
SECURITE
INFORMATIQUE

Attention

“ Les méthodes et procédés d’attaque expliqués dans ce cours ont pour objectif de vous faire comprendre les enjeux de la sécurité et l’importance de la protection du système d’information. Pour rappel, l’utilisation de ces attaques sur des systèmes réels est strictement interdit et passible de peines d’emprisonnement ainsi que d’amendes lourdes. Plus d’informations ici : atteintes aux systèmes de traitement automatisé de données.

Attention

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Legifrance.gouv.fr
LE SERVICE PUBLIC DE LA DIFFUSION DU DROIT

Article 323-1

“ Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 30 000 euros d'amende.

Article 323-2

“ Le fait d'entraver ou de fausser le fonctionnement d'un système de traitement automatisé de données est puni de cinq ans d'emprisonnement et de 75 000 euros d'amende.

La Sécurité web

Les technologies liées à la sécurité web

Analyse de code statique (SAST)

L'analyse de code statique, en anglais SAST (Static Application Security Testing), est un élément essentiel dans un cycle de développement sécurisé ainsi que les méthodes d'intégration continue telles que le DevOps. Son objectif est de lire et d'analyser le code afin d'y trouver d'éventuels bugs et vulnérabilités avant la mise en production d'une application.

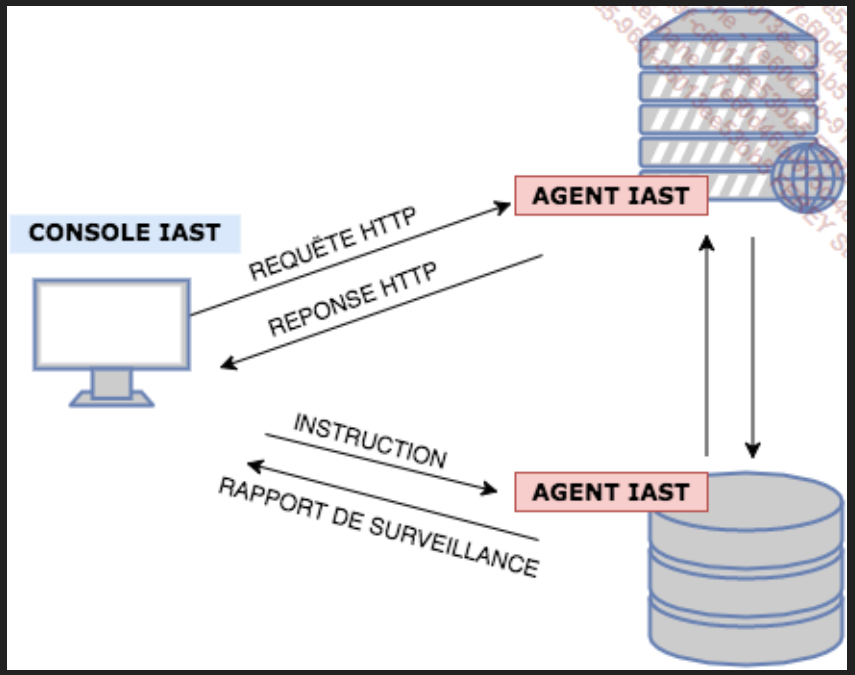
Analyse de code dynamique (DAST)

L'analyse des applications dynamiques, en anglais, *Dynamic Application Security Testing* (DAST), est, contrairement à l'analyse statique, utilisée avec une vision extérieure (*black box*) à l'application. Tel un pentest, les outils DAST essayent de pénétrer et identifier des vulnérabilités dans l'application par différentes techniques automatisées.

Nikto OWASP WebScarab WATOBO

Tests interactifs de la sécurité des applications (IAST)

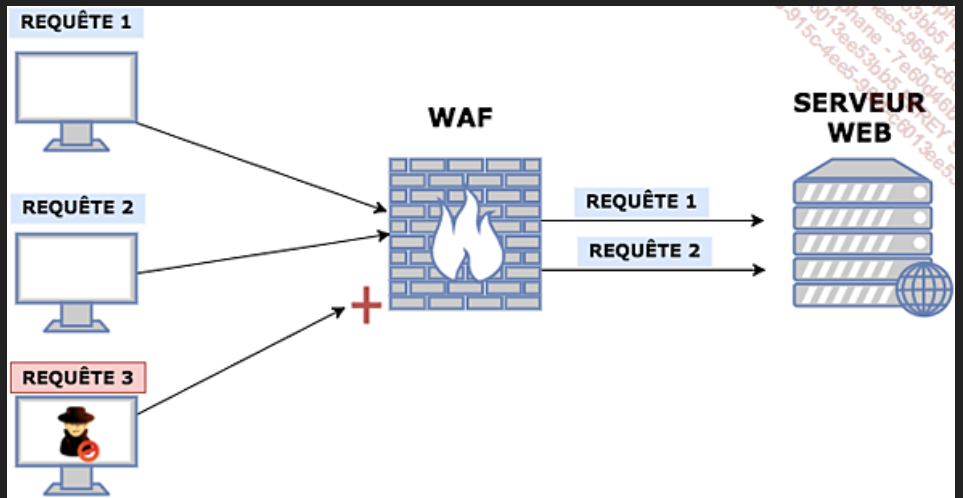
IAST exploite les informations à l'intérieur de l'application en cours d'exécution comme la transition des flux de données, les librairies et les connexions afin de repérer des comportements étranges lors de l'exécution de l'application.



