Web Application Penetration Testing (WAPT)

Masterclass Ethical Hacking

Frijlande Instituut voor Privacy en Security Management
Masterclass Cybersecurity – Ethical Hacking
Docent: Khalid Nakhli
Subject Matter Assist: Hafida Aboulbarakat
5 & 6 februari 2026

Web Application Penetration Testing (WAPT)
Welkom bij deze masterclass. Deze masterclass is specifiek ontworpen om u o.a. de diepte in te nemen van Web Application Penetration Testing, afgekort als WAPT.
Binnen de security-industrie is WAPT de meest voorkomende vorm van testing. De reden hiervoor is simpel: vrijwel elke organisatie heeft tegenwoordig één of meerdere assets op het web staan—of het nu gaat om complexe webapplicaties of webservers. Omdat het web het primaire platform is voor moderne software, is het waarborgen van de veiligheid hiervan een absolute prioriteit.

Onze Filosofie: De *Hackers Mindset* begrijpen
De kern van deze masterclass rust op de overtuiging dat technische expertise en strategisch management onlosmakelijk met elkaar verbonden zijn. Voor o.a. de Privacy Manager, DPO en CISO betekent dit de vertaling van theoretische kaders naar de praktijk. In plaats van security enkel als beleidsstuk te benaderen, leert u de technische realiteit van de *adversarial mindset* begrijpen. Dit stelt u in staat om risico's niet alleen te benoemen, maar de werkelijke impact van kwetsbaarheden op de organisatie en de privacy van betrokkenen effectief te valideren en te communiceren.

Hands-on Methodologie: Deze masterclass is zeer *practice-based*. We werken met een live applicatie genaamd 'Secure Bank', een omgeving die specifiek is ontworpen met kwetsbaarheden om als optimaal trainingsveld te dienen. Ondanks de tweedaagse opzet en de nadruk op kennisoverdracht kent deze masterclass *ethical hacking* een steile leercurve. In lijn met de professionele praktijk wordt niet verondersteld dat deelnemers alle kennis vooraf beheersen, maar wel dat zij in staat zijn om zelfstandig relevante informatie te identificeren, te duiden en zich deze effectief eigen te maken. Gezien de omvang en diversiteit van het vakgebied *ethical hacking* is het onmogelijk om alle onderwerpen binnen dit domein volledig te behandelen binnen deze tweedaagse masterclass.

Tot slot: Binnen de hackers wereld, hoor je vaak het motto "*try harder*". Deze omarmt vele zaken dan alleen harder werken. Het betreft het totale plaatje van; er is niet maar 1 weg maar meerdere oplossingen, *kan niet* bestaat niet, als iets niet lukt is het feedback dat dit je huidige skills zijn en kansen creëert om iets nieuws te leren, verander je aanpak als iets niet werkt.

NLP: "*Als je altijd doet wat je deed, dan krijg je altijd wat je kreeg!*"

Frijlande
INSTITUUT VOOR
PRIVACY EN SECURITY MANAGEMENT

Inhoudsopgave

1. Belangrijke Informatie: Ethical Disclosure

Doel van de Masterclass Het primaire doel van deze masterclass is om studenten een uniek inzicht te geven in de denkwijze en methodieken van een hacker (**the adversarial mindset**). Door kwetsbaarheden vanuit dit perspectief te benaderen, ontwikkel je het vermogen om complexe technische risico's te identificeren en deze effectief te vertalen naar strategisch advies binnen jouw specifieke rol in de organisatie.

Ethische en Educatieve Doeleinden De kennis en vaardigheden die tijdens deze masterclass worden overgedragen, zijn **strikt en uitsluitend bedoeld voor ethische en educatieve doeleinden**. Het is de verantwoordelijkheid van de student om deze kennis integer toe te passen met als doel de digitale weerbaarheid van organisaties te vergroten.

Wettelijke Kaders en Beperkingen Het uitvoeren van penetratietesten of andere vormen van security-onderzoek op systemen, netwerken of applicaties van derde partijen is **ten strengste verboden**, tenzij er vooraf expliciete, schriftelijke toestemming is verkregen van de rechtmatige eigenaar (bijvoorbeeld in de vorm van een *Rules of Engagement* document). Ongeautoriseerde toegang tot computersystemen is een strafbaar feit.

Gebruik van de Lab-omgeving Alle oefeningen, tools en technieken die tijdens deze masterclass worden gedoceerd, mogen **alleen worden uitgevoerd op je eigen computeromgeving**. Dit betreft hardware en virtuele omgevingen (zoals de Docker-containers van Secure Bank) waarvan jij de enige eigenaar en beheerder bent. Het is niet toegestaan om deze tools te richten op netwerken van de onderwijsinstelling, werkgevers of publieke internet-assets.

Door deelname aan deze masterclass verklaart de student bovenstaande voorwaarden te hebben begrepen en verbindt de student zich ertoe te allen tijde te handelen binnen de grenzen van de wet en de ethische beroepscode.

2. VMware Workstation (Desktop) downloaden, installeren en instellen

Dit praktische en stap-voor-stap handboek begeleidt je o.a. bij het downloaden van VMware Workstation Pro (desktop-virtualisatie-software) en het vervolgens installeren en configureren op een Windows- of Linux-systeem. VMware Workstation Pro is een krachtige desktop-hypervisor waarmee je meerdere besturingssystemen als virtuele machines (VM's) kunt draaien.

VMware Workstation Pro is een desktop-hypervisor waarmee je meerdere virtuele besturingssystemen naast elkaar kunt draaien op één fysieke computer (Windows, Linux etc.). Het ondersteunt onder meer:

- Meerdere gelijktijdige VM's
- Snapshots en geavanceerde netwerk- en opslagconfiguraties
- Gast-OS-ondersteuning voor Windows, Linux, enz.

Stap 1 — VMware Workstation Pro downloaden
Officiële startpagina (Broadcom Support Portal)
https://support.broadcom.com/
(Je moet hier een gratis account aanmaken en inloggen om te downloaden.)
Downloadprocedure:
Ga naar de Broadcom Support Portal en log in (gratis account).
Open de link voor VMware Workstation Pro downloads:
https://support.broadcom.com/group/ecx/productdownloads?subfamily=VMware%20Workstation%20Pro&freeDownloads=true
Selecteer de nieuwste versie (bijv. 17.6.4 voor Windows of Linux).
Klik op I agree / accepteer de EULA om de download te starten.

Bewaar het installer-bestand lokaal.
Bestanden:
Windows: .exe-installer

Stap 2 — Installatie op Windows
Voorbereiding
• Je hebt Administrator-rechten nodig.
• Schakel eventuele antivirus- of UAC-meldingen tijdelijk uit indien nodig.

Installatie Stappen
Dubbelklik op het gedownloade .exe-bestand (bv. VMware-workstation-full-17.x.x.exe).
Klik Next bij de wizard-welkomstpagina.
Lees en accepteer de License Agreement → Next.
Kies de installatiemap of laat de standaardlocatie staan → Next.
Kies optionele componenten als gewenst → Next.
Start de installatie → Install.
Wacht tot de setup voltooid is → klik Finish.
De installatie duurt meestal slechts enkele minuten.

Je hoeft geen licentiesleutel in te voeren omdat Workstation Pro nu gratis is.

Stap 3 — Installatie op Linux (Ubuntu/Fedora/etc.)
Als je Linux gebruikt:
3.1 Open een terminal.
3.2 Maak het .bundle-bestand uitvoerbaar:

```
bash

chmod +x VMware-Workstation-*.bundle
```

3.3 Voer de installer uit met root-rechten:

```
bash

sudo ./VMware-Workstation-*.bundle
```

3.4 Volg de grafische wizard om de installatie te voltooien.
**Zorg dat je eerdere VMware-versies verwijderd hebt voordat je een upgrade installeert.*
Stap 4 — Eerste keer starten & configureren
Open VMware Workstation Pro via Start-menu (Windows) of desktop-menu (Linux).
Bij de eerste start kun je instellingen zoals:
Update-controle
Verbeterde functie-opties
Gebruikservaring-instellingen aanpassen.
Stap 5 — Virtuele machine aanmaken
Klik bovenin op File > New Virtual Machine.
Kies Typical (aanbevolen) of Custom (voor gevorderden).
Selecteer het ISO-bestand van het gast-OS (bijv. Windows 10/11, Ubuntu).
Volg de wizard:
Opslaglocatie

Schijfgrootte
RAM / CPU-toewijzing
Klik Finish en start de VM.

Handige Tips & Fouten oplossen (Hafida)
Systeemvereisten
64-bit host-OS (Windows 10/11, Linux x86_64).
Virtualisatie ingeschakeld in BIOS/UEFI (VT-x/AMD-V).
Minimaal 8 GB RAM aanbevolen.

Veelvoorkomende installatieproblemen
Installer start niet: probeer met rechts-klik → Run as Administrator.

Foutmelding over runtimes: installeer eerst Microsoft Visual C++ runtimes.
✓ Extra functionaliteit
Snapshots: sla de huidige VM-status op en herstel later.
Netwerk-instellingen: bridged, NAT of host-only.

USB-doorvoer: direct toegang tot USB-apparaten vanuit je VM.

