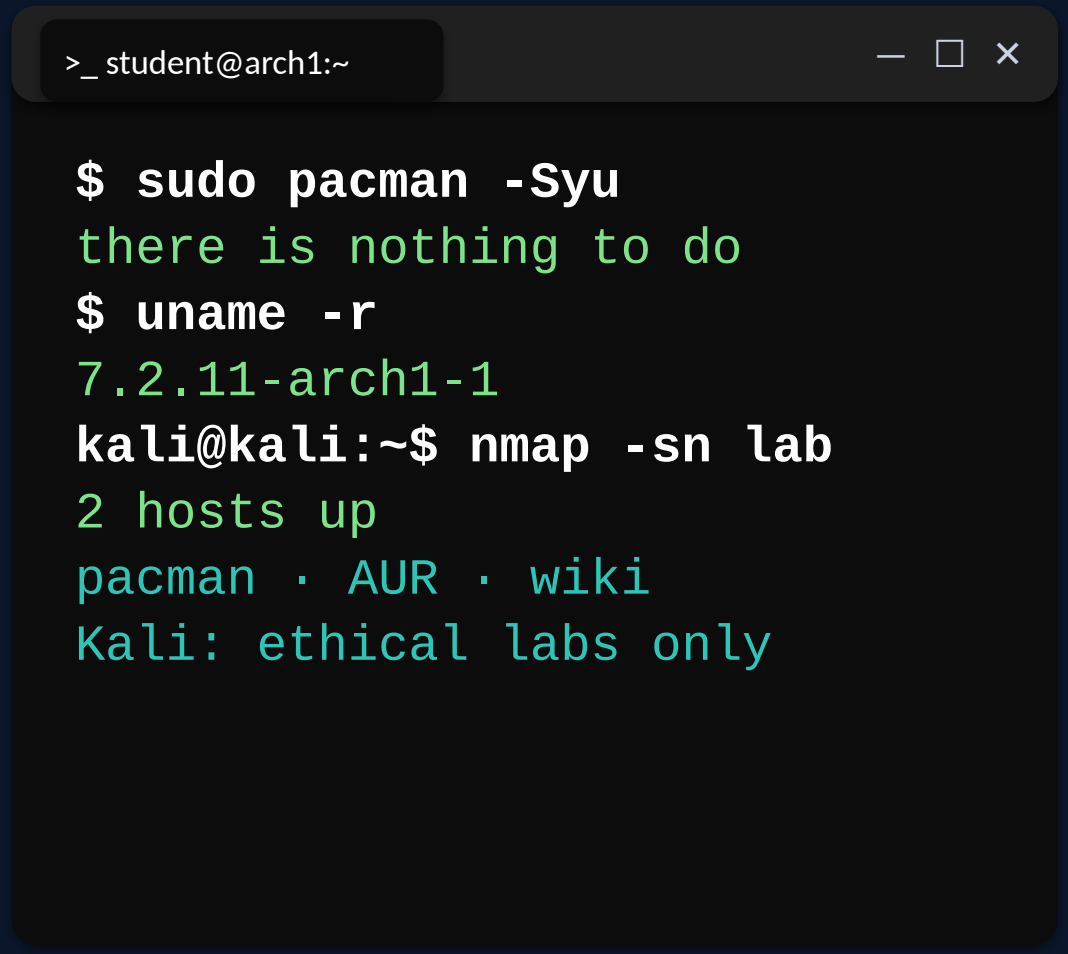
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 7 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

Kali Linux: Purpose, Ethics, Law & Editions

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 7

- What Kali is for — and what it is not for
- A short history: BackTrack to Kali 2026.3
- The law: unauthorized access is a crime
- How professionals work: authorization, scope, rules of engagement
- Legal ways to practice: labs, CTFs, bug bounty programs
- Kali editions: installer, live, VM, WSL, cloud, ARM, NetHunter, Purple
- Choosing the right edition for your lab

What Kali Is For — and Not For

Kali is for	Kali is not
Authorized penetration tests and vulnerability assessments	A daily desktop or a beginner's first Linux
Security learning in your own lab, CTFs, certifications	A tool to access other people's accounts or devices
Digital forensics (forensic boot mode doesn't touch disks)	Anonymous by default (it is not Tails)
Defensive work: Kali Purple (SOC, IDS, SIEM)	A replacement for patching and good defenses
Researchers, students, red, blue and purple teams	'Hacking' tools that work without skill and permission

A Short History of Kali

Year	Milestone
2006–2012	BackTrack (based on Ubuntu/Slackware) — the predecessor
2013	Kali Linux 1.0 by Offensive Security, rebuilt on Debian
2016	Kali Rolling: continuous updates from Debian testing
2019.4	Xfce becomes the default desktop; 'Undercover' mode
2020.1	Non-root default user (kali); single installer image
2022–2023	Kali Purple (defensive) introduced; NetHunter expands
2026.3	Oct 2026: Linux 7.1, refreshed build scripts, 6 new tools, desktop updates

The Law: Unauthorized Access Is a Crime

Place	Example law
United States	Computer Fraud and Abuse Act (CFAA); state computer-crime laws
United Kingdom	Computer Misuse Act 1990
European Union	Directive on attacks against information systems; national laws
Afghanistan & region	National penal and cybercrime laws — check your own country
Everywhere	Scanning, guessing passwords or intercepting traffic WITHOUT permission can be illegal
Note	This is general information, not legal advice — ask a lawyer for your situation

How Professionals Work

Element	What it contains
Written authorization	Signed by the system OWNER (not just an employee); who, what, when
Scope	Exact IP ranges, domains, apps in scope — and what is out of scope
Rules of engagement	Allowed techniques, testing hours, no-DoS, data handling, stop conditions
Contacts	Emergency contact if something breaks; how to report critical findings now
Data handling	Encrypt evidence, minimize personal data, delete after the engagement
Report	Findings, risk, evidence, and how to fix — the real product of a test

Legal Ways to Practice

Option	Examples
Your own lab	Kali + intentionally vulnerable VMs on a host-only network (lesson 11)
Vulnerable apps	DVWA, OWASP Juice Shop, WebGoat, Metasploitable 2/3
Online labs	TryHackMe, Hack The Box, PortSwigger Web Security Academy, OffSec Proving Grounds
CTF competitions	picoCTF (beginners), CTFtime.org calendar, school and university CTFs
Bug bounty programs	HackerOne, Bugcrowd, Intigriti — ONLY inside each program's published scope
Free training	MyWebUniversity, OWASP, SANS Cyber Aces, vendor docs

