Roadmap: From Zero to AI/ML Cybersecurity Engineer

A complete, web-research-based learning path for a beginner who knows only basic C++ and Python — no networking, security, Linux, or cloud background assumed.

How This Roadmap Was Built

This roadmap was assembled by searching the web for current roadmaps, official certification objectives, and learning resources from reputable, institutionally backed sources: TryHackMe (official paths), OverTheWire, Professor Messer (CompTIA-aligned free training), Khan Academy, Stanford Online / Stanford Engineering Everywhere, DeepLearning.AI, Coursera, OWASP (the official GenAI Security Project and AI Exchange), AWS Skill Builder, INE Security, Hack The Box, Cybrary, NVIDIA (Garak), and Microsoft (PyRIT). Random blogs, unverified opinion pieces, and sources with no institutional or community backing were excluded. Every topic below names the exact concept, tool, or protocol to learn rather than a vague category, and every step has its own resource section directly underneath it.

Honesty Note on Gaps

Where no sufficiently reputable free resource could be confirmed, this is stated explicitly inside the relevant phase rather than filled in with an assumption. The clearest example is Phase 9, Step 9.3: there is currently no widely recognized FREE certification specifically for AI/ML security. The two named certifications found in research (INE's eAIS and Practical DevSecOps's CAISP) are both paid. The free substitute path — OWASP documentation plus hands-on practice with Garak and PyRIT plus a self-built portfolio project — is explained there as the realistic alternative.

Total Realistic Timeline

Approximately 15-18 months of consistent part-time study (roughly 10-15 hours per week, realistic for a full-time student), broken into 10 phases below. This is not rushed (a genuine beginner cannot skip the Linux/networking/web foundations and still safely or competently do offensive security or AI security work) and not artificially stretched (each phase has a defined, achievable end point with a real resource, not an open-ended "keep learning" instruction).

Legal and Ethical Note

Every hands-on platform named below (TryHackMe, OverTheWire, Hack The Box) provides legally authorized practice environments. Techniques learned here must never be used against any system without explicit written permission.

---

1. Phase 1: Linux and Command-Line Fundamentals

Duration: Weeks 1-4. Why first: virtually every security tool, every hands-on lab platform, and every machine learning training pipeline in later phases runs on Linux. Without comfortable command-line use, every later phase will be slower and more confusing than it needs to be.

1.1 Step 1.1 — How computers, operating systems, and the internet work conceptually

What to learn specifically: binary and hexadecimal number basics, how an operating system manages processes, memory, and files, the client-server model, what an IP address and a port number actually are, and the basic idea of how a web request reaches a server.

Resources for this step:

- TryHackMe — “Pre Security” learning path (free): tryhackme.com/path/outline/presecurity. Built specifically for absolute beginners; covers computer basics, networking basics, web fundamentals, Linux fundamentals, and Windows fundamentals from zero, in that order.

1.2 Step 1.2 — Core Linux commands and the filesystem

What to learn specifically: navigation (cd, ls, pwd), viewing and editing files (cat, nano, less), searching (grep, find), permissions (chmod, chown, the rwx model), piping and redirection (|, >, >>, 2>/dev/null), and connecting to a remote machine over SSH.

Resources for this step:

- OverTheWire — “Bandit” wargame (free): overthewire.org/wargames/bandit. A 30+ level, SSH-based, hands-on challenge series that teaches real Linux commands through practical puzzles rather than passive reading. Long-standing, community-validated, and explicitly recommended as a companion to TryHackMe by multiple independent learner write-ups.
- TryHackMe — “Linux Fundamentals” rooms (free, part of the Pre Security path above).

---

2. Phase 2: Networking Fundamentals

Duration: Weeks 5-10. Why second: both security work and AI/cloud infrastructure run over networks. You cannot understand an attack, a defense, or a cloud architecture diagram without the OSI model, TCP/IP, and the core protocols below.

2.1 Step 2.1 — OSI model, TCP/IP, subnetting, routing, and core protocols

What to learn specifically: the 7 OSI layers, IPv4 and IPv6 addressing, subnetting and CIDR notation, the difference between TCP and UDP, DNS, DHCP, HTTP/HTTPS, common port numbers (21, 22, 23, 25, 53, 80, 443, 3389, and others), NAT, basic VPN concepts, and 802.11 wireless basics.

