Life as a pentester

Kashmir54

Summer School 2025

Kashmir54@0xdeusto# id

- Manuel Sánchez Paniagua, 97
- Senior Pentester @ A2SECURE
- Focus: Web and Mobile
- Messing around with CTFs @ *ISwearIGoogledIt*
- Computer Science and Researcher @ *Unileon*
- OSCP and OSWP Certified

@ManuSanchez54

EA1RDM

Kashmir54

kashmir54.github.io

Index

Summer School 2025

- ID
- Cybersecurity and its impact
- Pentesting: Goals, stages and reality
- Kind of pentests
- Redteam and Pentest: Close yet so far
- Miscellaneous

Cybersecurity and its impact

Quick basics

- **Confidentiality**: Only authorized parties should see the information.
- **Integrity**: Information is stored/sent as it is, no unauthorized modifications.
- **Availability**: Access the information anytime

Quick basics

Summer School 2025

- Apply the **CIA concept** to all **information**:
  - Emails
  - Banking
  - Health
  - Photos / Videos / Docs
  - Private conversations

- And to **all the gates to access** (attack surface):
  - Webs
  - Mobile apps
  - Desktops
  - Networks

Quick basics: The (ideal) theory

- **Programmers:** Secure coding.
- **Systems:** Design secure infrastructures and configurations.
- **Providers:** Both.
- **Users:** Avoid deceitful content.

Quick basics: The reality

- **Programmers:** ~~Secure coding~~. Quick fixes to ensure the deadline:
  - It works, right? Leave it like that, we will improve it later (kek)
- **Systems:** ~~Design secure infrastructures and configurations~~. No design stage, firemen all around:
  - 90% CPU usage. Auto-scale kubernetes. Latency is high, refactoring is needed (never ending technical debt)
- **Providers:** ~~Both~~. Same problems, but now on millions of products:
  - Is your product secure?
  - × Yes !!! (Does not know what product does).

Quick basics: The reality

Summer School 2025

- **Users:** ~~Avoid deceitful content.~~ Prompting creds on worst phishing websites:
  - *I saw Pablo Motos' new crypto strategy on Facebook and logged in with Gmail* (he is on the police station, -300€ on his account).

