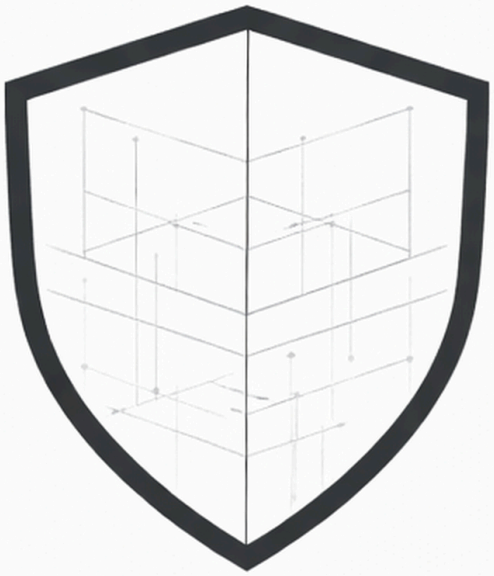
cyberwithjohann



Offensive Security
Transition Roadmap

CYBERWITHJOHANN

Offensive Security Transition Roadmap

A structured progression from security foundations to offensive capability.

"Every system carries the memory of its design.
Every weakness reveals the limits of that design."

CONTENTS

What's inside.

BEFORE YOU BEGIN

How to use this roadmap.

This is not a list of tools to download or certifications to collect. It is a structured way of thinking about how offensive capability is actually built.

Most people trying to move into offensive security do it backwards. They start with tools, chase certifications, and grind random labs, hoping skill will emerge from volume. It rarely does. What emerges is a hobbyist who can follow a walkthrough but cannot reason through an unfamiliar environment.

This roadmap is built on a different premise: offensive capability is layered, and the layers must be built in order. Skip a layer and everything above it becomes fragile.

READ THIS FIRST

Each section builds on the one before it. Resist the urge to jump ahead to exploitation. The professionals who progress fastest are the ones who build the foundation most deliberately.

The four pillars

Everything in this document maps to four pillars that define a real offensive practitioner:

→ **Technical foundations** — networking, operating systems, and systems thinking that let you reason about how environments actually behave.

→ **Security reasoning** — threat modeling, attack surface, and identity logic that give your testing direction.

→ **Offensive methodology** — structured reconnaissance, enumeration, and exploitation grounded in understanding, not imitation.

→ **Professional communication** — the ability to translate technical findings into business risk that organizations act on.

> *The difference between a technician and an operator is not the tools they use. It is whether they understand why those tools work.*

What offensive security actually is.

Offensive security is the disciplined practice of evaluating systems by simulating realistic adversarial behaviour under explicit legal authorisation.

It is not experimentation. It is not curiosity-driven hacking. It is not automated scanning of arbitrary targets. It is structured risk evaluation through controlled adversarial simulation.

At a professional level, offensive security is centred on a single strategic question:

> *If a capable, motivated attacker targeted this organisation, what would realistically occur?*

Answering this requires more than technical skill. It requires systems literacy, architectural awareness, and adversarial reasoning grounded in operational reality.

What professional practitioners understand

- Infrastructure topology and how environments are connected
- Identity propagation paths across systems
- Authentication and authorisation models
- Trust relationships between systems
- Points of boundary enforcement
- Where assumptions exist, and where they can fail

Offensive security is not an isolated discipline. It is built upon cybersecurity fundamentals. The most effective offensive professionals understand defensive controls, governance constraints, regulatory exposure, operational dependencies, and business impact thresholds. Without this foundation, offensive work becomes theatrical rather than strategic.

The five characteristics

Offensive security is defined by five characteristics that separate it from chaotic experimentation:

→ **Structured methodology** — engagements follow disciplined phases, not improvisation.

→ **Legal authority** — activity is contractually scoped and authorised.

→ **Business-impact orientation** — findings are evaluated through risk, not novelty.

→ **Methodology over tools** — tools assist thinking, they do not replace it.

→ **Professional communication** — clarity of reporting is as critical as technical depth.

THE CORE IDEA

Offensive security is the systematic evaluation of whether an organisation's control environment can withstand intelligent pressure. Everything in this roadmap builds toward that capability.

The remainder of this document maps the path to that capability and the foundation it requires. Each layer is deliberate. Each layer matters.

02 — THE FOUNDATION LAYER

Networking, Linux, systems thinking.

Offensive capability is not acquired through exposure to tools. It is built upon structural technical literacy.

