HACK THE PLANET

Ethical Hacking Roadmap 2026

From absolute zero to job-relevant security skills.
A step-by-step path with real tools, real platforms,
and zero fluff.

FREE GUIDE · 2026 EDITION

hackthepalnte.co.uk

What's Inside

This guide is built to be read with a notebook open. Here is the full journey:

How to Use This Roadmap	Read this first — the rules of the road
Phase 1 — Computer & Network Foundations	Weeks 1–3 · TCP/IP, ports, subnets
Phase 2 — Linux & the Terminal	Weeks 3–6 · Commands, scripting, Kali in a VM
Phase 3 — How the Web Really Works	Weeks 6–9 · HTTP, logins, sessions, SQL
Phase 4 — Core Hacking Methodology	Weeks 9–14 · The 5-step loop + OWASP Top 10
Phase 5 — Tools of the Trade	Weeks 14–18 · Nmap, Burp, Nuclei & the recon pipeline
Phase 6 — Bug Bounty & Specialisation	Weeks 18–24 · Platforms, reports that get paid
Phase 7 — Practise Legally & Get Certified	Ongoing · Labs, certs, staying legal
Your First 30 Days	A week-by-week starter plan
7 Mistakes That Kill Beginners	Avoid these and you beat the odds
Quick Answers	Degree? Laptop? Earnings? — honest answers

> Tip: print the "Your First 30 Days" page and pin it where you study. Crossing off weeks by hand is strangely motivating.

How to Use This Roadmap

This roadmap is a **sequence, not a menu**. Each phase builds on the one before it. Rushing to "hacking tools" in week one is the single most common reason beginners quit — they can run a tool but cannot understand its output, so every result looks like magic and nothing sticks.

- **Time commitment:** 1–2 focused hours a day beats 8 distracted hours on a Sunday. Consistency is the entire game.
- **Rule of the road:** only practise on systems you own or that explicitly invite testing (see Phase 7). Unauthorised access is a crime in almost every country, including Pakistan — and one mistake can end a career before it starts.
- **Honest expectation:** 4–6 months of consistent work takes a beginner to a level where bug-bounty platforms and junior roles start making sense. Anyone promising "hacker in 30 days" is selling you something.
- **English matters less than you think:** you need *technical* English — the 500 most common words in documentation and error messages. You will absorb it naturally by reading docs.

The seven phases at a glance

PHASE 1	Computer & Network Foundations	Weeks 1–3
PHASE 2	Linux & the Terminal	Weeks 3–6
PHASE 3	How the Web Really Works	Weeks 6–9
PHASE 4	Core Hacking Methodology	Weeks 9–14
PHASE 5	Tools of the Trade	Weeks 14–18
PHASE 6	Bug Bounty & Specialisation	Weeks 18–24
PHASE 7	Practise Legally & Get Certified	Ongoing

Phase 1 — Computer & Network Foundations

You cannot break what you do not understand. Networking is the ground everything else stands on — skip it and every tool you run later will feel like a black box.

What to learn

- **The journey of a URL:** what happens, step by step, when you type an address and press Enter — DNS lookup, TCP three-way handshake, TLS negotiation, HTTP request and response. If you can narrate this, you understand networking.
- **IP addresses and subnets:** IPv4 basics, CIDR notation (/24, /16), private vs public ranges (192.168.x.x, 10.x.x.x).
- **Common ports and what listens on them:** memorise the big ones below — you will see them every single day.

Port	Service	Why it matters
21	FTP	File transfer; often misconfigured, anonymous login still common
22	SSH	Secure remote shell; brute-force target, key-based auth
80	HTTP	Unencrypted web traffic
443	HTTPS	Encrypted web traffic; where most testing happens
3306	MySQL	Database; SQL injection's favourite neighbour
3389	RDP	Windows remote desktop; ransomware's front door

Free resources

- Professor Messer's Network+ course (free on YouTube) — the clearest networking explainer on the internet.
- Practical Networking by David Bombal (YouTube) — for packet-level, hands-on understanding with Wireshark.

What NOT to learn yet

Do not touch Metasploit, do not buy a hacking course, do not install Kali yet. Foundations first — the tools will still be there in week 9, and then they will actually make sense.

