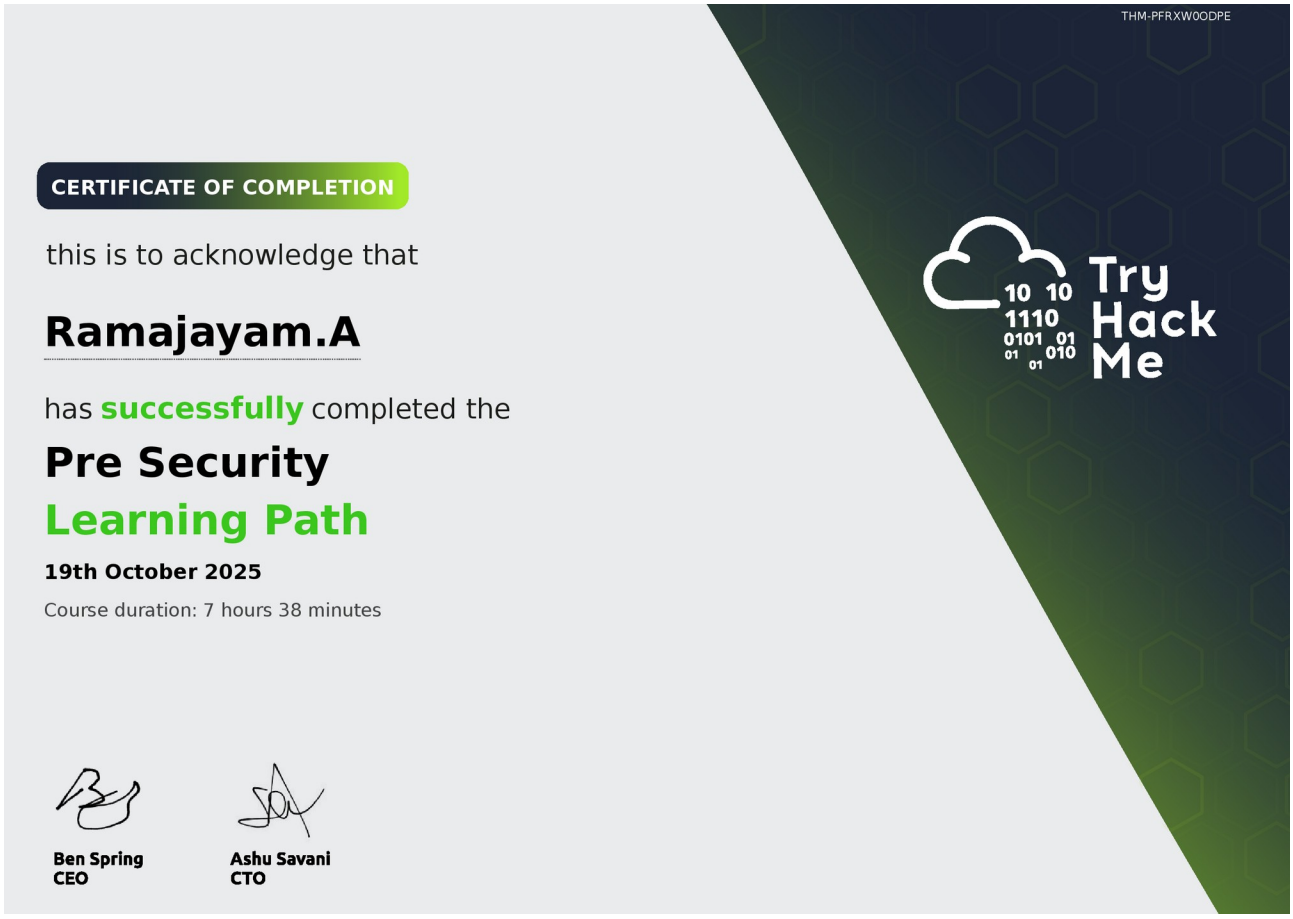

Section 1: Introduction to Cybersecurity

Module Parts: Offensive Security Intro · Defensive Security Intro · Careers in Cyber Security

In this first section of the TryHackMe Introduction to Cyber Security module,

I learned about what cybersecurity really means, how both offensive and defensive sides work, and what kinds of careers are out there.

Offensive Security

Offensive security is like the “red team” side — you think like an attacker, try to find weaknesses in systems, applications or networks. The module gave me a taste of hacking a simulated vulnerable web app, for example, which was pretty eye-opening.

What I learnt:

The mindset of probing for vulnerabilities rather than just building defenses.

That offensive work often involves scanning, enumeration, exploitation, and post-exploitation steps.

Using tools and concepts to break into systems (in a safe, legal lab) helps you understand what defenders must prepare for.

It’s important to stay ethical: the goal is to find and fix vulnerabilities, not exploit them maliciously.

Defensive Security

Defensive security is the flip side: protecting systems, monitoring, incident response, threat intelligence, etc. The module introduced concepts like how defenders work in a SOC (Security Operations Center), how investigations happen, and how tools like SIEMs or forensics might be used.