Before adversarial reasoning can develop, foundational understanding must be deliberate, deep, and operational. Without this layer, offensive work becomes reactive, superficial, and dependent on automation rather than judgement.

THIS PHASE IS NOT OPTIONAL

This is the layer that determines whether an individual becomes a practitioner or remains a hobbyist. You cannot accelerate past it. You must build it.

Networking depth

Every meaningful attack path crosses infrastructure boundaries. Applications communicate across subnets, authentication traverses identity providers, data flows through segmented networks, and services depend on routing logic and name resolution.

Professional-level networking literacy requires more than certification familiarity. You need the ability to:

- Visualise packet flow across multiple hops
- Understand TCP handshake mechanics and session establishment
- Interpret routing decisions, NAT behaviour, and firewall rule logic
- Reason about VLAN segmentation and trust zones
- Recognise how DNS resolution impacts reachability and redirection
- Understand TLS negotiation, certificate validation chains, and failure scenarios

True competence is behavioural, not theoretical. It requires comfort in observing traffic captures, recognising abnormal session persistence, understanding retransmissions and timing irregularities, and identifying where encryption terminates in a layered architecture.

> *Networking depth enables a crucial shift: thinking in terms of reachability, control enforcement, and boundary traversal.*

NETWORKING — BUILD THESE SKILLS

- [ ] Capture and read traffic in Wireshark until anomalies become obvious
- [ ] Trace a packet mentally from your machine to a remote service and back
- [ ] Explain why a firewall rule permits or denies a given connection
- [ ] Map how DNS, NAT, and routing combine to make a host reachable

Linux and operating system competence

Operating systems are where privilege is enforced or silently inherited. Offensive security evaluates how control boundaries behave under pressure. Without understanding their construction, their resilience cannot be assessed.

A comprehensive understanding of the following is essential:

- File system permission models — owner, group, inheritance
- Execution contexts and privilege levels
- Process spawning and parent-child relationships
- Service startup logic and configuration dependencies
- Scheduled task execution — cron and systemd mechanics
- Environment variable exposure and PATH resolution
- Logging pipelines and event tracing

You should be able to explain why a service runs under a specific user, how privileges propagate between processes, where configuration files introduce implicit trust, and what logging mechanisms would reveal suspicious behaviour.

> *Without operating system literacy, privilege escalation appears arbitrary. With it, it becomes predictable.*

Offensive practitioners do not try things until something works. They understand the intended behaviour and identify where that expectation is violated.

Systems thinking — from components to relationships

This is the layer where most transitions fail. Many professionals can identify individual vulnerabilities. Few can reason through interconnected systems.

Modern environments are not linear. They are graphs.

- Applications depend on databases
- Databases depend on service accounts
- Service accounts inherit group privileges
- Identity providers federate across cloud services
- Cloud IAM policies cascade through nested roles

THE KEY DISTINCTION

A list-based mindset produces vulnerability reports. A graph-based mindset produces attack chains. This is the difference between a technician and an operator.

Systems thinking requires continuous evaluation of relationships: What trusts what? Where does identity originate? Where is privilege elevated, and where is it inherited without verification? What happens if one node in this graph is compromised?

Why this foundation matters

If this layer is weak, tool usage replaces reasoning, enumeration becomes blind scanning, exploitation becomes copy-and-paste, and real-world environments feel overwhelming.

If this layer is strong, exposure becomes predictable, trust relationships become visible, misconfigurations stand out, and lateral movement pathways become logical.

> *This foundation determines the ceiling of your offensive capability.*

RECOMMENDED FOUNDATION RESOURCES	
Networking	Professor Messer Network+, Practical Packet Analysis, Wireshark labs
Linux	OverTheWire Bandit, Linux Journey, TryHackMe
Practice	TryHackMe (start here), HackTheBox Academy (next)
Timeframe	2 to 3 months of deliberate, daily practice

Threat modeling, attack surface, identity.

Before simulating attacks, you must understand security architecture. This is what gives your testing direction instead of randomness.

Threat modelling

Threat modelling provides direction by identifying critical assets, realistic adversaries, potential attack vectors, and business impact. Without it, testing becomes random. With it, it becomes strategic. Offensive professionals must be able to align technical findings with business risk.

Attack surface analysis

Attack surface analysis is the structured mapping of all exposed components within scope. This includes public-facing services, authentication portals, APIs, administrative interfaces, and third-party integrations. The objective is to create a map of exposure that guides structured testing.