Samenvatting — Stappen

Stap	Actie
1	VMware Workstation Pro downloaden via Broadcom Support Portal
2	Installatie uitvoeren (Windows of Linux)
3	First-time configuratie
4	Nieuwe virtuele machine aanmaken en starten

Downloadlinks
Broadcom Support Portal (download & account):
https://support.broadcom.com/

Workstation Pro downloads (free):
https://support.broadcom.com/group/ecx/productdownloads?subfamily=VMware%20Workstation%20Pro&freeDownloads=true

3. Kali Linux installeren als image in VMware Workstation Pro (Desktop)

Stap voor stap hoe Kali Linux als kant-en-klare image (virtuele machine) wordt geïnstalleerd in VMware Workstation Pro (desktopversie). De handleiding is geschikt voor beginners en vereist geen eerdere Linux-ervaring.

Stap 1 – *VMware Workstation Pro* controleren

Open *VMware Workstation Pro*.

Stap 2 – *Kali Linux VMware image* downloaden

Ga naar de officiële Kali Linux downloadpagina:

https://www.kali.org/get-kali/#kali-installer-images

Kies voor:

Windows: *x86_64* of

Apple M(chip): Apple Silicon (*ARM64*)

Download het bestand:

kali-linux-2025.4-installer-amd64.iso

Stap 3 – Kali Linux openen in VMware

Open VMware Workstation Pro.

Klik op File > Open.

Navigeer naar de uitgepakte map.

Selecteer het .iso bestand en klik op Open.

Stap 5 – Virtuele machine instellingen controleren

Klik op '*Edit virtual machine settings*'.

Aanbevolen instellingen:

- Geheugen: 4 tot 8 GB RAM

- Processors: minimaal 2 CPU's

- Netwerkadapter: NAT

Klik op OK om de instellingen op te slaan.

Stap 6 – Kali Linux starten

Selecteer de Kali Linux virtuele machine.

Klik op ‘Power on this virtual machine’.

Kies bij de vraag ‘Did you copy or move this virtual machine?’ voor ‘I Copied It’.

Stap 7 – Inloggen in Kali Linux

Gebruik de standaardgegevens:

Gebruikersnaam: kali

Wachtwoord: kali

Stap 8 – Updates uitvoeren

Open een terminal en voer uit:

sudo apt update

sudo apt upgrade

Resultaat

Kali Linux draait nu volledig als virtuele machine binnen VMware Workstation Pro en is klaar voor gebruik in opleiding, testing en security labs.

3.1. Linux

Linux is opgebouwd uit lagen.

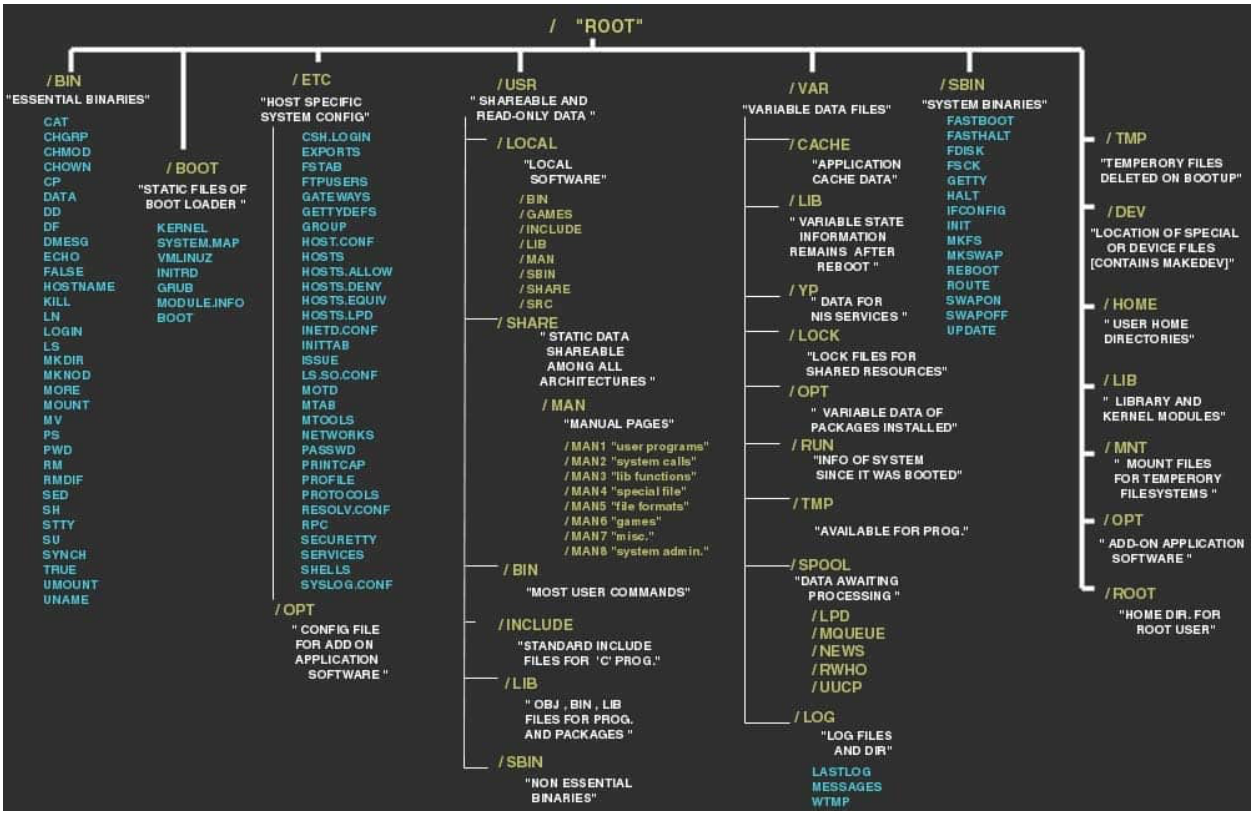
- **De Kern (Kernel):** Dit is het hart van Linux. Het praat direct met de hardware van je laptop (de processor, het geheugen etc). Het zorgt ervoor dat alles draait.
- **De Schil (Shell/Terminal):** Dit is de laag waar hackers van houden. In plaats van klikken op icoontjes, geef je hier directe tekstcommando's aan de computer. Het is sneller, efficienter, krachtiger en geeft je totale controle.
- **De Buitenkant (Desktop Environment):** Dit is de grafische laag met vensters en menu's, zodat het er voor een normale gebruiker gewoon uitziet als Windows.

Alles is een Bestand.
Dit is het belangrijkste principe van Linux: **"*Everything is a file*"**. In Windows heb je ingewikkelde instellingenmenu's. In Linux is alles - van je muisinstellingen tot de tekst van een website - opgeslagen als een simpel tekstbestandje.
Waarom is dit handig voor een hacker? Je kunt alles inzien en of veranderen naar eigen wens. Maar ook als je weet waar het tekstbestandje van de wachtwoorden staat, hoef je alleen dat bestandje te "lezen" om binnen te komen.

Rechten: Wie is de baas?
In Linux is veiligheid ingebouwd via een streng rechtensysteem. Er zijn drie niveaus:
Jij (User): Mag alleen bij je eigen documenten.
De Groep (Group): Een verzameling gebruikers die samen bij bestanden mogen.
De Superuser (Root): De "God-modus". Deze gebruiker mag alles, inclusief het hele systeem wissen. In de masterclass gebruiken we vaak het commando `sudo` (SuperUser Do) om tijdelijk deze macht op te eisen.

Waarom gebruiken hackers Linux (Kali)?
Kali Linux (de versie die wij gebruiken) is eigenlijk gewoon Linux waar al 600+ hack-tools (zoals Nmap en Burp Suite) vooraf zijn ingebouwd. Het wordt vooral als uitgangspunt gebruikt voor o.a. hackers en digitaal forensisch onderzoekers. Daar waar nodig worden meerdere tools aan toegevoegd.

De Directory Structuur



4. Nmap

Verkenning, ook wel *Reconnaissance* of *Information Gathering* genoemd, is de belangrijkste fase van een penetratietest. Zonder goede verkenning schiet je met hagel in het donker.

De fundamenten van de aanval:
- Voorkomen van ontdekking
- Specifiek aanvallen
- De juiste inzet van tools

In de wereld van cybersecurity is *Nmap* (*Network Mapper*) het ultieme gereedschap voor verkenning. Als de bankapplicatie een gebouw is, dan is *Nmap* de tool waarmee je om het gebouw heen loopt om te kijken welke deuren en ramen (poorten) openstaan en welk slot (beveiliging) er op de deur zit.
Het is een open-source netwerkscanner die speciaal gecreerde pakketjes naar een doelwit stuurt en vervolgens het antwoord daarvan analyseert. Hierdoor krijg je een verslag van het systeem.

De belangrijkste vragen die Nmap o.a. beantwoordt:

1. **Leeft het doelwit?**
2. **Welke poorten staan open?**
3. **Welke software en versie draaien er?**

4.1. Korte Gebruikershandleiding (Cheat Sheet)

In de masterclass gebruiken we Nmap via de **CLI** (terminal) in Kali Linux. De basisopbouw van een commando is altijd: nmap [opties] [doelwit].