Quick basics: How to fix it

Summer School 2025

Programmers

Systems

Providers

Users

SDLC

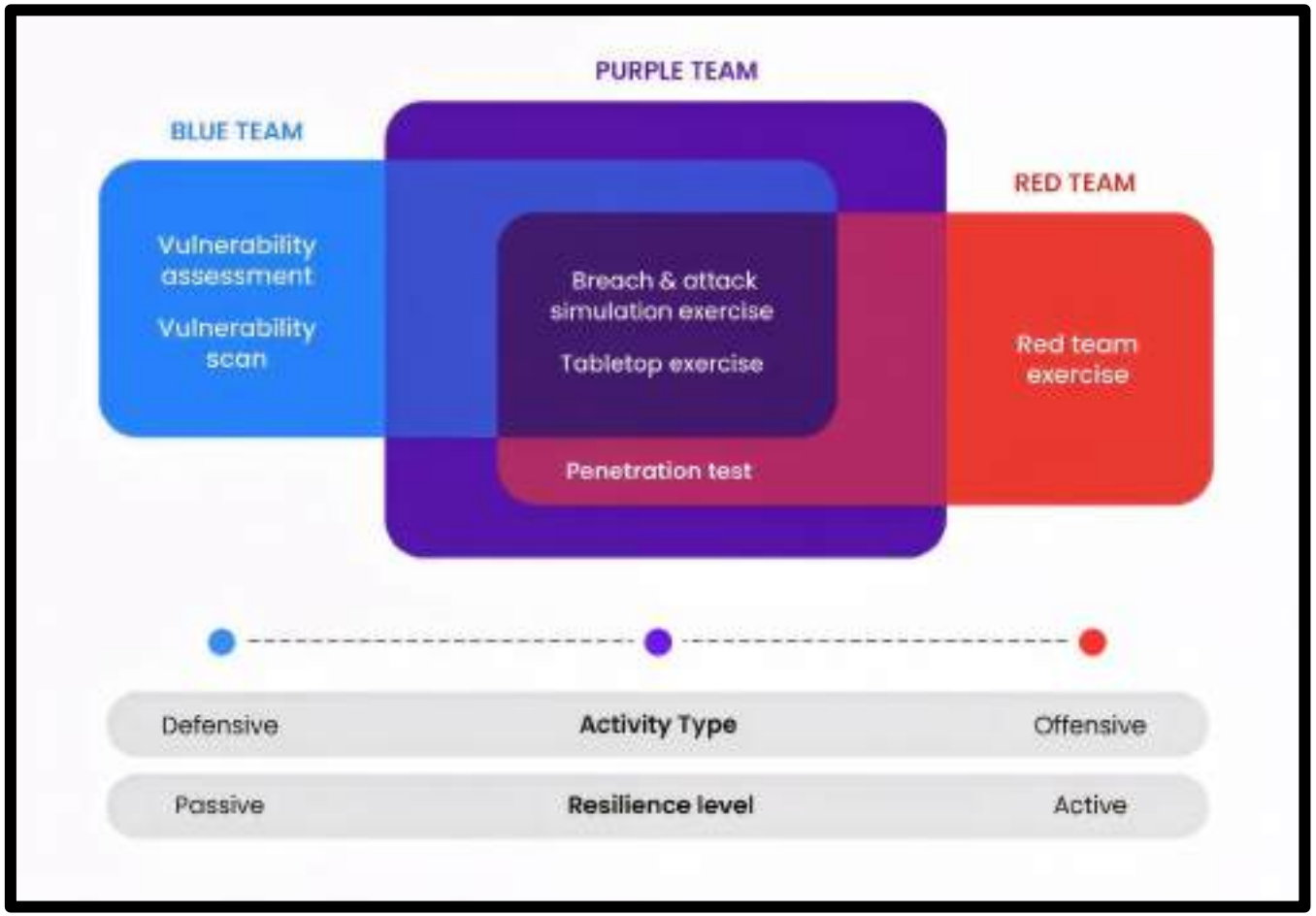
Cybersec

Awareness

S-SDLC

Quick basics: Cybersec Industry

Summer School 2025



Pentesting: Goals, stages and kinds

Pentesting: WWW (What, when, why)

Summer School 2025

What? An asset that the company wants to verify:

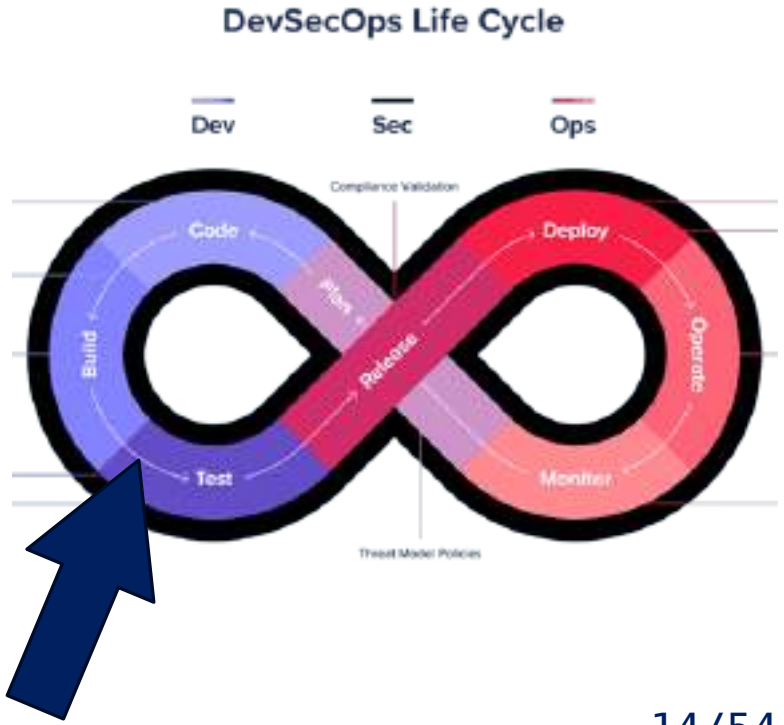
- A new development
- A new feature
- A new device
- A new deployment / server
- A new network config
- Any of the above but old or never tested before (yikes)
- Any of above from a provider

Pentesting: WWW (What, **when**, why)

Summer School 2025

When? The timing to perform a pentest (from ideal to usual):

- On final QA stage (1 month before go-live date)
- Last development stages, logic is final (along with QA tests)
- Go-live is tomorrow (f**k we forgot the pentest)(asap plz)
- We got breached!!! (lesson learnt)
- Product is live.
- When policy states it (PCI, ISO27001, etc…)



Pentesting: WWW (What, when, why)

Summer School 2025

Why?

- Because **we care about our users and their data** (do they?)
- Because policy says so (first bullet is not required)
- Because we need support data for stakeholders/investments
- Because we don’t trust our provider
- Because we got breached (need to fix things and build up reputation)