Identity and access control

Modern security is identity-driven. A thorough understanding of authentication and authorisation is critical. Many significant breaches originate not from complex exploits but from misconfigured access control.

- Role-based access control models
- Token-based authentication systems
- Federation and single sign-on
- Privilege inheritance
- Session lifecycle management

REMEMBER THIS

Identity misconfigurations often enable privilege escalation and lateral movement. Mastery of identity logic is one of the highest-leverage skills in offensive security.

SECURITY FUNDAMENTALS — BUILD THESE SKILLS

- [ ] Build a threat model for a simple web application from scratch
- [ ] Map the full attack surface of a target you have authorisation to test
- [ ] Explain how OAuth, SAML, and SSO actually pass identity between systems
- [ ] Identify three ways access control commonly breaks in real applications

RECOMMENDED SECURITY FUNDAMENTALS RESOURCES	
Threat modeling	OWASP Threat Modeling, "Threat Modeling" by Adam Shostack
Identity	OAuth 2.0 Simplified, Okta developer docs, Microsoft identity platform docs
Practice	PortSwigger Web Security Academy (free, excellent)
Timeframe	1 to 2 months alongside continued foundation work

> *Many significant breaches originate not from complex exploits but from misconfigured access control.*

Recon, enumeration, exploitation logic.

Once technical and security foundations are solid, structured offensive methodology begins. This is where analysis transitions into controlled adversarial simulation.

Offensive security is not a sequence of disconnected actions. It is a progression: awareness, then context, then boundary failure.

Reconnaissance

Reconnaissance is controlled information gathering within authorised scope. Its purpose is not volume but clarity. It builds environmental awareness before interaction deepens, allowing you to understand the technology stack, infrastructure patterns, deployment architecture, trust boundaries, and exposure points.

> *Without reconnaissance, engagement direction is assumed. With it, engagement direction is intentional.*

Enumeration

Enumeration moves beyond visibility into interpretation. Where reconnaissance identifies what exists, enumeration seeks to understand how it behaves. It involves interpreting service responses, identifying configuration weaknesses, understanding authentication flows, and analysing how applications respond to unexpected input.

Enumeration is analytical, not mechanical. It requires patience, observation, and the ability to connect subtle behavioural cues to architectural assumptions.

Exploitation logic

Exploitation is not a matter of luck. It is the identification of flawed assumptions within system design. A mature practitioner does not simply exploit a vulnerability. They understand the reason for its existence, the boundary that was bypassed, the control that failed, the trust relationship that degraded, and the realistic impact that follows.

> *Without understanding, exploitation becomes imitation. With understanding, it becomes controlled validation.*

Professional capability is demonstrated not by gaining access, but by explaining precisely how and why that access was possible.

OFFENSIVE METHODOLOGY — BUILD THESE SKILLS

- [ ] Run a full recon-to-exploitation chain on an authorised lab box and document your reasoning at each step
- [ ] Explain not just what worked, but why the control failed
- [ ] Practice enumeration until you can predict what you will find before you find it
- [ ] Solve machines without walkthroughs — struggle is where skill forms

RECOMMENDED METHODOLOGY RESOURCES	
Practice	HackTheBox, TryHackMe offensive paths, PortSwigger labs
Certification	eJPT first, then OSCP when foundations are solid
Reference	The Hacker Playbook, Penetration Testing by Georgia Weidman
Timeframe	3 to 4 months of structured practice

Web application and access control concepts.

Web applications dominate modern enterprise environments. They are the primary interface between users, identity systems, APIs, databases, and cloud services. Consequently, they are often the primary attack surface.

Offensive professionals must understand web logic thoroughly, not only HTTP mechanics but the behavioural and architectural assumptions embedded within applications:

- HTTP protocol behaviour and request/response lifecycle
- Session management mechanisms
- Cookie security attributes and token storage models
- Client-side versus server-side validation
- API authentication and authorisation flows
- Cross-origin access behaviour
- Access control enforcement logic

But understanding protocol mechanics is only the beginning. Modern vulnerabilities frequently stem not from memory corruption or exotic payloads but from flawed business logic:

- Broken access control
- Insecure direct object references
- Workflow manipulation
- Improper state transitions
- Inconsistent authorisation checks

These issues are rarely discovered through brute force scanning. They require reasoning. You must be able to ask: What was this workflow designed to prevent? Where is authorisation actually enforced? Is validation server-side or only client-side? Can the order of operations be manipulated?