Kali Editions

Edition	Best for
Installer (+ NetInstaller)	Bare metal or a custom VM install; choose desktop and tool sets
Pre-built VM images	VirtualBox, VMware, QEMU/KVM, Hyper-V — fastest start (user/pass: kali/kali)
Live USB (+ persistence)	Carry Kali; forensic mode; optional encrypted persistence
WSL (Windows)	Kali inside Windows; Win-KeX for a desktop
Cloud & containers	AWS/Azure images; docker.io/kalilinux/kali-rolling
ARM	Raspberry Pi and other boards
NetHunter	Android phones (lesson 12)
Kali Purple	Defensive / SOC lab edition (lesson 11)

Summary and Homework

- Kali: authorized testing, learning, forensics, defense — not a daily desktop
- BackTrack → Kali (2013) → rolling (2016) → non-root (2020) → 2026.3
- Unauthorized access is a crime (CFAA, Computer Misuse Act, local laws)
- Professionals: written authorization, scope, rules of engagement, report
- Practice legally: own lab, DVWA/Juice Shop, TryHackMe/HTB, CTFs, bug bounties in scope
- Homework: read one bug bounty program's scope page; download the Kali VM image for your hypervisor

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

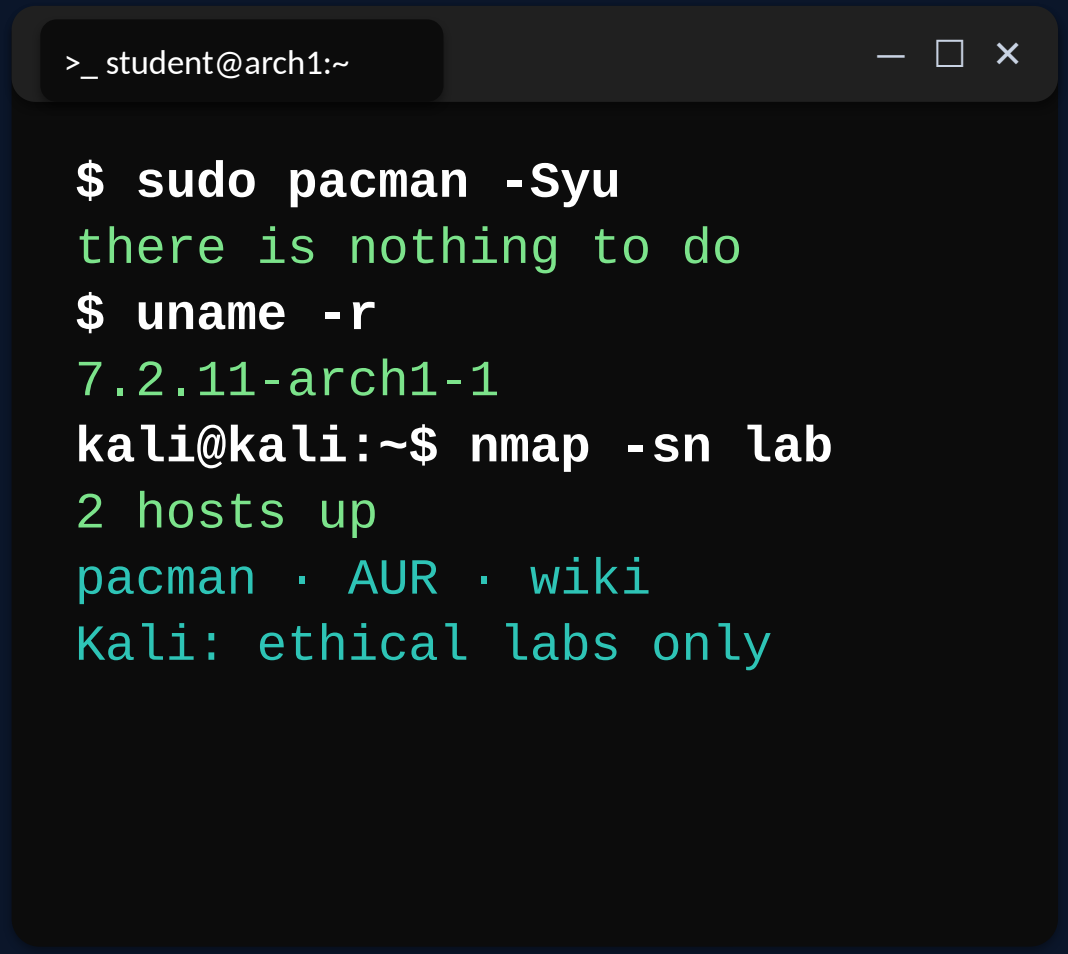
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 8 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

Installing Kali: VM, Bare Metal, Live USB & WSL

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 8

- Why a VM is best for learning
- Download and verify Kali images
- Import the pre-built VM; first-login security steps
- VM networking: NAT + host-only lab network
- The installer: desktop and tool selection
- Live USB with encrypted persistence
- Kali on WSL and in Docker

Where to Run Kali for Learning

Option	Pros	Cons
VM (recommended)	Isolated, snapshots, host-only lab network, easy to reset	Needs RAM (4 GB+ for Kali)
Live USB	Portable, no install, forensic mode	Slower; persistence must be set up
Bare metal	Full hardware access (Wi-Fi cards, GPU cracking)	No snapshots; mixes with daily use
WSL	Quick CLI tools on Windows	Limited raw network access; not for Wi-Fi testing
Cloud	Fast internet, temporary	Costs money; providers have testing policies

Download and Verify

```
# kali.org/get-kali — choose: Virtual Machines / Installer / Live / WSL / ARM
$ ls
kali-linux-2026.3-virtualbox-amd64.7z   SHA256SUMS   SHA256SUMS.gpg
$ wget -q -O - https://archive.kali.org/archive-key.asc | gpg --import
$ gpg --verify SHA256SUMS.gpg SHA256SUMS
gpg: Good signature from "Kali Linux Archive Automatic Signing Key ..."
$ sha256sum -c SHA256SUMS --ignore-missing
kali-linux-2026.3-virtualbox-amd64.7z: OK
$ 7z x kali-linux-2026.3-virtualbox-amd64.7z
# Download only from kali.org (or its official mirrors / torrents)
```