> Milestone: explain, without notes, the full journey of one web request from your laptop to a server and back.

Phase 2 — Linux & the Terminal

Almost every security tool on earth runs on Linux. If the terminal scares you, this phase fixes that permanently — and it is the phase most beginners try to skip and later regret skipping.

What to learn

- **Filesystem and permissions:** how Linux organises files, what rwx means, chmod, chown, sudo — and *why* permissions are a security boundary, not just trivia.
- **Core commands** until they are muscle memory:

Command	Does what	You will use it for
ls / cd / pwd	Navigate the filesystem	Everything, daily
grep	Search inside files	Reading logs, finding secrets in code
find	Locate files by name/type	Hunting config files on targets
chmod / chown	Change permissions/owners	Understanding privilege issues
ps / kill / top	Manage processes	Seeing what runs on a machine
curl / wget	Fetch URLs and files	Testing web endpoints quickly
ssh / scp	Remote shell and file copy	Connecting to lab machines

- **Bash scripting basics:** variables, loops, if-statements — just enough to automate boring recon tasks later. A 10-line script that checks 500 subdomains will save you days.
- **Install Kali Linux in a virtual machine** (VirtualBox is free). Snapshot the clean install on day one. Never install Kali as your main OS as a beginner — you will break it within a month.

Milestone: live in the terminal for one full week — no file manager, no GUI text editor.

Phase 3 — How the Web Really Works

Well over half of beginner bug bounties are web vulnerabilities. You must understand the web the way a *developer* does before you can test it the way an *attacker* does.

What to learn

- **HTML, CSS and JavaScript:** enough to read any page's source and follow what its JavaScript does. You do not need to become a front-end developer — you need to *read* code fluently.
- **HTTP in depth:** methods (GET, POST, PUT, DELETE), status codes (200, 301, 403, 500), headers, cookies.
- **Sessions vs tokens:** how a server remembers you are logged in — session cookies vs JWTs. Half of authentication bugs live in this distinction.
- **How logins really work:** password hashing (bcrypt — never plain text), multi-factor authentication, OAuth "login with Google" flows.
- **Databases and SQL:** what a query looks like and how a login form talks to a database. This single piece of knowledge is the foundation SQL injection is built on.

See it yourself

Open any login page, press F12, go to the Network tab, and log in. Watch each request. A typical login looks like this:

```
POST /api/login  →  200 OK
Request: {"username":"admin","password":"***"}
Response headers: Set-Cookie: session=abc123; HttpOnly
Response: {"token":"eyJhbGciOi..."}
```

If you can read that exchange and explain every line — what was sent, what came back, and where the session lives now — you are ready for Phase 4. If not, stay here another week. This is the highest-leverage knowledge in the entire roadmap.

> Milestone: explain every network request your browser makes during one login, using only the developer tools.

Phase 4 — Core Hacking Methodology

Hacking is a **method**, not a bag of tricks. Professionals run the same five-step loop on every engagement, whether it is a 2-hour bounty hunt or a 3-week pentest:

- **1. Reconnaissance** — gather information. Start passive (OSINT: subdomains, employee names, tech stack) then go active. Great hackers spend *more* time here than anywhere else.
- **2. Scanning & enumeration** — find live hosts, open ports, running services, hidden directories. Turn the target from a black box into a map.
- **3. Vulnerability analysis** — match what you found against known weakness patterns. "This login form reflects my input" → possible XSS.
- **4. Exploitation** — prove the weakness is real, carefully, *only* on systems you are authorised to test. Proof, not damage.
- **5. Reporting** — write it so clearly a developer can fix it on Monday morning. This skill alone gets people hired — most beginners never practise it.