Resources for this step:

- Professor Messer — free CompTIA Network+ N10-009 video course: professormesser.com/network-plus/n10-009. 87 free videos (about 13 hours total), covering every Network+ exam objective with specific, named protocol-level detail (this is not a vague overview course).

2.2 Step 2.2 — Apply networking knowledge hands-on

What to learn specifically: capturing and reading packets with Wireshark, and discovering hosts and open ports with Nmap (including basic NSE scripts).

Resources for this step:

- TryHackMe — “Cyber Security 101” path (free): tryhackme.com/path/outline/cybersecurity101 — includes dedicated Network Fundamentals and Nmap rooms with guided hands-on practice.

---

3. Phase 3: Core Security Concepts and Web Fundamentals

Duration: Weeks 11-14 (can overlap with the start of Phase 4). Why third: before attacking or defending anything you need the shared vocabulary of the field, and an understanding of how the web actually works, since most real-world attack surface today is web-based.

3.1 Step 3.1 — Security fundamentals

What to learn specifically: the CIA triad (confidentiality, integrity, availability), AAA (authentication, authorization, accounting), the difference between a risk, a threat, and a vulnerability, symmetric vs asymmetric encryption at a conceptual level, and common attack types such as phishing, malware, and social engineering.

Resources for this step:

- TryHackMe — “Cyber Security 101” path (free, same path as Step 2.2, used here for its security-concept rooms).
- Professor Messer — free CompTIA Security+ SY0-701 video course: professormesser.com (optional deeper dive; covers governance, risk, cryptography, and security operations in full, entirely free).

3.2 Step 3.2 — Web application fundamentals

What to learn specifically: the HTTP request/response cycle, cookies and sessions, HTML/JavaScript basics from a security perspective, and the OWASP Top 10 web vulnerabilities (SQL injection, cross-site scripting, broken access control, and the rest of the list).

Resources for this step:

- TryHackMe — “Web Fundamentals” path (free).
- PortSwigger — Web Security Academy (free): portswigger.net/web-security — the de facto standard free, hands-on resource for OWASP Top 10 web vulnerabilities; widely referenced inside the eJPT preparation community specifically for this reason.

---

4. Phase 4: Python for Security and Automation

Duration: Weeks 11-16, run in parallel with Phase 3. Why here: you already know Python at a surface level. This phase converts that into the specific scripting patterns security work actually uses, before you need them in Phase 5.

4.1 Step 4.1 — Python refresher applied to security contexts

What to learn specifically: file I/O, regular expressions, and the socket, requests, and argparse modules.

Resources for this step:

- Cybrary — “Python for Cybersecurity Professionals” (free): cybrary.it/course/python-for-cybersecurity-professionals. Listed on CISA's official NICCS national training catalog (niccs.cisa.gov), beginner level, project-based.
- Cybrary — “Intro to Python” (free): cybrary.it/course/python. About 3 hours, designed specifically for non-coders entering cybersecurity; useful as a quick refresher even though you already know basic Python.

4.2 Step 4.2 — Build basic security tools

What to learn specifically: writing a simple port scanner, a basic brute-forcer, and a log parser in Python.

Resources for this step:

- Cybrary — “Developing Ethical Hacking Tools with Python” (free): cybrary.it/course/developing-ethical-hacking-tools-with-python.

---

5. Phase 5: Offensive Security Core / Practical Penetration Testing

Duration: Months 4-7. Why here: this is your stated career direction, and it is the foundation every later specialization (including AI red teaming in Phase 9) builds on. You now have the Linux, networking, web, and scripting base needed to do this safely and effectively.

5.1 Step 5.1 — Structured offensive security path

What to learn specifically: reconnaissance and enumeration methodology, vulnerability scanning, exploitation with Metasploit, Linux and Windows privilege escalation, basic Active Directory attack concepts, and professional reporting.

Resources for this step:

- TryHackMe — “Jr Penetration Tester” learning path: tryhackme.com/path/outline/jrpenetrationtester. Free tier available; a paid subscription unlocks every room. Aligned to CompTIA PenTest+ exam objectives.
- Hack The Box — “Starting Point” (free tier): hackthebox.com. Guided, beginner-friendly machines designed as a first step before tackling unguided HTB content.

5.2 Step 5.2 — Certification milestone: eJPT

What it covers: host and network penetration testing, assessment methodologies, host/network auditing, and web application penetration testing. The exam is entirely hands-on (no multiple-choice theory questions), 35 practical questions against a live lab, 48-hour window, 70% passing score.