Import the VM and First Login

```
# VirtualBox: Machine > Add > kali-linux-2026.3-virtualbox-amd64.vbox
# VMware: open the .vmx ; QEMU/KVM: import the .qcow2 in virt-manager
# Login: kali / kali  — change it NOW
kali@kali:~$ passwd
kali@kali:~$ sudo apt update && sudo apt full-upgrade -y
kali@kali:~$ sudo dpkg-reconfigure keyboard-configuration   # if needed
kali@kali:~$ sudo timedatectl set-timezone Asia/Kabul
kali@kali:~$ sudo systemctl disable --now ssh 2>/dev/null   # off unless needed
kali@kali:~$ cat /etc/os-release | grep VERSION
VERSION="2026.3"
# Take a VM snapshot: 'clean-2026.3'
```

VM Networking for a Safe Lab

Adapter	Type	Purpose
Adapter 1	NAT	Internet for updates (can be disabled during practice)
Adapter 2	Host-only 192.168.56.0/24	Lab network: Kali + intentionally vulnerable targets
Targets	Host-only ONLY	Vulnerable VMs must never reach your home network or the internet
Avoid	Bridged	Puts Kali (and targets) on your real LAN — not for practice
Check	ip -br a ; ip route	eth0 = NAT (10.0.2.x), eth1 = lab (192.168.56.10)
Snapshots	Before each exercise	Restore after — keeps the lab clean

The Kali Installer

Screen	Choice
Install type	Graphical install (Debian Installer based)
User	Your own username and strong password (no root login)
Partitioning	Guided — use entire disk; encrypted LVM recommended on laptops
Desktop	Xfce (default), GNOME, KDE Plasma, or none
Tools	top10, default, large, everything (metapackages, lesson 9)
Bootloader	GRUB on the main disk
Tip	NetInstaller downloads current packages during install

Live USB with Encrypted Persistence

```
# Write the live ISO (destroys the USB stick!)
$ sudo dd if=kali-linux-2026.3-live-amd64.iso of=/dev/sdX bs=4M status=progress conv=fsync
# Create a persistence partition in the free space
$ sudo fdisk /dev/sdX            # n, p, 3, <enter>, <enter>, w
$ sudo cryptsetup luksFormat /dev/sdX3
$ sudo cryptsetup open /dev/sdX3 my_usb
$ sudo mkfs.ext4 -L persistence /dev/mapper/my_usb
$ sudo mkdir -p /mnt/my_usb && sudo mount /dev/mapper/my_usb /mnt/my_usb
$ echo '/ union' | sudo tee /mnt/my_usb/persistence.conf
$ sudo umount /mnt/my_usb && sudo cryptsetup close my_usb
# Boot menu: 'Live USB Encrypted Persistence'  |  'Live (forensic mode)'
```

Kali on WSL and in Docker

```
PS C:\> wsl --install -d kali-linux
PS C:\> wsl -d kali-linux
kali@DESKTOP:~$ sudo apt update && sudo apt full-upgrade -y
kali@DESKTOP:~$ sudo apt install -y kali-linux-default     # tools (large download)
kali@DESKTOP:~$ sudo apt install -y kali-win-kex && kex --win -s   # desktop

$ docker pull docker.io/kalilinux/kali-rolling
$ docker run --rm -it kalilinux/kali-rolling /bin/bash
root@1a2b3c:/# apt update && apt install -y kali-linux-headless
# Containers start minimal — install only the tools you need
```

Summary and Homework

- VM is best for learning: isolated, snapshots, host-only lab
- Download from kali.org only; verify SHA256SUMS + gpg
- First login: change kali/kali, full-upgrade, snapshot
- Two adapters: NAT (updates) + host-only (lab); targets host-only only
- Installer: Xfce/GNOME/KDE + tool sets; live USB with encrypted persistence; WSL; Docker
- Homework: import the Kali VM, change password, set up host-only network, take 'clean' snapshot

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

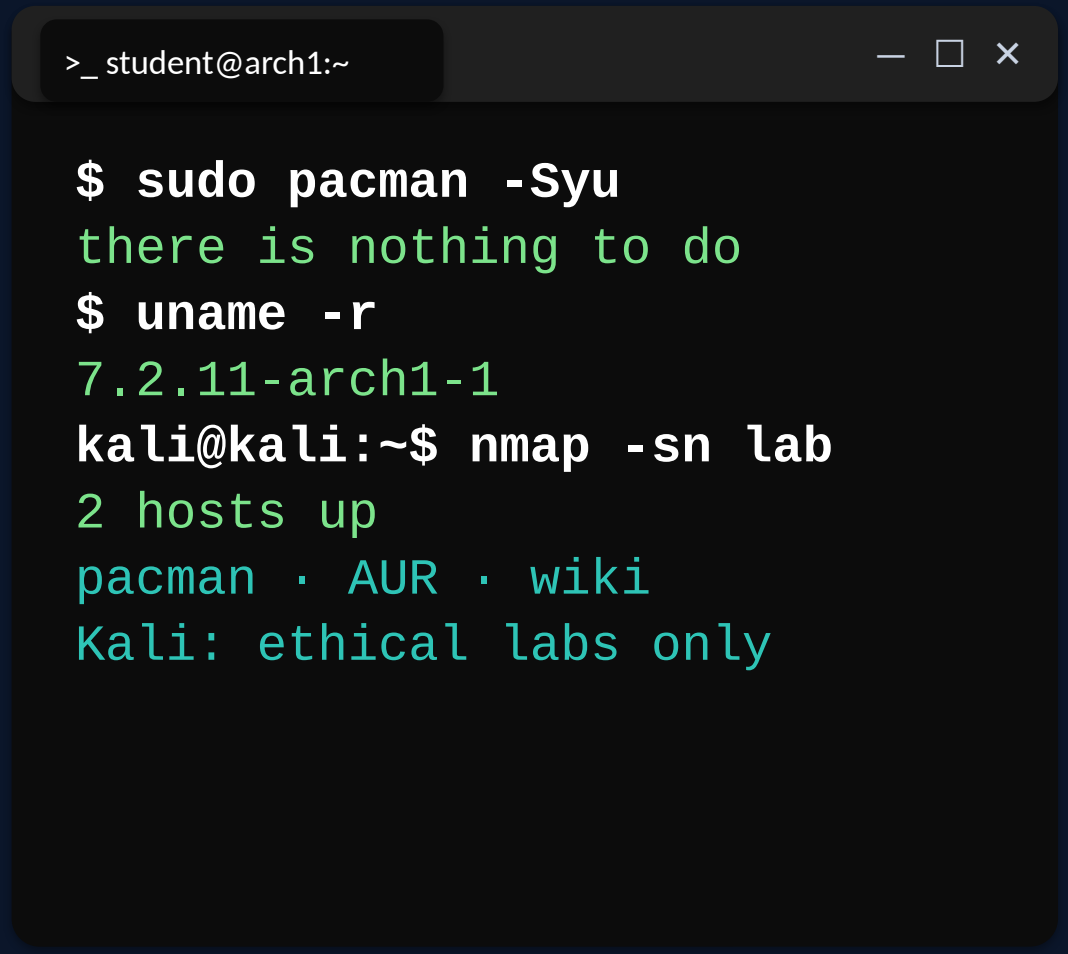
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 9 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

