Uptake of Innovative Security-as-a-Service Solutions

Practical Penetration Testing

with the Montimage Pentesting Toolbox (MMT-Pentester)

Montimage · CyberSuite Academy | Intermediate · ~20 hours

Instructor: Edgardo Montes de Oca

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

Agenda

01 Course Introduction

02 Module 1: Pentesting Foundations, Ethics & Methodology

03 Module 2: Getting Started with MMT-Pentester

04 Module 3: Reconnaissance & Attack Scenarios

05 Module 4: Monitoring, Detection & Reporting

06 Module 5: Advanced: AI Agents & Tool Integration

07 Final Hands-On Assessment (Capstone)

08 Course Wrap-Up and Evaluation

Uptake of Innovative Security-as-a-Service Solutions

Course Introduction

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

Overall picture

- What is pentesting

  Penetration testing is how organisations find their weaknesses before attackers do.

- Objectives of course

  Learning the complete penetration-testing lifecycle — reconnaissance, controlled exploitation, monitoring, and reporting

  Using the Montimage MMT-Pentester toolbox, a web platform that brings recon, attack simulation, monitoring, and reporting together in one guided interface.

- Advertisement:

  Everything is hands-on, but every exercise runs **only against provided, authorised lab targets** — never against real or third-party systems.

What you will learn

- Describe the penetration-testing lifecycle and the legal and ethical rules of authorised testing.
- Perform reconnaissance with Nmap and map a target's attack surface.
- Configure and run controlled attack scenarios (SSH brute-force, HTTP content discovery (dirb)) in a safe lab.
- Monitor executions in real time and interpret intrusion-detection alerts.
- Generate professional reports and write actionable remediation recommendations.
- Use AI-assisted testing and understand the platform's integrated tool ecosystem.

Who this course is for

- This course suits IT and cybersecurity students, SOC analysts, network security engineers, and professionals seeking practical penetration-testing skills.

- A basic understanding of networking (IP addresses, ports, HTTP, SSH), comfortable using command lines, and familiarity with cybersecurity concepts are recommended. No prior experience with machine learning or AI is required, every concept is introduced before it is used.

How the course is structured

The course is organised as five modules plus a final hands-on assessment. Each module is a short lesson. Modules 1-5 and Capstone are followed by an exercise or quiz.

Module	Title	Estimated Time
1	Pentesting Foundations, Ethics & Methodology	3-4 hours
2	Getting Started with MMT-Pentester	2 hours
3	Reconnaissance & Attack Scenarios	3-4 hours
4	Monitoring, Detection & Reporting	2 hours
5	Advanced: AI Agents & Tool Integration	2 hours
Capstone	Final Hands-On Assessment	4-5 hours
Wrap-up	Course Wrap-Up and Evaluation	30 minutes

You can follow the course at your own pace; the full effort is less than **20 hours**, ideally spread over **several days**.

Recommended procedure to be followed

1. View each video of each module (or, if you prefer, go through the PowerPoint slide set and read the Word text).

2. Pause the video if some term or concept is not clear for you and search the web or ask a chatbot for more details. Active participation is important for acquiring higher knowledge.

3. For Modules 2-5 and Capstone, follow the instructions to do the exercises and quiz.

4. At the end, provide your feedback. This is important for us to be able to improve the course.

Assessment, Certification & Ethics

Assessment and certificate

Each module ends with a quiz; the pass mark is 7/10. After the capstone you complete a final quiz that draws on all the modules. Complete every module and the final assessment with the required grades and you earn a digital certificate of completion from the CyberSuite Academy.

A note on safety and ethics

When working with network traffic data, you may encounter sensitive information. Treat all data as confidential. The sample PCAP files provided for this course are sanitized datasets designed for educational purposes. Always follow your organization's data handling policies when working with real network captures.

Uptake of Innovative Security-as-a-Service Solutions

Module 1

Pentesting Foundations, Ethics & Methodology

PART 1
Foundations of Penetration Testing

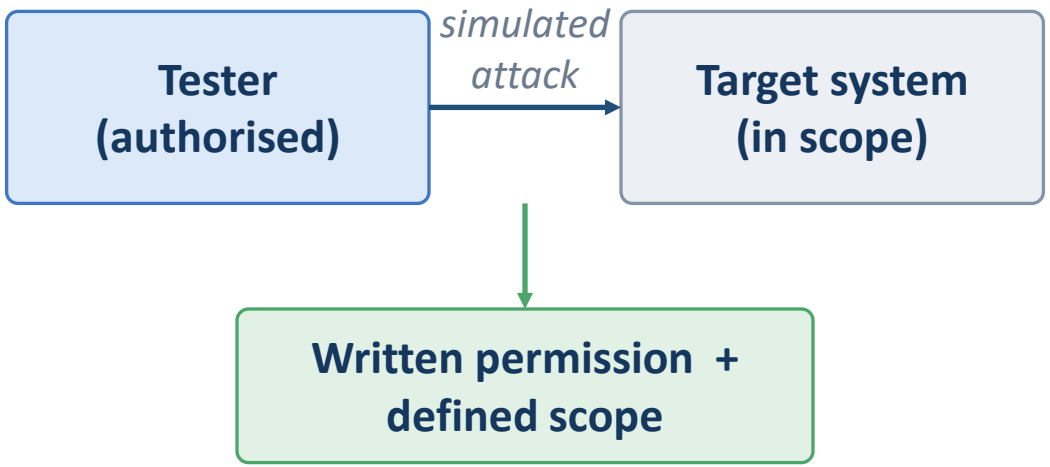
Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

What Is Penetration Testing?

- A penetration test ("pentest") is an authorised, simulated attack on a system, network, or application — to find and demonstrate exploitable weaknesses before a real attacker does.
- It answers a practical question: "If someone tried to break in, could they — and how far would they get?"
- The goal is constructive: surface real risk, prove impact, and give defenders a prioritised list of fixes.
- Always conducted with explicit, written permission and a defined scope.

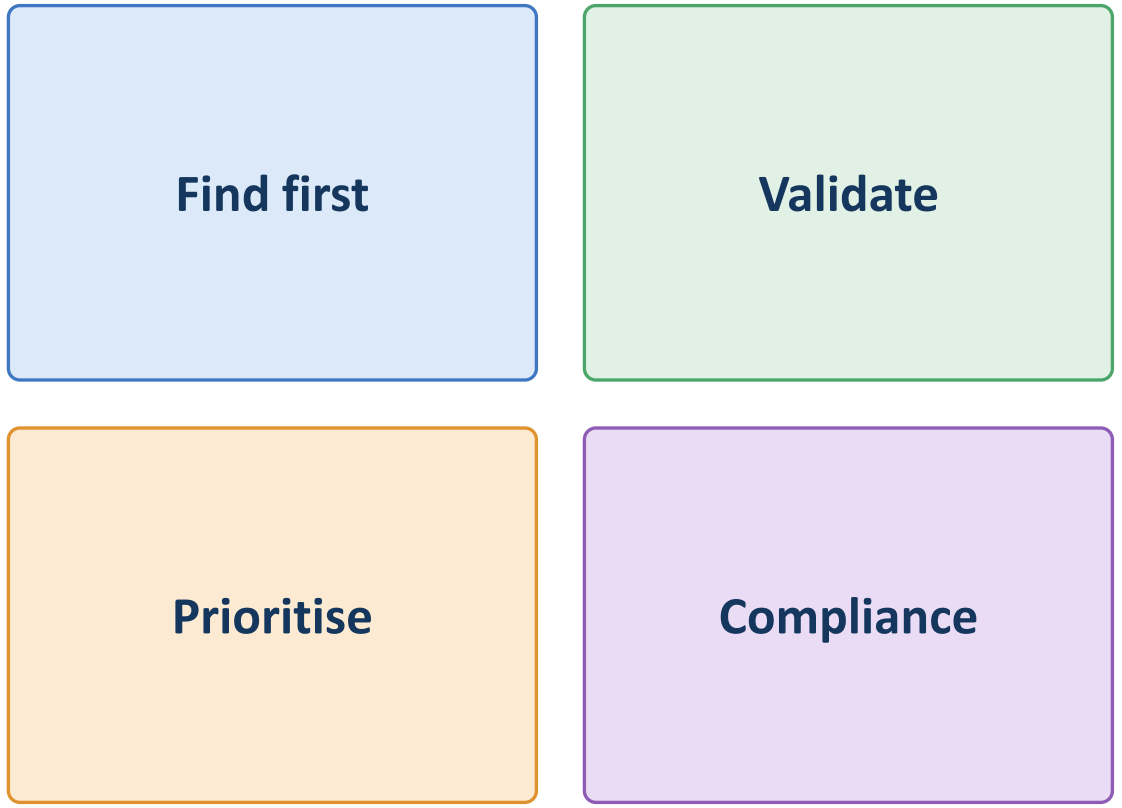
An authorised, simulated attack



Key idea: a pentest proves real, exploitable risk under controlled, authorised conditions — it is not random hacking, and not just a list of theoretical issues.

Why Penetration Testing Matters

- Find before they do
  - discover exploitable flaws while you still control the outcome.
- Validate defences
  - confirm that firewalls, patching, and monitoring actually work under attack.
- Prioritise risk
  - turn a long list of theoretical issues into a short list of what truly matters.
- Compliance & trust
  - PCI-DSS, ISO 27001 and SOC 2 expect regular testing; clients and regulators ask for proof.



Scanning vs Pentest vs Red Teaming

Three points on one spectrum of depth, realism and cost — not three unrelated activities.

Vulnerability Scanning

- Automated, broad, fast
- Lists potential weaknesses
- Little/no exploitation
- Run often

Penetration Testing

- Human-driven, scoped, deeper
- Confirms flaws by safe exploitation
- Shows real impact
- Run periodically

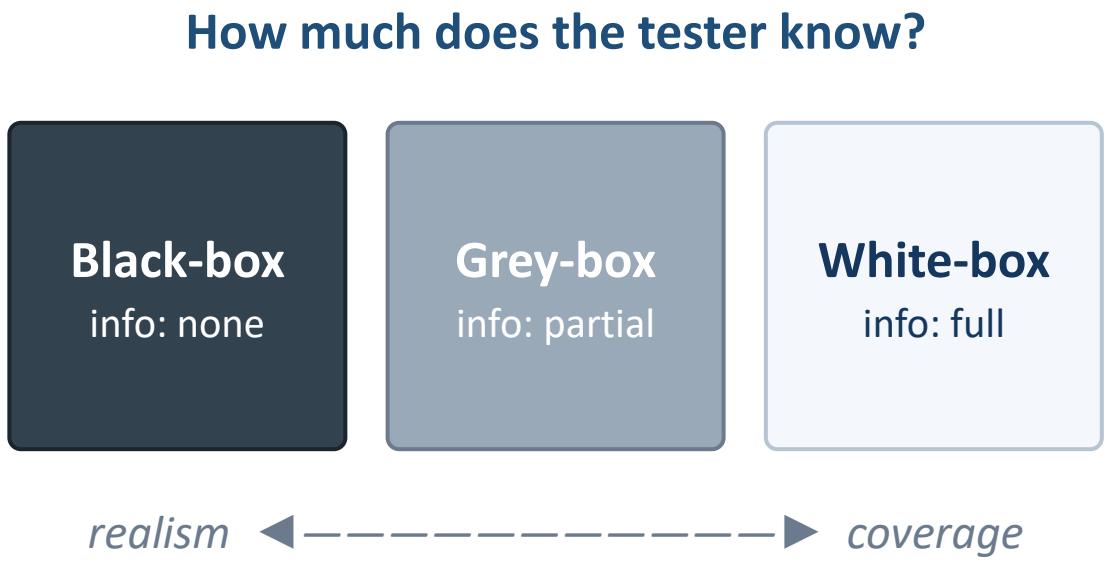
Red Teaming

- Goal-oriented adversary emulation
- Tests people, process & technology
- Stealthy, often over weeks
- Run occasionally

increasing depth · realism · cost · time

Knowledge Depth: Black / Grey / White Box

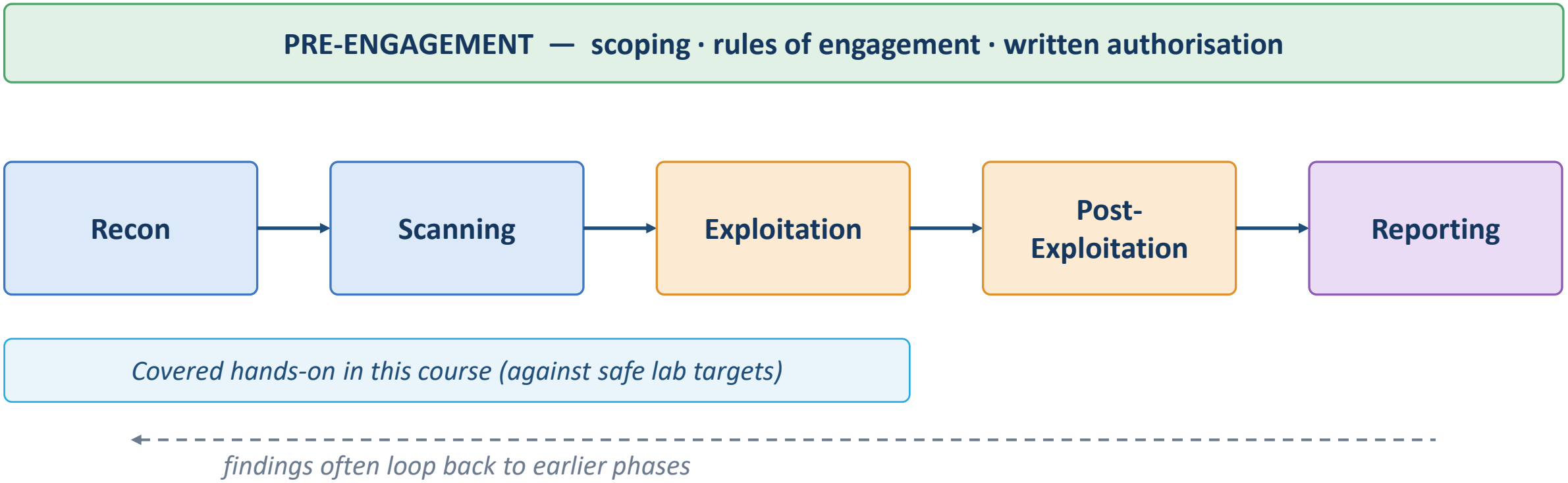
- Black-box — no inside information; simulates an external attacker. Realistic, but slower and may miss internal flaws.
- Grey-box — partial information (a low-privilege account or a network diagram). Balances realism and coverage — most common in practice.
- White-box — full information: source code, configs, credentials. Most thorough coverage, less about realism.
- The right choice depends on the question being asked and the time/budget available.