THINK IN THREE MINDS AT ONCE

A legitimate user navigating workflows. A malicious actor probing for logic gaps. A developer who may have made convenience-based design decisions. Mature web testing happens at the intersection of all three.

> *Exploitation in web contexts is often less about breaking the system and more about identifying where assumptions were trusted without enforcement.*

Subtle logic flaws often create more severe real-world risk than technical injection vulnerabilities. Understanding this distinction separates surface-level testers from mature offensive practitioners.

WEB SECURITY — BUILD THESE SKILLS

- [ ] Complete every lab in the PortSwigger Web Security Academy
- [ ] Find a broken access control flaw through reasoning, not scanning
- [ ] Map where a real application enforces authorisation, and where it only assumes it

Privilege escalation and lateral thinking.

Initial access is rarely the objective. It is a foothold, not a conclusion. The true measure of offensive capability is how far that foothold can realistically be extended.

Privilege escalation

Privilege escalation is the structured assessment of whether an account, process, or system possesses more authority than intended. It is not about forcing elevation. It is about evaluating whether privilege boundaries degrade under scrutiny.

- Group memberships and inherited roles
- Service account permissions
- Misconfigured access control lists
- Token privileges and execution contexts
- Configuration dependencies that implicitly grant authority

> *The gap between intended privilege and actual privilege is where escalation emerges.*

A mature practitioner evaluates not only what an identity can do, but what it was intended to do.

Lateral movement

Lateral movement evaluates the breadth of compromise. It asks: if this identity is controlled, how far does its authority propagate? This requires reasoning across system boundaries — domain trust structures, shared credentials, identity federation pathways, and trust relationships between cloud and on-premise systems.

> *Lateral movement is not random probing. It is structured identity tracing.*

You are mapping where credentials are valid, where tokens are accepted, where trust is implicitly granted, and where logging visibility may be limited. Creative reasoning is essential, but it must remain constrained by architectural reality.

WHERE IT BECOMES ARCHITECTURAL

Privilege escalation and lateral movement transform a local issue into an enterprise risk scenario. This is where technical competence matures into strategic adversarial reasoning.

ESCALATION AND MOVEMENT — BUILD THESE SKILLS

- [ ] Practice Active Directory attack paths in a lab environment
- [ ] Trace how a single compromised credential could propagate through an org
- [ ] Predict escalation paths before testing them, then verify

Reporting and professional communication.

Offensive security creates value only when findings are clearly understood, properly contextualised, and translated into action. Technical discovery alone does not improve security posture. Clarity does.

Translating findings into business risk

A vulnerability is a technical observation. A report transforms it into risk intelligence. Effective reporting answers three questions: What was observed? Why does it matter? What should be done about it?

The most common failure in offensive reporting is excessive technical detail without contextual relevance. Executives do not require payload structure. Engineers do not need vague summaries. Professional communication requires dual fluency: executive-level clarity on business impact, and technical precision for remediation teams.

> *Without exaggeration. Without theatrics. Without ambiguity.*

Structure and discipline

Each finding should include a clear description, reproducible evidence, an explanation of the failed control, a business impact assessment, justification for the risk severity, and practical remediation guidance.

Communicating to multiple audiences

An effective offensive professional understands audience segmentation. Executive stakeholders require risk clarity, prioritisation, and strategic remediation direction. Technical stakeholders require exact reproduction context, configuration details, and validation steps. A senior practitioner moves fluidly between both.

REPUTATION IS BUILT HERE

In enterprise environments, reputation is built as much on communication discipline as on technical depth. Trust determines long-term credibility.

Professional conduct and trust

Offensive security operates on trust. Organisations grant controlled access under legal authorisation. Professional communication reinforces that trust through accurate scope adherence, clear documentation, transparent methodology, and responsible handling of sensitive data.

COMMUNICATION — BUILD THESE SKILLS

- [ ] Write a finding that an executive and an engineer could both act on
- [ ] Translate one technical vulnerability into clear business risk
- [ ] Practice writing without exaggeration or alarmist language

08 — MINDSET

How to think like an ethical attacker.

Ethical offensive work requires discipline. The distinction between a malicious actor and a professional practitioner is not technical skill. It is structure, authorisation, and accountability.