Wil je weten of een IP-adres actief is en wat de meest voorkomende poorten zijn?

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.1
```

4.2. Versies achterhalen

Wil je weten of de bank een verouderde webserver gebruikt? Met de **-sV** parameter dwing je Nmap om de versienummers op te vragen.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.1
```

> ***Output:*** In plaats van alleen "Port 80 is open", zie je nu bijvoorbeeld "Apache 2.4.41". Hierop kun je vervolgens gaan zoeken naar lekken.

4.3. De "Agressieve" Scan

De “agressieve” scan combineert meerder opties. Het combineert o.a. versiedetectie, besturingssysteem-herkenning en een route-analyse (traceroute).

```
┌──(kali㉿kali)-[~]

└─$ nmap -A 192.168.1.1
```

4.4. Specifieke poorten scannen

Soms wil je alleen weten of de database (poort 3306) of het webverkeer (poort 80/443) openstaat.

```
┌──(kali㉿kali)-[~]

└─$ nmap -p 80,443 192.168.1.1
```

5. CTF (Capture The Flag)

CTF (Capture The Flag) de ultieme oefenterrein voor beginners en professionals --ethische hackers en cybercriminelen--. Het is een competitie waarbij deelnemers technische puzzels oplossen om een verborgen stukje tekst te vinden: de "*flag*" (bijvoorbeeld: CTF{Y0u_H4ck3d_Frijl}).

5.1. Waarom is een CTF nuttig?

Een CTF is veel meer dan een spelletje; het is een versnellingsbak voor je leerproces:

- **Veilige Leeromgeving:** Je kunt legaal experimenteren met gevaarlijke technieken zonder dat je bang hoeft te zijn dat je een echte server platlegt of de wet overtreedt.
- **Probleemoplossend Denken:** Het leert je om "*out-of-the-box*" te denken. Je leert hoe software werkt door het kapot te maken ☺
- **Up-to-date Technieken:** *CTF*-makers verwerken vaak de allernieuwste beveiligingslekken in hun puzzels, waardoor je kennis altijd relevant blijft.
- **Portfolio opbouwen:** Voor sommige opdrachtgevers/werkgever zegt een hoge score op een platform als Hack The Box vaak meer dan een certificaat.

5.2. CTF vs. Echt Hacken: Wat is het verschil?

Hoewel de technieken (zoals bijvoorbeeld *SQL Injection* of *Nmap*) hetzelfde zijn, verschilt de context enorm:

Kenmerk	*CTF (Capture The Flag)*	*Echt Hacken (Pentesting)*
Het Doel	Het vinden van een specifieke "flag".	Het vinden en rapporteren van *alle* kwetsbaarheden.
De Route	Er is vaak één specifiek "bedoeld" pad naar de oplossing.	Er zijn talloze manieren om binnen te komen (of helemaal geen).
Realiteit	Systemen zijn vaak opzettelijk kwetsbaar gemaakt.	Systemen zijn (meestal) zo goed mogelijk beveiligd.
Rapportage	Je voert de vlag in en krijgt punten.	Je schrijft een uitgebreid rapport over de impact en de oplossing.

5.3. Stappenplan: *Hack The Box (HTB)* Account aanmaken

Hack The Box is het meest populaire platform voor *CTF's*. Vroeger moest je de website "hacken" om een account te krijgen, maar tegenwoordig is het proces toegankelijker gemaakt.

Stap 1: Registratie

1. Ga naar hackthebox.com.
2. Klik op **"Join Now"**.
3. Kies voor **"Individual"**.
4. Vul je gegevens in (gebruik bij voorkeur een apart e-mailadres voor je hacking-hobby).

Stap 2: Je Kali Linux voorbereiden

Om de machines op HTB aan te vallen, heb je een VPN-verbinding nodig zodat jouw computer "denkt" dat hij in hetzelfde netwerk zit als de doelwit-server.

1. Log in op je HTB Dashboard.
2. Ga naar *"Connect to HTB"* (meestal rechtsboven).
3. Kies *"Starting Point"* of *"Machines"* en selecteer *"OpenVPN"*.
4. Download het .ovpn configuratiebestand.

Stap 3: De VPN-verbinding starten

Open je terminal in Kali Linux:

```
┌──(kali㉿kali)-[~]
└─$ sudo openvpn jouw_bestandsnaam.ovpn
```

Stap 4: Je eerste "Flag" veroveren

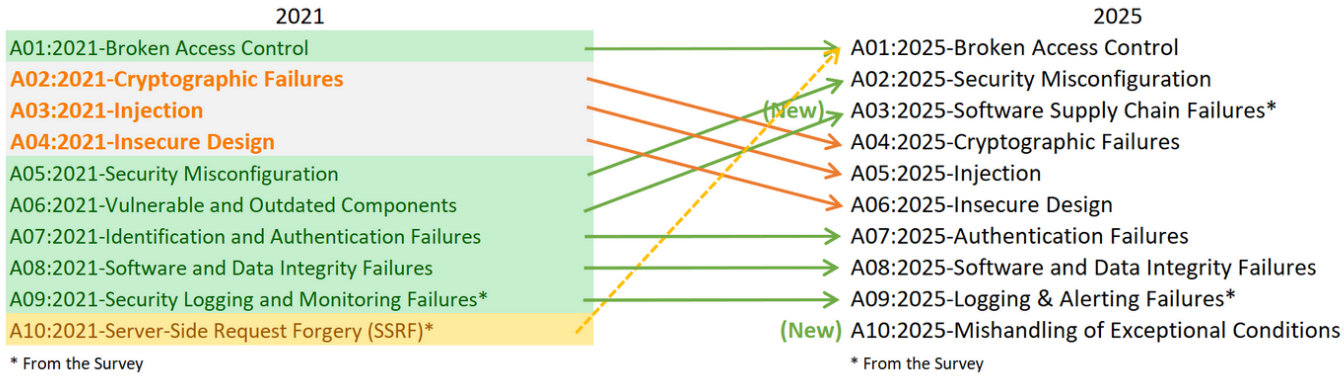
1. Ga op de website naar de sectie **"Starting Point"**. Dit zijn speciaal ontworpen machines voor beginners.
2. Start de machine (zie presentatie).
3. Gebruik je nieuwe skills (**Nmap**, **ls**, **cat**) om de flag.txt op de server te vinden.
4. Kopieer de code naar de HTB website en verdien je eerste punten!

6. Installatie Docker en Webapplicatie *Securebank*

6.1. SecureBank

SecureBank is een *.NET-applicatie* geschreven in *C#*, ontworpen om te leren over softwarebeveiliging en ethisch hacken. De licentie betreft een *MIT*-licentie en stelt gebruikers in staat de software onbeperkt te gebruiken, wijzigen, verkopen en distribueren in commerciële en besloten projecten ontwikkeld door SSRD
Orginele versie *SecureBank* te downloaden via *Github*: https://github.com/ssrdio/SecureBank
Gewijzigde versie *SecureBank*: https://github.com/Casaoui2/WAPT/raw/refs/heads/main/secure-bank.zip

Het is gebasseerd op *OWASP top 10 Web Application Security Risks*
https://owasp.org/Top10/2025/0x00_2025-Introduction/



Bron: The OWASP Foundation

De wijzigingen die voor deze versie zijn doorgevoerd, zijn onder andere:
- Wijzigingen in uiterlijk en stijl
- Meer kwetsbaarheden
- Kleine *bugfixes*

Deze webapplicatie van *SecureBank* bevat meer dan 40 kwetsbaarheden.
Om de webapplicatie te activeren, dien je *docker* en *docker compose* geïnstalleerd hebben.
Om te controleren of je docker en docker compose al hebt geïnstalleerd, kun je de volgende commando's uitvoeren in de terminal:

```
┌──(kali㉿kali)-[~]
└─$ which docker
/usr/bin/docker
```

```
┌──(kali㉿kali)-[~]
└─$ which docker-compose
/usr/local/bin/docker-compose
```

6.2. Installeren docker en docker compose

Stap 1: open je terminal in Kali Linux

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
```

Deze optie vertelt *apt* om de pakketlijst bij te werken naar de nieuwste versies die beschikbaar zijn in de *repositories* (opslagplaatsen) waaruit uw systeemsoftware haalt.

Dit staat voor "*Advanced Packaging Tool*" en is een pakketbeheerprogramma voor *Debian*-gebaseerde distributies. "*apt*" wordt gebruikt om software te installeren, up-to-date te houden en te verwijderen. Het ondersteunt ook het oplossen van afhankelijkheden tussen pakketten.

Dit staat voor "*Super User DO*" en geeft de gebruiker tijdelijk supergebruikersrechten. Met "*sudo*" kan een gebruiker bepaalde taken uitvoeren die normaal gesproken alleen beschikbaar zijn voor de *root-gebruiker* (de systeembeheerder). Dit wordt gedaan om te voorkomen dat ongewilde veranderingen worden aangebracht in het systeem zonder expliciete toestemming.

Stap 2: Kali Linux terminal

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install -y docker.io
```

Dit is de naam van het pakket dat geïnstalleerd moet worden. In dit geval is het Docker, een populaire containerisatietechnologie die het mogelijk maakt om toepassingen te ontwikkelen, testen en implementeren in containers. De "*.io*" extensie verwijst naar de officiële Docker-pakketnaam in de *Ubuntu-repositories*.