In practice: grey-box is the usual sweet spot — enough context to be efficient, enough realism to be meaningful.

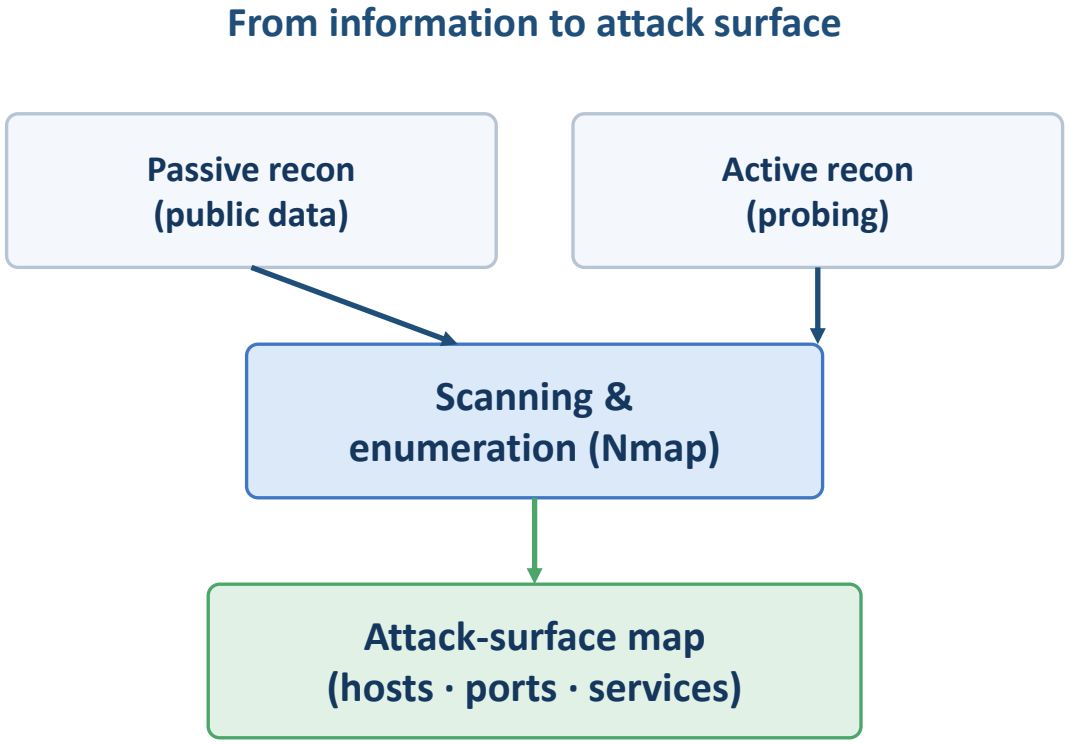
The Phases of an Engagement

A pentest follows a repeatable lifecycle. Earlier phases inform later ones — and findings often loop you back.



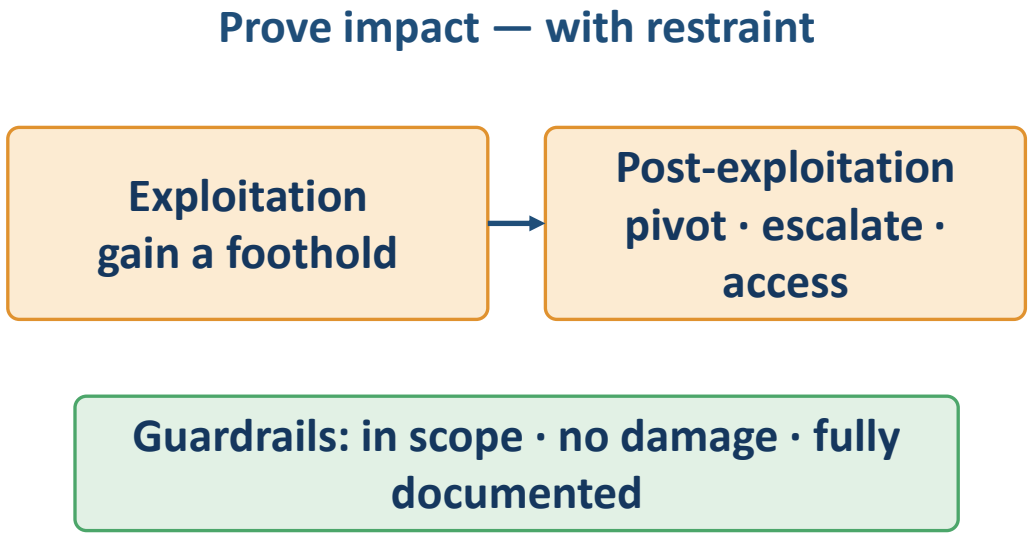
Phase Detail: Recon & Scanning

- Reconnaissance — gather information about the target. Passive (public data, no direct contact) and active (directly probing the target).
- Scanning & enumeration — map the live attack surface: hosts, open ports, services, versions and likely entry points.
- Output: a clear picture of what exists and where the soft spots might be.
- In the labs you will use Nmap to enumerate services and map a minimal attack surface against a provided lab target.



Phase Detail: Exploitation & Post-Exploitation

- Exploitation — safely use a weakness to gain access or cause a defined effect, proving the risk is real.
- Post-exploitation — once in, assess what an attacker could actually do: pivot, escalate privileges, access data — all within scope.
- Restraint matters: avoid damage, stay in scope, and document every action.

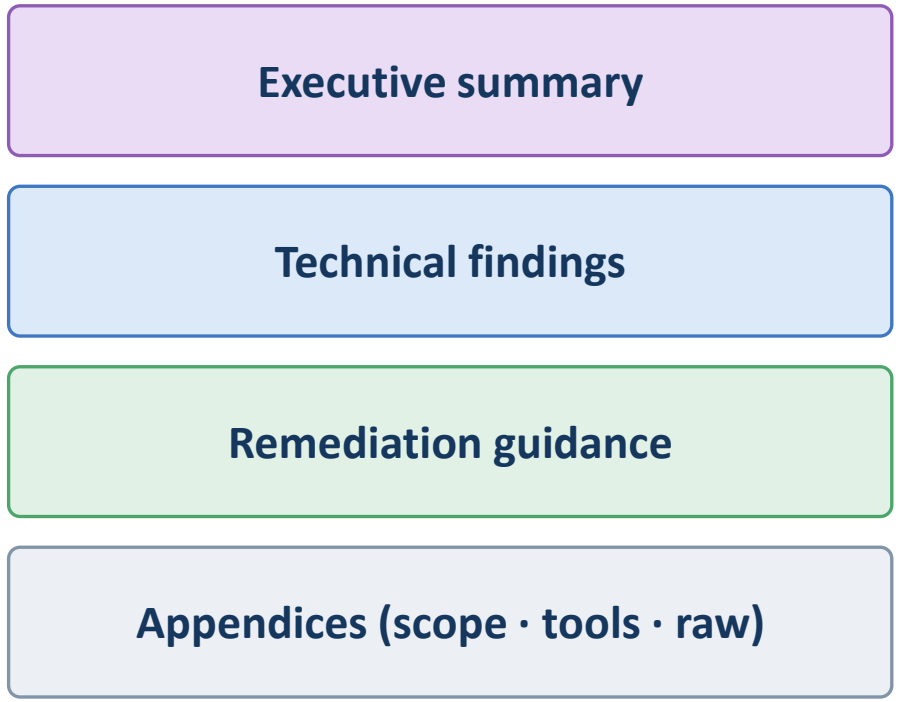


Lab examples: an SSH brute-force against a discovered service, and an HTTP content-discovery scan with dirb against a discovered web service.

Phase Detail: Reporting & Deliverables

- Reporting turns activity into value — often the most important phase for the client.
  - Executive summary — risk in plain language for decision-makers.
  - Technical findings — each issue with evidence, severity and reproduction steps.
  - Remediation guidance — prioritised, actionable fixes.
  - Appendices — scope, methodology, tools, raw output.
- MMT-Pentester auto-generates PDF and Excel reports from your executions.

Anatomy of the deliverable



Ethics, Authorisation & Scope

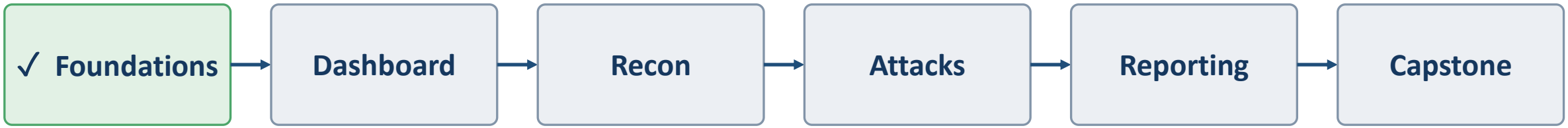
- Everything in this course assumes explicit authorisation and a defined scope — the rules of engagement.
- Test only the provided, isolated lab targets. Never test real or third-party systems.
- Stay in scope, minimise harm, protect any data you encounter, and document what you do.
- Unauthorised testing is illegal in most jurisdictions, regardless of intent.

⚠ Non-negotiable

- Only the provided, isolated lab targets.
- Always with written authorisation and a defined scope.
- Minimise harm · protect data · document everything.
- Unauthorised testing is illegal — regardless of intent.

Where This Course Goes Next

- You now have the mental model: why we test, the types of testing, and the phases of an engagement.
- Next parts introduce the MMT-Pentester Dashboard and its data model (Project → Campaign → Scenario → targets + attacks).
- You will then run the three guided scenarios hands-on and observe results on the dashboard.
- Every tool and click maps back to a phase you learned today.



You are here — Part 1 complete

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Module 1
Pentesting Foundations, Ethics & Methodology

PART 2
Legal, Ethics & Rules of Engagement

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

Why this module comes before the tools

- Penetration testing means deliberately attacking a system — the only thing that separates it from a crime is permission.
- A skilled tester with no authorisation is a liability; an authorised tester with modest skills is an asset.
- Everything in MMT-Pentester runs ONLY against provided, isolated lab targets, with explicit authorisation — never real or third-party systems.
- This is for education and authorised testing only.

Permission is the whole game: the same action is a professional service with authorisation, and a crime without it.

Authorisation is mandatory

- "Authorisation" = explicit, written permission from someone who actually owns or controls the target.
- A verbal "go ahead", or an email from the wrong person, is NOT authorisation — get it in writing, from an authorised signatory.
- Authorisation must exist BEFORE the first packet is sent, and must cover exactly what you intend to do.
- No authorisation, no test. There is no grey area.

The legal risk of unauthorised testing

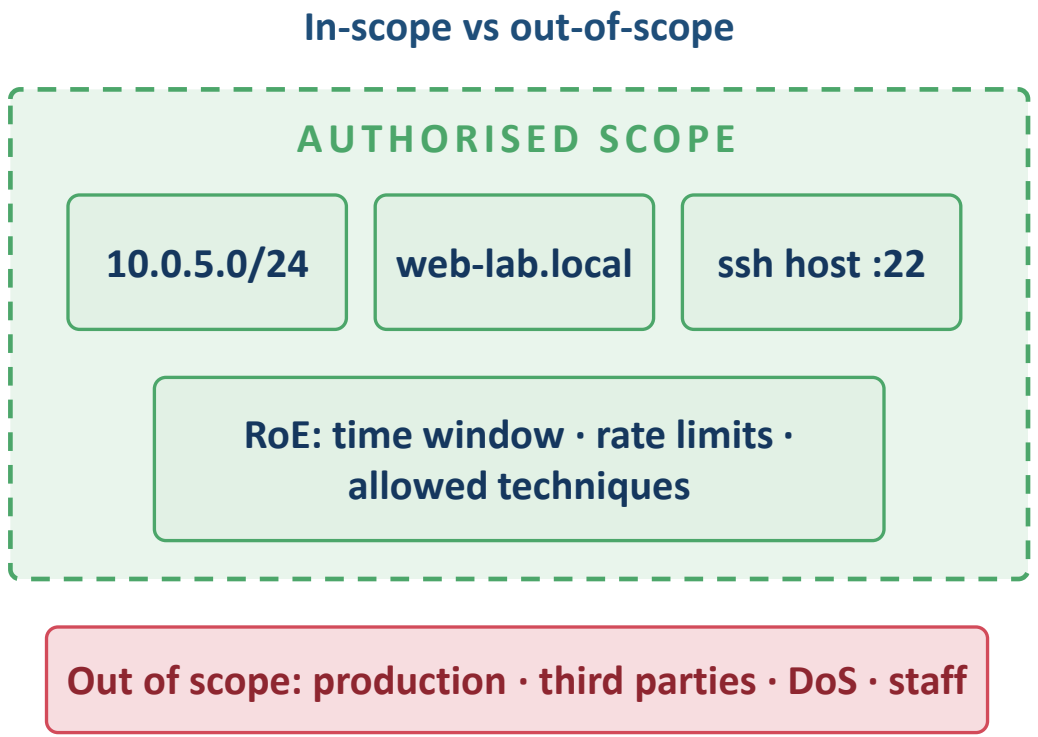
- Unauthorised access, even "just scanning", can breach computer-misuse laws (CFAA in the US, the Computer Misuse Act in the UK, EU national equivalents).
- Consequences are real: criminal charges, civil liability, fines, and a destroyed career.
- "I was only curious / I didn't change anything" is not a defence — intent and access are what matter.
- Consent + a signed contract are what turn an attack into a legitimate service.