Cost honesty note: no free official eJPT path exists. The exam itself costs approximately $249 standalone, or is included with a $299/year INE subscription. All preparation for it, however, can be done entirely free using the TryHackMe, Hack The Box Starting Point, and PortSwigger resources already listed above before paying only for the exam voucher itself.

Resources for this step:

- INE Security — official eJPT certification page: ine.com/security/certifications/ejpt-certification.

---

6. Phase 6: Mathematics for Machine Learning

Duration: Months 6-8, may overlap with the later part of Phase 5. Why here: machine learning cannot be understood conceptually without linear algebra and probability/statistics. Covering this before the ML course in Phase 7 prevents math from becoming a blocker once you are inside that course.

6.1 Step 6.1 — Linear algebra

What to learn specifically: vectors, matrices, matrix multiplication, the dot product, and an intuitive (not fully rigorous) understanding of eigenvalues and eigenvectors, since these directly explain how a neural network layer transforms data.

Resources for this step:

- Khan Academy — “Linear Algebra” (free): khanacademy.org/math/linear-algebra.

6.2 Step 6.2 — Statistics and probability

What to learn specifically: probability distributions, mean/variance/standard deviation, conditional probability, and Bayes' theorem (directly relevant later to anomaly detection and threat-scoring models in security).

Resources for this step:

- Khan Academy — “Statistics and Probability” (free): khanacademy.org/math/statistics-probability.

---

7. Phase 7: Machine Learning Fundamentals

Duration: Months 8-11. Why here: this is the “ML” half of “AI/ML Cybersecurity Engineer.” You need to understand how models are actually built and trained before you can meaningfully secure them, which is the entire point of Phase 9.

7.1 Step 7.1 — Core supervised and unsupervised machine learning

What to learn specifically: linear regression, logistic regression, decision trees, neural network basics (forward pass and backpropagation at a conceptual level), clustering, the

train/validation/test split, overfitting versus underfitting, and building models using NumPy and scikit-learn.

Resources for this step:

- “Machine Learning Specialization” by Andrew Ng, Stanford Online and DeepLearning.AI (Coursera): coursera.org/specializations/machine-learning-introduction. Rated 4.9/5 and taken by over 4.8 million learners; the course content can be accessed via Coursera's free audit option (verify current audit availability at enrollment, since Coursera occasionally changes this). A paid certificate is optional and not required to learn the material.
- Stanford CS229 full lecture series, Stanford Engineering Everywhere (free, no audit restriction): see.stanford.edu/Course/CS229, also widely indexed at classcentral.com. The original, complete Stanford graduate machine learning course taught by Andrew Ng, free with no paywall.

7.2 Step 7.2 — Practical ML coding

What to learn specifically: implementing the models from Step 7.1 hands-on in Python.

Resources for this step:

- Same two resources as Step 7.1 — both include programming assignments built around NumPy and scikit-learn.

---

8. Phase 8: Cloud Computing and Cloud Security Fundamentals

Duration: Months 10-12, overlap with the end of Phase 7 is fine. Why here: nearly all production AI/ML systems run on cloud infrastructure. You need to understand how cloud computing works, and how to secure it, before you can meaningfully secure an AI pipeline running on top of it in Phase 9.

8.1 Step 8.1 — Cloud computing fundamentals

What to learn specifically: the IaaS / PaaS / SaaS service models, core AWS services (EC2, S3, IAM, VPC), and the AWS shared responsibility model.

Resources for this step:

- AWS Skill Builder — “AWS Cloud Practitioner Essentials” (free, official AWS training): skillbuilder.aws. Part of AWS's official library of 600+ free digital courses.

8.2 Step 8.2 — Cloud security fundamentals

What to learn specifically: IAM policies and the principle of least privilege, encryption at rest and in transit, security groups and network ACLs, and logging/monitoring services (CloudTrail, CloudWatch).

Resources for this step:

- AWS Skill Builder — “AWS Security Fundamentals” (free, official, self-paced AWS course): skillbuilder.aws, also independently indexed and confirmed free at classcentral.com.

---

9. Phase 9: AI/ML Security Foundations

Duration: Months 12-15. Why only now: you cannot meaningfully secure something you do not understand how to build. This phase only works because Phases 6 through 8 already gave you the ML fundamentals and the cloud security fundamentals it depends on.