The OWASP Top 10

The Open Worldwide Application Security Project publishes the definitive catalogue of web risks (free at owasp.org). Learn all ten — one per day — and find one real disclosed CVE write-up for each:

#	Category	In plain words
1	Broken Access Control	Seeing or changing things your account should not reach (e.g. viewing another user's orders)
2	Cryptographic Failures	Sensitive data exposed or weakly protected
3	Injection	Tricking the server into running your input as code — SQLi, XSS, command injection
4	Insecure Design	Security never considered in the design itself
5	Security Misconfiguration	Default passwords, open debug pages, verbose errors
6	Vulnerable Components	Outdated libraries with known CVEs
7	Auth Failures	Broken login, weak sessions, missing MFA
8	Data/Software Integrity Failures	Unverified updates and third-party code
9	Logging Failures	Attacks nobody notices because nothing is logged
10	SSRF	Tricking the server into requesting internal resources

> Milestone: describe the five steps from memory and explain any three OWASP categories with a real-world example each.

Phase 5 — Tools of the Trade

Tools **amplify** methodology; they never replace it. The rule: **one tool per week, learned deeply**. Read its official docs, run it against your own lab, and note what every flag does.

- **Nmap** — network scanning and service enumeration. The single most-used tool in the industry. Learn scan types (-sS, -sV, -sC) and *when* each is appropriate.
- **Burp Suite** (Community edition, free) — intercept and modify your own web traffic. This becomes your web-testing home base; get comfortable with the Proxy and Repeater.
- **Subfinder + httpx** — subdomain discovery, then probing which hosts are actually alive. The bug-hunter's daily pair — most recon pipelines start here.
- **Nuclei** — template-based scanning with thousands of community templates. Fast way to check known vulnerability patterns across many targets.
- **Gobuster** — directory and DNS brute-forcing during enumeration. Simple, fast, essential.
- **Wireshark** — see the actual packets on the wire. Makes everything from Phase 1 concrete — run it while you browse and watch the theory come alive.

A typical recon workflow, tying it together

subfinder finds subdomains → httpx checks which are alive → nuclei scans the live ones for known issues → anything interesting gets manual testing in Burp Repeater → findings become a report (Phase 4, step 5). That pipeline — automated discovery plus manual verification — is what working bug hunters actually do all day.

> Tool collectors know fifty tools shallowly. Professionals know five deeply. Be the second kind.

Phase 6 — Bug Bounty & Specialisation

Bug bounty is where beginners earn their first real money and their first real reputation — legally, and from anywhere in the world.

The path

- **Join HackerOne and Bugcrowd** (free). Before hunting, *read disclosed reports* — hundreds are public and each one is a free masterclass in how a real bug was found, step by step.
- **Pick ONE vulnerability class** and go deep for a month: XSS, IDOR, or SSRF are the classic beginner-friendly starting points with plenty of learning material.
- **Start on VDPs** (vulnerability disclosure programs — no money, just thanks) to practise the full find-and-report workflow without financial pressure.
- **Honest expectations:** your first valid report may take weeks. Most hunters' first payouts are $50–$500. The hunters earning five figures studied *one* bug class deeply for months. There are no shortcuts here either.

Anatomy of a report that gets paid

A good report has five parts — practise this format on every lab machine from Phase 7:

- **Title:** one line, precise — "Stored XSS in profile bio via SVG upload".
- **Steps to reproduce:** numbered, starting from a fresh login, so exact that a stranger can follow them blindly.
- **Impact:** what an attacker could *actually* do with this — be concrete, not dramatic.
- **Proof:** screenshots or a short video showing the exploit working.
- **Suggested fix:** one or two lines a developer can act on. This is what separates paid reports from ignored ones.

Specialisations for later: web application pentesting, network pentesting, mobile security, cloud security, digital forensics. Depth in one beats breadth in five.

Phase 7 — Practise Legally & Get Certified

Where to train — all legal, most free

- **PortSwigger Web Security Academy** — free labs from the makers of Burp Suite. Widely considered the best web-hacking training on the internet. Non-negotiable — finish it.
- **TryHackMe** — guided rooms from absolute zero to advanced; the free tier is generous and the learning paths hold your hand early on.
- **HackTheBox** — harder, realistic machines for when TryHackMe starts feeling easy. The "Starting Point" tier eases you in.

Certifications worth your money

Certification	Focus	Best for
CompTIA Security+	Security fundamentals, theory-heavy	Getting past HR filters; the baseline CV cert
eJPT (INE)	Hands-on junior pentesting	Proving practical skill affordably
CEH	Broad theory, recognised name	Compliance-driven job markets

A blunt truth: **certificates open doors; proof of skill walks through them.** Pair any certification with a public portfolio — ten write-ups on a blog, lab notes, a GitHub with your recon scripts. Interviewers remember the candidate who *showed* work.