Contracts and consent

- A proper engagement is backed by a written agreement: a contract / Statement of Work plus an authorisation ("get-out-of-jail") letter.
- Key clauses: who authorises, what is permitted, liability limits, confidentiality (NDA), and how findings are handled.
- The authorisation letter is the document you keep at hand during testing as proof of permission.
- Consent can be withdrawn — if the client says stop, you stop immediately.

Scope and Rules of Engagement (RoE)

- Scope = exactly which assets you may touch: IP ranges, hosts, domains, applications, accounts.
- Anything not explicitly listed as in-scope is out-of-scope by default.
- RoE define the "how and when": time windows, allowed techniques, prohibited actions, rate limits, escalation contacts.
- Commonly prohibited: social engineering, physical access, destructive payloads, DoS, touching production data.



Data handling and confidentiality

- During a test you will see sensitive data: credentials, customer records, internal configs. Treat all of it as confidential.

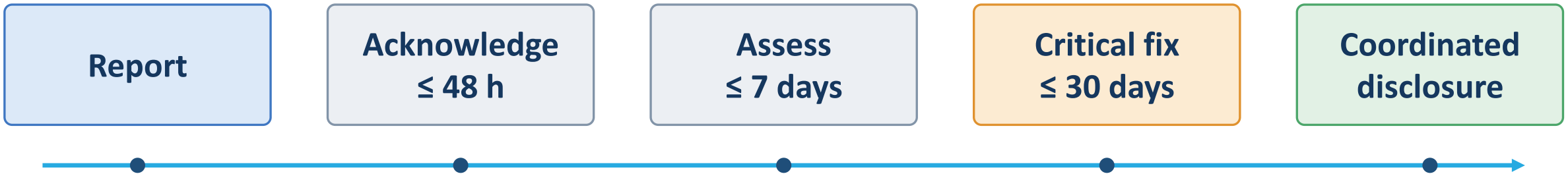
- Collect only what you need to prove a finding; capture minimal evidence (screenshots, hashes, redacted samples) rather than exfiltrating real data.

- Store evidence and reports encrypted and access-controlled, and delete them per the agreed retention period.

- The platform follows the same principle: change default credentials, use a strong JWT secret, run behind HTTPS, restrict network access.

Responsible disclosure (SECURITY.md)

- If you find a real vulnerability (in MMT-Pentester or elsewhere), report it responsibly — do not post it publicly.
- MMT-Pentester process: email security@montimage.com; never open a public GitHub issue.
- Include a description, steps to reproduce, impact/severity, and optional mitigations.



The MMT-Pentester boundary: authorised use only

- The platform is designed for educational and research use in authorised, controlled environments only.
- Unauthorised use against systems you do not own or have explicit permission to test is illegal and outside this project's scope.
- The three guided scenarios (Nmap recon, SSH brute-force, HTTP content discovery (dirb)) run ONLY against the provided lab targets.
- Before every exercise, confirm: is this target authorised, in scope, and within the allowed time window?

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Module 1
Pentesting Foundations, Ethics & Methodology

PART 3
Methodologies & Frameworks

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

Why methodologies matter

- A methodology is a repeatable, agreed plan for how a test is run — not just which tools you use.
- Benefits: full coverage (you don't forget steps), repeatability, comparable results, and a shared vocabulary with clients and defenders.
- Without one, testing becomes "poke at random things until something breaks" — unscoped, unauditable, hard to report.
- Frameworks also map to authorisation and scope: a phase-based plan is what you agree with the client.

The four frameworks at a glance

They overlap on purpose — PTES is what a tester does; Kill Chain & ATT&CK describe what an attacker does; OWASP zooms into web specifics.

PTES

Penetration Testing Execution Standard — end-to-end engagement process (7 phases).

OWASP Testing Guide

Deep checklist for testing web applications (pairs with the OWASP Top 10).

Cyber Kill Chain

Lockheed Martin — 7 steps of how an attacker progresses through an intrusion.

MITRE ATT&CK

Knowledge base of real-world adversary tactics and techniques.

PTES: the engagement process

Seven phases, in order — note the bookends: it starts with authorisation/scope and ends with a report.

MMT-Pentester's Project → Campaign → Scenario + auto-reports line up with phases 2, 5 and 7

Process-oriented: PTES tells you the order of work and what each phase should produce — the same discipline this course teaches.

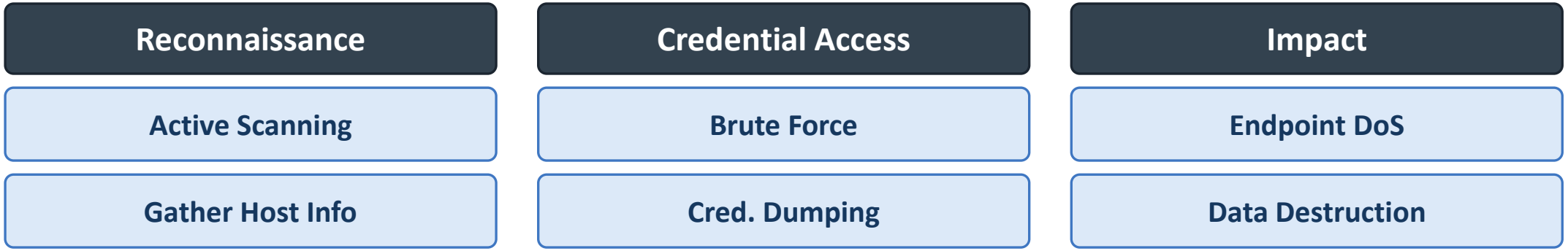
OWASP Testing Guide & the Cyber Kill Chain

- OWASP Testing Guide (WSTG): a structured catalogue of web tests — info gathering, configuration, authentication, session management, input validation. Use it when the target is a web app.
- It pairs with the OWASP Top 10: the guide is how to test, the Top 10 is what commonly goes wrong.
- The Cyber Kill Chain models a single linear intrusion — and it is defender-friendly: break any link and the attack stops.

Lockheed Martin Cyber Kill Chain — a linear intrusion, broken at any link

MITRE ATT&CK: tactics vs techniques

- Tactic = the adversary's goal for a step (the "why"): Reconnaissance, Initial Access, Credential Access, Impact …
- Technique = how they achieve it (the "how"). A tactic holds many techniques; a technique can serve several tactics.
- Sub-technique = a more specific variant. Unlike the Kill Chain's fixed 7 steps, ATT&CK is granular and non-linear.



tactics (columns) · techniques (cells)

How to read a technique ID

- Format: T#### [.###] — T then a 4-digit technique number, optionally a dot + 3-digit sub-technique.
- T1595 = Active Scanning (Reconnaissance). T1499 = Endpoint Denial of Service (Impact).
- IDs are stable identifiers — cite them in reports so clients and defenders can look up detection and mitigation directly.

T1110.001

T = technique prefix	1110 = Brute Force (Credential Access)	.001 = Password Guessing (sub-technique)

Mapping MMT-Pentester to the frameworks

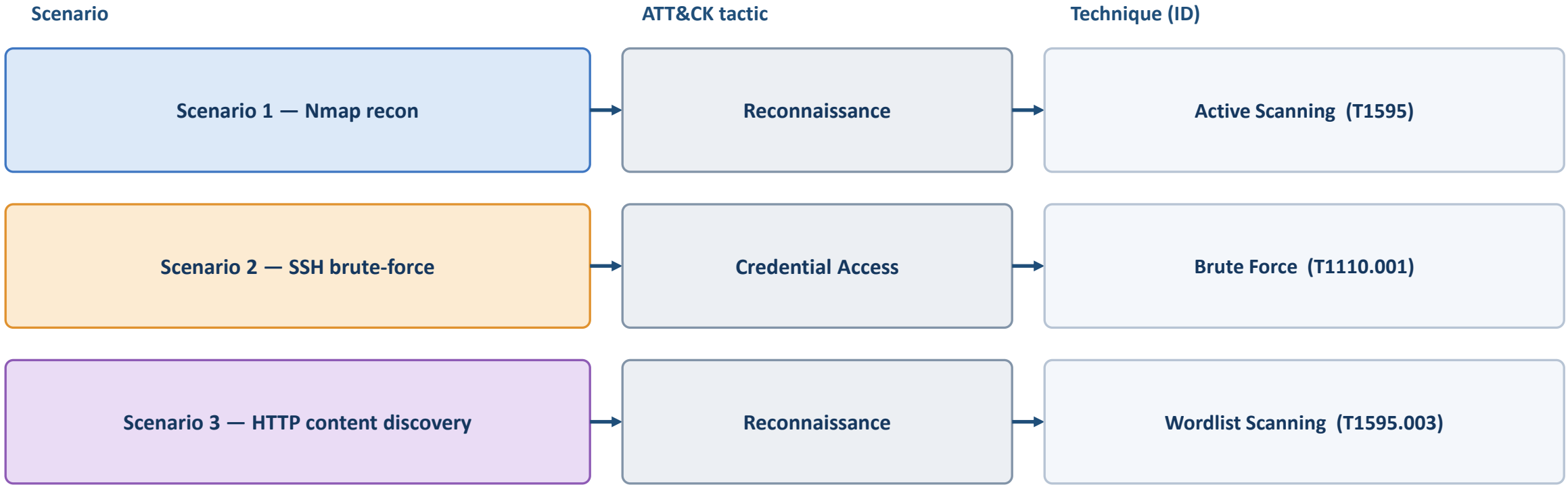
- The platform's flow is recon → attack → report — PTES compressed: intelligence gathering → exploitation → reporting.

- Each guided Scenario you run = one or more ATT&CK techniques exercised against an authorised lab target.

- The Reports view (PDF/Excel) is your PTES phase-7 deliverable — tag findings with ATT&CK IDs there.

- Same workflow, three lenses: PTES tells you the order, ATT&CK names what you did, the report communicates it.

The 3 guided scenarios as ATT&CK tactics



A mini kill chain in three labs: recon to find the door, credential access to open it, content discovery to map it — only ever against provided, isolated lab targets.

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Module 2
Getting Started with MMT-Pentester

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

What Is MMT-Pentester?

- The MMT-Pentester Dashboard (Montimage Pentesting Toolbox, v1.0.0) is a web platform to plan, orchestrate and execute security tests and attack simulations.

- Instead of running scanners and attack tools by hand in separate terminals, you drive them all from one browser interface — and watch results stream back live.

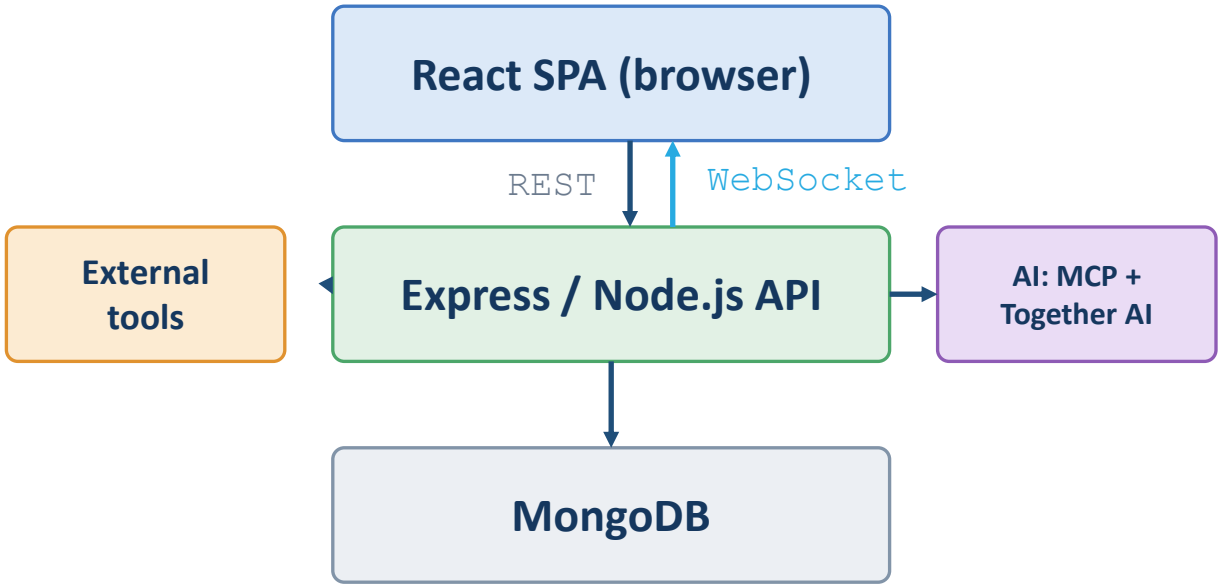
- It is built for controlled labs: every test runs against provided, isolated targets you are authorised to test.

- Think of it as mission control for an authorised engagement: define the work, launch it, watch it, report on it.

The Architecture in Beginner Terms

- Front end — a React single-page app in your browser (fast, responsive, dark/light).
- Back end — an Express / Node.js API: logins, storage, starting the tools.
- Database — MongoDB stores projects, campaigns, scenarios, users, audit logs.
- Live updates — WebSocket channels push events and raw tool output, no refresh.
- Tools & AI — external tools run as processes; an AI assistant (MCP + Together AI) analyses output.

How the platform fits together



How the Pieces Talk

- Your browser sends requests to the REST API (e.g. "log me in", "create a project") — dev API at http://localhost:3000/api.