9.1 Step 9.1 — Understand the AI/ML attack surface

What to learn specifically: data poisoning, model inversion and model extraction attacks, adversarial examples (evasion attacks), and supply chain risks inside ML pipelines (compromised pretrained models, poisoned datasets, vulnerable dependencies).

Resources for this step:

- OWASP — “AI Exchange,” part of the OWASP GenAI Security Project (free): genai.owasp.org. A 300+ page, community-maintained, open resource covering both traditional ML and LLM-specific risks.
- OWASP — “Machine Learning Security Top 10” (free, part of the same OWASP AI/ML security initiative) — covers traditional ML-specific risks, distinct from the LLM-specific list in Step 9.2 below.

9.2 Step 9.2 — LLM-specific security (the fastest-growing sub-area as of 2026)

What to learn specifically: prompt injection, insecure output handling, training data poisoning, model denial of service, supply chain vulnerabilities, sensitive information disclosure, and excessive agency (directly relevant to agentic AI systems).

Resources for this step:

- OWASP — “Top 10 for Large Language Model Applications,” version 2.0, 2025 (free, official): genai.owasp.org/llm-top-10. The de facto industry-standard checklist for this area, referenced by regulators and auditors even though it is not itself a regulation.

9.3 Step 9.3 — Hands-on adversarial testing

What to learn specifically: how to actually run structured adversarial probes against a deployed language model, rather than only reading about the risks above.

Resources for this step:

- NVIDIA — “Garak” (free, open-source): github.com/NVIDIA/garak. An LLM vulnerability scanner with 37+ probe modules covering prompt injection, jailbreaks, data leakage, and encoding-based bypass testing.
- Microsoft — “PyRIT”, the Python Risk Identification Toolkit (free, open-source, Microsoft's official AI Red Team framework): github.com/Azure/PyRIT. Used for building automated, multi-turn adversarial attack chains against conversational AI systems.

Certification honesty note: no free, widely recognized certification specifically for AI/ML security currently exists. The two named AI-security certifications found in this research — INE's eAIS and Practical DevSecOps's CAISP — are both paid. The realistic free substitute path is the OWASP documentation above, combined with hands-on Garak/PyRIT practice and the portfolio project in Phase 10, Step 10.1.

---

10. Phase 10: Specialization, Certification, and Portfolio

Duration: Months 13-18, and ongoing afterward. Why last: hiring managers and certifications validate what you have already built in Phases 1-9. This phase converts learning into something employable.

10.1 Step 10.1 — Build a portfolio project bridging both halves of the role

What to do specifically: stand up a small open-source LLM application (for example, a chatbot built on an open model), then run a documented Garak and PyRIT red-team assessment against it, map every finding to the OWASP LLM Top 10 from Step 9.2, and write the results up as a professional security report (scope, methodology, findings, severity ratings, and remediation recommendations — the standard structure used industry-wide for penetration test reports).

Resources for this step:

- Same tools as Phase 9, Step 9.3 (Garak and PyRIT). No single official free report-writing template was found as a reputable, citable source during this research; the scope/methodology/findings/severity/remediation structure named above reflects the documented industry-standard format referenced across multiple penetration-testing certification bodies, including Hack The Box's CPTS exam requirements.

10.2 Step 10.2 — Decide on further certification investment

What to do: eJPT (Phase 5) is your first checkpoint and is genuinely industry-recognized for entry-level offensive security roles. Beyond that, as stated in Phase 9, no free AI-security-specific certification was found. A fully unpaid alternative is to rely entirely on the OWASP resources, Garak/PyRIT hands-on work, and the Step 10.1 portfolio project, since available labor-market research indicates the practical skillset is currently outpacing formal credentialing in this specific niche.

10.3 Step 10.3 — Stay current

What to do: follow the OWASP GenAI Security Project's ongoing updates (genai.owasp.org/meetings) and NIST's AI Risk Management Framework publications, since this sub-field is documented to be changing faster than most established cybersecurity domains.

Summary of the Full Sequence

Phase 1 Linux → Phase 2 Networking → Phase 3 Security & Web Fundamentals → Phase 4 Python for Security (parallel to Phase 3) → Phase 5 Offensive Security Core + eJPT →

Phase 6 Math for ML → Phase 7 Machine Learning Fundamentals → Phase 8 Cloud & Cloud Security → Phase 9 AI/ML Security Foundations → Phase 10 Specialization, Certification, and Portfolio.