Kali Essentials: kali-rolling, Metapackages & kali-tweaks

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 9

- Kali repositories and branches: kali-rolling, last-snapshot, experimental
- Updating Kali safely
- Metapackages: core, headless, default, large, everything
- Tool groups: kali-tools-web, -forensics, -wireless …
- kali-tweaks: shell, metapackages, repos, virtualization, hardening
- Kali's network-service policy and the non-root user
- Finding documentation for every tool

Kali Repositories and Branches

```
kali@kali:~$ cat /etc/apt/sources.list
deb http://http.kali.org/kali kali-rolling main contrib non-free non-free-firmware
# kali-rolling        : default, continuously updated from Debian testing
# kali-last-snapshot  : the last quarterly release (2026.3) — more stable
# kali-experimental   : packages under test (not for daily use)
# kali-bleeding-edge  : built from upstream git (for testers)
kali@kali:~$ apt policy | grep -E 'kali-(rolling|last)' | head -3
# NEVER add Debian/Ubuntu repos to Kali — it breaks the system ("FrankenDebian")
```

Updating Kali Safely

```
kali@kali:~$ sudo apt update
kali@kali:~$ apt list --upgradable | wc -l
kali@kali:~$ sudo apt full-upgrade -y           # Kali needs full-upgrade, not upgrade
kali@kali:~$ sudo apt autoremove --purge
kali@kali:~$ [ -f /var/run/reboot-required ] && echo 'reboot needed'
kali@kali:~$ grep VERSION= /etc/os-release
# VM? Take a snapshot before big upgrades; old VM (months)? update keyring first:
kali@kali:~$ sudo wget https://archive.kali.org/archive-keyring.gpg \
      -O /usr/share/keyrings/kali-archive-keyring.gpg
```

Metapackages: How Much Kali Do You Want?

Metapackage	What you get	Size
kali-linux-core	Base Kali system, no tools	Small
kali-linux-headless	Default tools without a GUI (servers, WSL, containers)	Medium
kali-linux-default	The standard set on installer/VM images	~ 10 GB+
kali-linux-large	Default + many more tools	Large
kali-linux-everything	Every packaged tool	Very large
kali-tools-top10	The ten most popular tools	Small

Tool Groups (kali-tools-*)

```
kali@kali:~$ apt-cache search --names-only '^kali-tools-' | sort
kali-tools-crypto-stego         kali-tools-information-gathering
kali-tools-database             kali-tools-passwords
kali-tools-forensics            kali-tools-reverse-engineering
kali-tools-fuzzing              kali-tools-sniffing-spoofing
kali-tools-hardware             kali-tools-social-engineering
kali-tools-identify             kali-tools-vulnerability
kali-tools-protect / respond / detect / recover   (Kali Purple, defensive)
kali-tools-web                  kali-tools-wireless
kali@kali:~$ apt show kali-tools-forensics | grep Depends | tr ',' '\n' | head
kali@kali:~$ sudo apt install -y kali-tools-forensics
```

kali-tweaks

Menu	What it does
Metapackages	Install/remove tool sets with checkboxes
Network Repositories	Switch mirror / enable kali-experimental or bleeding-edge (carefully)
Shell & Prompt	Zsh (default) or Bash; one-line or two-line prompt
Virtualization	Install guest tools (VirtualBox, VMware, QEMU/Spice)
Hardening	Strong SSH defaults, wide compatibility mode for old protocols (lab use)
Kernel / misc	Kernel settings, e.g. privileged ports for non-root users

Services, the Non-root User and Undercover

```
# Kali policy: network services are DISABLED by default and don't start at boot
kali@kali:~$ systemctl is-enabled ssh postgresql apache2 2>/dev/null
disabled
kali@kali:~$ sudo systemctl start postgresql        # start only when needed
kali@kali:~$ sudo systemctl enable --now ssh        # only if you really need it
# Non-root by default since 2020.1 — use sudo for admin tasks
kali@kali:~$ id ; sudo -l | tail -2
kali@kali:~$ sudo apt install -y kali-root-login     # not recommended
# Undercover mode: Windows-like look in public places
kali@kali:~$ kali-undercover
```

Documentation for Every Tool

Source	How
man pages	man nmap ; man -k password
Built-in help	tool --help or tool -h
kali.org/tools	Every packaged tool: description, packages, commands, examples
kali.org/docs	Installation, troubleshooting, policies, NetHunter, Purple
/usr/share/doc/<pkg>	README, examples and changelogs on your system
Upstream projects	Official docs (e.g. nmap.org/book, OWASP guides)

Summary and Homework

- kali-rolling (default) or kali-last-snapshot; never add Debian/Ubuntu repos
- Update with apt full-upgrade; snapshot first; refresh keyring on old VMs
- Metapackages: core < headless < default < large < everything; top10
- kali-tools-* groups by task; Purple groups for defense
- kali-tweaks: metapackages, repos, shell, virtualization, hardening
- Homework: run kali-tweaks; install kali-tools-forensics; read kali.org/tools for 3 tools