What I learnt:

Defenders must understand attacker methods to build proper protections.

Defensive work often means detecting anomalies, responding quickly to incidents, and patching systems.

The role is less flashy than hacking, maybe, but critical and continuous.

Both sides (offense + defense) are needed for a real-world security posture.

Careers in Cybersecurity

Finally, the module walked through different career paths in cyber: things like Penetration Tester (offense), SOC Analyst (defense), Incident Responder, Malware Analyst, Red Team, etc. I realised there are many options depending on your interest.

Key take-aways:

You should pick the side (offensive or defensive) depending on whether you like “breaking things” or “protecting things”.

Many roles overlap: someone in defense might use offensive knowledge and vice-versa.

Knowing basic tech (Linux, networking, web) is essential whichever career you pick.

This module helped me narrow my focus — I’m leaning towards [state your interest here] because [state reason].

What I Learned from This Section

I now understand the big picture: cybersecurity isn’t just hacking, it also isn’t just installing antivirus — it’s a broad field involving planning, execution, detection, and response.

I gained vocabulary: “offensive”, “defensive”, “SOC”, “red team”, “blue team”, “incident response” etc.

I realised why fundamentals (networking, OS, web) matter so much — they are common across both sides.

I started building a security mindset: always ask “what if an attacker did this?” and “how would I detect/respond if this happened?”.

I found some direction: I’m now more motivated to continue to the next modules knowing what kind of career path I might aim for.

Notes & Minor Mistakes I Made

I assumed hacking is always “cool & fast” — but I realised defence work is slower, many times boring, but super important.

I skipped writing notes at first and lost some details; now I’m documenting everything.

I tried a scan too early without enumeration — I learnt that proper order (recon → enumeration → exploit) matters.

I found some commands confusing at first (on Linux terminal) but writing them down and repeating helped.

Section 2: Network Fundamentals

This section covers how networks work — because in cyber security, if you don’t know how devices talk and how data flows, you’re missing the basis for everything else.

What is Networking?

In this room I learnt that a network is simply a set of connected devices — more than just computers in an office, but thousands or even millions of things linked together.

A big part was understanding that the internet is just a massive network of many smaller networks.

What I learnt:

Devices need unique addresses (like IP and MAC) so they can be found and communicated with.

That networks exist in everyday life (traffic lights, power grids, social networks) and computing networks follow the same ideas.

In cyber security, understanding networking fundamentals matters before you go after an exploit: you must know how data travels.

Minor mistakes I made: I initially thought “network = internet only”, but then learnt networks can be local/private too.

Intro to LAN

This room introduced Local Area Networks (LANs) — basically networks in a limited area (home, school, office). Key points: topologies, devices like hubs/switches/routers, and how devices in a LAN talk to each other.

What I learnt:

Different topologies have different strengths and weaknesses (star, bus, ring etc).

Protocols like ARP (address resolution) and DHCP (automatic address assignment) are part of how LAN devices find each other and get on the network.

Devices like switches vs hubs operate differently — a hub broadcasts to all, a switch is smarter.

What I found interesting: I realised that many attacks/pivoting start inside a LAN (if the attacker gets into one machine, they might move laterally).

Minor mistake: I mixed up the concept of “router vs switch” at first — writing the notes helped fix that.

OSI Model

Here I learnt about the OSI model (Open Systems Interconnection) — a 7-layer framework for network communication.

The layers are (from 7 to 1): Application, Presentation, Session, Transport, Network, Data Link, Physical.

What I learnt:

Why the OSI model is so useful: it gives a common language for how data moves, and when I see exploitation or defence steps later, I can map them to layers.

Things like “encapsulation” (data gets additional headers/footers at each layer) make sense now.

For example: when I use a web browser (App layer), the data eventually goes down to wires and signals (Physical layer).

Mistake I made: I tried to memorise all layers in the wrong order; now I know to use a mnemonic and practise the correct sequence.

Packets & Frames

This room helped me understand the basic units of network communication: packets and frames.

What I learnt:

A packet is a block of data at the network layer or above, which includes IP addresses. A frame is at the data link layer (below network) and may not contain IP address info.

The difference between TCP and UDP: TCP is reliable (three-way handshake → SYN, SYN/ACK, ACK) and UDP is faster/less reliable (used for video, voice) in this context.

Ports, protocols, network addressing details all matter when thinking about how systems communicate.

What I found useful: Realising that when I see “moving laterally” in a pentest, it’s often about understanding how these packets/frames are structured and how devices on the network trust each other.

Mistake I made: Initially I thought “frame = packet” but now I see the difference and why it matters.

Extending Your Network

The final room in this section covered how you can broaden or link networks — beyond a simple LAN — including devices, routers, forwarding, VLANs etc.

What I learnt:

What port forwarding is: when you forward a port on your router so external traffic can reach a specific internal device.

That routers connect networks (layer 3) and switches primarily work in a LAN (layer 2) and this distinction matters when you're trying to exploit or defend.