Stay legal. Always.

Test only systems you own, lab environments, or programs with a published scope that explicitly authorises you. "I was just learning" is not a legal defence anywhere. The entire security industry runs on trust — protect yours from day one, because one reckless test can close every door this roadmap opens.

Your First 30 Days — A Starter Plan

Overwhelmed? Do exactly this. One week at a time:

- **Week 1 — Networking:** Professor Messer's Network+ playlist, 1 hour daily. Write a one-page summary of the URL journey in your own words.
- **Week 2 — Linux:** install VirtualBox + Kali (or Ubuntu to start gentler). Do the command table from Phase 2 until you stop googling ls.
- **Week 3 — Web basics:** freeCodeCamp's HTML/CSS/JS intro. Then spend two evenings in browser dev tools on real sites.
- **Week 4 — First lab:** create a TryHackMe account, finish the "Pre Security" path. You will touch Nmap, Burp and Gobuster for the first time — on legal targets, with guidance.

End of day 30, you will not be a hacker. You will be something better: a beginner with *direction* — which puts you ahead of 90% of people who "want to learn hacking" and never start.

7 Mistakes That Kill Beginners

- **1. Tool-first learning.** Running Kali tools with no networking knowledge produces screenshots, not skills.
- **2. Tutorial hell.** Watching 200 hours of videos without touching a lab. Rule: for every hour of video, two hours of practice.
- **3. Skipping writing.** If you cannot explain a concept in writing, you do not understand it. Notes and write-ups compound.
- **4. Chasing every new tool.** A new scanner every week means zero depth anywhere. Master the Phase 5 list first.
- **5. No legal lab.** "Practising" on random websites is how beginners get police visits instead of jobs. Use the Phase 7 platforms.
- **6. Comparing day 30 to someone's year 5.** The experts you admire were exactly where you are. The only difference is they did not quit.
- **7. Learning alone, silently.** Join communities, ask questions, share write-ups. Security is a team sport — hiring happens through networks.

Avoid these seven and you have already beaten the odds. Most people quit in Phase 1 — not because it is hard, but because nobody told them what to do next. Now you know.

Quick Answers

- **Do I need a degree?** For bug bounty: no, nobody asks. For jobs: it helps with HR filters, but skills + write-ups regularly beat degrees in security hiring.
- **What laptop do I need?** Any machine that runs VirtualBox smoothly — 8 GB RAM minimum, 16 GB comfortable. No gaming laptop required.
- **Is it too late to start in 2026?** The industry has a chronic talent shortage and AI is *increasing* demand for people who understand how systems break. Late is a myth.
- **How much can beginners earn?** Honestly: $0 for the first months while learning, then small bounties ($50–$500), growing with skill. Anyone guaranteeing income is lying.

- **Urdu/Hindi resources?** Start in English for documentation (it is unavoidable), and use structured video courses in your own language for the concepts — that combination is exactly what Hack The Planet was built for.
- **Can I do this from Pakistan?** Yes. Bug bounty platforms accept hunters from everywhere and pay internationally; remote security jobs hire on skill, not location. Your location is not the barrier — consistency is.
- **How do I stay motivated for 6 months?** Make progress visible: keep a daily log, publish write-ups, and join a community. Motivation follows action, not the other way round.

Your Next Step

A roadmap tells you *what* to learn. A teacher shows you *how* — spotting your mistakes, answering your questions, and keeping you moving when tutorial hell tempts you.

That is what we built **Hack The Planet** for: structured, step-by-step video lessons that follow this exact path, from absolute zero to bug bounty — in English, with select courses in Urdu and Hindi.

- **49 practical video courses** — ethical hacking, bug bounty, OSINT, web security. $9–$25 each.
- **3 courses free** to start. No card required — just curiosity.
- Direct **WhatsApp support** with every course — a human replies, not a bot.

hackthepalnte.co.uk

Free resources every day on Telegram: t.me/hackthepl

Good hunting. Stay ethical.