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

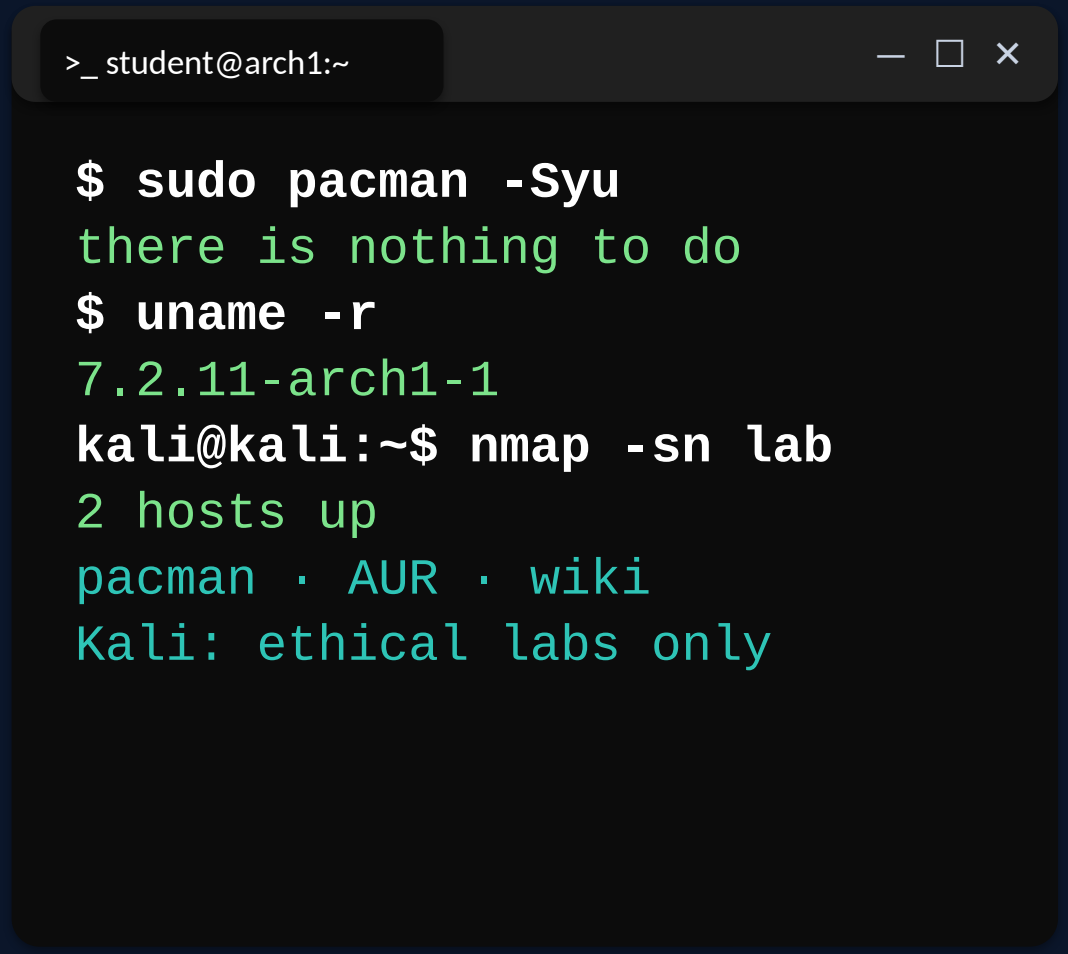
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 10 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

A Tour of Kali Tools: Recon, Web, Passwords, Wireless & Forensics

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 10

- The phases of an authorized assessment
- Reconnaissance: Nmap on your lab network
- Web testing basics: Burp Suite, Nikto, Gobuster on DVWA
- Passwords: why strong hashes and long passwords matter
- Traffic analysis with Wireshark
- Wireless: what Kali can test — and the legal limits
- Forensics: Autopsy and the forensic boot mode

Phases of an Authorized Assessment

Phase	Goal	Example tools
0. Authorization	Written permission, scope, rules	Documents — not software
1. Reconnaissance	Find hosts and services in scope	nmap, whatweb, theHarvester (OSINT)
2. Vulnerability analysis	Find weaknesses	nikto, nuclei, OpenVAS/GVM
3. Verification	Prove risk safely, within the rules	Burp Suite, Metasploit (lab)
4. Reporting	Explain findings and fixes	Notes, screenshots, report template
Defender view	Same steps show what to patch and monitor	Kali Purple (lesson 11)

Reconnaissance: Nmap (lab only)

```
kali@kali:~$ ip -br a | grep eth1            # confirm you are on the lab network
eth1  UP  192.168.56.10/24
kali@kali:~$ nmap -sn 192.168.56.0/24        # which lab hosts are up?
Nmap scan report for 192.168.56.20  Host is up.
Nmap scan report for 192.168.56.30  Host is up.
kali@kali:~$ nmap -sV -T3 192.168.56.20      # services and versions on target1
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
22/tcp open  ssh     OpenSSH 4.7p1
80/tcp open  http    Apache httpd 2.2.8
kali@kali:~$ nmap -sV -oA ~/lab/target1 192.168.56.20   # save results for the report
```

Web Testing Basics (on DVWA / Juice Shop)

Tool	What it does	Defender lesson
Burp Suite (Community)	Proxy to view and modify browser requests	Never trust input from the client
whatweb	Identify web technologies and versions	Hide/patch version banners
nikto	Check for common web server misconfigurations	Remove default files, harden configs
gobuster / ffuf	Find hidden directories in your lab app	Don't rely on 'secret' URLs
sqlmap	Test SQL injection in the lab	Use parameterized queries
OWASP Top 10	The main classes of web risks	Secure coding guidance

Passwords and Hashes (your own test hashes)

```
# Create YOUR OWN test hashes to learn why password strength matters
kali@kali:~$ echo -n 'summer2026' | sha256sum
kali@kali:~$ mkpasswd -m yescrypt 'summer2026'     # modern Linux hash (slow on purpose)
$y$j9T$...
kali@kali:~$ hashid '<hash>'                      # identify a hash type
kali@kali:~$ ls /usr/share/wordlists/              # rockyou.txt.gz etc.
# Lesson for defenders:
#   short/common passwords fall to wordlists in seconds
#   long passphrases + slow hashes (yescrypt, bcrypt, Argon2) + MFA resist attacks
#   never store plain or fast-hashed (MD5/SHA1) passwords
```