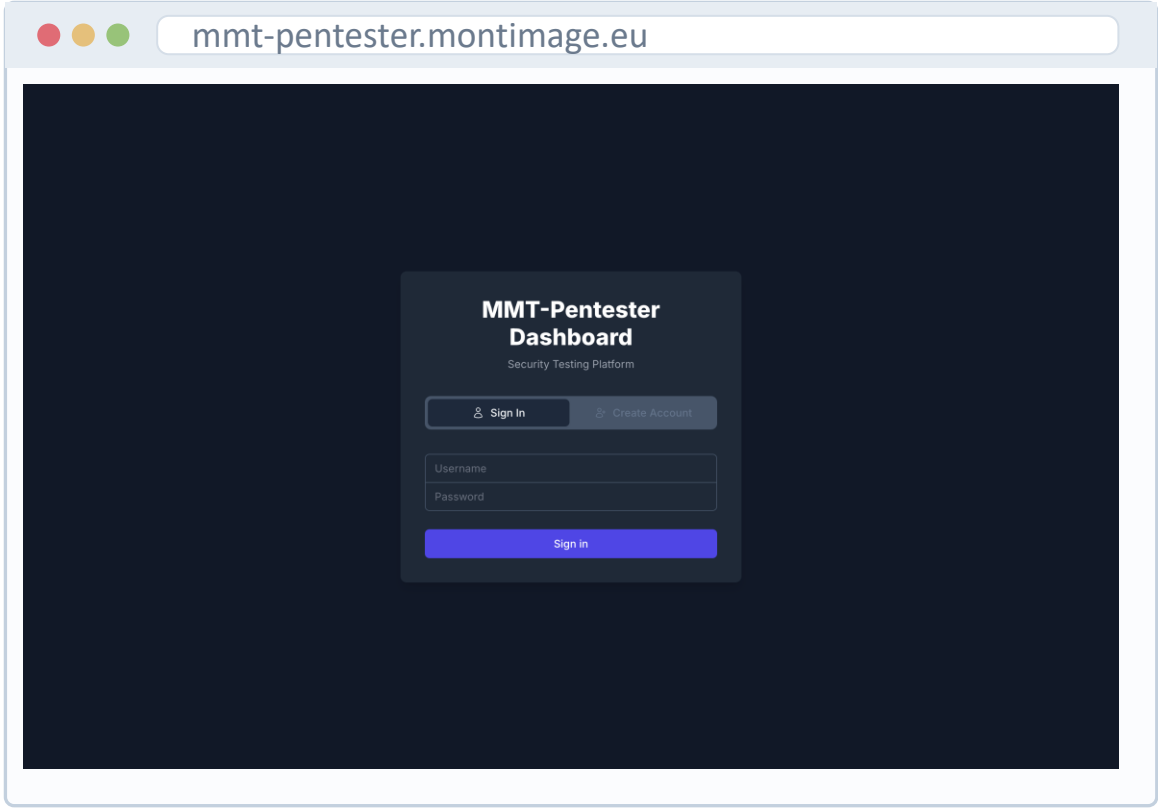
- For anything live, two WebSocket channels run in parallel: Socket.io for campaign/scenario events, and a raw ws channel (port 9090) for live tool output.

- The API starts each external tool as a child process and streams its output back over WebSocket in near real time.

- The AI layer (MCP + Together AI) sits beside the API to analyse results and power the autonomous Agent.

First Login & Changing the Password

- The platform ships with a single super-admin account: username admin, password admin123456.
- On first login you are required to change this default password — every install, first thing.
- Behind the scenes: bcrypt-hashed passwords (12 rounds), JWT sessions that are IP-tracked, account lockout after 5 failed logins.
- The default credentials are public knowledge — leaving them unchanged is the same as having no password.



UI Tour: The Five Views

1. Dashboard

Status overview, recent activity, quick entry points.

2. Projects

Where engagements live: projects, campaigns, team access.

3. Scenarios

Define & run tests: targets, attacks, live output.

4. Agent

AI-assisted autonomous mode that orchestrates steps.

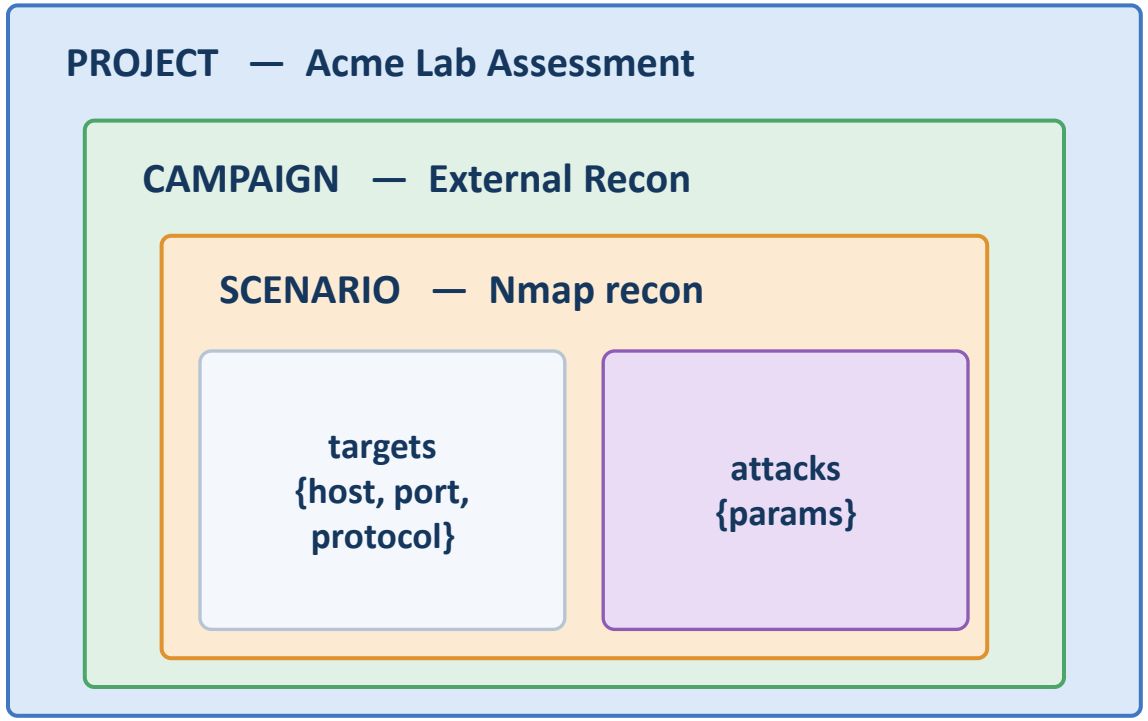
5. Reports

Auto-generated PDF and Excel reports.

The five top-level views you navigate throughout the course

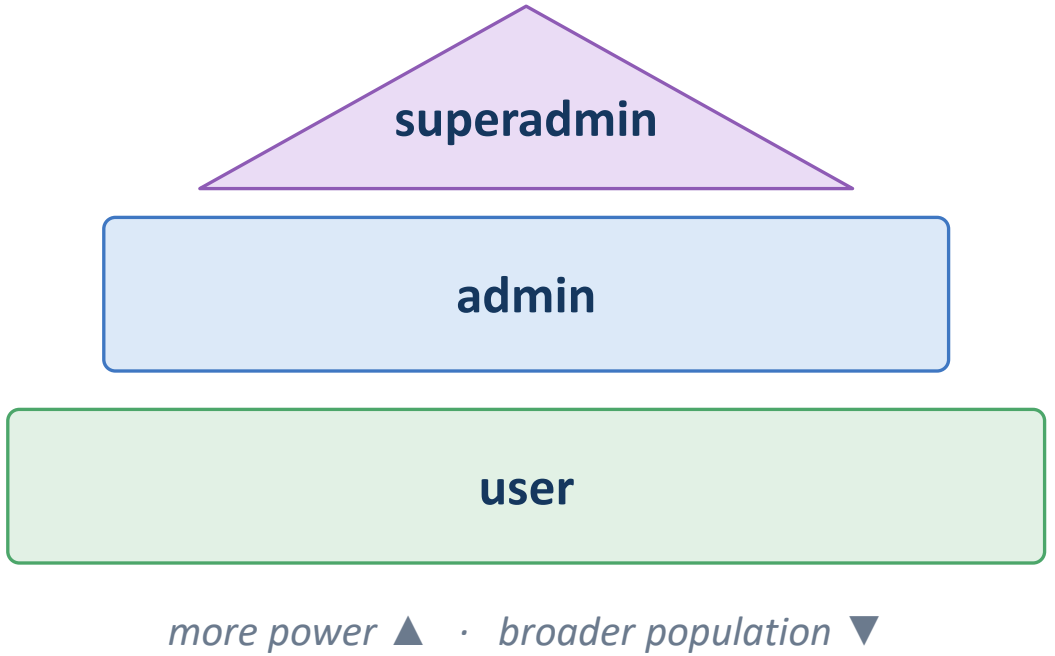
The Data Model: Project → Campaign → Scenario

- Project — the top container for an engagement or client.
- Campaign — a phase or themed batch of work inside a project.
- Scenario — one concrete test: targets (host, port, protocol) + attacks (with params).
- Scenarios run sequentially or in parallel; output streams live over WebSocket.



Roles & Access (RBAC)

- Three roles, in order of power: superadmin > admin > user.
- superadmin — full control, including other admins and global settings (the default admin account).
- admin — manages users and projects within their scope.
- user — works inside projects they are granted access to.
- Every action is recorded in audit logs, so who-did-what is always traceable.



Joining the Platform: Invite-Code Registration

- New accounts are created through invite-code-gated registration: you need a valid code to sign up.
- This stops random strangers from creating accounts on a platform that can launch attack tooling.
- Admins can disable registration entirely and create accounts manually for tighter control.
- For this course your instructor provides the invite code (self-paced learners use the default super-admin account).

Putting It Together & What's Next

- You now have the mental model: one browser app, backed by an API, a database, live WebSocket feeds, the tools, and an AI assistant.
- You can log in, secure the account, navigate the five views, and understand how work is organised.
- This part's lab: create your first Project and Campaign — the containers everything else lives in.
- After that, Parts 5–7 fill those containers with the three guided scenarios and teach reporting.

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Module 3
Reconnaissance & Attack Scenarios

PART 1
Reconnaissance & Enumeration

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

What Is Reconnaissance?

- Reconnaissance ("recon") is the first phase of an engagement: gather information about a target before touching it hard.
- Goal: build an accurate picture of what exists — hosts, services, versions — so later phases are targeted, not blind.
- Good recon saves time: you exploit the soft spots you found, instead of guessing.
- Footprinting is the broad, early sub-step — sketching the outline (address ranges, exposed services, technologies) before zooming in.

Passive vs Active Recon

- Passive — gather information without touching the target: public DNS, WHOIS, search engines, certificate logs, leaked data. Low risk, low noise.
- Active — directly probe the target: ping it, scan ports, grab banners. Faster and more accurate, but generates traffic the target can log.
- Nmap scanning is active recon — the target and any monitoring can detect it.

PASSIVE
no direct contact · low noise · hard to detect

ACTIVE (Nmap)
direct probing · accurate · detectable

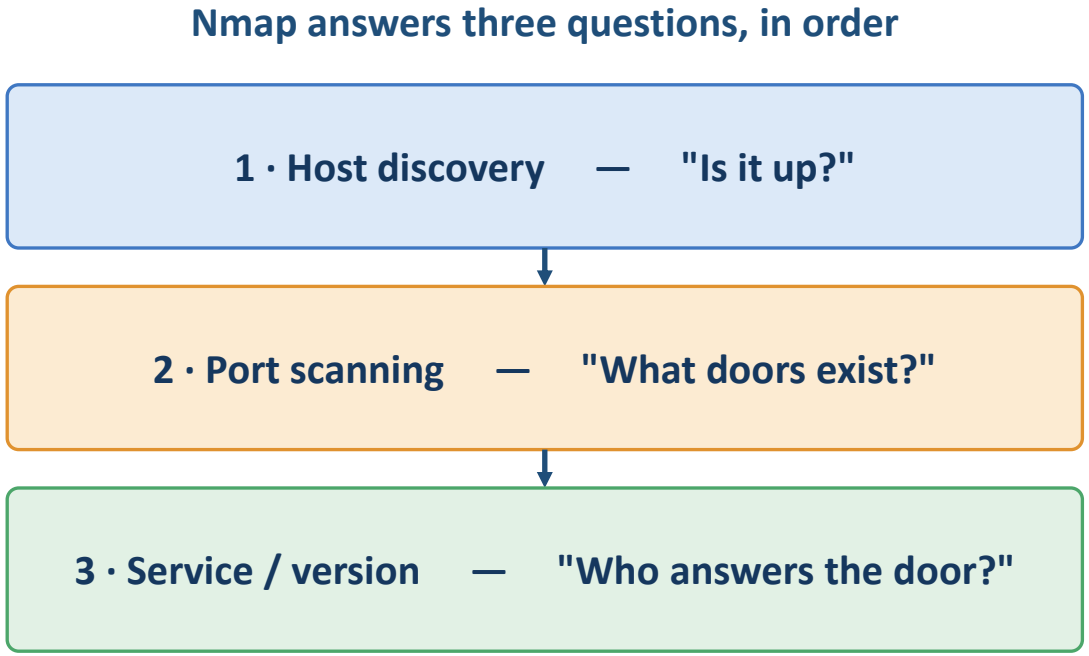
Rule of thumb: start passive to scope, go active only inside an authorised, isolated lab.

Safety & Authorisation (read before scanning)

- Every scan in this course runs only against the provided, isolated lab target.
- Never scan real or third-party systems — even "just a port scan" can be unlawful and is detectable.
- The difference between a pentester and an attacker is permission and scope, not technique.
- MMT-Pentester executes tools against the targets you define in a Scenario — keep those targets inside the lab.

Meet Nmap

- Nmap (Network Mapper) is the standard open-source tool for active recon and enumeration.
- Its output is the backbone of your attack-surface map and feeds directly into the next parts.



