Forebrain — 6-Month Cybersecurity Professional (24 weeks)

Duration: 24 weeks

Intensity: ~20–25 hrs/week (live lessons, labs, project)

Mode: Cohort (instructor + TA) — hands-on, red+blue mindset, job-ready + research capstone

Assessment & Grading (Master)

Weekly labs & practical quizzes: 40%

Module projects & reports (6 smaller + 1 midterm): 30%

Capstone (team red/blue + research): 20%

Participation, CV/mocks: 10%

Rubrics (Master Templates)

Weekly lab rubric (out of 10):

- Completion (all steps): 4

- Correctness of commands/output: 3

- Documentation & explanation: 2

- Timeliness & Git submission quality: 1

Project rubric (out of 100):

- Technical correctness: 40

- Documentation & reproducibility: 30

- Creativity: 20

- Presentation: 10

Capstone rubric (team):

- Offensive effectiveness & evidence: 25

- Defensive detection & response quality: 25

- Research & metrics: 20

- Report & remediation: 20

- Presentation: 10

Module A — Foundation & Environment (Weeks 1–4)

Week 1 — Course intro, lab environment & Linux basics

Learning objectives:

- Install/boot provided lab VMs, set up host-only networking, basic Linux navigation and file perms, Git basics.

Topics: VirtualBox/VMware, Kali & Windows lab OVA, host-only vs NAT, basic shell, chmod, chown, ssh.

Tools: VirtualBox/VMware, Kali Linux, Windows 10 VM, Git, VS Code, Putty/WSL.

Lab (step-by-step):

1. Install VirtualBox (or VMware). Import Forebrain lab OVA.

2. Create two VMs: fore-kali and fore-win. Configure host-only network.

3. Start Kali, set network, create user student. From host, ssh student@<kali_ip> to verify SSH.

4. Run basic commands: ls -la, chmod 700 script.sh, chown student:student file.

5. Initialize a Git repo, commit a README describing VM IPs and networking steps.

Deliverable / assessment: Git repo link with README + 10-question practical quiz.

Week 2 — Networking fundamentals & packet capture

Objectives: TCP/IP, ports, subnetting, capture & analyze packets, identify 3 handshake packets.

Topics: OSI vs TCP, IPv4 subnetting, ICMP, TCP 3-way handshake, basic routing.

Tools: Wireshark, tcpdump, ip / ifconfig, netstat.

Lab:

1. On Kali, run tcpdump -i <iface> -w week2.pcap while browsing example.com.

2. Open pcap in Wireshark, filter tcp.handshake and export three handshake packets as evidence.

3. Subnetting exercise: calculate networks for /24, /26 given addresses; submit a one-page solution.

Assessment: PCAP file + one-page analysis.

Week 3 — Windows internals & basic scripting

Objectives: Read Windows event logs, PowerShell basics, simple Bash & Python automation for parsing logs.

Topics: Event Viewer, Windows services, Get-EventLog, Python file parsing.

Tools: PowerShell, Python3, Notepad++, Git.

Lab:

1. In Windows VM: Get-EventLog -LogName Application -Newest 200 | Export-Csv applogs.csv.

2. In Kali: write parse_logs.py that reads applogs.csv and summarizes error counts by event ID.

3. Push script and sample output to Git.

Assessment: Script correctness + README + 5-min demo.

Week 4 — Security fundamentals & cryptography primer

Objectives: CIA triad, symmetric/asymmetric crypto, hashing, TLS handshake basics.

Topics: AES vs RSA, SHA family, TLS/SSL basics, certificates.

Tools: openssl, Wireshark (TLS filter), test HTTPS server.

Lab:

1. Generate RSA keypair & self-signed cert: openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365.

2. Configure simple HTTPS server with cert and connect via browser; capture TLS handshake in Wireshark.

3. Short quiz: match crypto methods to use cases.

Assessment: Generated cert files + 10-question crypto quiz.

Module B — Core Security Concepts (Weeks 5–8)

Week 5 — OWASP Top 10 & vulnerable web app

Objectives: Identify OWASP Top 10, exploit basic web vulnerabilities on a lab app, document remediation.

Topics: Injection, XSS, CSRF, auth flaws.

Tools: OWASP Juice Shop (Docker), DVWA, Burp Suite (Community/Pro), sqlmap.

Lab:

1. Deploy Juice Shop: docker run -d -p 3000:3000 bkimminich/juice-shop.

2. Complete first 5 challenges; for each exploited vuln write stepwise PoC and remediation notes in Git.

Assessment: Challenge reports + peer review.

Week 6 — Threat modeling & secure design

Objectives: Create a threat model for a 3-tier app using STRIDE, propose mitigations.

