CYBERSECURITY ROADMAP

A practical step-by-step path from beginner to job-ready security skills.

STEP	FOCUS	WHAT YOU WILL BUILD
01	Foundation	Computer, networking, Linux and programming fundamentals
02	Core Security	Web, systems, cryptography, vulnerabilities and security operations
03	Specialization	A focused offensive, defensive, cloud or application-security path
04	Portfolio	Labs, projects, write-ups and professional security reports
05	Career	Resume, GitHub, interview preparation and job applications
06	90-Day Plan	A structured weekly schedule to turn knowledge into ability

How to use this roadmap

Do not rush through every topic. Learn the concept, practice it in a legal lab, build something with it, document what you learned, and then move forward. Revisit weak areas every month.

01 FOUNDATION

Goal: understand computers and networks well enough that security concepts make sense.

AREA	LEARN	PRACTICAL TASK	TARGET
Computer Basics	CPU, RAM, storage, processes, filesystems, permissions, virtualization	Set up a Linux VM and explain its components	Understand how a system works
Networking	OSI/TCP-IP, IPv4, ports, TCP/UDP, DNS, DHCP, NAT, routing, HTTP/HTTPS	Inspect your own traffic with Wireshark	Explain a connection end-to-end
Linux	CLI, users, groups, chmod, processes, services, logs, packages, SSH	Use Linux for daily security practice	Work comfortably in terminal
Python	Variables, functions, files, JSON, requests, regex, APIs, exceptions	Build a log parser and HTTP/API script	Automate simple tasks
Git	Repositories, commits, branches, README files	Publish a small learning project	Track and document work

What to practice every week

• Build and break small Linux environments. • Read network packets. • Practice Bash commands. • Write one small Python script. • Explain one networking concept without notes.

Milestone

You can install Linux, navigate the terminal, inspect processes and network connections, understand common ports, and write simple scripts without following a tutorial.

02 CORE SECURITY

Goal: understand the common ways applications, systems and users become exposed to risk.

DOMAIN	LEARN	PRACTICE / OUTPUT
Web Security	HTTP methods, headers, cookies, sessions, authentication, authorization, input validation, APIs	Use a legal web lab. Explain the request, weakness, impact and fix.
Vulnerabilities	Injection, broken access control, insecure design, security misconfiguration, vulnerable components	Map a vulnerability to its root cause and remediation.
System Security	Users, privileges, services, patching, hardening, endpoint controls, backups	Harden a lab machine and record every change.
Cryptography	Hashing vs encryption, symmetric/asymmetric keys, certificates, TLS, digital signatures	Inspect a certificate and explain what it protects.
Security Operations	Logs, alerts, indicators, incident lifecycle, threat modeling, risk	Analyze sample logs and write a short incident summary.
Tools	Nmap, Wireshark, Burp Suite, Linux utilities, basic SIEM concepts	Learn why each tool works instead of memorizing commands.

Important mindset

A tool finding is not the final answer. Always ask: What is exposed? Why is it happening? What could the impact be? How would a defender fix or detect it?

Milestone

You can read a basic security finding, reproduce it safely in a lab, explain the risk in plain English, and recommend a practical fix.

03 SPECIALIZE

Goal: become strong in one area while keeping your general security foundation.

PATH	CORE TOPICS	PROJECT IDEAS	ENTRY ROLES
Offensive Security	Web testing, enumeration, privilege escalation, Active Directory labs, reporting	Build a legal pentest lab; write assessment reports	Pentest Intern / Junior Tester
Defensive Security	SIEM, logs, detection rules, endpoint telemetry, threat hunting, response	Build a home SOC dashboard; create detection rules	SOC Analyst / Junior Blue Team
Cloud Security	IAM, cloud networking, storage, secrets, logging, containers, misconfigurations	Secure a small cloud lab and document controls	Cloud Security Intern / Junior
Application Security	Secure SDLC, code review, API security, dependencies, CI/CD, threat modeling	Review a demo app and produce remediation notes	AppSec Intern / Security Engineer

How to choose

Choose the work you enjoy practicing repeatedly. Offensive security is not the only cybersecurity career path. If you like investigation, choose defensive security; if you like development, consider AppSec; if you enjoy infrastructure, consider cloud security.

Study ratio

70% specialization • 20% core security • 10% new technologies and industry awareness

04 BUILD YOUR PORTFOLIO

Goal: turn knowledge into visible proof of skill.

PROJECT	WHAT TO BUILD	DOCUMENTATION SHOULD SHOW
01 — Security Lab	A small virtual lab with Linux systems, networking and security tools	Architecture, setup, controls, lessons learned
02 — Automation	A Python/Bash utility for log parsing, asset inventory or repetitive analysis	Problem, approach, code, example output
03 — Legal Labs	A series of CTF or training-platform exercises	Concept learned, methodology, remediation
04 — Security Report	A simulated professional assessment	Executive summary, finding, evidence, impact, fix
05 — Defensive Project	Simple log analysis or detection workflow	Data source, detection logic, false positives, response

Career checklist

TASK	TARGET
Resume	One page; skills supported by projects
GitHub / Portfolio	3–5 clean projects with READMEs
Write-ups	Clear, legal and focused on learning
Interview Prep	Networking, Linux, web security and scenario questions
Applications	Internships, SOC, junior security, AppSec or testing roles

What makes a strong project?

A strong project has a clear problem, a repeatable setup, evidence of your work, a short explanation of security impact, and a section describing how the issue could be prevented or detected.

05 90-DAY EXECUTION PLAN

Goal: create momentum without trying to learn everything at once.

WEEKS	FOCUS	WEEKLY ACTION	OUTPUT
1–2	Linux + Networking	Terminal practice, IP/ports, DNS, packet analysis	Linux notes + packet analysis
3–4	Python + Bash	Automation, files, APIs, parsing and error handling	2 useful scripts
5–6	Web Security	HTTP, sessions, authentication, APIs and safe labs	Lab notes + findings
7–8	Security Operations	Logs, alerts, risk, incident response basics	Incident summary
9–10	Specialization	Focused labs and one small project	Project draft
11–12	Portfolio + Career	Finish project, documentation, resume, interview practice	Published portfolio

Daily routine

60–90 minutes: 20 min theory → 40–50 min hands-on practice → 10–20 min notes/documentation. On weekends, spend one longer session building a project instead of only watching tutorials.

Rules to follow

1. Practice only on systems you own or have explicit permission to test.
2. Learn concepts before tools.
3. Keep a personal knowledge base.
4. Build something every month.
5. Review weak topics instead of endlessly starting new courses.

FINAL TARGET

Learn → Practice → Build → Document → Explain → Repeat

The goal is not to know every hacking tool. The goal is to understand systems well enough to find, prevent, detect and explain security problems.

JOB-READY TARGET: 12–18 MONTHS OF CONSISTENT PRACTICE