Traffic Analysis with Wireshark

```
kali@kali:~$ sudo usermod -aG wireshark kali     # capture without root (log in again)
kali@kali:~$ wireshark &                          # GUI: capture on eth1 (lab)
kali@kali:~$ tshark -i eth1 -c 20 -f 'port 21'    # CLI capture of lab FTP
# Display filters:
#   ip.addr == 192.168.56.20      http.request      ftp      tls.handshake
# Lesson: FTP and plain HTTP send passwords in clear text — use SFTP/HTTPS
# Capture only on networks you own or are authorized to monitor
```

Wireless: What Kali Can Test — and the Limits

Topic	Details
Hardware	USB Wi-Fi adapter with monitor mode support (VMs need USB passthrough)
Tools	aircrack-ng suite, kismet, wifite, bettercap
What it shows	Weak WPA2 passphrases, WPS flaws, rogue access points
Defender lesson	WPA3 or WPA2 with a long passphrase; disable WPS; update router firmware
LEGAL LIMIT	Only YOUR OWN access point or written authorization — never neighbors, cafés, schools
Also illegal	Jamming or deauthenticating others' devices

Forensics: Autopsy and Forensic Mode

```
# Boot Kali live in 'forensic mode': no auto-mount, no swap, no disk writes
kali@kali:~$ sudo apt install -y kali-tools-forensics
kali@kali:~$ sudo dc3dd if=/dev/sdb hash=sha256 of=/cases/usb.img log=/cases/usb.log
kali@kali:~$ sha256sum /cases/usb.img            # verify the image matches the log
kali@kali:~$ autopsy                              # web UI: new case, add usb.img
kali@kali:~$ binwalk firmware.bin ; exiftool photo.jpg
# Work on the IMAGE, never the original evidence; keep a chain-of-custody record
```

Summary and Homework

- Phases: authorization, recon, analysis, verification, report
- nmap -sn / -sV on 192.168.56.0/24 only; save with -oA
- Web: Burp, whatweb, nikto, gobuster, sqlmap on DVWA/Juice Shop
- Passwords: long passphrases + slow hashes + MFA; Wireshark shows clear-text risks
- Wireless & capture only on your own networks; forensics on images
- Homework: scan target1, list its services, and write 3 'how to fix' recommendations

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

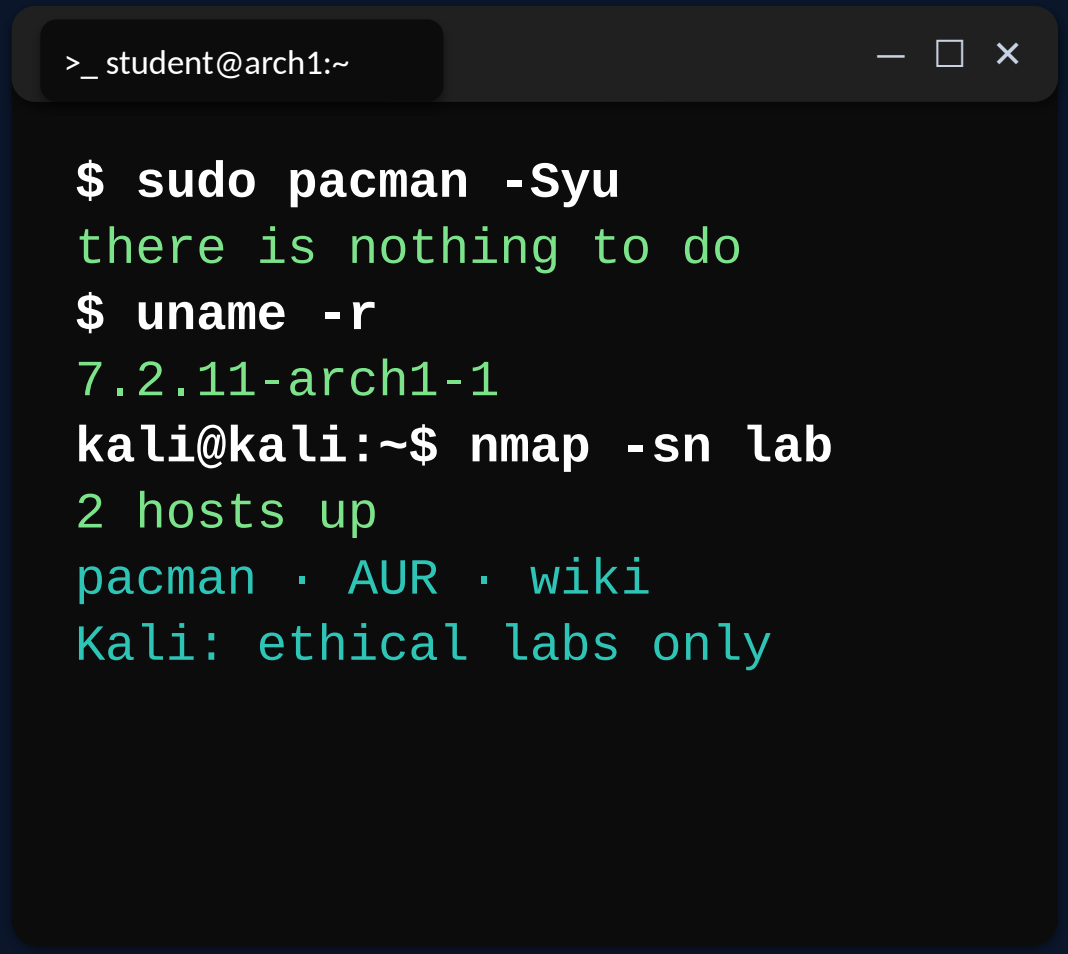
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 11 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

Building a Safe Practice Lab & Defending with Kali Purple

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 11

- Lab design: isolation, targets, snapshots
- Building the host-only network
- Practice targets: Metasploitable, DVWA, Juice Shop in containers
- Lab safety rules and a lab notebook
- Kali Purple: the defensive edition
- Detecting lab activity with Suricata and logs
- Purple teaming: attack, detect, fix, repeat

Lab Design

Part	Our choice	Why
Hypervisor	VirtualBox or KVM/virt-manager	Free, snapshots, internal networks
Network	Host-only 192.168.56.0/24, no DHCP to the internet	Targets can't reach your LAN or the internet
Attacker	kali (192.168.56.10) + NAT for updates	Disable NAT during exercises if you like
Targets	target1 Metasploitable 2 (.20), target2 containers (.30)	Intentionally vulnerable — never expose
Defender	Kali Purple VM (.40) or Suricata on target2	See your own traffic from the blue side
Snapshots	'clean' on every VM	Reset after each exercise