Pentesting: Goals

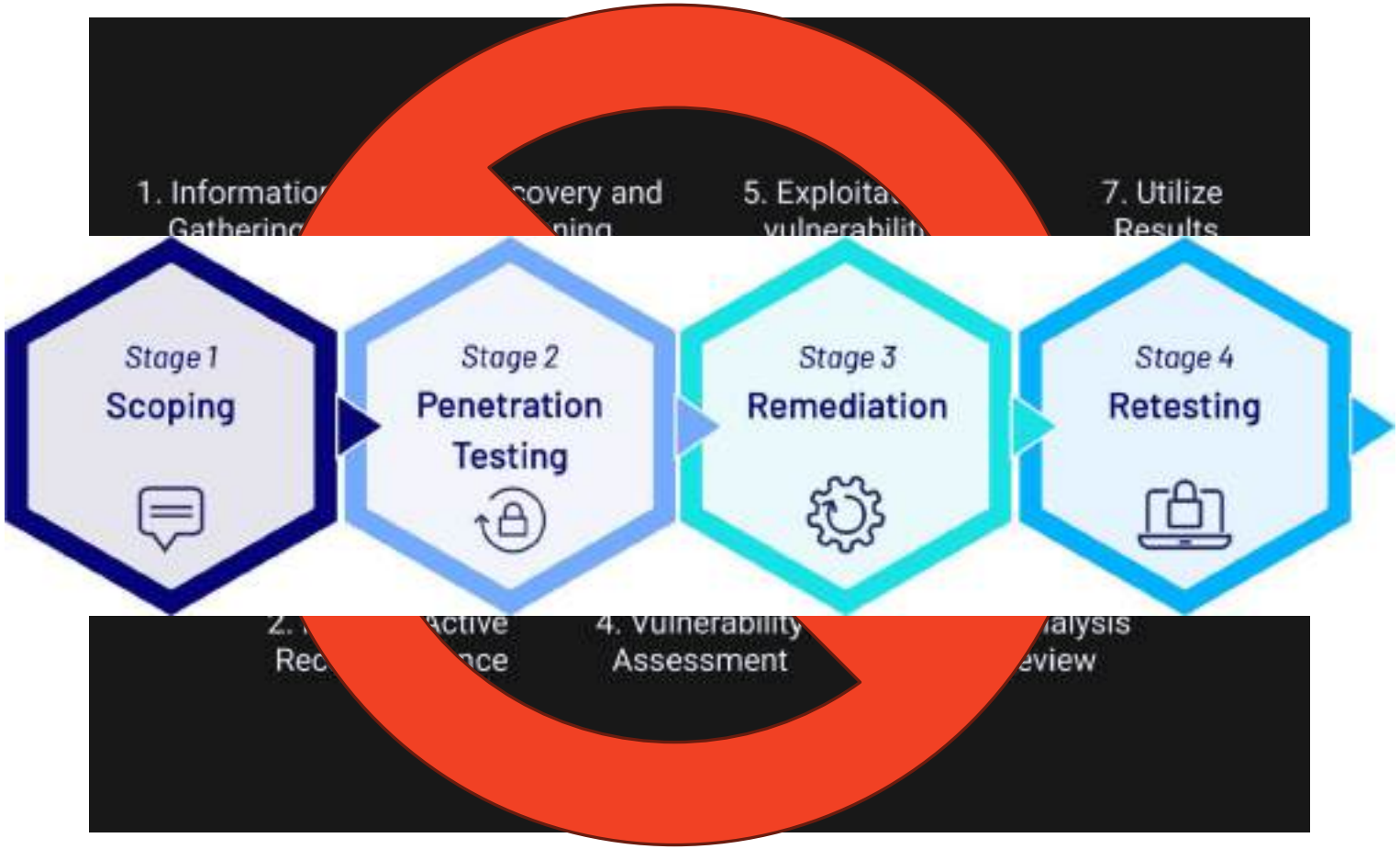
Summer School 2025

To find vulnerabilities in the asset/s for the client to fix them. In summary, to improve the product/environment. Period.

Easy, right?

Pentesting: Stages

Summer School 2025



Pentesting: Kinds

Summer School 2025

External Penetration Test

- Web, Web3, APIs
- Mobile (Android, iOS)
- Infrastructure, Networks
- IoT, Radio
- WiFi
- Binaries, Desktop Apps
- …

Internal Penetration Test

- Cloud: AWS, Azure, GCP
- Infra, Networks (Cloud/Prem)
- Offices
- Printers, OT
- …