Tools: draw.io, threat modeling template.

Lab:

1. Create system diagram for supplied app; apply STRIDE to identify top 8 threats.

2. Submit threat model PDF with prioritized mitigations.

Assessment: Threat model + peer feedback.

Week 7 — Web app recon & manual testing

Objectives: Use Burp for manual discovery, parameter tampering, session analysis.

Tools: Burp Suite, Firefox devtools, sqlmap (as needed).

Lab:

1. Intercept login flow, tamper parameters (roles, IDs), document impact.

2. Use Burp intruder for a simple fuzz test and record findings.

Assessment: Lab report + Burp logs.

Week 8 — OSINT & passive reconnaissance

Objectives: Passive recon techniques and asset discovery for engagement prep.

Tools: Amass, Subfinder, Shodan, theHarvester.

Lab:

1. Run amass enum -d forebrain-lab.local -o amass.txt.

2. Build an asset list (domains, IPs, potential services) and produce OSINT report with at least 10 findings and risk notes.

Assessment: OSINT report + accuracy check.

Module C — Offensive Tooling & Exploitation (Weeks 9–12)

Week 9 — Scanning & vuln assessment

Objectives: Use Nmap, OpenVAS/Nessus to identify services & vulnerabilities; triage findings.

Tools: Nmap, OpenVAS (or Nessus), Nikto.

Lab:

1. nmap -sS -sV -p- -oA scan_week9 <target_ip>.

2. Run OpenVAS scan, export top 10 findings, create remediation priority list.

Assessment: Scan outputs + prioritized remediation doc.

Week 10 — Exploitation & Metasploit basics

Objectives: Learn Metasploit workflow: search exploit → configure → obtain shell → post-exploit.

Tools: Metasploit, searchsploit, msfvenom.

Lab:

1. Use searchsploit for a lab VM vuln. Setup exploit in Metasploit, configure LHOST/LPORT, and get a reverse shell.

2. Document steps and cleanup.

Assessment: Lab demo + logbook.

Week 11 — App exploitation: SQLi, XSS & business logic

Objectives: Exploit SQLi and XSS, craft PoCs, and recommend fixes.

Tools: Burp, sqlmap, custom Python PoC.

Lab:

1. Exploit SQLi to extract a sample user table (lab DB).

2. Demonstrate stored XSS in Juice Shop and show impact (cookie theft simulation).

Assessment: PoC artifacts + remediation plan.

Week 12 — Post-exploitation & professional reporting

Objectives: Persistence techniques, lateral movement basics, and professional pentest report writing.

Tools: Meterpreter, SSH, Markdown/LaTeX templates.

Lab:

1. Simulated post-exploitation: establish persistence, enumerate the network for pivot.

2. Produce a full pentest report with executive summary, technical findings, CVSS scoring, and remediation steps.

Assessment: Graded report (use Project rubric).

Module D — Blue Team & Detection (Weeks 13–16)

Week 13 — SIEM fundamentals (ELK / Splunk)

Objectives: Deploy ELK or Splunk trial, ingest logs, write searches and dashboards.

Tools: ELK stack in Docker, Filebeat, Splunk Free trial.

Lab:

1. Deploy ELK via Docker Compose. Configure Filebeat on fore-win and fore-kali to forward logs.

2. Create a Kibana dashboard showing authentication events and top IPs.

Assessment: Dashboard screenshot + saved search queries.

Week 14 — IDS & network detection (Zeek/Suricata)

Objectives: Deploy Suricata/Zeek and write detection rules.

Tools: Suricata, Zeek, Wireshark.

Lab:

1. Run Suricata against provided PCAP; examine eve.json.

2. Write a Suricata rule to detect a custom pattern (e.g., specific user-agent + URI), test on PCAP.

Assessment: Rule + explanation + test results.

Week 15 — Endpoint forensics basics

Objectives: Memory triage, disk artifact discovery, basic timeline reconstruction.

Tools: Volatility 3, Autopsy, FTK Imager (or OS analog).

Lab:

1. Acquire a memory image from a lab VM (or use supplied memory dump). Use Volatility to list running processes and extract suspicious DLLs.

2. Load a disk image into Autopsy and find evidence of suspect binaries and timeline events.

Assessment: Forensic findings report + timeline.

Week 16 — Threat hunting & detection engineering

Objectives: Build hunting hypotheses, write Sigma rules / detection logic, reduce false positives.

Tools: YARA, Sigma rule templates, Kibana/Splunk searches.