Professional offensive engagements operate within legal authorisation, clearly defined scope, controlled timelines, and documented methodology. There is no improvisation outside agreed boundaries.

An ethical attacker exercises restraint, documenting actions precisely, avoiding unnecessary disruption, and protecting sensitive information encountered during testing. The objective is not to demonstrate personal capability but to evaluate organisational resilience.

> *Technical skill may earn attention. Professional discipline earns credibility.*

Professional maturity is measured by respect for scope, responsible handling of access, clear documentation, and controlled execution. Trust is foundational to long-term success in offensive security.

THE REAL SIGNAL OF SENIORITY

Anyone can gain access in a lab. The professionals organisations trust are the ones who operate with precision, restraint, and accountability under real authorisation.

09 — AVOID THESE

Common mistakes people make.

Most professionals do not struggle due to a lack of intelligence, but because of structural errors in how they build capability.

Jumping into tools without architectural understanding

Tools cannot compensate for weak foundations. A scanner in the hands of someone who does not understand the environment produces noise, not insight.

Over-prioritising certifications over competence

Exams validate knowledge but do not guarantee operational maturity. A certificate proves you passed a test. It does not prove you can reason through an unfamiliar environment.

Ignoring identity systems

Modern breaches frequently stem from access control failures rather than complex exploits. Practitioners who skip identity logic miss the most common real-world attack paths.

Confusing lab success with enterprise readiness

Real environments involve segmentation, monitoring, and operational constraints that labs rarely replicate. Lab skill is necessary but not sufficient.

Neglecting documentation and communication

Discovery without articulation creates limited value. The finding you cannot explain clearly is a finding the organisation cannot act on.

> *Offensive security is layered. Skipping layers produces fragile skillsets. Structured progression produces durable capability.*

10 — THE PATH FORWARD

Your structured path forward.

A deliberate transition requires progression through defined stages. Here is the sequence, with realistic timeframes.

01

Strengthen foundations

2 TO 3 MONTHS

Deepen networking, operating systems, and systems architecture until reasoning replaces guesswork.

02

Strengthen security understanding

1 TO 2 MONTHS

Master threat modelling, exposure analysis, and identity systems.

03

Develop offensive methodology

3 TO 4 MONTHS

Practice structured reconnaissance, analytical enumeration, and logical exploitation.

04

Refine through feedback

ONGOING

Engage guided environments, refine reporting clarity, and develop strategic thinking.

WHY STRUCTURE BEATS VOLUME

Many professionals struggle not because they lack intelligence, but because they lack structure and feedback. Offensive capability develops faster under structured guidance than through isolated experimentation.

FINAL NOTE

Built deliberately, layer by layer.

Offensive security is not accelerated through shortcuts. It is built deliberately, one layer at a time.

Technical foundations. Security reasoning. Structured methodology. Mastery of identity. Clear communication. Professionals who progress intentionally develop durable capability. Those who rush often plateau.

> *You cannot accelerate past the foundation. You can only build it.*

Wherever you are in this progression right now, the next step is the same: build the current layer deliberately before reaching for the next one. That discipline is what separates the people who make this transition from the people who talk about it.

This roadmap shows you the path. The next step is having the full plan to walk it.

FIND YOUR STARTING POINT

Where are you right now?

Before you walk the path, you need to know where you are standing on it. Most people waste months learning the wrong layer because they never honestly assessed their foundation.

Read each statement below. Check the ones you can do confidently and explain to someone else. Be honest. The gaps are your starting point.

Layer 1 — Foundations

- [ ] I can trace a packet from my machine to a remote service and back, naming each hop.
- [ ] I can explain why a Linux service runs under a specific user and how privilege propagates.
- [ ] I can read a traffic capture and identify what is abnormal.
- [ ] I think about environments as graphs of trust, not lists of machines.

Layer 2 — Security reasoning

- [ ] I can build a threat model for an application from scratch.
- [ ] I understand how OAuth, SAML, and SSO pass identity between systems.
- [ ] I can map a full attack surface and explain where exposure concentrates.

Layer 3 — Offensive methodology

- [ ] I can run a full recon to exploitation chain and explain why each control failed.
- [ ] I can find a broken access control flaw through reasoning, not scanning.
- [ ] I can write a finding an executive and an engineer could both act on.

How to read your score

0–3 **Start at the foundation**

You are at the beginning. This is the best place to be, because everything you build now will be solid. Begin with Phase 1 and do not skip ahead.