Core Nmap Flags

- -sV service/version detection (e.g. OpenSSH 8.9, Apache 2.4.52). -p- scan all 65535 ports (slower).
- -p 22,80,443 specific ports (fast). -sC default safe NSE scripts. -sn host discovery only. -Pn skip discovery.

```
$ nmap -sC -sV -p- <target>
```

Common combo: nmap -sC -sV -p- <target> — thorough enumeration with versions and default scripts.

Reading Nmap Output

- Each line: PORT STATE SERVICE VERSION. The VERSION column (from -sV) tells you exactly what to look up next.
- Always record open ports + services verbatim — that record is your attack surface.

```
PORT     STATE     SERVICE   VERSION
22/tcp   open      ssh       OpenSSH 8.9p1
80/tcp   open      http      Apache 2.4.52
443/tcp  closed    https
3306/tcp filtered  mysql
```

open — service accepting connections (the interesting ones)

closed — reachable, nothing listening

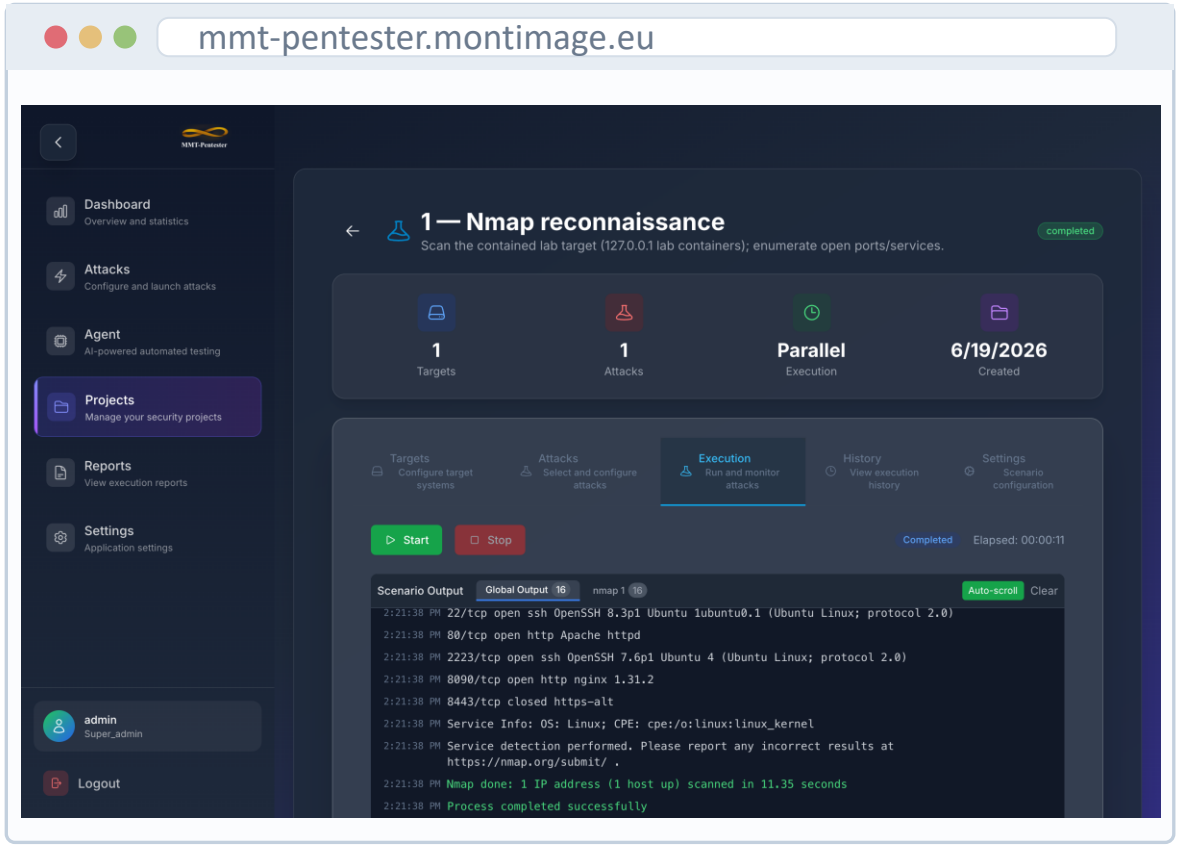
filtered — firewall blocking; Nmap can't tell

From Output to Attack Surface

- An attack surface is the set of exposed services an attacker could try to reach.
- For each open port ask: what is this service, what version, and is it a known entry point?
- Prioritise services that commonly allow access: SSH (22) → credential attacks; HTTP/HTTPS (80/443) → web attacks / DoS.
- In this course the two services you look for are SSH (Part 6 — brute-force) and HTTP (Part 7 — dirb (HTTP content discovery)).
- A "minimal attack surface map" = a short, ranked list of services worth testing, with reasons.

Running Recon in MMT-Pentester

- Recon runs as a Scenario under a Project/Campaign: targets {host, port, protocol} + attacks {params}.
- Steps: create/open a Project → add a Scenario → use the Reconnaissance-with-Nmap template → set the lab target → execute.
- Watch live tool output stream over WebSocket (raw ws, port 9090) — you see Nmap's results as they arrive.
- Review results in the Reports view; outcomes appear on the MMT-Dashboard.



Recap & What's Next

- Recon = gather info first; passive (no contact) vs active (direct probing). Nmap scanning is active.
- Nmap workflow: host discovery → port scan → service/version detection.
- Read output by port state and version; turn open services into a ranked attack surface.
- You will hand in a recon report identifying SSH and HTTP for Parts 6 and 7.

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Module 3
Reconnaissance & Attack Scenarios

PART 2
Running Attack Scenarios

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

From Recon to Attack

- In Part 5 you ran Nmap recon and mapped a minimal attack surface: a live host with an SSH service and an HTTP service.

- This module turns those findings into action: two guided attack scenarios — SSH brute-force and HTTP content discovery (dirb).

- Everything runs only against the provided, isolated lab target. Authorised use only.

SAFETY FIRST (non-negotiable)

⚠ Run attacks ONLY against the provided, isolated lab target

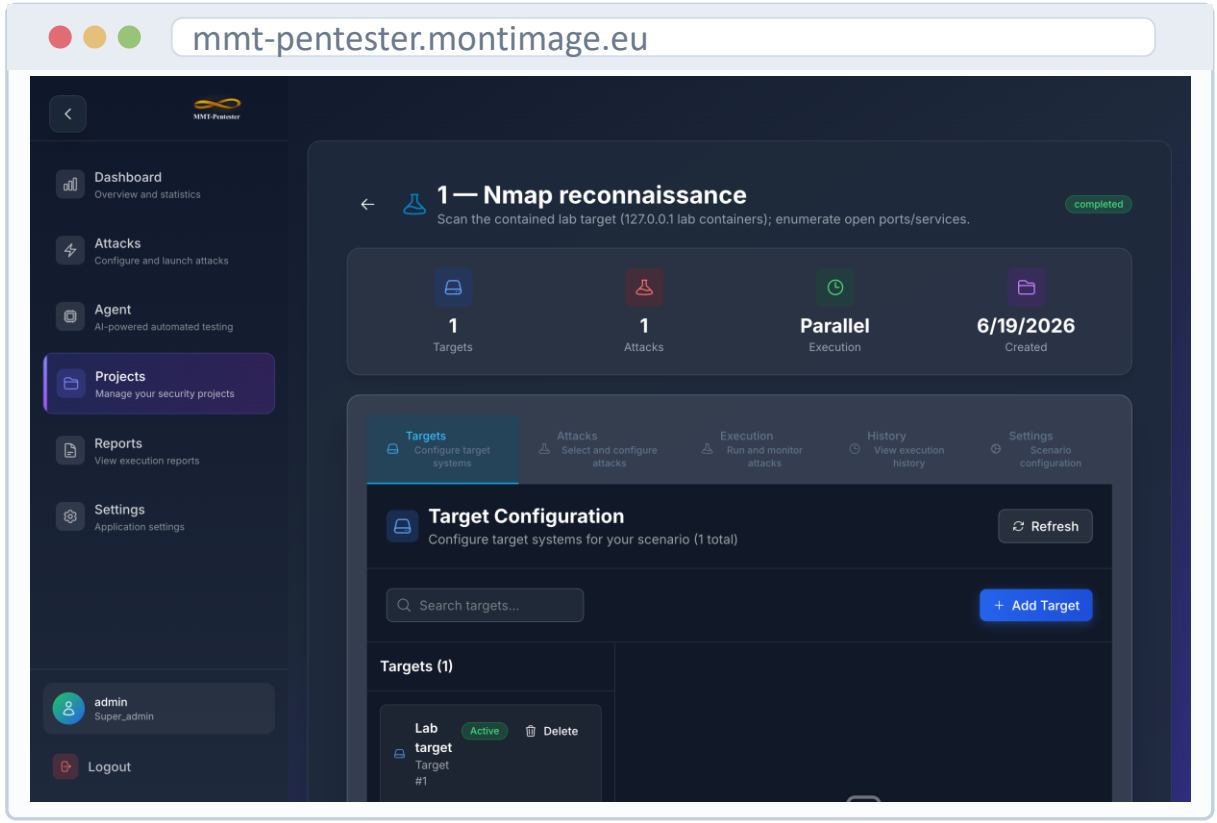
- Explicit authorisation is required before any attack.
- Never run brute-force or DoS against real, production, or third-party systems — it is illegal and harmful.
- A real DoS can take down a business; real credential attacks are unauthorised access.
- **The lab makes this safe and legal to learn.**

The Data Model: Targets and Attacks

- Hierarchy: Project → Campaign → Scenario → (targets + attacks).
- A target is { host, port, protocol } — what you are testing.
- An attack is a tool invocation with { params } — how you are testing it.
- A Scenario binds one or more targets to one or more attacks, then executes them.

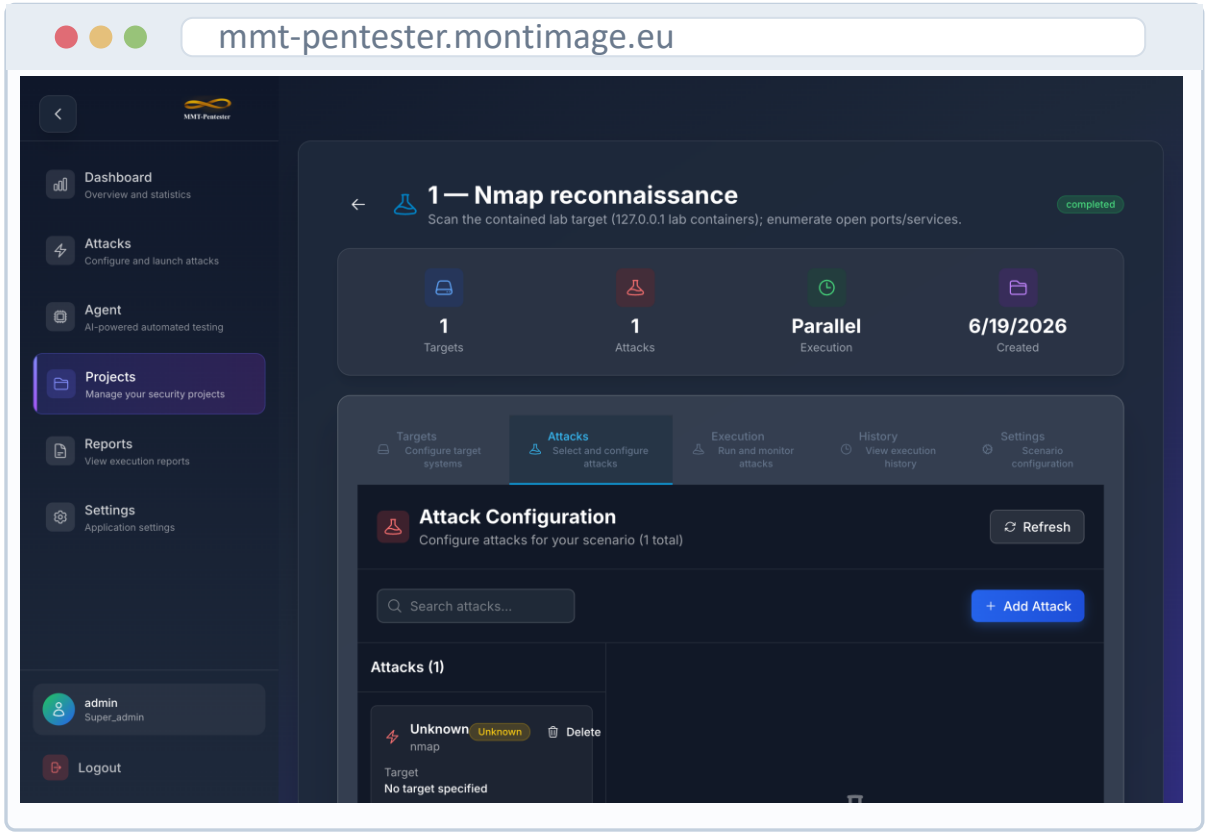
Configuring a Target

- Set host = the lab target IP/hostname from your recon results.
- Set port = the discovered service port (e.g. 22 for SSH, 80 for HTTP).
- Set protocol = the service type (ssh, http, tcp …).
- Targets are reusable across attacks within the Scenario.