Dit is een optie die "*apt*" instrueert om "*yes*" te antwoorden op alle vragen die normaal gesproken tijdens het installatieproces worden gesteld, zoals of je zeker bent dat je de pakketten wilt installeren en of je akkoord gaat met de hoeveelheid schijfruimte die nodig is. Deze optie wordt vaak gebruikt in scripts om het installatieproces te automatiseren zonder menselijke interactie.

Stap 3: Kali Linux terminal

```
┌──(kali㉿kali)-[~]
└─$ sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -S)-$(uname -m)" -o /usr/local/bin/docker-compose
```

curl: Dit is een *command-line tool* voor het overdragen van gegevens met *URL's* (*Uniform Resource Locators*). *curl* wordt gebruikt om bestanden te downloaden of uploads naar servers te verzenden.

-L: Deze optie vertelt *curl* om doorverwijzingen (*redirects*) te volgen. Dit betekent dat als de *URL* die je specificeert een doorverwijzing is naar een andere locatie, *curl* automatisch naar die nieuwe locatie gaat en het downloadproces daar voortzet.

https://github.com/docker/compose/releases/latest/download/: Dit is de basis-*URL* van waaruit *Docker Compose* wordt gedownload en verwijst naar de laatste release van *Docker Compose* op *GitHub*.

docker-compose-$(uname-s)-$(uname -m)": Dit gedeelte specificeert het exacte bestand dat moet worden gedownload. De variabelen *$(uname -s)* en *$(uname -m)* zijn commando's die dynamisch de juiste waarden oplevert. De *-s* geeft het besturingssysteemtype (*operating system name*), zoals "*Linux*". De *-m* geeft de architectuur van de processor, zoals "*x86_64*" voor 64-bits systemen of "*arm64*" voor *ARM*-gebaseerde systemen. Zo download je wat overeenkomt met jouw systeemarchitectuur.

-o /usr/local/bin/docker-compose: Deze optie specificeert de output-bestandsnaam van de download. In plaats van naar de standaarduitvoer (*stdout*) te schrijven, wordt het gedownloade bestand *docker-compose* opgeslagen in */usr/local/bin/* .
Dit is een veelgebruikte locatie voor het installeren van uitvoerbare bestanden die niet door het pakketbeheersysteem zijn geïnstalleerd. Door Docker Compose hier te plaatsen, maakt het beschikbaar als een systeembreed commando.

Stap 4: Kali Linux terminal

```
┌──(kali㉿kali)-[~]
└─$ chmod +x /usr/local/bin/docker-compose
```

chmod: staat voor change mode en wordt gebruikt om bestandsrechten te wijzigen zoals lezen (*read*), schrijven (*write*) en uitvoeren (*execute*). De optie **+x** waarbij "+" de betekenis heeft voeg een recht toe en de "*x*" de betekenis maak het uitvoerbaar.

6.3. Deploy Securbank Applicatie

Om de applicatie te gebruiken voor ons testdoeleinde, zullen we deze moeten "*deployen*" (uitrollen). Door de applicatie uit te rollen wordt de applicatie voorbereid voor gebruik door het te configureren, te starten en beschikbaar gesteld voor de gebruiker(s).

```
┌──(kali㉿kali)-[~]
└─$ docker compose -f docker-compose.yml up -d
```

Als je de broncode van de applicatie aanpast en deze opnieuw wilt opbouwen om de wijzigingen te zien, dan kun je het build-commando als volgt gebruiken:

```
┌──(kali㉿kali)-[~]
└─$ docker compose -f docker-compose.yml build
```

En vervolgens de applicatie opnieuw opstarten

* Credentials
  Default credentials for admin user.

Username: admin@hexdump.sh
Password: admin

7. Installatie Burpsuite & Wat doet de tool?

7.1. Burp Suite Community Edition

Burp Suite is de industry-standard HTTP proxy. Het wordt gebruikt voor het analyseren, onderscheppen en modificeren van webverkeer tussen de browser en de server. Het programma is geschreven in ***Java*** en is ontwikkeld door *PortSwigger*.

In deze masterclass gebruiken we de ***Community Edition*** om te begrijpen hoe een hacker het verkeer tussen zijn browser en de bankapplicatie manipuleert.
Om *Burp Suite* te gebruiken is het noodzakelijk om een *Java* omgeving te hebben in ons systeem.

Ga naar de terminal in Kali Linux en controleer of *Java* op je systeem staat

```
┌──(kali㉿kali)-[~]
└─$ java –version
```

7.2. Installatie Burp Suite

Als *Java* is geïnstalleerd kunnen we vervolgens *Burp Suite Community Edition* downloaden. Omdat we Hackers zijn, doen wij dit uiteraard niet volgens de gewone weg via de website https://portswigger.net/burp/releases#community, maar gaan onze terminal hiervoor gebruiken.

```
┌──(kali㉿kali)-[~]
└─$ curl
"https://portswigger.net/burp/releases/download?product=community&version=2025.12.5&type
=Linux" > burpsuite-installer.sh
```

```
┌──(kali㉿kali)-[~]
└─$ chmod +x burpsuite-installer.sh
```

```
┌──(kali㉿kali)-[~]
└─$ ./burpsuite-installer.sh
```

Volg vervolgens de Installatie Wizard voor installatie. *Burp Suite* is nu klaar voor gebruik.

7.2. Installatie Burp Suite

Wanneer je *Burp Suite* start, krijg je twee beginschermen te zien.
Het eerste scherm stelt je in staat om het project te kiezen waarmee je gaat werken.
Een *Burp Suite-project* bevat alle *HTTP*-verzoeken die binnen dat specifieke project zijn uitgevoerd.
In de *Community Edition* is het echter niet mogelijk om met permanente projecten te werken. Je hebt alleen toegang tot een tijdelijk project, dat wordt opgeslagen in het **RAM-geheugen** van het proces. Zodra *Burp Suite* wordt afgesloten, gaat alle informatie verloren.

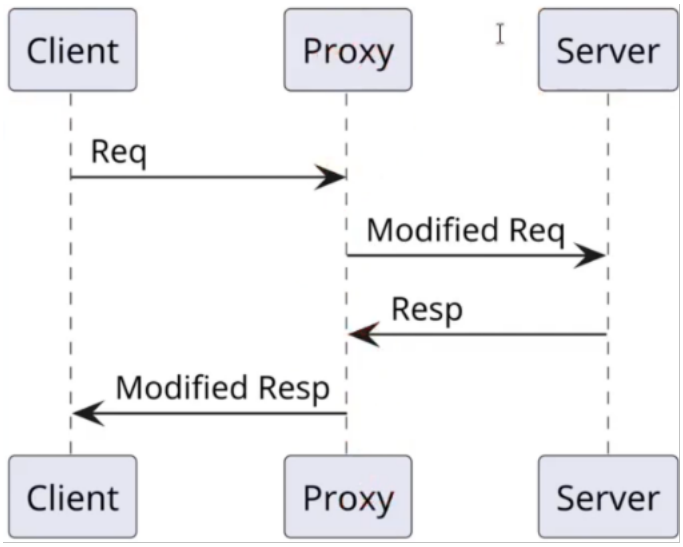
Tijdens een penetratietest maken we altijd aantekeningen. We slaan de *HTTP*-verzoeken en -antwoorden uit *Burp Suite* op in een teksteditor of een ander programma, zodat we deze kunnen archiveren. Op die manier behouden we overzicht over wat we hebben gedaan en raken we de verzamelde data niet kwijt.

Het tweede beginscherm is het *projectselectiescherm*. In de *Community Edition* kunnen we hier in de praktijk weinig mee doen; dit scherm dient voornamelijk als promotie voor de Pro Edition. We klikken daarom op ***Next***.

Het tweede scherm heeft betrekking op de configuratie van *Burp Suite* die we willen gebruiken. In dit geval kunnen we de standaardinstellingen van *Burp Suite* gebruiken.

7.3. Wat is Burp Suite? (*The Man-in-the-Middle*)

Burp Suite fungeert als een *Interception Proxy*. Normaal gesproken praat je browser direct met de server van de betreffende website. Met *Burp Suite* ertussen wordt elk verzoek (*request*) eerst opgevangen.



Dit diagram is de visuele weergave van hoe *Burp Suite* (de *Proxy*) werkt tijdens een *Web Application Penetration Test*. Het laat zien dat een hacker niet simpelweg een bezoeker is, maar een "***man-in-the-middle***" die het verkeer beheerst.

Het proces, stap voor stap:

Het Verzoek (*Req*)
Normaal gesproken stuurt jouw browser (***Client***) een verzoek rechtstreeks naar de (***Server***). Met *Burp Suite* ertussen stopt het verzoek echter eerst bij de *Proxy*.
De browser is zo ingesteld dat al het verkeer via Burp Suite loopt.

In Burp staat de functie "**Intercept**" aan. Het verzoek blijft hier "hangen" tot de hacker beslist wat er gebeurt.

De Aanpassing (*Modified Req*)
Dit is waar het "hacken" begint. In de Proxy-tab van Burp Suite kan de student de ruwe tekst van het verzoek aanpassen.