Build the Host-only Network

```
# VirtualBox
$ VBoxManage hostonlyif create                          # -> vboxnet0
$ VBoxManage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1 --netmask 255.255.255.0
$ VBoxManage modifyvm target1 --nic1 hostonly --hostonlyadapter1 vboxnet0
$ VBoxManage modifyvm kali --nic2 hostonly --hostonlyadapter2 vboxnet0

# KVM / libvirt: an isolated network (no <forward>)
$ cat lab.xml
<network><name>lab</name><ip address='192.168.56.1' netmask='255.255.255.0'/></network>
$ sudo virsh net-define lab.xml && sudo virsh net-start lab && sudo virsh net-autostart lab
# Verify from target1: ping 8.8.8.8 must FAIL
```

Practice Targets

```
# target1: Metasploitable 2 (download from its official SourceForge page)
#   Import the VMDK, host-only NIC ONLY, login msfadmin/msfadmin (lab only!)

# target2: vulnerable web apps in containers (on an isolated VM)
student@target2:~$ docker run -d --name dvwa -p 8080:80 vulnerables/web-dvwa
student@target2:~$ docker run -d --name juice -p 3000:3000 bkimminich/juice-shop
student@target2:~$ docker ps --format '{{.Names}} {{.Ports}}'
dvwa   0.0.0.0:8080->80/tcp
juice  0.0.0.0:3000->3000/tcp
# Browse from Kali: http://192.168.56.30:8080  and  http://192.168.56.30:3000
```

Lab Safety Rules

Rule	Practice
Targets never on the internet	Host-only only; verify with a ping test
No real data in the lab	Fake users, fake passwords, fake files
Snapshot before, restore after	Every exercise starts clean
Keep a lab notebook	Date, goal, commands, results, what you learned
Malware samples only in isolated VMs	No shared folders or clipboard to the host
Update Kali, not the targets	Targets stay vulnerable on purpose — and isolated

Kali Purple: The Defensive Edition

Area (NIST CSF)	Example tools in Kali Purple
Identify	Asset and vulnerability discovery (nmap, GVM/OpenVAS)
Protect	Hardening, firewalls, secure configurations
Detect	Suricata and Zeek (network IDS), Elastic/OpenSearch SIEM, Arkime
Respond	TheHive (case management), Cortex, playbooks
Recover	Backups, restore and lessons learned
Install	Kali Purple ISO, or: sudo apt install kali-linux-purple / kali-tools-detect

Detect Your Own Lab Activity with Suricata

```
purple@purple:~$ sudo apt install -y suricata
purple@purple:~$ sudo vim /etc/suricata/suricata.yaml    # af-packet: interface: eth1
purple@purple:~$ sudo suricata-update                   # download open rule sets
purple@purple:~$ sudo systemctl enable --now suricata
# From kali: nmap -sV 192.168.56.20   (your own lab scan)
purple@purple:~$ sudo tail -f /var/log/suricata/fast.log
[**] ET SCAN Suspicious inbound to mySQL port 3306 [**] 192.168.56.10 -> .20
purple@purple:~$ jq 'select(.event_type=="alert") | .alert.signature' \
      /var/log/suricata/eve.json | sort | uniq -c | sort -rn | head
```

Purple Teaming: Attack, Detect, Fix, Repeat

Step	Example in our lab
1. Attack (red)	Scan target1 with nmap -sV; find the old FTP service
2. Detect (blue)	Did Suricata alert? Does target1's log show the connections?
3. Improve detection	Tune rules, add log forwarding to the SIEM
4. Fix	On a copy of target1: disable/upgrade the service, add a firewall rule
5. Repeat	Re-run the scan: service gone, alert still fires
6. Report	Write what you found, what you saw, and what you fixed

Summary and Homework

- Host-only, isolated network; verify targets cannot reach the internet
- Targets: Metasploitable 2, DVWA, Juice Shop — deliberately vulnerable, never exposed
- Safety: no real data, snapshots, lab notebook, malware only in isolated VMs
- Kali Purple: NIST CSF — identify, protect, detect, respond, recover
- Suricata detects your own lab scans; purple teaming = attack, detect, fix, repeat
- Homework: build the lab, verify isolation, run one purple-team cycle and write a 1-page report

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

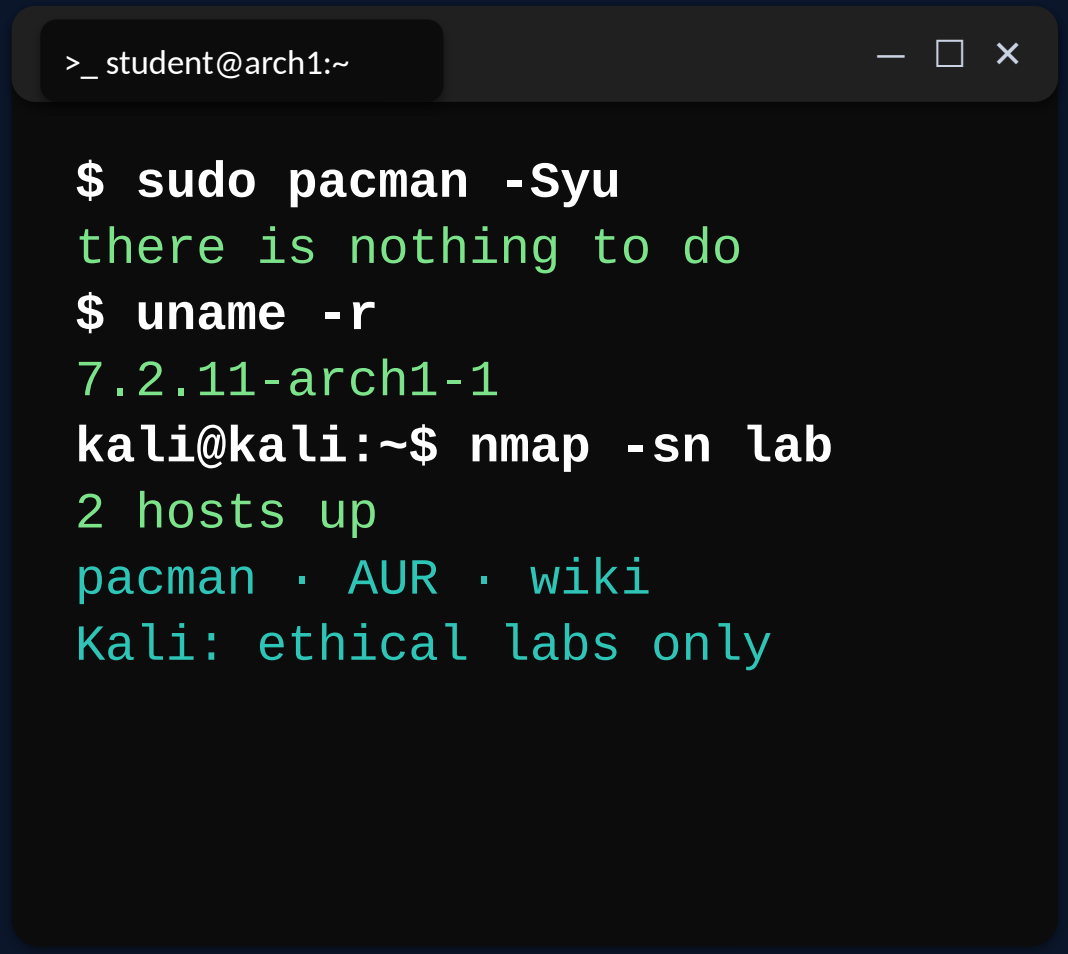
Slides for this lesson: MyWebUniversity.com/presentations