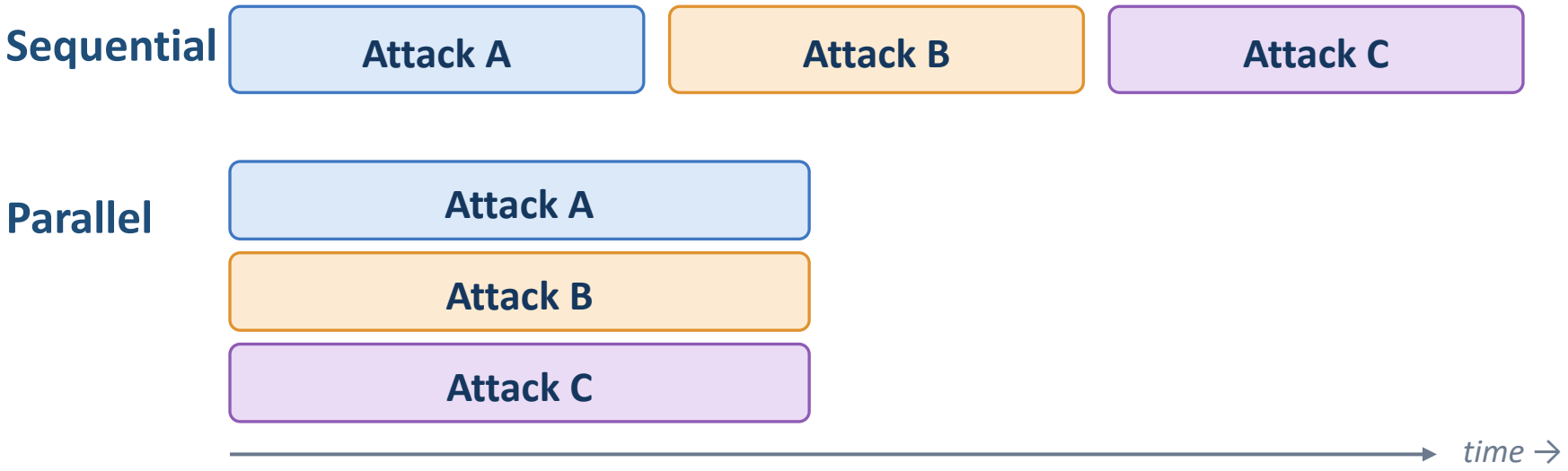
Configuring an Attack and its Params

- Pick the attack/tool, then fill its params (wordlists, thread count, rate, duration).
- Params control intensity and behaviour — sensible defaults are provided for the lab.
- Each attack is bound to a target so the tool knows where to point.



Sequential vs Parallel Execution

- Sequential — attacks run one after another: predictable, easier to read, lower load.
- Parallel — attacks run at the same time: faster, models multi-vector pressure, noisier output.
- Choose sequential while learning; parallel to model a realistic combined assault.



Live Output over WebSocket

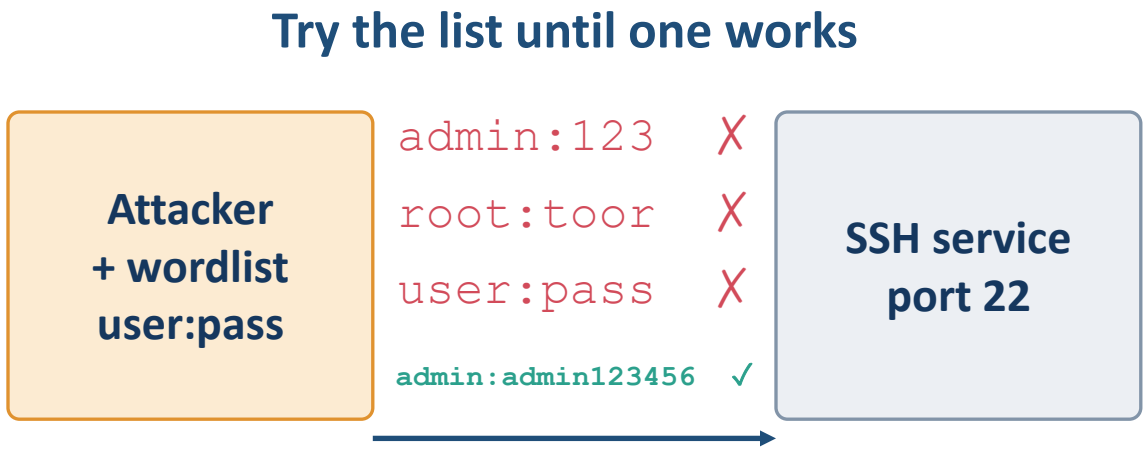
- The Dashboard streams tool output live: Socket.io carries campaign/scenario events; a raw WebSocket on port 9090 carries live tool stdout/stderr.

- You watch attempts, responses and errors scroll in real time — no need to wait for a final report.

- After the run, the same outcomes appear as results and any monitor/IDS alerts.

Lab A: SSH Brute-Force (concept)

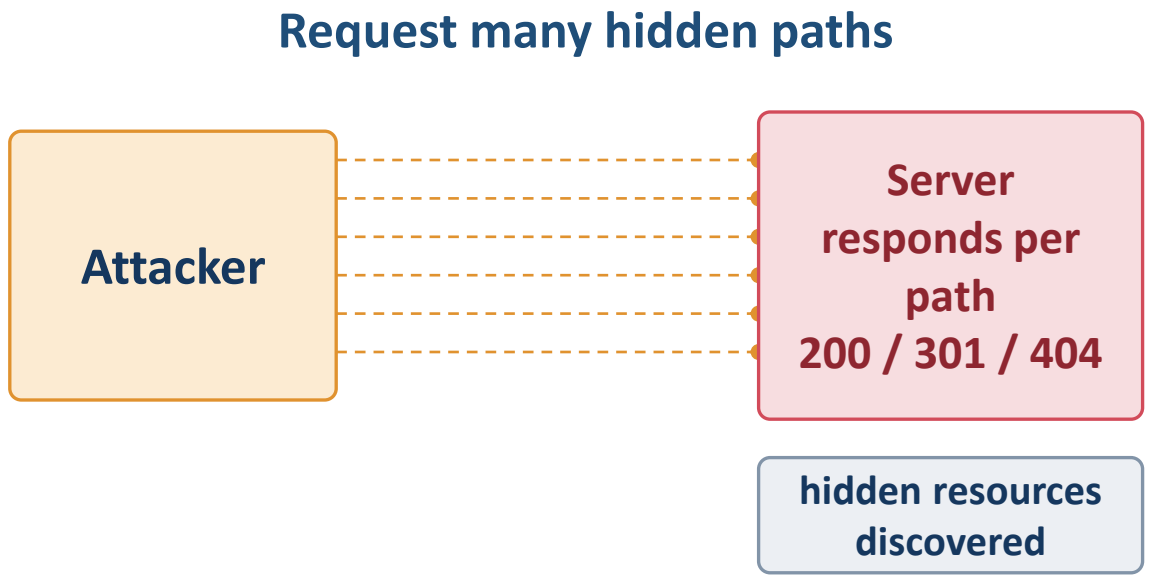
- Brute-force / credential attack: repeatedly try username+password pairs until one authenticates.
- Targets weak / default / reused credentials on the discovered SSH service (port 22).
- You will see authentication attempts stream live, and detection of repeated failures.



MITRE ATT&CK: Credential Access — Brute Force (T1110), incl. Password Guessing (T1110.001).

Lab B: HTTP Content Discovery with dirb (concept)

- Content discovery (forced browsing): send many HTTP requests for common paths (/admin, /login, /backup, robots.txt) to enumerate hidden or unlinked resources on the web server.
- Noisy and highly observable — a rapid burst of requests to many distinct paths.
- You will see requests to many paths, discovered resources (200/301), and a web-scanning / IDS alert fire.



MITRE ATT&CK: Reconnaissance — Active Scanning — Wordlist Scanning (T1595.003).

Observing Results and Detection

- Watch the MMT-Dashboard: traffic seen, expected vs actual behaviour, and IDS/monitor alerts.
- Brute-force: failed-auth bursts, possible lockout/throttling, an alert on repeated attempts.
- dirb: a burst of HTTP requests to many paths, discovered resources (200/301), and a web-scanning / enumeration alert.
- Capture evidence as you go — this is the graded deliverable.

Reporting Your Run

- The platform auto-generates reports: PDF (Puppeteer) and Excel (ExcelJS).
- Use these alongside your own screenshots to assemble evidence.
- A good run is reproducible: target, attack, params, observed output, and detection all recorded.

Recap and What's Next

- You configured targets and attacks, chose execution mode, and watched live WebSocket output.
- You ran a Credential Access attack (SSH brute-force) and a web content-discovery attack (dirb, Active Scanning), and observed detection.
- Next: deeper analysis and reporting of what these scenarios produced.

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Module 4
Monitoring, Detection & Reporting

Co-funded by the European Union

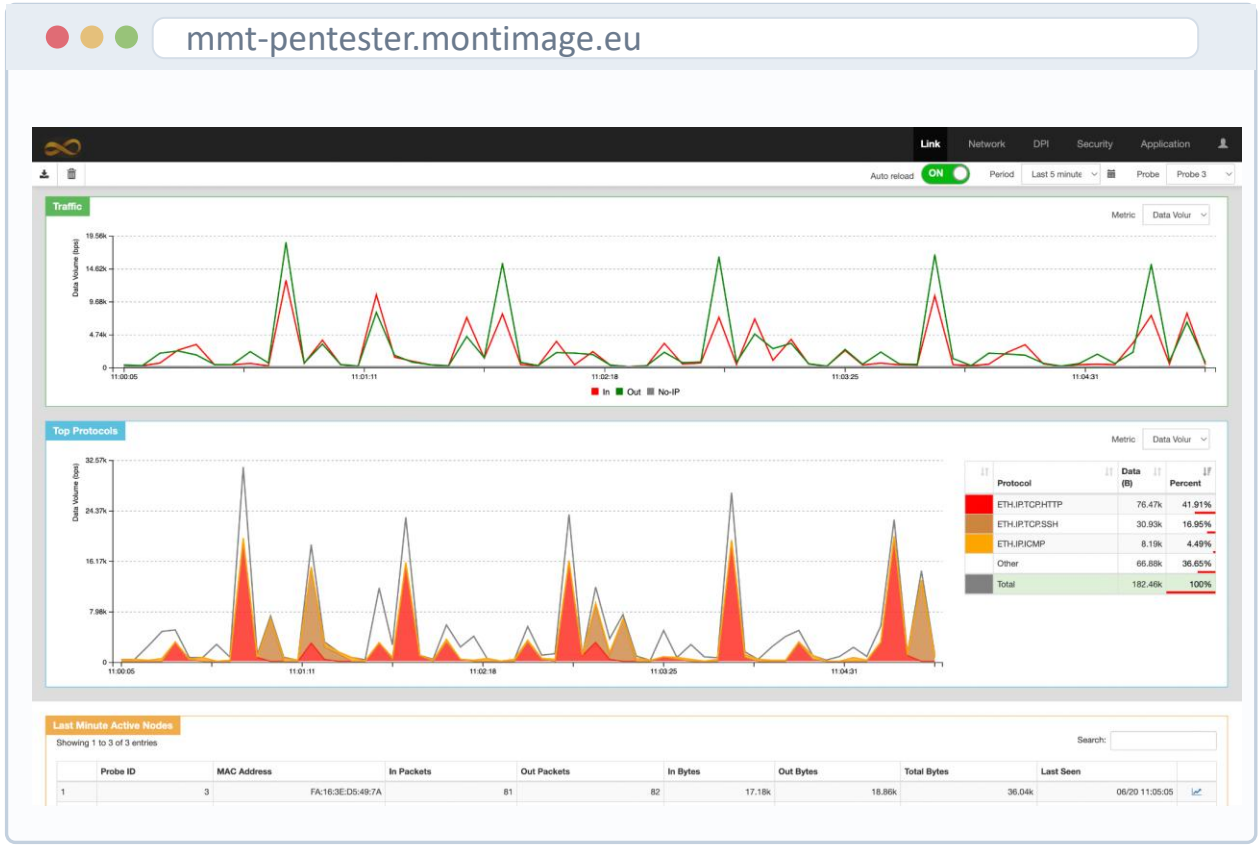
Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

From Running Attacks to Understanding Them

- Parts 4–6 launched scenarios; Part 7 is about reading the result.
- An execution produces three things to watch: live tool output, dashboard state, and detection alerts.
- A pentest is only as valuable as the report that comes out of it — evidence beats opinion.

Reading the Live MMT-Dashboard

- During an execution the Dashboard shows campaign/scenario status (queued, running, completed, failed).
- Two live channels feed the view: Socket.io for events, raw WebSocket (port 9090) for tool stdout/stderr.
- Watch status badges, the live log pane, and per-target progress as the test runs.



Expected vs Actual Behaviour

- Each guided scenario has an expected signature: Nmap → ports listed; SSH brute-force → auth attempts logged; dirb → many path requests, resources discovered.

- "Actual" = what the live output and target response really show.

- Mismatch is information: a brute-force with zero auth attempts means the target was unreachable, not secure.