Lab:

1. Formulate a hypothesis (e.g., credential dumping via mimikatz). Create a Sigma rule to detect relevant behavior.

2. Run a hunt over lab logs and submit IOC list + proof.

Assessment: Sigma rule correctness + hunt report.

Module E — Advanced Topics (Weeks 17–20)

Week 17 — Active Directory attacks & defenses

Objectives: Understand AD architecture and common attacks: Kerberos, Pass-the-Hash, Golden Ticket; map attack paths using BloodHound.

Tools: BloodHound, PowerView, Rubeus, Mimikatz (lab only, controlled).

Lab:

1. Run BloodHound collector (Sharphound) on lab AD to generate graph.

2. Identify top 3 attack paths and propose mitigations (ACL hardening, tiered admin model).

Assessment: AD report + mitigation checklist.

Week 18 — Cloud security: AWS/GCP basics

Objectives: Cloud IAM review, discover misconfigurations, audit cloud security posture.

Tools: AWS Free Tier sandbox (lab account), ScoutSuite, Prowler.

Lab:

1. Run ScoutSuite on the lab AWS account and identify top 5 risky items.

2. Implement a policy change to fix one critical misconfig and document.

Assessment: Cloud audit report.

Week 19 — API & mobile security basics

Objectives: Test REST APIs for auth flaws, test mobile app static analysis basics.

Tools: Postman, Burp, MobSF (Mobile Security Framework).

Lab:

1. Intercept API calls to lab app with Burp and demonstrate broken auth (e.g., predictable JWT key).

2. Run MobSF on supplied APK and report findings.

Assessment: API findings + remediation plan.

Week 20 — Research methods & experiment design

Objectives: Frame a research question, build reproducible experiment, write a 2-page research proposal for capstone.

Tools: Google Scholar, Zotero, Jupyter Notebook.

Lab:

1. Draft a one-page research proposal for capstone (question, method, dataset, metrics).

2. Implement a small reproducible experiment in Jupyter (e.g., test detection time for two rules).

Assessment: Proposal + Jupyter notebook demonstrating experiment.

Module F — Capstone & Job Prep (Weeks 21–24)

Week 21 — Capstone design, RoE & team formation

Objectives: Form teams, set goals, finalize scope and Rules of Engagement (RoE).

Activities: Team roles (lead pentester, defender, researcher), finalize lab topology, prepare a capstone schedule.

Deliverable: Capstone plan with timeline + RoE signed by all.

Week 22 — Red team offensive phase

Objectives: Execute offensive phase: recon → exploit → persistence; produce timestamped logs and evidence.

Lab:

1. Red team performs controlled attacks as per RoE. Keep detailed logbook (timestamps, commands, outputs).

2. Preserve artifacts and evidence for blue team review.

Assessment: Offensive logbook + PoC artifacts.

Week 23 — Blue team defense & IR phase

Objectives: Detect, triage, contain, and remediate attacks; produce incident report & timeline.

Lab:

1. Blue team uses SIEM/IDS logs and forensics to detect red activity. Contain hosts, remediate, and restore.

2. Produce formal incident report (exec summary, timeline, root cause, remediation).

Assessment: Incident report (graded), detection metrics (time to detect, time to contain).

Week 24 — Final presentations, career day & graduation

Objectives: Present capstone (red/blue timelines + research), portfolio demos, mock interviews & hiring day.

Activities:

1. Team presentations: 20 min presentation + 10 min Q&A.

2. Employer demo booths / mock interviews.

3. Final grading & certificates.

Deliverable: Final capstone submission (report + evidence + research paper) and polished portfolio.

Module-level Slide Decks — Full Slide Text (copy-paste ready)

Each module includes 8–10 slides of copy-ready text for PowerPoint. Examples (Module A shown):

Module A slides include: Title, Learning Objectives, Environment Setup Steps, Linux Quick Commands, Windows Quick Commands/PowerShell, Networking Basics, Lab Deliverables & Assessment, Resources & Cheatsheets.

Weekly instructor notes & class cadence

Live sessions: 2–3 × week (lecture + demo). Labs: scheduled 2× week supervised + weekend self-lab.

Maintain lab reset scripts to restore VMs after destructive labs.

Each week: 1 live mini-quiz (15 min) + lab submission to Git with evidence.

Mandatory policies (to present to students)

All testing must be in Forebrain lab or explicitly authorized targets.

Students must sign Rules of Engagement (RoE) and acceptable use before Week 5.

No external unauthorized scanning or disclosure.

Next steps / Options

1. Export a branded PDF and PPTX from this Word doc.

2. Generate detailed lab playbooks (commands + provisioning scripts) for Weeks 1–8.

3. Create instructor slide decks (PPTX) using the provided slide text.