4–7 **Reinforce, then advance**

You have real foundations but gaps remain. Identify which unchecked items are holding you back and close them before moving into methodology.

8–10 **Refine and specialise**

You are operating at a real level. Your next gains come from feedback, specialisation, and professional communication. This is where structured guidance compounds fastest.

THE HONEST TRUTH

Most people who think they are at Layer 3 are actually at Layer 1 with good tool familiarity. There is no shame in that. There is only cost in pretending otherwise. Build the layer you are actually on.

REFERENCE

The tools that actually matter.

Tools do not make a practitioner. But knowing which ones professionals actually reach for, and why, saves you from drowning in the hundreds that do not matter.

Learn what each category does before you learn any specific tool. The category is the concept. The tool is just one implementation of it.

CATEGORY	WHAT IT DOES	WHERE TO START
Recon	Maps the target environment and exposure	**Nmap**
Web testing	Intercepts and manipulates web traffic	**Burp Suite**
Directory discovery	Finds hidden paths and endpoints	**ffuf, gobuster**
Traffic analysis	Reads packets to understand behaviour	**Wireshark**
Exploitation	Validates findings under authorisation	**Metasploit**
Password attacks	Tests credential strength	**Hashcat, John**
AD enumeration	Maps Active Directory relationships	**BloodHound**
Scripting	Builds custom tooling for the task	**Python, Bash**

> *A tool in the hands of someone who understands the environment is leverage. The same tool in the hands of someone who does not is noise.*

BUILD YOUR ENVIRONMENT

Setting up your practice lab.

You cannot build offensive capability on theory alone. You need an environment where you can break things safely and repeatedly, with no legal risk.

1. **Get a virtualisation layer**
   Install VirtualBox or VMware. This lets you run isolated machines on your own computer with no risk to anything real.

2. **Set up an attack machine**
   Kali Linux or Parrot OS. This is your operating base. Learn to live inside it until the command line feels natural.

3. **Add vulnerable targets**
   Start with intentionally vulnerable machines from VulnHub, then progress to the structured environments on TryHackMe and HackTheBox.

4. **Build an Active Directory lab**
   Once foundations are solid, build a small Windows domain. Most enterprise compromise happens through identity, and this is where you learn it safely.

ONE RULE, NO EXCEPTIONS

Only ever test systems you own or have explicit written authorisation to test. Everything in this roadmap assumes legal, authorised environments. There are no exceptions to this. Your career depends on it.

CERTIFICATIONS

Which certifications actually matter.

Certifications do not make you capable. But the right ones, taken at the right time, open doors and prove you can perform under pressure. The wrong ones waste money and time.

CERT	BEST FOR	WHEN	DIFFICULTY
eJPT	First real offensive credential	After foundations	Entry
PNPT	Practical, report-driven testing	Intermediate	Medium
OSCP	The industry baseline for pentest roles	After methodology is solid	Hard
CRTP	Active Directory and red team focus	After OSCP path	Medium
Burp Certified	Web application specialists	Web focus track	Medium

> *Take eJPT to prove you can start. Take OSCP to prove you can perform. Everything between is about your chosen direction.*

THE MISTAKE TO AVOID

Do not collect certifications to feel like you are progressing. A certification you rushed to pass without building the underlying skill is a liability in an interview. Earn the skill first. The certificate should confirm what you already know.

TRAIN YOUR REASONING

Think like an operator.

Offensive capability is reasoning under uncertainty. These scenarios are not about tools. They are about how you think. Sit with each one before reading the principle beneath it.

SCENARIO ONE

You gain access to a low-privilege account on a server. What is your first instinct?

The technician immediately runs privilege escalation scripts. The operator first asks: what is this account for, what does it trust, and what was it never supposed to reach? Reasoning before action.

SCENARIO TWO

An application returns a slightly different error for valid versus invalid usernames. Why does this matter?

A small behavioural difference reveals an assumption the developer trusted. Enumeration is the art of noticing what the system tells you without meaning to.

SCENARIO THREE

You find a critical vulnerability. The engagement ends tomorrow. What matters most?

Not the exploit. The explanation. A finding nobody can understand or act on creates no value. Your ability to communicate the risk clearly is the deliverable.

> *Tools change every year. Reasoning compounds for a career.*

YOUR FIRST 90 DAYS

A plan you can start tomorrow.

Direction without a schedule is just a wish. Here is a concrete structure for your first 90 days, whatever layer you are starting from.