What IDS / Monitor Alerts Mean

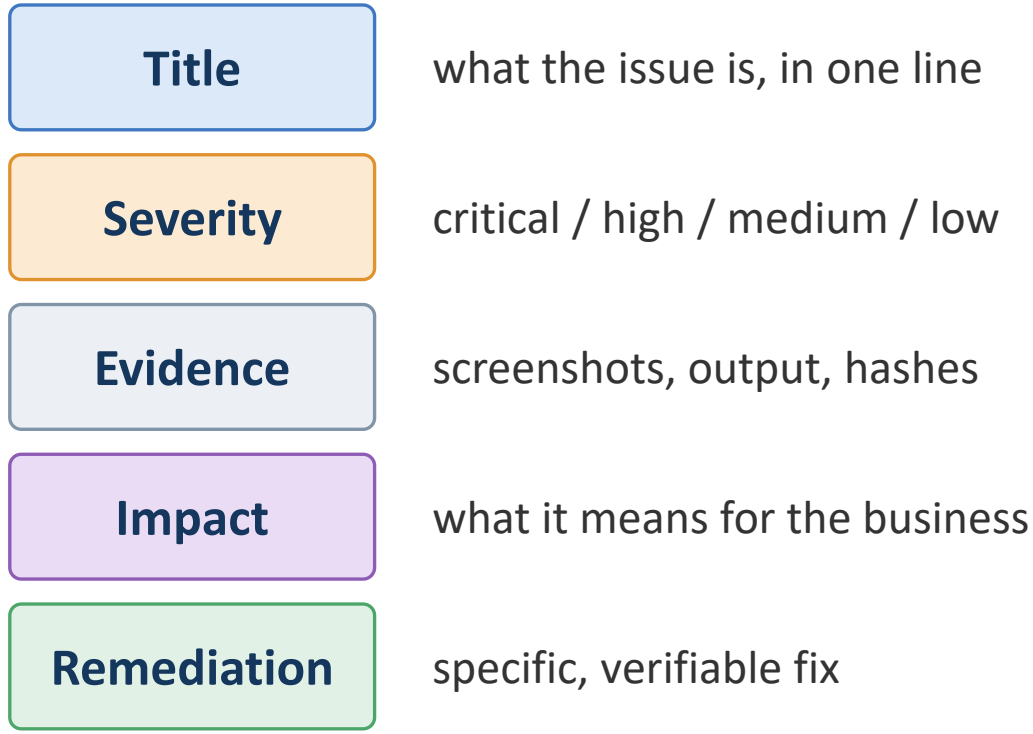
- IDS = Intrusion Detection System: software that inspects traffic/behaviour and raises alerts on suspicious patterns.

- Detection is mostly signature (matches a known bad pattern, e.g. a burst of SSH logins) or anomaly (deviates from a normal baseline).

- On the Dashboard, alerts appear alongside execution output — they tell you whether a defender would have noticed your attack.

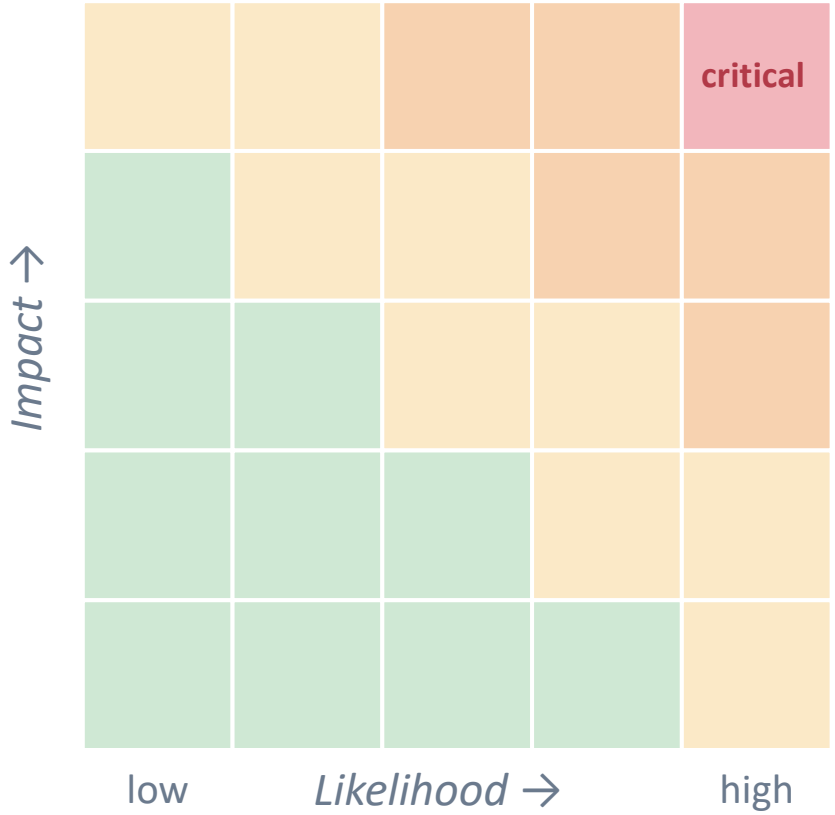
Anatomy of a Good Finding

- Five required parts — the platform's report data model mirrors this (severity, evidence[], impact, remediationInstructions).
- Severity scale used by the platform: critical / high / medium / low.



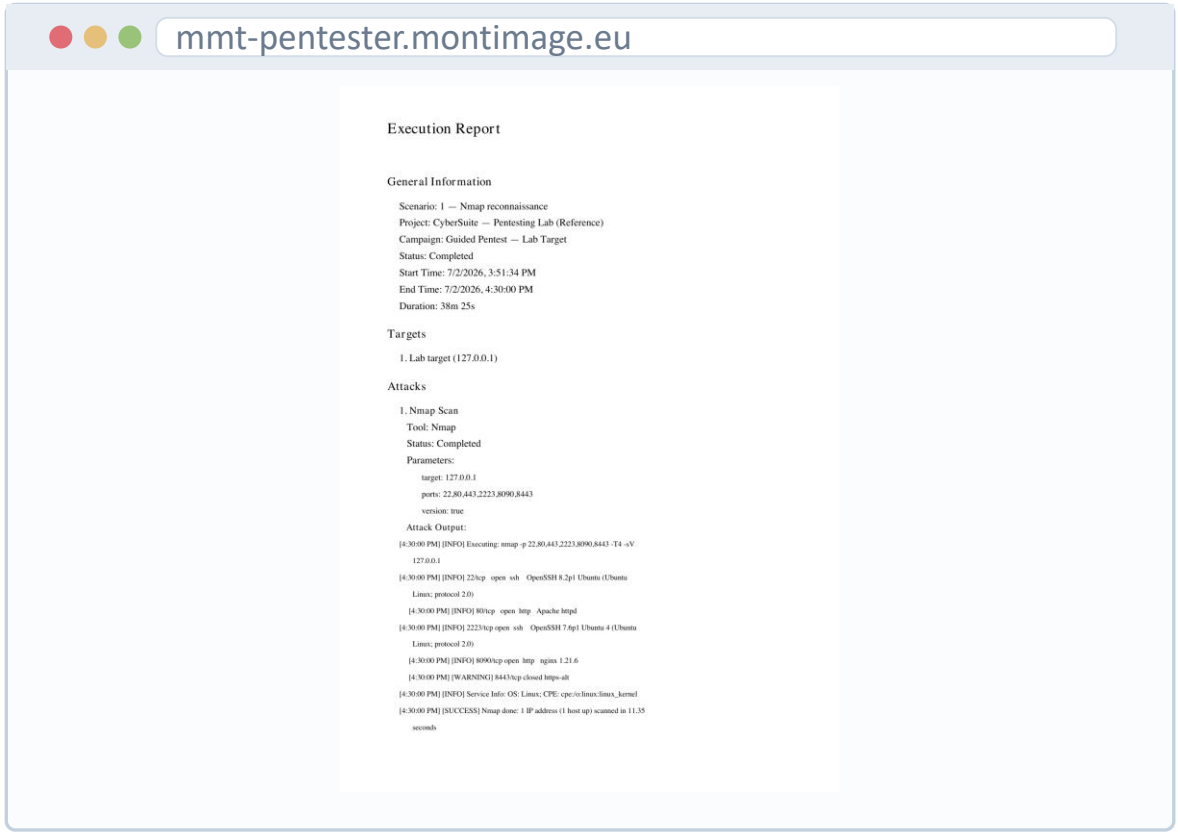
Severity & Risk Score

- Severity reflects how bad a single finding is; the report also rolls findings into an overall riskScore.
- The statistics block counts findings by tier: critical / high / medium / low.
- Judge severity by impact × ease of exploitation — not by how impressive the attack felt.



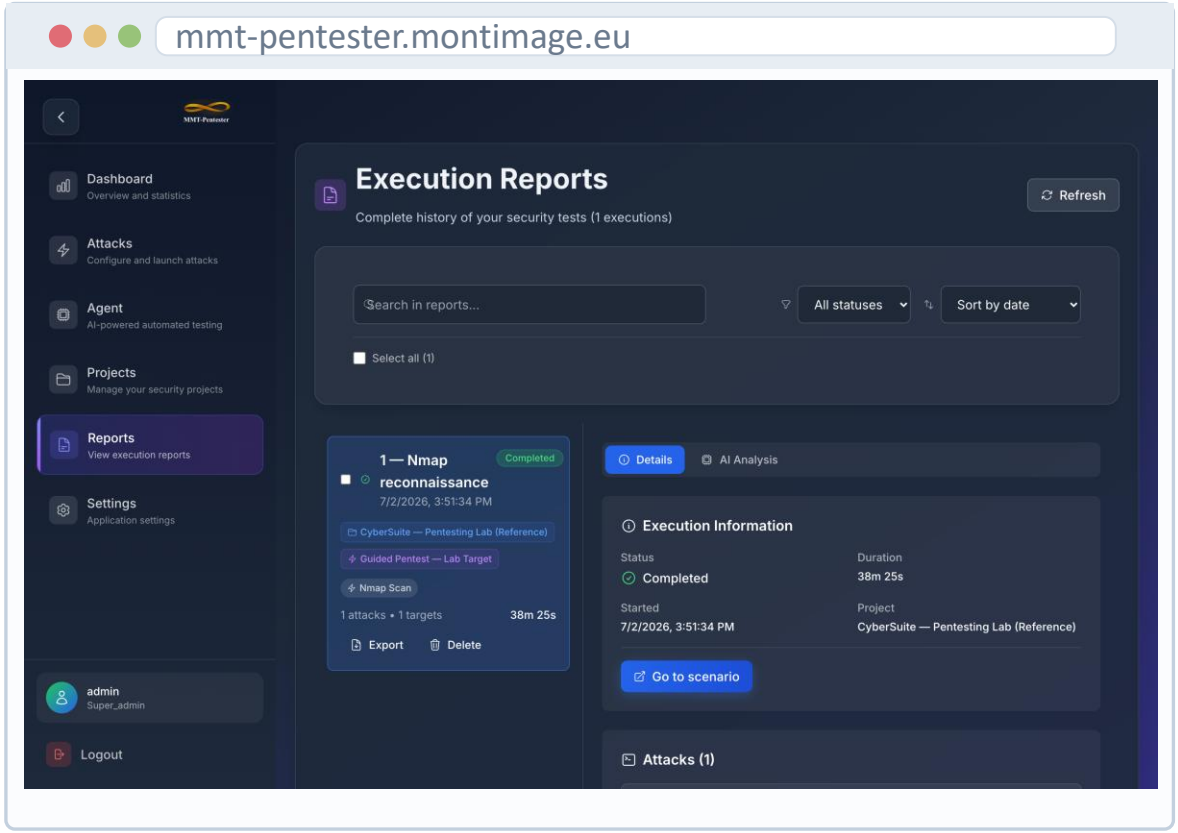
The PDF Report (Puppeteer)

- The platform renders a PDF via Puppeteer (headless Chrome) — a polished, client-ready document.
- Contents: executive summary, attack narrative, methodology, per-finding detail, remediation plan, statistics.
- Filename pattern: mmt-pentester-rapport-securite-<target>-<date>.pdf. Download from the Reports view.



The Excel Report (ExcelJS)

- The platform also exports structured data via ExcelJS (.xlsx), with JSON/CSV options.
- Excel is the machine/tracking deliverable: one row per finding, sortable and filterable.
- Use it to triage: sort by severity, assign owners, track remediation status over time.



Writing Actionable Remediation

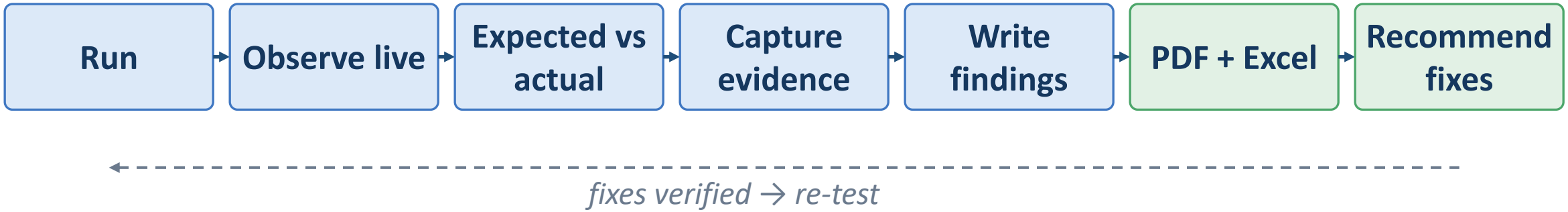
- Bad: "Harden SSH." Good: "Disable password auth, enforce key-based login, add fail2ban with a 5-attempt lockout."

- A remediation entry should be specific, verifiable, and prioritised (the platform stores priority, fix, estimatedEffort, businessImpact).

- Always pair the what with the why (impact) so the owner can justify the work.

Putting It Together: The Reporting Loop

- A good report is reproducible: someone else should follow your evidence and see the same thing.
- Reporting is not the end of the engagement; it is the start of the fix.



Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Module 5
Advanced: Autonomous Agents & Tool Integration

Co-funded by the European Union

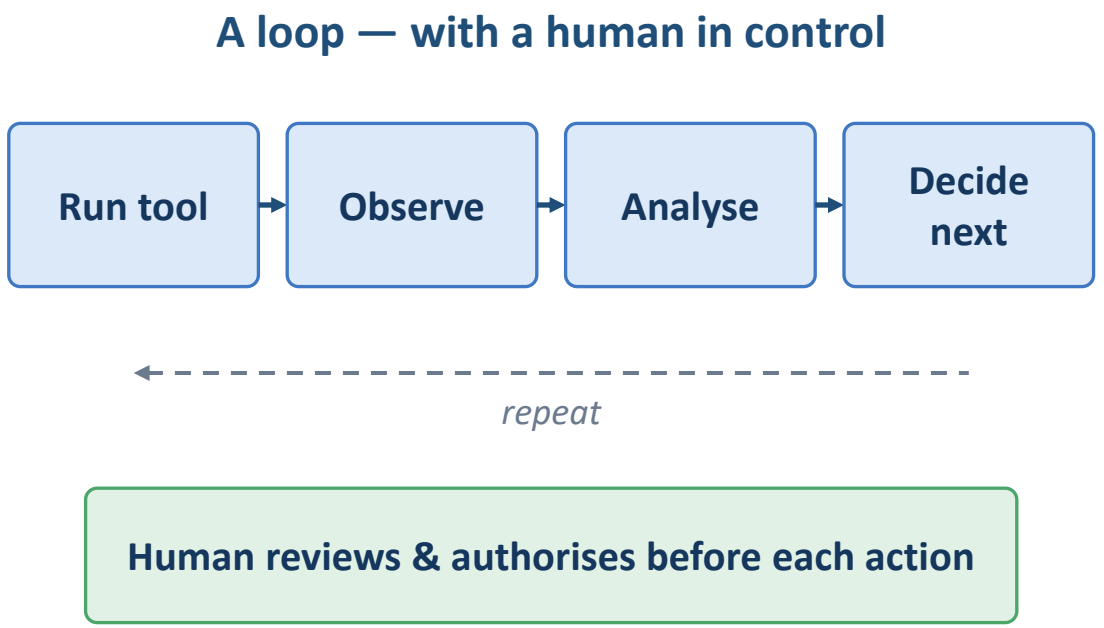
Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

From guided scenarios to autonomous agents

- Parts 4–7 walked through the three guided scenarios you drove step by step.
- This module looks at the Agent view, where the platform can coordinate a run and use AI to suggest the next step.
- Goal: understand the concept and the tool ecosystem — not to overstate what the AI can do on its own.
- Reminder: everything here runs ONLY against provided, isolated lab targets, with explicit authorisation.

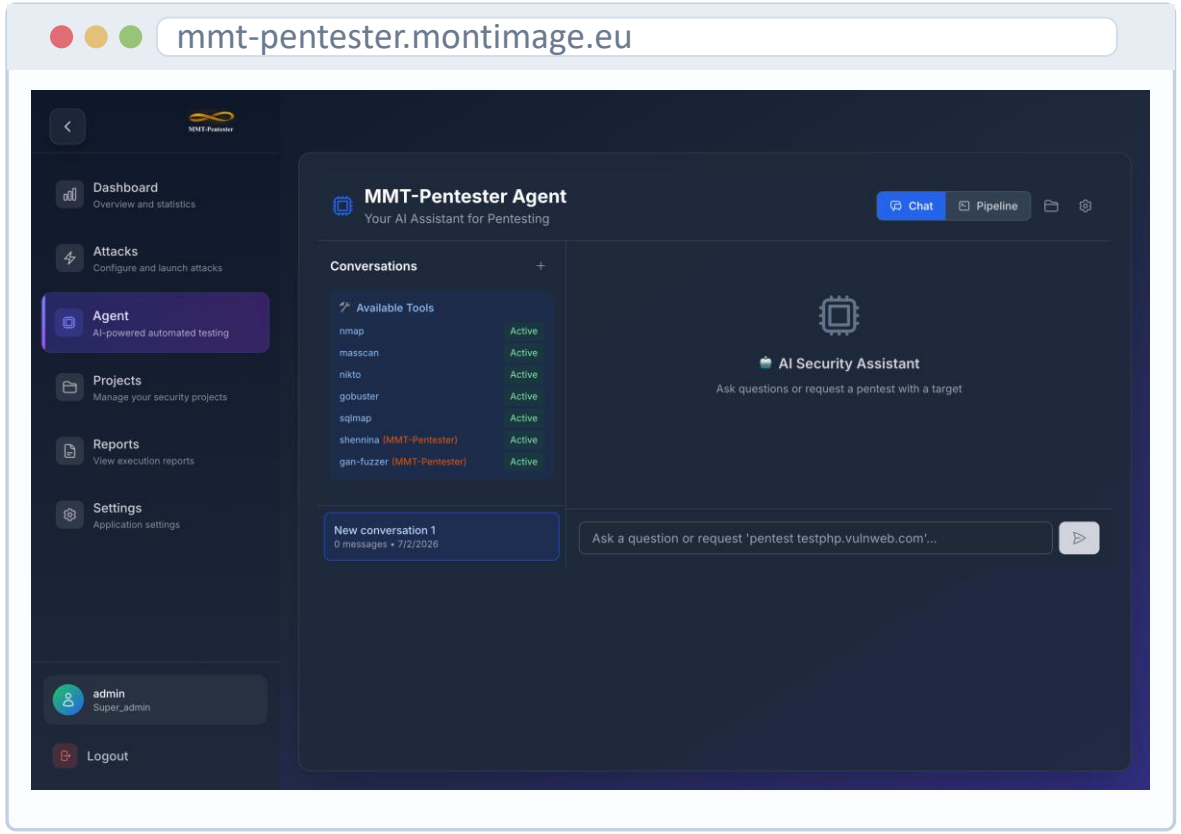
What "autonomous agent" means here

- An autonomous agent is a loop: run a tool → observe output → analyse → decide a reasonable next action → repeat.
- In MMT-Pentester the human stays in control: the agent proposes; you review and authorise.
- It does not replace the pentester — it reduces busywork between obvious steps. Think "assisted orchestration", not "hands-off hacking".



The Agent view

- One of the five platform views: Dashboard, Projects, Scenarios, Agent, Reports.
- The Agent view is where you start, watch, and review an AI-coordinated session against a lab target.
- It ties into the same data model: Project → Campaign → Scenario → targets + attacks.
- Live tool output streams over WebSocket, just like the guided scenarios.



AgentOrchestrator: coordinating a run

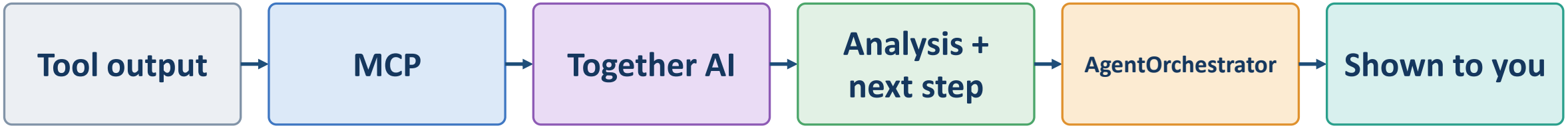
- AgentOrchestrator is the backend service that coordinates an autonomous agent run.
- It launches/sequences tool executions, collects their output, and routes that output to the AI layer for analysis.
- It manages run state and feeds results back into the platform so you see progress and recommendations in the Agent view.
- It works alongside the same execution machinery used elsewhere (sequential or parallel runs, live WebSocket output).

What you will learn

- You will understand what an autonomous agent, and the human-in-the-loop model
- You will see how AgentOrchestrator coordinates a run, and how MCP turns raw tool output into readable insight
- And you will learn more about the integrated tool ecosystem

MCP + Together AI: turning output into insight

- MCP (Model Context Protocol) is the standard interface that hands structured context (tool output) to an AI provider.
- MCPService connects to the MCP server; Together AI is the provider that performs the analysis.



Raw scan/attack output becomes a readable "here is what I see, here is what you might do next"

Reading recommendations critically

- AI recommendations are grounded in the observed output, but they can be wrong, generic, or incomplete.
- Always validate against what you actually see: the live output, IDS/monitor alerts, and the MMT-Dashboard observations.
- You decide whether to accept, adjust, or ignore a suggested next step.
- Recommendations never override the authorised-scope and safety rules.

The integrated tool ecosystem

MMT-Pentester runs external tools as child processes; the agent can orchestrate them. Each answers a different question.

Caldera

adversary simulation

MAIP

attack injection into traffic

Shennina

automated exploitation

KNX Smart Fuzzer

KNX / building automation

GAN Fuzzer

ML-generated inputs

MAG / MMT-Attacker

attack traffic & payloads

When to use which tool

- Adversary simulation / TTP chains → Caldera.
- Injecting attacks into network traffic → MAIP.
- Automated exploitation experiments → Shennina.
- Protocol robustness / unexpected input → KNX Smart Fuzzer (building automation), GAN Fuzzer (ML-generated inputs).
- Generating attack traffic / payloads → MAG / MMT-Attacker.

Putting it together: a typical agent session

- Pick a lab target and start a session in the Agent view.
- AgentOrchestrator runs an initial step (e.g. recon), streams output, and sends it via MCP to Together AI.
- The AI returns analysis + a suggested next step (e.g. test a discovered service).
- You review the recommendation, decide, and let the loop continue — observing results on the Dashboard and in Reports.

Recap and boundaries

- The Agent view + AgentOrchestrator coordinate AI-assisted runs; MCP + Together AI provide analysis and next-step recommendations.
- The tool ecosystem covers simulation, injection, exploitation, fuzzing, and traffic generation — pick by purpose.
- Human-in-the-loop and authorised-scope rules always apply; AI suggestions are inputs, not orders.
- Next: the exploration lab — run or describe an agent session and capture its recommendations.

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

CAPSTONE
Final Hands-On Assessment

End-to-End Guided Pentest

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

The Capstone: Put It All Together

- This is the final module. You will run a complete mini-engagement end to end against a fresh lab target.

- The arc: Recon (Nmap) → pick two services → two attacks (SSH brute-force + HTTP content discovery (dirb)) → observe detections → final report → final quiz → certificate.

- Everything you need you already learned in Parts 1–7; today you assemble it under your own steam.

- This lab is graded against a rubric, so work methodically and document as you go.

Safety, Authorisation & Scope (Non-Negotiable)

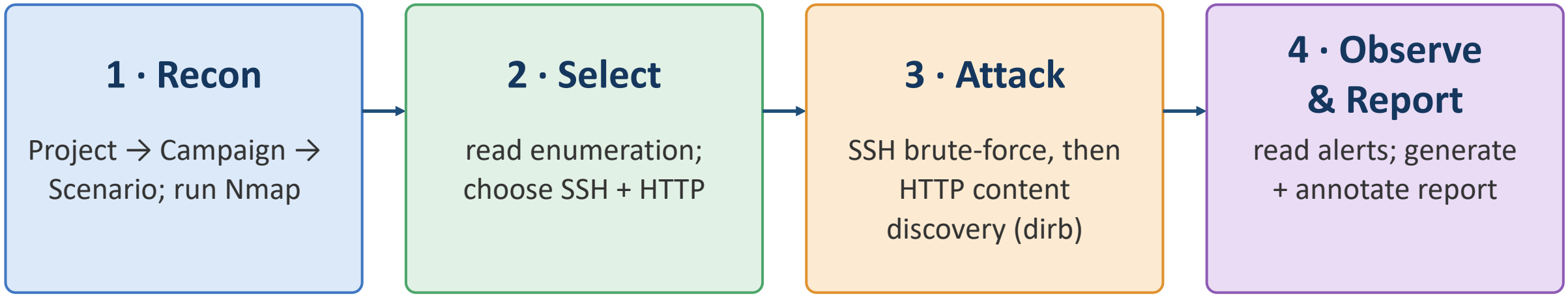
⚠ The lab target + this exercise are your authorisation and scope

- Run everything only against the provided, isolated lab target — never real or third-party systems.
- Minimise harm, document every action, and stop if anything behaves unexpectedly outside the lab.
- Unauthorised testing is illegal regardless of intent — the licence to act comes from explicit permission.

Your Mission Brief

- Target: a fresh lab host provided for this capstone (host/IP given in the brief).
- Mission: map the attack surface, demonstrate two distinct weaknesses, and report on risk with prioritised fixes.
- Constraints: use only the three guided scenario types; observe results on the dashboard; produce one report.
- Output: a final report (PDF/Excel export + your written analysis) and a completed final quiz.

Workflow at a Glance



How You Are Graded

- A rubric scores five areas; "Excellent" = complete, evidence-backed, clearly communicated.
- The final quiz pass threshold is 70% (7 of 10). Pass the quiz and submit the deliverable to complete the course.
- Passing triggers the platform's certificate flow → a certificate of completion.

Recon	*Excellent → Missing*
Execution	*Excellent → Missing*
Observation	*Excellent → Missing*
Report quality	*Excellent → Missing*
Recommendations	*Excellent → Missing*

Go Build It

- Open the Dashboard at the lab URL, log in, and begin with recon.
- Take screenshots of key moments — scan results, attack output, alerts — for evidence.
- Use the submission checklist in the Supporting Material before you hand in.
- **When the lab is done, take the final quiz to unlock your certificate. Good luck!**

Cyber Suite

montimage

Uptake of Innovative Security-as-a-Service Solutions

Course Wrap-Up and Evaluation

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission.

Course wrap-up

Congratulations on completing **Practical Penetration Testing with the Montimage Pentesting Toolbox**!

What you learned from this course

- Understanding the penetration-testing lifecycle and the legal and ethical rules of authorised testing.
- Reconnaissance using Nmap and mapping a target's attack surface.
- Configuring and running controlled attack scenarios.
- Monitoring executions in real time and interpreting intrusion-detection alerts.
- Generating professional reports and writing actionable remediation recommendations.

We Value Your Feedback!

Please take a few minutes to complete our short course-evaluation survey. Your honest feedback helps us improve the lessons, labs, and the Pentesting tools for future learners.

What the survey asks about

- The clarity and usefulness of each module.
- Whether the hands-on labs were easy to follow and worked as expected.
- The pacing and overall difficulty of the course.
- Any technical issues you encountered with the platform.
- Suggestions and features you would like to see.

Take the survey

Please complete the evaluation form. The survey is anonymous and takes about five minutes. Thank you for helping us make the course better and good luck applying your new network threat detection and response skills, responsibly and only on systems you are authorised to test.

Thank you for your attention!

Montimage

Edgardo Montes de Oca, Maxime La

Maxime.vinhhoa.la@montimage.eu
edgardo.montesdeoca@montimage.eu

https://www.montimage.eu/

Co-funded by the European Union

Co-Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Digital Europe Programme. Neither the European Union nor Digital Europe Programme can be held responsible for them.