Pas nadat de hacker op de knop **"*Forward*"** drukt, wordt het aangepaste verzoek naar de server gestuurd. De server denkt dat dit het originele verzoek van de browser is.

Het Antwoord (*Resp*)
De server verwerkt het (aangepaste) verzoek en stuurt een antwoord terug. Ook dit antwoord gaat niet direct naar de browser, maar komt eerst weer in de Proxy terecht.

Hier kan de hacker zien of de aanval is geslaagd. Krijgt hij een foutmelding, of stuurt de server gevoelige data terug die eigenlijk niet zichtbaar had mogen zijn?

Het Antwoord Manipuleren (*Modified Resp*)
Niet alleen wat je *stuurt* kun je aanpassen, maar ook wat je *ontvangt*. De hacker kan dit in de Proxy wijzigingen aanbrengen voordat het de browser bereikt. Hiermee kun je bijvoorbeeld soms beveiligingen in de interface van de website omzeilen.

Hacking gaat vaak over het verbreken van het vertrouwen tussen de Client en de Server. De Proxy (Burp Suite) is de tool waarmee we dat vertrouwen testen door te kijken wat er gebeurt als we de data "onderweg" veranderen.

8. Web Application Penetration Test Kwetsbaarheden

Tijdens de testing activity identificeren we kwetsbaarheden in de volgende categorieën:

Categorie	*Aantal Issues*
1. Information Disclosure	7 issues
2. Injection Vulnerabilities	7 issues
3. Authentication	5 issues
4. Authorization	3 types
5. Validation Vulnerabilities	6 issues (niet in masterclass)
6. Session Management	4 issues (niet in masterclass)
7. Cryptography	4 issues (niet in masterclass)
8. Business Logic & Configuration	4 issues (niet in masterclass)

Conclusie: *Training on the Job*
Deze masterclass fungeert als een simulatie van een echte werksituatie. Hoewel we een kwetsbare trainingsapplicatie gebruiken, is de aanpak identiek aan een *real-world* scenario. De intuïtie en methodiek die u hier opbouwt, zijn direct vertaalbaar naar applicaties van echte cliënten.
**Categorie 5 tot en met 8 zijn buiten scope van deze masterclass*

8.1. Information Disclosure

Information Disclosure (of *Information Leakage*) is het onbedoeld onthullen van gevoelige informatie aan ongeautoriseerde gebruikers. We analyseren hierbij de informatie op drie niveaus: de *UI*, de *Source Code* en de *HTTP-traffic*.

Veelvoorkomende Kwetsbaarheden
• *Error Messages & Stack Traces:* Het lekken van technische details door applicatiefouten.
• *Directory Listing:* Het tonen van bestandslijsten door de webserver.
• *Metadata Leakage:* Verborgen informatie in PDF- of Office-bestanden.
• *Exposed Configuration Files*: Publiekelijk toegankelijke configuratiebestanden.

Een blootgestelde *Swagger API* (*OpenAPI*) fungeert als een blauwdruk voor aanvallers. Het onthult *endpoints*, methoden (*GET*, *POST*) en datastructuren, wat verdere exploitatie aanzienlijk vergemakkelijkt.

8.2. Injection Vulnerabilities

In de wereld van hacking zijn "*injections*" een van de meest voorkomende en gevaarlijke soorten kwetsbaarheden. Het basisprincipe is altijd hetzelfde: een aanvaller stuurt kwaadaardige data naar een applicatie, die deze data vervolgens onbedoeld uitvoert als een commando of query.

8.2.1. SQL Injection (SQLi)

SQL (*Structured Query Language*) is de taal die wordt gebruikt om te communiceren met een database. In de context van webapplicaties, zoals de bankapplicatie in de masterclass, fungeert de database als de "kluis" waarin alle klantgegevens en saldi worden bewaard.

Je inlogt op de bank-app. De webserver moet aan de database vragen of de gebruiker bestaat en of het wachtwoord klopt. Dit gebeurt met een *SQL-query* (een zoekopdracht).

Een standaard *SQL-opdracht* ziet er vaak zo uit:

```
SELECT * FROM users WHERE username = 'admin' AND password = 'secretpassword';
```

SELECT *: "Pak alle gegevens..."
FROM users: "...uit de tabel met gebruikers..."
WHERE: "...waar de volgende voorwaarden gelden..."

Als hacker willen wij de opdracht zodanig manipuleren zodat we opdrachten meegeven. De aanvaller "injecteert" *SQL-code* om ongeautoriseerde toegang te krijgen tot data of deze te manipuleren.

Dit gebeurt wanneer de applicatie de invoer van een gebruiker (bijvoorbeeld in een loginveld of zoekbalk) niet goed filtert en direct in de *SQL*-opdracht uitvoerd.

Een hacker kan dan speciale *SQL*-tekens invoeren om de logica van de opdracht te veranderen. In plaats van een wachtwoord, vult de hacker dit in: `' OR '1'='1`

De query wordt dan: `SELECT * FROM users WHERE username = 'admin' AND password = '' OR '1'='1';`

Omdat `'1'='1'` altijd waar is, negeert de database het wachtwoord en geeft de hacker toegang tot het account.

8.2.2. Command Injection (OS injection)

Hierbij misbruiken hackers een invoerveld om commando's rechtstreeks op het besturingssysteem (bijv. Linux) van de server uit te voeren.

Voorbeeld: Een website die je een IP-adres laat pingen. Een hacker vult in:

`8.8.8.8 ; cat /etc/passwd`.

De server voert de ping uit, maar door de `;` voert hij daarna ook het commando uit om het bestand met gebruikersnamen te tonen.

Impact: Volledige overname van de server, *RCE* (*Remote Code Execution*).

8.2.3. XSS (Cross-Site Scripting)

Hoewel vaak apart genoemd, is XSS technisch gezien een **HTML/JavaScript Injection**. De aanval is niet gericht op de server, maar op de browser van andere gebruikers.

Voorbeeld: Een hacker plaatst een script

`<script>alert('Hack')</script>` in een opmerking of profielveld.

Elke andere gebruiker die die pagina bekijkt, voert dat script onbewust uit.

Impact: Stelen van sessie-cookies, overnemen van accounts van mede-gebruikers of het tonen van valse inlogformulieren.

8.2.4. LDAP Injection

Veel grote organisaties gebruiken LDAP (Lightweight Directory Access Protocol) voor het beheren van gebruikersrechten en adresboeken. Een query ziet er vaak uit als een reeks haakjes:

`(&(USER=admin)(PASSWORD=secret))`

Vergelijkbaar met SQLi, maar gericht op de mappenstructuur van de organisatie.

Voorbeeld: Een hacker vult de volgende gebruikersnaam in het inlogscherm:

`Admin)(&)`

De server plakt dit in de query, waardoor de rest van de controle (zoals het wachtwoord) buiten beschouwing wordt gelaten:

`(&(USER=admin)(&)(PASSWORD=input_pw))`

De `(&)` is een "altijd waar" conditie in LDAP. De database ziet dat de gebruiker 'admin' bestaat en dat de tweede conditie `(&)` ook waar is. Hij stopt met kijken naar het wachtwoordveld.
Impact**:** Het omzeilen van inlogschermen of het opvragen van de volledige organisatiestructuur en rechten van werknemers. De hacker logt in als admin zonder het wachtwoord te weten.

8.2.4. NoSQL Injection

Met de opkomst van moderne databases zoals MongoDB is er een nieuwe variant ontstaan die geen gebruik maakt van SQL, maar van JSON-achtige structuren. In plaats van `' OR '1'='1`, gebruiken we hier operatoren zoals `$gt` (Greater Than) of `$ne` (Not Equal).

Een inlogveld dat data verwacht in JSON-formaat: `{"username": "admin", "password": "password123"}`.

De hacker stuurt via Burp Suite de volgende data naar de server: `{"username": "admin", "password": {"$ne": ""}}`

Wat er gebeurt is:

`$ne`: "" betekent: Wachtwoord is **Niet Gelijk** aan een lege tekst.

Aangezien elk echt wachtwoord *niet gelijk* is aan een lege tekst, geeft de database de waarde "True" terug voor de hele query.

Impact**:** is dat de hacker is ingelogd op het admin-account omdat de voorwaarde technisch gezien klopt voor elk account met een wachtwoord.

Hetzelfde resultaat als SQLi (dataverlies), maar de gebruikte techniek in Burp Suite ziet er anders uit (bijv. gebruik van `$gt` filters).

8.2.5. Hoe test je deze Injection Vulnerabilities in Burp Suite

Tijdens de masterclass gebruiken we de *Burp Suite Repeater* om deze verschillende tekens één voor één naar de server te sturen.

- Krijg je een foutmelding? Dan "praat" je met de database.
- Word je opeens ingelogd? Dan heb je de logica succesvol omzeild.

In de *Burp Suite Repeater* kun je dus precies zien hoe een verzoek verandert en hoe de server daarop reageert. Dit is de plek waar je als hacker experimenteert: je past de "voor" (het origineel) aan naar de "na" (de injectie) en drukt op **Send**.

SQL Injection (Login Omzeilen)

Hier proberen we binnen te komen zonder wachtwoord door de logica aan te passen.