How network segmentation (VLANs, subnets) can help security (or if misconfigured, can be a point of weakness).

What I liked: This room felt more "applied" — like how you'd actually connect home/office devices, or how an attacker might pivot between networks.

Mistake I made: I skipped the quiz once thinking it's easy, but the port forwarding questions were trickier than I expected.

What I learned from Section 2

My understanding of networking grew from vague to structured: I now know how devices connect, why addresses matter, and what layers they pass through.

Networking is not just cable + internet: it has structure, layers, units, devices, and each part can be attacked or defended.

These fundamentals are crucial for cyber security: without them, many tools feel like “magic” rather than understandable.

I started to build a workflow: when I encounter a network in a lab I’ll now think — “What layer? What unit (packet/frame)? What device? How is networking configured?”

Mistakes are normal: I improved by documenting, revisiting topics, and doing the labs again.

Section 3: How the Web Works

Topics: DNS in Detail · HTTP in Detail · How Websites Work · Putting It All Together

In this section I got shown how the web actually functions behind the scenes — not just “open browser & go” but all the plumbing that makes websites possible.

DNS in Detail

DNS (Domain Name System) is basically the internet’s phone-book: converting human-friendly domain names into IP addresses so your computer knows where to send the request.

What I learnt:

Every domain has an IP address behind it; without DNS you’d need to remember numeric addresses.

Hierarchy of domains (TLDs, subdomains etc) matters; DNS record types like A, AAAA, MX, CNAME exist.

How the lookup happens: local cache → recursive/resolver → authoritative server → return address.

Mistake I made: I skipped reading the quiz question about “what is the maximum length of a subdomain” (it was 63) and got it wrong.

HTTP in Detail

This room deep-dived the HTTP protocol — how browsers send requests, servers respond, status codes, headers, methods (GET/POST) etc.

What I learnt:

How a request flows: browser → DNS resolution → TCP connection → HTTP request → server responds with HTML/CSS/JS etc.

Understanding status codes (200, 404, 500) helps when you do web-app vuln labs.

Headers and methods mean a lot: knowing them helps you craft better exploit attempts or spot misconfigurations.

Mistake I made: I originally thought “POST = always bad” but now I know it’s just a method; security depends on how it’s used.

How Websites Work

Here I learnt how browsers render websites, how front-end/back-end work together, and how web apps are layered.

What I learnt:

The difference between static content (HTML/CSS) and dynamic (JS + backend).

How web servers use virtual hosts so one server can host many domains.

That multiple components (CDN, DB, load-balancer) may be involved “behind the scenes”.

Mistake I made: I looked only at HTML and ignored network traffic for a while; now I inspect network tab in browser when studying.

Putting It All Together

The final room in the section connected DNS + HTTP + app logic + infrastructure into one big picture: how a simple “open this website” comes to life.

What I learnt:

Step-by-step chain: browser asks DNS → gets IP → establishes connection → HTTP request → server sends data → browser renders.

Real-world extras: load-balancers, CDNs, databases, WAFs can all influence how websites behave and how attackers might target them.

When doing web-pentesting, understanding this full flow helps you know where to look for weaknesses.

Mistake I made: I didn’t map the “database” part of a website at first; but now I know backend = big deal for web-security.

What I learned from Section 3

The web isn’t magic — there are concrete steps and components behind it.

Having network + OS fundamentals (which I already had or am building) directly helps in web labs.

I feel more confident now: when I see a “404” or a weird DNS record, I know what to ask next: “why did this fail?” “what’s the chain?”

I need to practice web-app modules next with this foundation.

Section 4: Linux Fundamentals

This section was about getting comfortable with the Linux operating system — very important because many servers are Linux, many pentest tasks target Linux.

What I learnt:

Linux history & distributions: knowing what makes Linux different and why it’s so popular in security.

Essential shell commands: ls, cd, cat, grep, find, chmod, sudo, etc.

File system & permissions: how Linux handles files/folders, how users/groups work, why permissions matter.

Basic operators and redirection: >, >>, |, &&, ; — chaining commands, filters, searching.

I realised I need to practise daily: breaking habit of “Windows mindset” and going deeper into command line.

Mistake I made: I tried using GUI tools too soon in Linux labs; this slowed me down. I now commit to using terminal for first 10-15 min of each session.

Section 5: Windows Fundamentals

Because Windows dominates in business environments, this section helped me get comfortable with the Windows OS, which is super useful for defence and attack scenarios.

What I learnt:

The Windows Desktop/UI, NTFS file system, UAC (User Account Control), Control Panel & Settings.

User accounts & permissions: Administrator vs Standard user, groups, how permissions affect system changes.

Key utilities: Task Manager, System32 folder, environment variables, registry, resource monitor.

For security: built-in tools matter (Windows Update, Windows Security, BitLocker etc) — attackers often abuse built-in tools (living off the land).

Mistake I made: I neglected studying the registry for a while; now I spend 5 minutes each session checking a registry key just to build familiarity.