Pentesting: Kinds

Summer School 2025

External Penetration Test

Final attack surface is exposed to the internet or via whitelist

Internal Penetration Test

Attack surface is within DMZ or the internal company network

Pentesting: The reality

Pentesting: The Real Stages

Summer School 2025

Kickoff

Setup

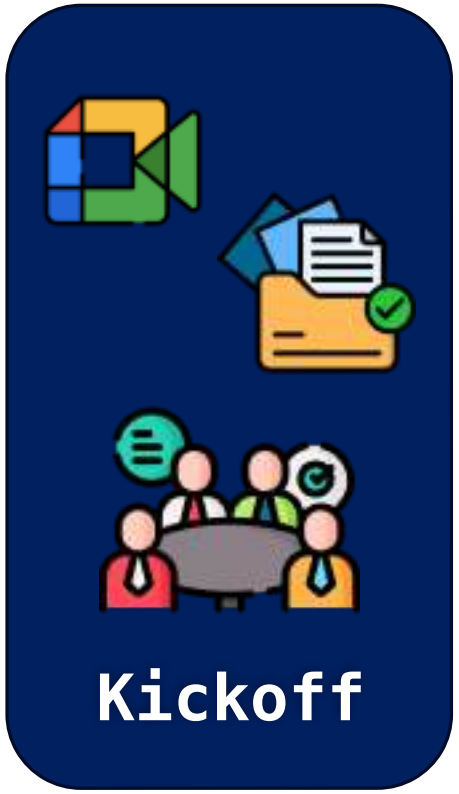
Pentest

Remediation

Retest

Pentesting: The Real Stages

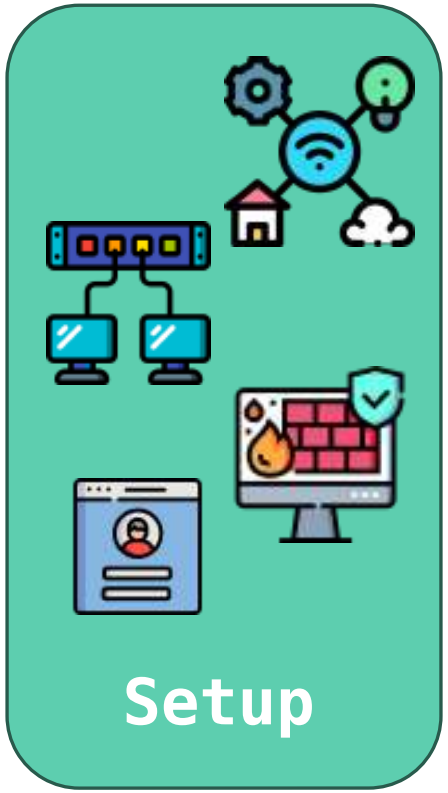
Summer School 2025



- Meet the scope, stablish ***RED LINES***
- Get to know the asset (show the app functionality…)
- Ask as much information as possible (depends):
  - Technologies, environment, roles, dependencies, arch…
- Ask for documentation (Swaggers, BOMs, Diagrams…)
- Establish an execution timeline (hard)
- “We don’t need a pentest”, “We do pentest every year”…

Pentesting: The Real Stages

Summer School 2025



- All the requirements must be clear:
  - The scope: Domains, endpoints, IPs, devices, etc…
  - The users required
  - The timeframe
  - The IPs for the whitelist / VPN to access
- Test the access to the assets and the users.
- Technical proposal?
- Get into the context beforehand.
- Issues: Back and forth. People not answering, client doesn’t understand its own infra, vibecoders, freelancers…

Pentesting: The Real Stages

Summer School 2025

Pentest

- Reconnaissance (attack surface).
- Implement the methodology: OWASP, MASTG, OSSTMM, NIST…
- Discover **as much vulns as possible**,
- Exploit the vulns, get in the dirt.
- Get all the evidences!!
- Report
- Issues: Unstable infra, half-implemented functionality…

Pentesting: The Real Stages

Summer School 2025

Remediation

- The company received the report and responsibilities are set.
- Sometimes, explanation meet is required.
- Policies establish time to fix vulns, ex. Critical – 2 weeks.
- Also problems: This is not a vuln, it’s a feature. Furious providers, back and forth. Not fixing the vulns (they appear the next year)…

Pentesting: The Real Stages

Summer School 2025

Retest

- Comeback over the pentest findings. Dig a bit deeper
- Revaluate the vulns (why?)
- Generate a new report with retest notes
- More problems: Devs tells to manager its fixed (its not), retest after 1+ years, not fixing vulns…

Kind of pentests (techs)

What do we pentest?

Basically, any asset. What I have done:

- Web: webs from the 2000s up to web3.
- Mobile apps: both from Android and iOS, from basic to banking apps.
- Android TPVs (Cards) and phone apps within.
- Hardware: Surveillance cameras, tills, light bulbs, display systems, laptops…
- IoT / Radio / WiFi: The above + LoRa/Zigbee sensors and gateways. WiFi nets.
- Offices: All kind of scrap, Hosts, VoIPs, printers, NAS, CCTVs, Keyboards.
- AI: Prompt injections, video summarizations
- AWS and cloud… I skip : K8s, Network policies, Firewall rules…

What do we pentest?

Summer School 2025

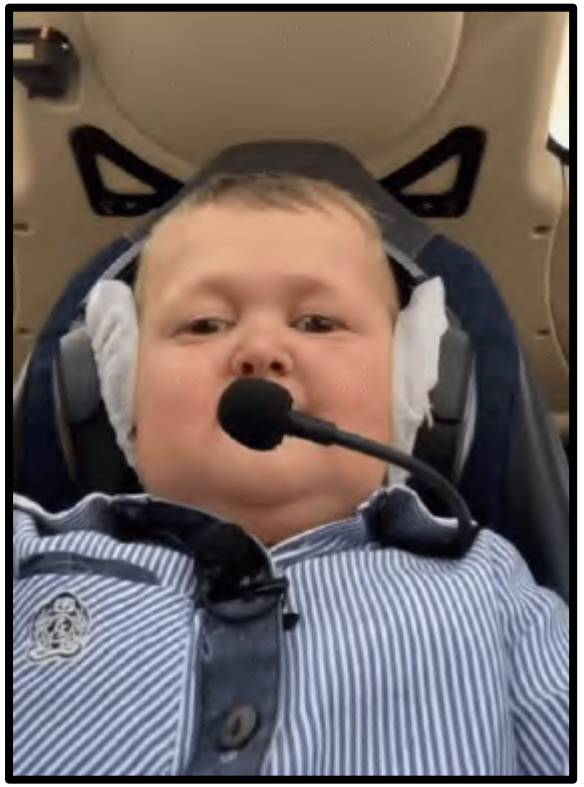
Too much theory, so me something technical please!!!

Web/API Pentest

Shopping list at the kickoff:

- Target URLs (host IP for further checks?)
- Users (as much roles as possible)
- Documentation (APIs)
- Whitelisted at WAF (why?)

Now we are ready!!!