- **Voor (Origineel):** De browser stuurt je wachtwoord netjes door. `POST /login HTTP/1.1 username=admin&password=Wachtwoord123`
- **Na (Injectie):** We voegen `' OR '1'='1` toe. De server ziet nu: "Gebruiker is admin EN (wachtwoord is leeg OF 1 is gelijk aan 1)". Omdat 1 altijd 1 is, laat de server je binnen. `POST /login HTTP/1.1 username=admin' OR '1'='1'--&password=niets`

LDAP Injection (Rechten omzeilen)

Hier manipuleren we de mappenstructuur om de wachtwoordcheck over te slaan.

- **Voor (Origineel):** `GET /userinfo?user=jansen HTTP/1.1`
- **Na (Injectie):** Door `admin)(&)` te sturen, sluit je de zoekopdracht voortijdig af met een "waar"-statement `(&)`. `GET /userinfo?user=admin)(&) HTTP/1.1` De server geeft nu de gegevens van de admin terug omdat de rest van de interne filteropdracht genegeerd wordt.

Hoe lees je de reactie (Response)?

In de Repeater zie je aan de rechterkant direct het resultaat:

- **Succes:** Je ziet een `HTTP 302 Redirect` (je wordt doorgestuurd naar het dashboard) of een `HTTP 200 OK` met gevoelige data in de tekst.
- **Mislukt:** Je ziet een `HTTP 401 Unauthorized` of een foutmelding waarin staat dat je syntax niet klopt.

8.2.6. Checklist is mijn injectie geslaagd?

Kijk na het drukken op *Send* in de *Repeater* naar de rechterkolom (de *Response*). Let op deze drie signalen:

1. De *HTTP* Status Code

- *302 Found / Redirect:* Dit is vaak het teken van een geslaagde login-omzeiling. De server zegt: "Je gegevens kloppen, ik stuur je nu door naar bijvoorbeeld de `/dashboard` pagina."
- *200 OK:* Kan twee dingen betekenen. Ofwel de aanval is mislukt en je ziet gewoon weer het inlogscherm, óf de aanval is geslaagd en de gevoelige data staat direct in de tekst onderaan.
- *500 Internal Server Error:* De database is gecrasht door jouw vreemde tekens (zoals `'` of `(`). Dit betekent dat het veld kwetsbaar is, maar dat je je aanvalscode (*payload*) nog moet verfijnen.

2. De *Response Body* (De inhoud)

Scroll naar beneden in de response-tekst of gebruik de zoekbalk onderaan:

- Zoek naar trefwoorden: Zoek op `admin`, `welcome`, `balance`, of `password`. Als je deze woorden ziet terwijl je nog niet officieel was ingelogd, is je injectie geslaagd.
- Lengte van de response: Let op de *Render*-tab in de response. Als een normale foutmelding 500 karakters lang is, maar jouw injectie een response geeft van 5000 karakters, dan heb je waarschijnlijk een hele tabel uit de database getrokken.

3. Foutmeldingen (De *Leaks*)

Soms geeft de server je letterlijk de routekaart naar de volgende stap:

- *"MySQL Error: check the manual that corresponds to your MySQL server version..."* -> Je weet nu dat het een *MySQL* database is en kunt je *SQL*-commando's daarop aanpassen.
- *"LDap error: size limit exceeded"* -> Je hebt zoveel data opgevraagd dat de server het niet meer aankan.

8.3. Authentication

Authentication Vulnerabilities zijn zwakheden in het systeem dat moet vaststellen of een gebruiker echt is wie hij beweert te zijn. In het diagram van de *Proxy* dat we eerder bespraken, is dit het moment waarop de *Server* beslist of de *Client* naar binnen mag.

Authenticatie gaat niet alleen over het wachtwoord, maar over het hele proces van inloggen tot uitloggen. Eén zwakke schakel in dit proces maakt de rest van de beveiliging overbodig.

De meest voorkomende kwetsbaarheden in de praktijk zijn:

8.3.1. Gebrekkige Wachtwoordbeveiliging (*Brute Force*)

Dit is de meest basale aanval. De server staat toe dat een hacker duizenden wachtwoordcombinaties probeert zonder de gebruiker te blokkeren.

In *Burp Suite* gebruiken we de *Intruder*-tool om een lijst met de 10.000 meest gebruikte wachtwoorden automatisch af te vuren op een gebruikersnaam. Als er geen "*rate limiting*" (snelheidsbeperking) is, is het slechts een kwestie van tijd voordat het account gekraakt is.

8.3.2. Broken Session Management

Nadat je bent ingelogd, geeft de server je een *Session Cookie*. Dit is je digitale toegangsbewijs.

Als deze cookies voorspelbaar zijn (bijvoorbeeld gebaseerd op je gebruikersnaam) of niet veilig worden verzonden, kan een hacker jouw sessie "stelen" (*Session Hijacking*).

Een hacker kan zijn eigen cookie in de Proxy-tab aanpassen om te proberen de sessie van een andere gebruiker over te nemen.

8.3.3. Onveilige "*Forgot Password*" Functies

Soms is de voordeur (het inlogscherm) zwaar beveiligd, maar staat de zijdeur (wachtwoord herstellen) wagenwijd open.

De server stelt een beveiligingsvraag die makkelijk te raden is, of verstuurt een herstellink die niet verloopt of voorspelbaar is.

8.3.4. Username Enumeration

De applicatie geeft te veel informatie prijs bij een foutieve inlogpoging.
Slechte melding, "*Wachtwoord is onjuist voor gebruiker admin.*" (De hacker weet nu dat *'admin'* een geldige gebruiker is).

Goede melding, "*Gebruikersnaam of wachtwoord is onjuist.*"

Hacker-techniek, door in de *Intruder* te kijken naar het verschil in responsetijd of de lengte van de response, kan een hacker achterhalen welke gebruikersnamen bestaan.

8.4. Authorisation

Waar *Authentication* gaat over de vraag "*Wie ben je?*", gaat *Authorisation* over de vraag: "*Wat mag je doen?*".

Authorisation vulnerabilities ontstaan wanneer een applicatie niet goed controleert of een ingelogde gebruiker wel de rechten heeft om een bepaalde actie uit te voeren of data in te zien. In de cybersecurity-wereld noemen we dit *Broken Access Control* (*BAC*).

Hier zijn de belangrijkste vormen:

8.4.1. IDOR (*Insecure Direct Object Reference*)

Dit is de meest voorkomende en begrijpelijke vorm. Het gebeurt wanneer een applicatie een ID in de URL of in een parameter gebruikt om data op te halen, zonder te checken of die data van jou is.

Voorbeeld:
Je logt in bij de bank en ziet je profiel op: `securebank.nl/profiel?id=1005`.

Je verandert het getal in de URL naar `1006`.

Als de server je nu het profiel van een andere klant laat zien, heb je een *IDOR*-lek gevonden. De server "vertrouwt" dat je alleen vraagt om wat van jou is.

8.4.2. Horizontal Privilege Escalation

Dit is de technische term voor wat we hierboven bij IDOR zagen. Je krijgt toegang tot data van iemand met **hetzelfde** rechtenniveau (bijvoorbeeld de gegevens van een mede-klant).

8.4.3. Vertical Privilege Escalation (*PrivEsc*)

Hierbij probeert een gebruiker met weinig rechten de functies van een gebruiker met ***meer*** rechten (zoals een *Admin*) te gebruiken.

Voorbeeld:

Je bent een gewone klant en je ziet dat de URL voor jouw instellingen `/user/settings` is.

Je raadt of vindt (via verkenning) de URL `/admin/panel` of `/admin/delete_user`.

Als de server deze pagina opent zonder te controleren of jij wel een *admin*-badge hebt, kun je functies uitvoeren die het hele systeem kunnen platleggen.

8.4.4. Bypassing Access Control via Parameter Manipulation

Soms zit de toegangscontrole verborgen in de data die je naar de server stuurt (de "*Body*" van je *request* in Burp Suite).

Voorbeeld:

Bij het aanmaken van een account stuurt je browser: `username=test&role=user`.

Je vangt dit pakketje op in de Burp Proxy en verandert het naar: `username=test&role=admin`.

De server accepteert de rol die de *client* opgeeft, in plaats van dit zelf veilig op de server te bepalen.

Hoe zie je dit in *Burp Suite*?

In de *Repeater* stuur je een verzoek naar een pagina waar je eigenlijk niet bij mag.

Krijg je een `403 Forbidden`? Dan is de autorisatie goed geregeld.

Krijg je een `200 OK` met de data van de admin? Dan heb je een groot lek gevonden.

***Gouden Regel voor Ontwikkelaars:** Vertrouw nooit op informatie die de gebruiker (client) zelf opstuurt over zijn eigen rechten.*

9. PoC – Proof of Concept

Een *PoC* (*Proof of Concept*) het onomstotelijke bewijs dat een theoretische kwetsbaarheid ook daadwerkelijk misbruikt kan worden.