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.



LESSON 12 OF 12

Arch & Kali Linux

Rolling · pacman · AUR · Security Lab

NetHunter, Cloud, Reporting & Security Career Next Steps

Full Course · Arch 2026.11 · Kali 2026.3 (Nov 2026)

What You Will Learn in Lesson 12

- Kali NetHunter: Rootless, Lite and full editions
- Kali in the cloud — and cloud providers' testing policies
- Writing a professional security report
- Responsible disclosure
- Security careers: red, blue, purple, GRC, forensics
- Certifications and free learning paths
- Course review and final project

Kali NetHunter Editions

Edition	Requirements	What you get
NetHunter Rootless	Any Android phone, Termux app (no root)	Kali command line in a chroot; safest start
NetHunter Lite	Rooted phone (e.g. Magisk)	More tools, NetHunter app, still no custom kernel
NetHunter (full)	Supported device + custom NetHunter kernel	USB/HID attacks, Wi-Fi injection on supported chips
Smartwatch	Supported watches (e.g. TicWatch Pro 3)	Kali 2026.3 added wardriving support
Rule	Same law and ethics as Kali on a laptop	Authorized testing and your own lab only

Kali in the Cloud

Topic	Details
Images	Official Kali images on AWS Marketplace and Microsoft Azure; Docker image
Use cases	Assess YOUR cloud resources, labs with fast internet, temporary engagements
Provider policies	AWS, Azure and Google Cloud publish penetration-testing policies for your own resources
Never	Scan other customers' resources or the provider's infrastructure
Security	SSH keys only, restrict security groups to your IP, stop instances when done
Cost	Shut down or delete instances and disks to avoid charges

A Professional Security Report

Section	Contents
Executive summary	1 page, plain language: overall risk, top issues, key recommendations
Scope & methodology	What was tested, when, how, and what was NOT tested
Findings	Title, severity (e.g. CVSS), affected systems, description, evidence
Impact	What an attacker could do — in business terms
Remediation	Concrete steps to fix, priority, and how to verify the fix
Appendix	Tool output, timelines; protect sensitive data in the report itself

Example Finding (from our lab)

```
Finding 1: Outdated FTP service with known vulnerabilities
Severity : Critical (CVSS v3.1 9.8)
Host     : target1 — 192.168.56.20, TCP/21
Evidence : nmap -sV reports 'vsftpd 2.3.4' (scan saved: lab/target1.xml)
Detected : Suricata alert ET SCAN at 21:14 (Kali Purple)
Impact   : Full remote compromise of the server is possible
Fix      : 1) Remove FTP or upgrade to a supported version
           2) Use SFTP instead of FTP (no clear-text passwords)
           3) Firewall: allow only required ports; monitor with IDS
Verify   : nmap -sV 192.168.56.20 shows port 21 closed
```

Responsible Disclosure

Step	What to do
Found something by accident?	Stop. Do not explore further or access data
Find the right contact	security.txt (/.well-known/security.txt), bug bounty page, vendor PSIRT
Report privately	Clear steps, impact, minimal evidence; no public posts
Give time to fix	Commonly about 90 days, or as agreed
Coordinate publication	Publish only after a fix, with the owner's agreement
Never	Demand money, threaten, or keep personal data

Security Career Paths

Path	Typical work	Skills to build
Red team / pentester	Authorized testing of networks, web, cloud	Linux, networking, web, scripting, report writing
Blue team / SOC analyst	Monitoring, alert triage, incident response	Logs, SIEM, IDS, Windows & Linux internals
Purple team	Improve detections with attack simulations	Both sides; ATT&CK framework
Digital forensics / IR	Investigate incidents, preserve evidence	File systems, memory, chain of custody
GRC / security engineer	Policies, risk, hardening, cloud security	Frameworks (NIST, ISO 27001, CIS), automation

Certifications and Free Learning

Level	Options
Foundations	CompTIA Linux+, Security+; LPI Linux Essentials; ISC2 Certified in Cybersecurity (CC)
Hands-on offense	eJPT, PNPT, OffSec OSCP (PEN-200 — uses Kali)
Defense	CompTIA CySA+, Blue Team Level 1, SANS/GIAC (GCIH, GCIA)
Linux depth	LFCS, RHCSA (our RHEL course)
Free practice	TryHackMe free rooms, picoCTF, OverTheWire, PortSwigger Academy, MyWebUniversity
Tip	Build a portfolio: lab write-ups, reports, scripts on GitHub (no real targets!)

Course Review and Final Project

- Arch: archinstall, pacman, AUR safety, snapshots, troubleshooting
- Kali: ethics & law, install, metapackages, tools tour, safe lab, Purple, NetHunter
- Final project part 1: Arch desktop with snapper + weekly routine documented
- Final project part 2: isolated Kali lab, one purple-team cycle, one professional report
- Next courses: Solaris/OVM, mobile OSes, web-based services — and RHEL, Debian, Fedora
- Learn it — ethically — then pass it on

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.