Web/API Pentest

Summer School 2025

Information Gathering

- Manually explore the site
- Spider/crawl for missed or hidden content
- Check for files that expose content, such as robots.txt, sitemap.xml, .DS_Store
- Check the caches of major search engines for publicly accessible sites
- Check for differences in content based on User Agent (eg, Mobile sites, access as a Search engine Crawler)
- Perform Web Application Fingerprinting
- Identify technologies used
- Identify user roles
- Identify application entry points
- Identify client-side code
- Identify multiple versions/channels (e.g. web, mobile web, mobile app, web services)
- Identify co-hosted and related applications
- Identify all hostnames and ports
- Identify third-party hosted content

Configuration Management

- Check for commonly used application and administrative URLs
- Check for old, backup and unreferenced files
- Check HTTP methods supported and Cross Site Tracing (XST)
- Test file extensions handling
- Test for security HTTP headers (e.g. CSP, X-Frame-Options, HSTS)
- Test for policies (e.g. Flash, Silverlight, robots)
- Test for non-production data in live environment, and vice-versa
- Check for sensitive data in client-side code (e.g. API keys, credentials)

Secure Transmission

- Check SSL Version, Algorithms, Key length
- Check for Digital Certificate Validity (Duration, Signature and CN)
- Check credentials only delivered over HTTPS
- Check that the login form is delivered over HTTPS
- Check session tokens only delivered over HTTPS
- Check if HTTP Strict Transport Security (HSTS) in use

Authentication

- Test for user enumeration
- Test for authentication bypass
- Test for bruteforce protection
- Test password quality rules
- Test remember me functionality
- Test for autocomplete on password forms/input
- Test password reset and/or recovery
- Test password change process
- Test CAPTCHA
- Test multi factor authentication
- Test for logout functionality presence
- Test for cache management on HTTP (eg Pragma, Expires, Max-age)
- Test for default logins
- Test for user-accessible authentication history
- Test for out-of channel notification of account lockouts and successful password changes
- Test for consistent authentication across applications with shared authentication schema / SSO

Session Management

- Establish how session management is handled in the application (eg, tokens in cookies, token in URL)
- Check session tokens for cookie flags (httpOnly and secure)
- Check session cookie scope (path and domain)
- Check session cookie duration (expires and max-age)
- Check session termination after a maximum lifetime
- Check session termination after relative timeout
- Check session termination after logout
- Test to see if users can have multiple simultaneous sessions
- Test session cookies for randomness
- Confirm that new session tokens are issued on login, role change and logout
- Test for consistent session management across applications with shared session management
- Test for session puzzling
- Test for CSRF and clickjacking

Authorization

- Test for path traversal
- Test for bypassing authorization schema
- Test for vertical Access control problems (a.k.a. Privilege Escalation)
- Test for horizontal Access control problems (between two users at the same privilege level)
- Test for missing authorization

Data Validation

- Test for Reflected Cross Site Scripting
- Test for Stored Cross Site Scripting
- Test for DOM based Cross Site Scripting
- Test for Cross Site Flashing
- Test for HTML Injection
- Test for SQL Injection
- Test for SOQL Injection
- Test for LDAP Injection
- Test for ORM Injection
- Test for XML Injection
- Test for XXE Injection
- Test for SSI Injection
- Test for XPath Injection
- Test for XQuery Injection
- Test for IMAP/SMTP Injection
- Test for Code Injection
- Test for Expression Language Injection
- Test for Command Injection
- Test for Overflow (Stack, Heap and Integer)
- Test for Format String
- Test for incubated vulnerabilities
- Test for HTTP Splitting/Smuggling
- Test for HTTP Verb Tampering
- Test for Open Redirection
- Test for Local File Inclusion
- Test for Remote File Inclusion
- Compare client-side and server-side validation rules
- Test for NoSQL injection
- Test for HTTP parameter pollution
- Test for auto-binding
- Test for Mass Assignment
- Test for NULL/Invalid Session Cookie

Denial of Service

- Test for anti-automation
- Test for account lockout
- Test for HTTP protocol DoS
- Test for SQL wildcard DoS

Business Logic

- Test for feature misuse
- Test for lack of non-repudiation
- Test for trust relationships
- Test for integrity of data
- Test segregation of duties

Cryptography

- Check if data which should be encrypted is not
- Check for wrong algorithms usage depending on context
- Check for weak algorithms usage
- Check for proper use of salting
- Check for randomness functions

Risky Functionality - File Uploads

- Test that acceptable file types are whitelisted
- Test that file size limits, upload frequency and total file counts are defined and are enforced
- Test that file contents match the defined file type
- Test that all file uploads have Anti-Virus scanning in-place.
- Test that unsafe filenames are sanitised
- Test that uploaded files are not directly accessible within the web root
- Test that uploaded files are not served on the same hostname/port
- Test that files and other media are integrated with the authentication and authorisation schemas

Risky Functionality - Card Payment

- Test for known vulnerabilities and configuration issues on Web Server and Web Application
- Test for default or guessable password
- Test for non-production data in live environment, and vice-versa
- Test for Injection vulnerabilities
- Test for Buffer Overflows
- Test for Insecure Cryptographic Storage
- Test for Insufficient Transport Layer Protection
- Test for Improper Error Handling
- Test for all vulnerabilities with a CVSS v2 score > 4.0
- Test for Authentication and Authorization issues
- Test for CSRF

HTML 5

- Test Web Messaging
- Test for Web Storage SQL injection
- Check CORS implementation
- Check Offline Web Application