Zonder *PoC* is een kwetsbaarheid slechts een "vermoeden". Met een PoC laat je zien hoe en welke methode en of tools je hebt gebruikt om bij bijvoorbeeld de gevoelige data te komen

9.1. Het doel van een PoC

Een goede *PoC* is bedoeld om de ernst van een probleem aan te tonen zonder schade aan te richten. In de rapportage dient het drie doelen:

Validatie: Bewijzen dat het lek geen '*false positive*' is (een foutieve melding van een scanner).
Reproductie: De developer helpen om het probleem na te bootsen, zodat hij kan testen of zijn oplossing werkt.
Overtuiging: Een CISO of manager sneller laten inzien dat actie nodig is. Een screenshot van een database met klantnamen is veel krachtiger dan een abstracte tekst over "*SQL Injection*".

9.2. Hoe ziet een PoC eruit?

Afhankelijk van de kwetsbaarheid kan een *PoC* verschillende vormen aannemen:

- **Een Screenshot:** Bijvoorbeeld van een `/admin` pagina waar je zonder in te loggen op bent gekomen.
- **Een Payload:** De specifieke code die je invult in een invoerveld, bijvoorbeeld: `<script>alert('XSS')</script>`.
- **Een HTTP-request/response:** Een *screenshot* uit *Burp Suite* die laat zien dat de server gevoelige tokens teruggeeft in de tekst.
- **Een Script:** Een klein *Python-script* dat automatisch een handeling uitvoert om het lek aan te tonen.

9.3. Exploit vs. PoC

Het is belangrijk om het verschil te kennen tussen een *Exploit* en een *PoC*.
PoC: Een minimale handeling om te bewijzen dat de deur openstaat (bijvoorbeeld: het commando `whoami` uitvoeren op een server).
Exploit: Een volledige aanval die gericht is op het overnemen van het systeem of het stelen van alle data (bijvoorbeeld: een ransomware-aanval).
Gouden regel: Een PoC in een professionele pentest moet altijd **non-destructief** zijn. Je wilt aantonen dat je de data *kunt* inzien, niet de hele database verwijderen.

10. CVSS

In de cybersecurity-wereld is de ***CVSS*** de universele meetlat voor risico's. Het staat voor ***Common Vulnerability Scoring System***. Het is een numerieke score tussen de **0.0** en de **10.0**, waarbij 10.0 de hoogste graad van kritiekheid vertegenwoordigt.
In de rapportage gebruik je *CVSS* om je bevindingen objectief te rechtvaardigen naar de klant, in plaats van te zeggen: "Ik vind dit een belangrijk kwetsbaarheid."

10.1. Hoe wordt de score berekend?

De *CVSS-score* wordt bepaald door drie groepen parameters, ook wel '*Metric'* genoemd. Deze zijn '*Base Score*', '*Temporal Score*' en '*Environmental Score*'. In deze masterclass zoomen we vooral in op de *Base Score*, zodat we de gevonden kwetsbaarheden kunnen aanduiden met een cijfer.

10.1.1. Base Score (De basis)

Dit is het belangrijkste onderdeel en kijkt naar de intrinsieke eigenschappen van het lek:
Attack Vector (AV): Hoe komt de aanval binnen? (Bijv. via het internet of moet de hacker fysiek bij de computer zijn?)

Attack Complexity (AC): Hoe moeilijk is de aanval uit te voeren?

Privileges Required (PR): Moet de hacker al een account hebben?

User Interaction (UI): Geen actie van de gebruiker nodig. Moet een slachtoffer ergens op klikken?

Impact (C, I, A): Wat is het gevolg voor de ***C****onfidentiality,* ***I****ntegrity en* ***A****vailability*?

10.1.2. De Severity Ratings

In rapportages vertalen we de cijfers naar een ernst-niveau:

Score	Rating	Betekenis voor de organisatie
0.0	None	Geen beveiligingsrisico.
0.1 - 3.9	Low	Kleine impact, lastig te misbruiken.
4.0 - 6.9	Medium	Serieus risico, moet op de planning voor reparatie.
7.0 - 8.9	High	Grote kans op misbruik, snelle actie vereist.
9.0 - 10.0	**Critical**	Directe actie nodig; grote kans op data-exfiltratie of systeemuitval.

10.1.3. Officiële calculators

Er zijn twee officiële calculators die gebruikt worden voor een penetratie rapportage: die van *FIRST.org* (*Forum of Incident Response and Security Teams*)
https://www.first.org/cvss/calculator/3.1
Naast *FIRST* wordt de calculator van de *National Vulnerability Database (NIST)* ook zeer veel gebruikt, vooral omdat zij de scores koppelen aan de officiële *CVE*-database van de Amerikaanse overheid https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

10.1.4. Vector String

Wanneer je de calculator op deze websites gebruikt, zie je onderaan een tekstregel verschijnen, bijvoorbeeld:

```
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
```

Dit noemen we de *Vector String*. Het is een verkorte notatie die alle gemaakte keuzes in de calculator samenvat. In professionele pentest-rapporten wordt deze string altijd vermeld bij een bevinding, zodat een andere security professional precies kan zien en valideren hoe jij op die specifieke score bent uitgekomen. Het is dus niet afdoende om alleen het cijfer te noemen, zodat het zorgt voor transparantie en reproduceerbaarheid.

10.1.5. Valkuilen bij CVSS-beoordelingen

- "**Scope Creep**" (De S-parameter)

  Men zet de *Scope* (S) vaak op 'Changed' (C) omdat ze denken dat een aanval grote gevolgen heeft.

  De werkelijkheid: De *Scope* verandert alleen als een kwetsbaarheid in de ene component (bijv. de webapplicatie) een directe impact heeft op een **andere** beveiligingsautoriteit (bijv. het onderliggende besturingssysteem of een andere cloud-tenant).

  ***Tip**:* Bij 90% van de web-vulnerabilities (zoals SQLi of XSS) blijft de Scope 'Unchanged' (U).

- Overschatting van de Impact (C, I, A)

  De fout: De impact op *Integrity* (I) of *Availability* (A) op '*High*' zetten bij een *Information Disclosure* lek.

  De werkelijkheid: Als je alleen data kunt lezen (zoals bij het Swagger-voorbeeld), is de impact op de *Confidentiality* (C) weliswaar 'High', maar de *Integrity* (I) en *Availability* (A) blijven 'None' (N). Je hebt immers niets aangepast of verwijderd.

  ***Tip**:* Wees eerlijk. Een *'High'* op alle drie de fronten betekent meestal dat je volledige controle over de server hebt (*Remote Code Execution*).

- De "Privileges Required" (*PR*) verwarring

De fout: De *Privileges Required* op 'None' (N) zetten terwijl de aanvaller eerst een gratis account moet aanmaken op de website.

De werkelijkheid: * None (N): De aanvaller kan de aanval uitvoeren zonder in te loggen (bijv. op de loginpagina zelf).

Low (L): De aanvaller heeft een standaard gebruikersaccount nodig. Zelfs als iedereen dat account simpel kan aanmaken, is het in CVSS-termen nog steeds *'Low'*.

***Tip**:* Denk aan de barrière. Moet ik een formulier invullen en op een activatiemail klikken? Dan is het (L).

11. CWE

CWE (*Common Weakness Enumeration*) is de universele "woordenlijst" voor programmeerfouten en kwetsbaarheden in software.

Waar de *CVE* (*Common Vulnerabilities and Exposures*) een lijst is van specifieke lekken in bestaande software (bijvoorbeeld: "Lek X in Windows 11"), is de *CWE* een lijst van de *soorten* fouten die tot die lekken leiden.

11.1. Wat is het doel van CWE?

De CWE-lijst wordt beheerd door de *MITRE Corporation* en dient als een gemeenschappelijke taal voor ontwikkelaars, hackers en security managers. Het helpt bij:

Identificatie: Precies benoemen wat er mis is (bijv. "Dit is een *CWE-89*").

Oplossing: Elke *CWE*-pagina bevat uitgebreide uitleg over hoe je de fout kunt herstellen.

Kennis: Ontwikkelaars leren welke patronen in hun code gevaarlijk zijn.

Elk jaar publiceert *MITRE* een lijst met de *25 gevaarlijkste softwarefouten*. Dit zijn de fouten die het vaakst voorkomen, het makkelijkst te misbruiken zijn en de grootste schade aanrichten.

De centrale plek voor alles wat met CWE te maken heeft, is de officiële website van *MITRE* https://cwe.mitre.org/. Het is een enorme database die toegankelijk is voor iedereen.

Als je een lek vindt in Burp Suite, zoek het **CWE-nummer** op. Door dit nummer in je rapport te zetten, laat je aan de klant zien dat je een professionele methodiek gebruikt en niet zomaar wat doet.

12. CVE

Waar de *CWE* de "soorten" beschrijft, is de *CVE* (*Common Vulnerabilities and Exposures*) de registratie van een ***specifiek geval*** bij een specifieke soort (bijvoorbeeld: Windows 11 heeft op 4 februari een virus opgelopen via deze specifieke app).

CVE is de wereldwijde standaard om een specifiek beveiligingslek in software of hardware een unieke naam te geven.

12.1. De Anatomie van een CVE-nummer

Elk lek krijgt een uniek identificatienummer dat er als volgt uitziet:

CVE-2021-44228 (een beruchte Log4Shell lek).

CVE: De afkorting van de standaard.