DAYS 1 TO 30

Foundations

- → Networking fundamentals: complete a structured course end to end
- → Live in a Linux terminal daily until it feels natural
- → Capture and read your own traffic in Wireshark
- → Solve one easy box per week and document your reasoning in writing

DAYS 31 TO 60

Security reasoning and web

- → Work through the PortSwigger Web Security Academy systematically
- → Build a threat model for one application from scratch
- → Study how identity and access control actually break
- → Continue weekly boxes, now without walkthroughs

DAYS 61 TO 90

Methodology and direction

- → Run full recon to exploitation chains and write a clean report for each
- → Begin eJPT preparation if foundations feel solid
- → Choose your direction: web, network, Active Directory, or cloud
- → Build the habit of explaining every finding in plain business terms

REFERENCE

Key terms, clearly defined.

Speak the language precisely. Vague terminology signals a vague understanding. Here are the terms that matter, defined the way professionals use them.

Attack surface

The complete set of points where an unauthorised actor could attempt to enter or extract data from an environment.

Enumeration

The analytical process of understanding how discovered systems behave, beyond simply knowing they exist.

Lateral movement

Extending compromise across systems by tracing where an identity's authority is accepted.

Privilege escalation

Assessing whether an account or process holds more authority than it was intended to have.

Threat model

A structured view of what matters, who might attack it, how, and what the impact would be.

Trust boundary

The line where one system begins relying on another, and where assumptions can fail.

EVERYTHING IN ONE PLACE

Your resource library.

Every platform, book, and channel referenced in this roadmap, organised so you never have to search for where to go next.

PRACTICE PLATFORMS	
Beginner	TryHackMe, OverTheWire, PortSwigger Academy
Intermediate	HackTheBox, VulnHub, HackTheBox Academy
Active Directory	GOAD, HackTheBox Pro Labs, home AD lab

FOUNDATIONAL LEARNING	
Networking	Professor Messer, Practical Packet Analysis
Linux	Linux Journey, OverTheWire Bandit
Web	PortSwigger Web Security Academy

REFERENCE READING	
Methodology	The Hacker Playbook, Penetration Testing (Weidman)
Threat modeling	Threat Modeling (Shostack)

WHAT YOU JUST RECEIVED

Let's be honest about what this is worth.

You did not pay for this roadmap. But that does not mean it has no value. Here is what you would have spent piecing this together on your own.

The structured progression Months of trial and error, mapped into a sequence	**Priceless**
Curated resource library No more drowning in 100 conflicting sources	**$200+ in time**
Certification roadmap Knowing what to take and when, saving wasted exams	**$500+ saved**
Self-assessment framework Knowing exactly where to start	**Months saved**
90-day action plan A schedule you can start tomorrow	**$100+ value**
You paid	**$0**

This roadmap gives you the path. But a path is not the same as a plan for getting hired. Knowing what to learn is different from knowing how to turn it into a job, a salary, and a career.

> *The roadmap teaches you the skill. The Complete Guide teaches you how to get paid for it.*

THE NEXT STEP

The roadmap shows the path. The guide gives you everything else.

COMPLETE GUIDE TO GET INTO CYBERSECURITY

From zero to hired.

99 pages covering exactly what to learn, in what order, and how to turn it into a real cybersecurity job. CV strategy, interview frameworks, hiring manager expectations, and a 90-day action plan.

$19.90

99 pages · Instant download · PDF and EPUB

Get the Complete Guide

cyberwithjohann.com/guide

If it saves you one month of going in the wrong direction, it has already paid for itself many times over.

ABOUT

CyberWithJohann.

An educational platform built to help serious IT and cybersecurity professionals transition into offensive security with structure, not shortcuts.

CyberWithJohann emphasises the development of comprehensive skill rather than tool memorisation. The focus is on building systems-level understanding, identity-driven security reasoning, structured adversarial methodology, professional reporting discipline, and enterprise-aligned offensive capability.

The work is created by Johann Lahoud, a cybersecurity engineer working across offensive security, security operations, and regulatory frameworks in large financial environments. The objective is simple: equip serious people with offensive skillsets that hold up in real organisational environments.

STAY CONNECTED

Free roadmap, career quiz, and the Complete Guide are all available at cyberwithjohann.com. New content published regularly across TikTok, Instagram, and LinkedIn.

---

contact@cyberwithjohann.com · cyberwithjohann.com