Web/API Pentest

Summer School 2025

- Information Gathering
- Configuration Management
- Data Validation
- Authentication
- Authorization
- Secure Transmission
- Session Management
- Denial of Service (?)
- Business Logic
- Cryptography
- Risky Functionality - File Uploads, Cards

Web/API Pentest

Summer School 2025

Wild things:

- GraphQL introspection > Unauth crypto transfers > 2M$
- Comment in JS > Secret URL > File naming convention > 10k DNIs
- Document metadata > Vuln PDF lib > RCE > 500k Invoices
- CompanyNameAdmin:CompanyNameWinter2020! > Full Dashboard Access > SSTI on Invoice Engine > LFI > 3k DNIs
- Vacation approval tool > Ofuscated actions with Ids > Reverse > Broken Access Control > Unlimited vacacions

Mobile Pentest

Summer School 2025

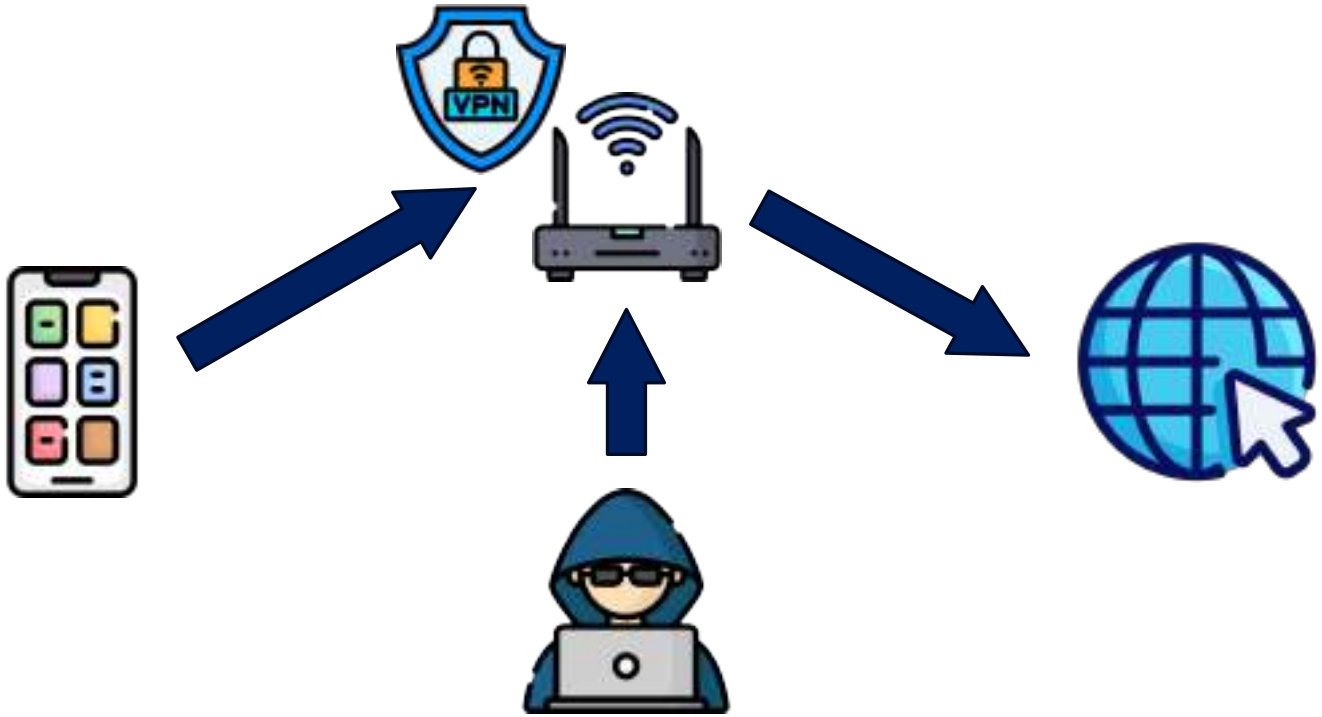
Mobile Pentest

Summer School 2025

Mobile Pentest

Summer School 2025

Mobile Setup



Jailbroken / Rooted Device

Lab setup:

- Physical device (?)
- Magisk
- LSPosed
- File Monitoring
- Drozer
- tcpdump

BurpSuite

Mobile Pentest

Summer School 2025

Basically API + New attack surface at the phone:

- IPCs (Inter-Process Communication)
- mydeust0app://thisisaninput
- Peripherals
- How data is stored
- Client-side functionality
- App resilience (environment, network…)
- Android != iOS

Mobile Pentest

Summer School 2025

OWASP MAS Testing Profiles:

- MAS-L1 - Essential Security: Low risk data, baseline
  - The device is not rooted/jailbroken
  - Not viewed as an adversary. Other apps are adversary.
- MAS-L2 - Advanced Security: High risk data
  - The device is rooted/jailbroken
  - Not viewed as an adversary. Other apps are adversary.
- MAS-R - Resilient Security: Safeguard their own assets and logic
  - The device is rooted/jailbroken
  - Viewed as an adversary. Other apps are adversary.