2021: Het jaar waarin het lek is ontdekt of gerapporteerd.

44228: Een uniek volgnummer voor dat jaar.

12.2. Waarom is CVE zo belangrijk?

Zonder CVE zou er totale chaos ontstaan. Als een hacker een lek vindt in bijvoorbeeld *Wordpress*, welk lek bedoelt hij dan? Er zijn er duizenden.

Eén taal: Security-scanners, IT-beheerders en *CISO's* weten precies over welk lek ze praten als ze het *CVE*-nummer gebruiken.

Prioriteit: Aan een *CVE* wordt bijna altijd een *CVSS-score* (0 tot 10) gekoppeld. Zo weet een klant direct: *CVE-2024-1234* heeft een score van 9.8, we moeten dit NU patchen!"

Database: Het is een openbaar archief waar je kunt opzoeken of de software die jouw bank gebruikt bekende gaten bevat.

12.3. Waar vind je CVE's?

Er zijn verschillende databases (vaak "feeds" genoemd) waar je *CVE's* kunt opzoeken:

1. **CVE.org:** De officiële lijst beheerd door MITRE.
2. **NVD (National Vulnerability Database):** De database van de Amerikaanse overheid (NIST). Deze is heel populair omdat zij de CVE's verrijken met de CVSS-score en links naar oplossingen.
3. **Exploit-DB:** Hier vind je niet alleen de beschrijving van het lek, maar vaak ook de "exploit" (de code om het lek daadwerkelijk te misbruiken).

13. Korte samenvatting verschil tussen CWE, CVE en CVSS

13.1. CWE (Common Weakness Enumeration) - *De Oorzaak*

Dit is een lijst van **typen softwarefouten**. Het is abstract en technisch. Het vertelt een programmeur wat hij fout heeft gedaan in de code.

Voor Ontwikkelaars en architecten.

Vraag: *"Welke fout is er gemaakt tijdens het programmeren?"*

13.2. CVE (Common Vulnerabilities and Exposures) - *Het Voorval*

Dit is een lijst van **specifieke kwetsbaarheden** in bestaande softwareproducten. Elk nummer hoort bij één specifiek probleem in één specifiek programma (zoals Windows, Chrome of WordPress).

Voor Systeembeheerders en hackers.

Vraag: *"Zit er een bekend kwetsbaarheid in de software die ik nu gebruik?"*

13.3. CVSS (Common Vulnerability Scoring System) - *De Impact*

Dit is een rekenmethode om een **score van 0 tot 10** te geven aan een lek. Hoe hoger de score, hoe gevaarlijker het lek. Er wordt gekeken naar:

- Hoe makkelijk is het te hacken? (Heb je internet nodig of moet je fysiek bij de server staan?)
- Wat is de schade? (Kan de hacker alles verwijderen of alleen een naam lezen?)

13.4. Hoe ze samenwerken in een rapport

Als je op Dag 2 een rapport schrijft voor de bank, ziet een regel er vaak zo uit:

“We vonden een *SQL Injection (CWE-89)* in de inlogpagina. Dit is een bekend lek in dit type webserver, geregistreerd als *CVE-2023-1234*. Vanwege de hoge impact heeft dit een *CVSS-score van 9.1 (Critical).”*

Kali Linux Commando Overzicht voor Ethical Hacking

Als *Ethical Hacker* is de command-line interface (CLI) je primaire gereedschap. Een diepgaande beheersing van deze commando's is de basis voor elke succesvolle penetratietest.

I. De Basis: Navigatie en Bestandssysteem
Deze commando's helpen je te bewegen binnen het Linux-bestandssysteem en bestanden te beheren.

Commando	*Beschrijving*	*Voorbeeld van Gebruik*
ls	Lijst de inhoud van een directory op (bestanden en mappen).	ls -la (Toon alles, inclusief verborgen bestanden (.) en gedetailleerde informatie).
cd	Wijzig de huidige directory (Change Directory).	cd /var/log (Ga naar de logboekenmap).
pwd	Print Working Directory. Toont het volledige pad naar de map waarin u zich bevindt.	pwd
mkdir	Maak een nieuwe directory (map).	mkdir mijn_project
cat	Toon de inhoud van een tekstbestand op de standaarduitvoer.	cat /etc/passwd (Toont de gebruikersdatabase).
nano / **vim**	Teksteditors om bestanden te bekijken of te bewerken.	nano exploit.py
cp	Kopieer bestanden of mappen.	cp /pad/naar/bron /pad/naar/doel
mv	Verplaats of hernoem bestanden of mappen.	mv oud_bestand nieuw_bestand
rm	Verwijder bestanden of mappen. **Wees voorzichtig!**	rm -rf mapnaam (Verwijder de map en de inhoud recursief (-r) zonder bevestiging (-f)).

II. Medior Commando's: Systeem & Netwerkinformatie
Deze commando's zijn cruciaal voor het verzamelen van *OSINT* (Open Source Intelligence) en het begrijpen van de netwerkconfiguratie van uw eigen Kali-machine en doelwitten.

A. Systeeminformatie

Commando	*Beschrijving*	*Waarom in Ethical Hacking?*
whoami	Toont de huidige gebruikersnaam.	Essentieel voor **Post-Exploitation** om te weten welke permissies u heeft (bijv. bent uroot?).
id	Toont de gebruikers- en groep-ID's.	Bepaalt welke privileges u kunt gebruiken of escaleren.

uname -a	Geeft gedetailleerde systeeminformatie over het OS en de kernelversie.	Zoekt naar **Kernel Exploitatie**-mogelijkheden.
service	Beheert systeemservices (starten, stoppen, herstarten).	service apache2 start (Start de webserver voor hosting van payloads).
systemctl	Moderne tool voor servicebeheer (opvolger van service).	systemctl status ssh (Controleert of SSH actief is).

B. Netwerkinformatie

Dit is de ruggengraat van elke pen-test, van **Reconnaissance** tot **Exploitation**.

- **ip a / ifconfig**
  - **Beschrijving:** Toont netwerkinterface-adressen (IP, MAC). ip a is de modernere variant.
  - **Gebruik:** U moet uw eigen IP-adres kennen om verbindingen terug te laten komen (reverse shells).
- **ping**
  - **Beschrijving:** Test de connectiviteit tussen twee hosts door ICMP-pakketten te sturen.
  - **Gebruik:** Bepaalt of een doelwit host **live** is en reageert.
- **netstat -tuln**
  - **Beschrijving:** Toont actieve netwerkverbindingen, luisterende sockets (TCP/UDP), numeriek (-n) en zonder resolveren (-t).
  - **Gebruik:** Identificeert welke services op uw of een doelwitmachine draaien en **welke poorten openstaan**.
- **ss**
  - **Beschrijving:** Sneller en moderner dan netstat voor het inspecteren van sockets.
  - **Gebruik:** Net als netstat, cruciaal voor het auditen van netwerkactiviteit.
- **route -n**
  - **Beschrijving:** Toont de IP-routeringstabel.
  - **Gebruik:** Begrijpen hoe pakketten door het netwerk van het doelwit worden gerouteerd (*Internal Network Reconnaissance*).

Frijlande
INSTITUUT VOOR
PRIVACY EN SECURITY MANAGEMENT

Bevindingenrapportage: Proof of Concept (PoC)

Opdracht: Web Application Penetration Testing (WAPT) – SecureBank

Student Naam: _____

Datum: _____

Algemene Informatie

- **Naam van de kwetsbaarheid:** (bijv. SQL Injection op Login)
- **Locatie (URL/Endpoint):** (bijv. `http://localhost:8080/login`)
- **Kwetsbare Parameter:** (bijv. `username`, `id`, `cookie`)
- **Classificatie (CWE):** CWE-_____
- **Risico Score (CVSS):** [ ] Low | [ ] Medium | [ ] High | [ ] Critical

---

Samenvatting van het lek

Beschrijf in 2-3 zinnen wat het probleem is en wat een hacker hiermee kan bereiken.

Reproductie Stappen (De PoC)

Noteer hier de exacte stappen die je hebt genomen in Burp Suite om het lek te bewijzen.

1. **Navigeer naar:** _____
2. **Onderschep het verkeer met Burp Suite Proxy.**
3. **Stuur het verzoek naar de REPEATER.**
4. **Pas de volgende Payload toe:** _____
5. **Observeer de Response:** (Wat zie je veranderen? bijv. statuscode 302 of data in beeld).

---

4. Bewijslast (Screenshot Logboek)

Plak hieronder (of beschrijf) de screenshots die je hebt gemaakt.

- **Screenshot 1 (Input):** Het verzoek in Burp Suite met de payload.
- **Screenshot 2 (Output):** Het bewijs dat je binnen bent of data hebt gestolen.

Impact voor de Bank (Privacy & CISO)

Vink aan wat van toepassing is bij misbruik:

- [ ] **Confidentiality:** Onbevoegden kunnen gevoelige klantdata inzien.
- [ ] **Integrity:** Gegevens (zoals saldi) kunnen worden aangepast.
- [ ] **Availability:** De applicatie kan onbruikbaar worden gemaakt.

---

6. Aanbeveling voor Ontwikkelaars

Hoe moet de bank dit oplossen? (Kijk in het handboek bij de betreffende CWE).