Mobile Pentest

Summer School 2025

Wild things:

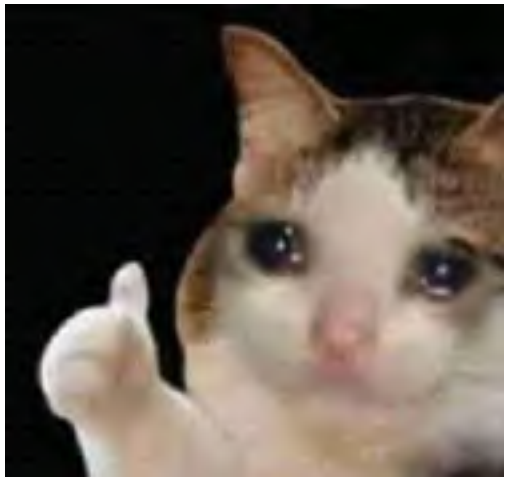
- Client-side logic > SMTP
- Card data into system logs
- Custom authorization > Reverse > Broken Access Control > Bypass email confirmation > Digital Key
- Export confidential data > Just print

Other Pentest

Summer School 2025

Other wild things:

- From provider VPN > Access internal network > Discover Conference gear > Default creds > CEO / StakeHolder meeting
- Hard OSINT over company > S3 Disclosure > Google Chat Screenshots > Unknown domain name > companyname:12345678 FTP with sensitive data



Red Team and Pentest: Close yet so far.

Red Team & Pentest

Summer School 2025

- **Penetration test:**
  - Limited assets (web, app, host…)
  - Short time (1-4 weeks)
  - No social engineering
  - No physical intrusion
  - Collaboration with DevSecOps (client)
- **Red Team:**
  - Any asset within the company
  - Long time (6-12 months)
  - Social engineering
  - Physical intrusion allowed (?)
  - No collaboration, only White Team communication

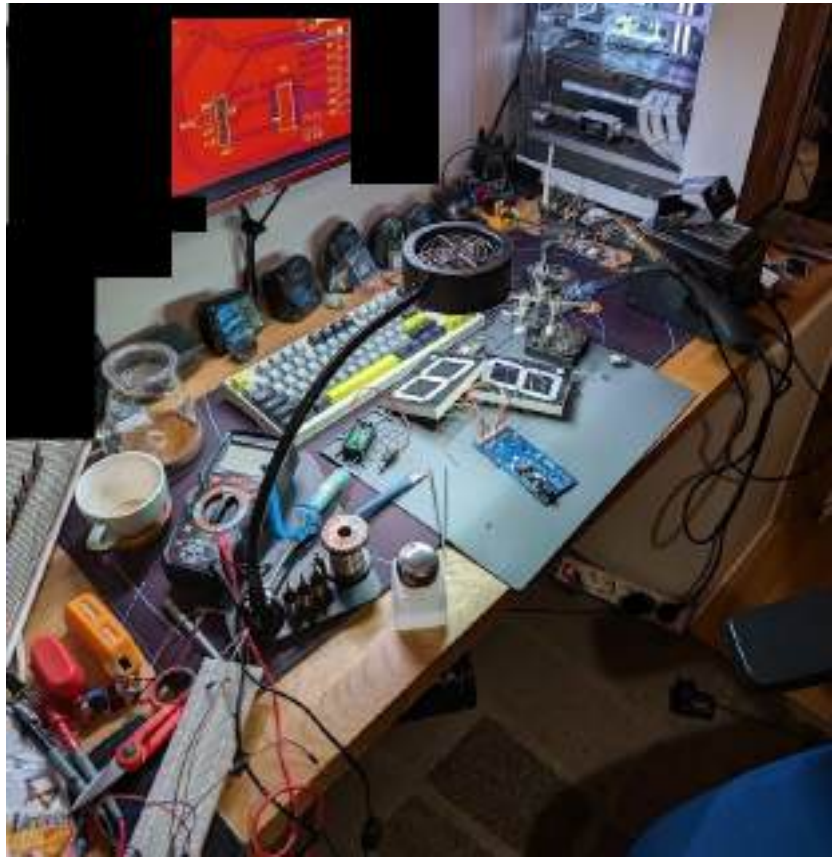
Radio / HW Gear

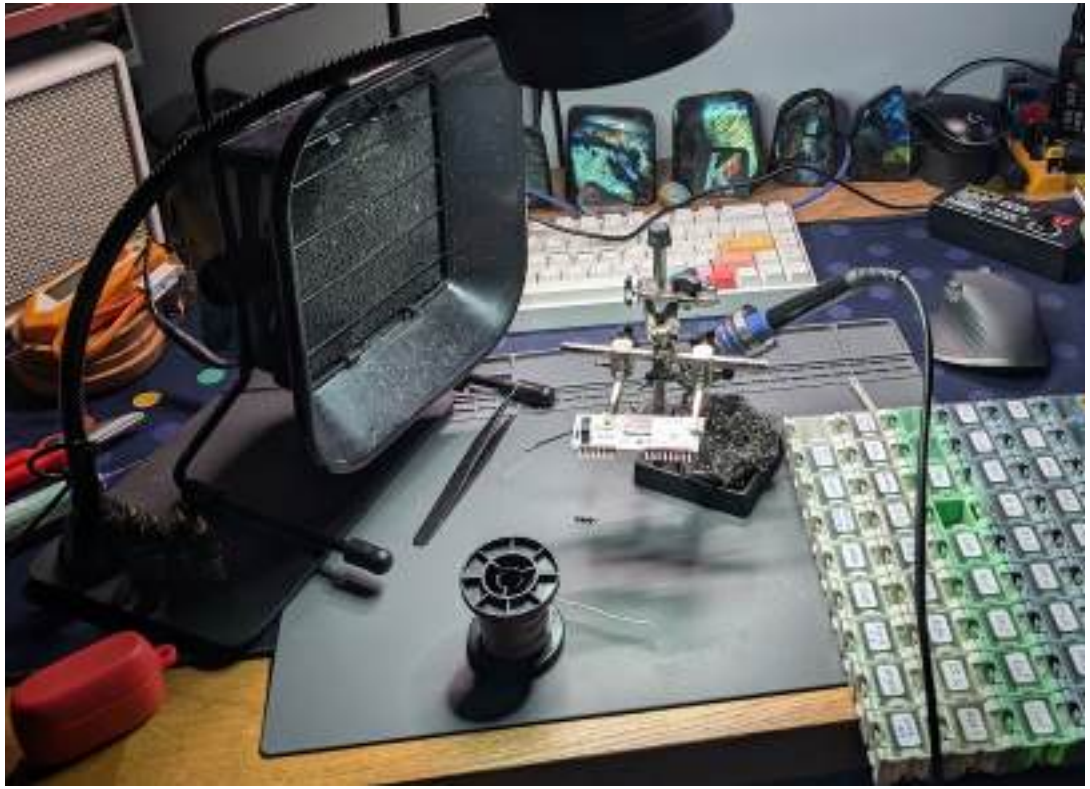
Miscellaneous

Summer School 2025

YouTube

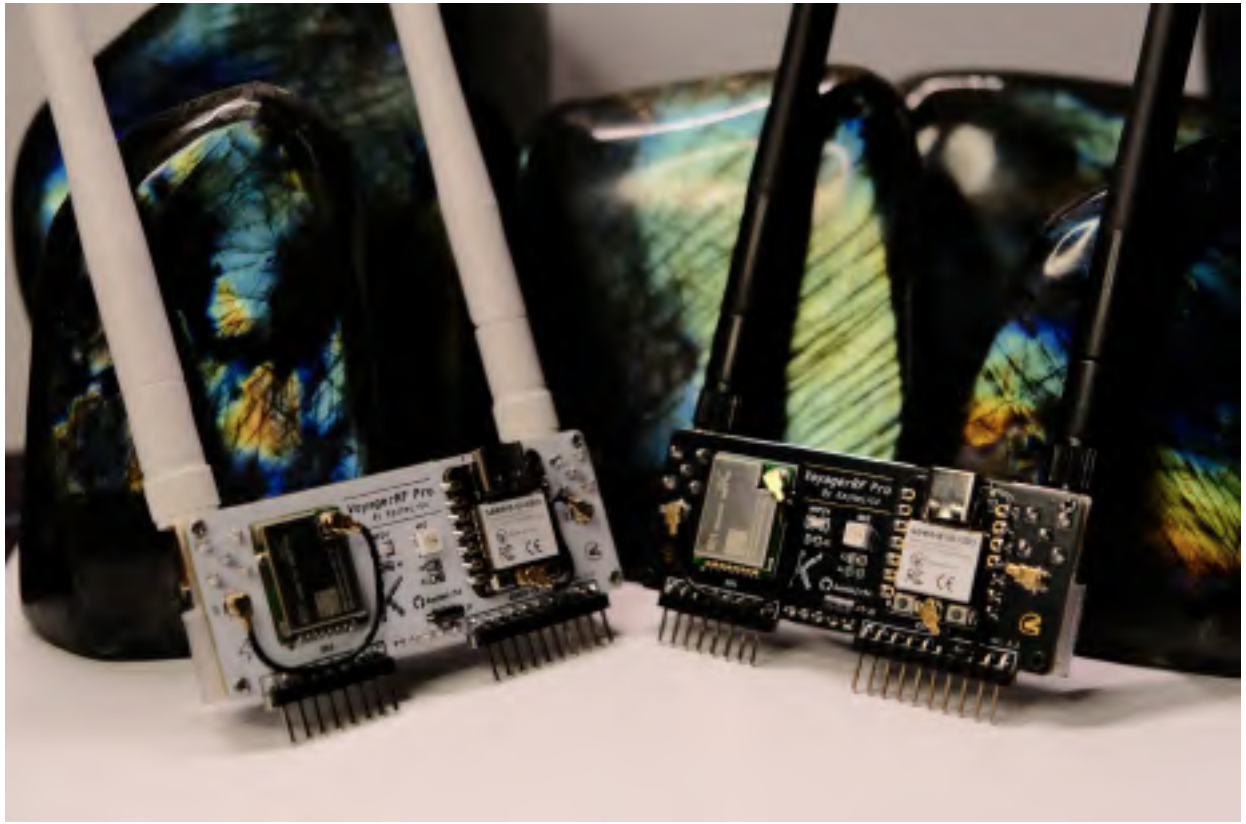
Miscellaneous





Miscellaneous

Summer School 2025



Thanks for your attention!

Thanks to 0xd3c0d3

Summer School 2025

A2SECURE