Pare-feu applicatif (WAF)

Un *Web Application Firewall* (WAF) est comme son nom l'indique un pare-feu applicatif qui a pour objectif de contrôler, filtrer et bloquer le trafic HTTP en direction d'une application web.

Un WAF peut être sous forme physique, avec l'ajout d'une machine dédiée à cette tâche (*appliance*), un plugin à installer sur le serveur web ou bien une solution cloud.



modsecurity
Open Source Web Application Firewall

Outil de suivi de bugs (issue tracking system)

Un logiciel de suivi de bugs/problèmes permet à une équipe de développeurs de communiquer, soumettre et suivre les bugs afin d'améliorer la qualité des applications.

GitHub

JIRA

REDMINE
flexible project management

Kali Linux

Kali Linux est une distribution GNU/Linux sortie le 13 mars 2013, basée sur Debian.

La distribution a pris la succession de Backtrack, qui était basé sur Slackware jusqu'à la version 3 puis Ubuntu depuis la version 4.

L'objectif de Kali Linux est de fournir une distribution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un système d'information, notamment le test d'intrusion.

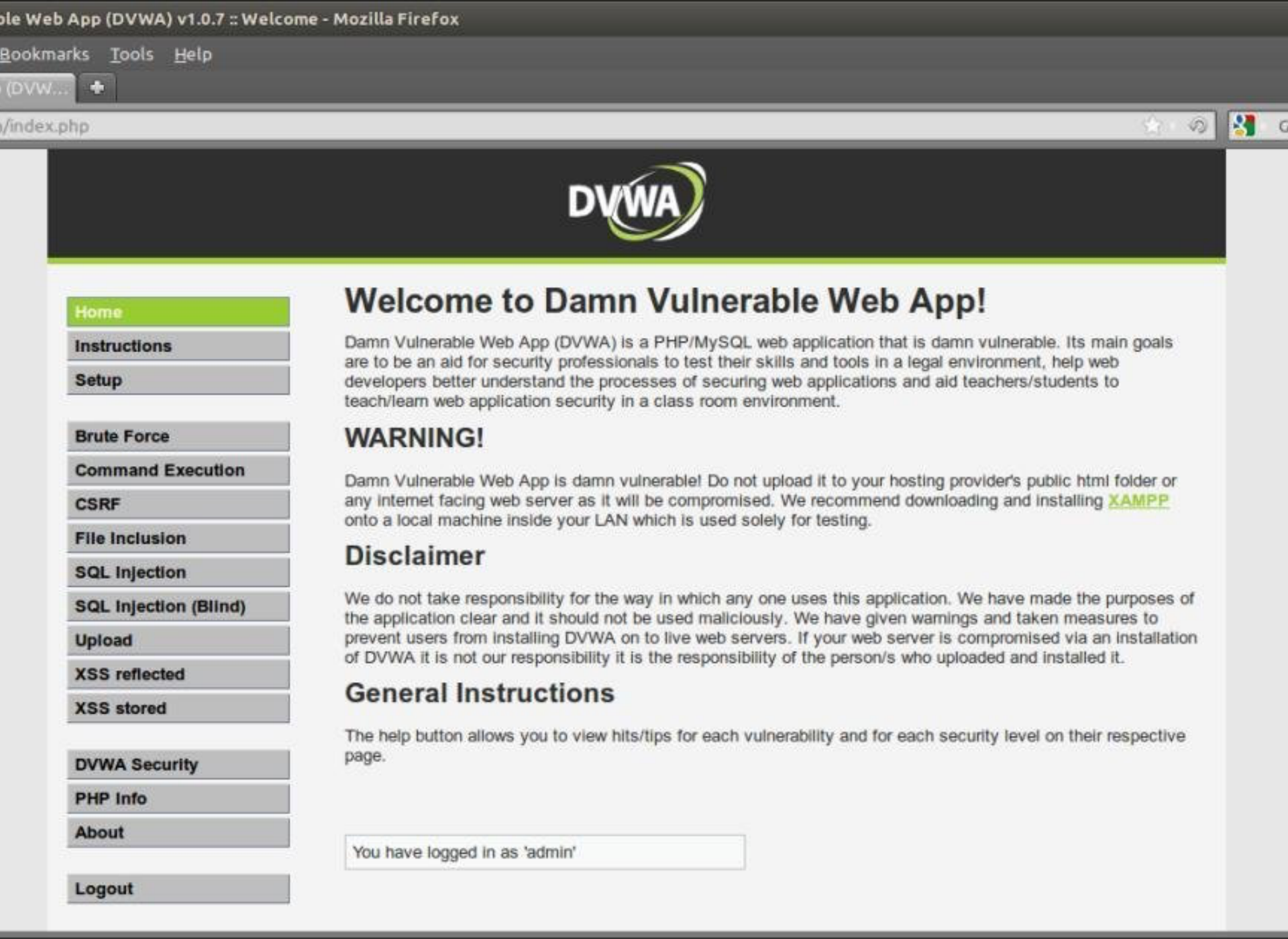
Pentest

Test d'intrusion

> *“ Un test d'intrusion (« penetration test » ou « pentest » en anglais) est une méthode d'évaluation de la sécurité d'un système ou d'un réseau informatique.*

Test d'intrusion

> *“ La méthode consiste généralement à simuler une attaque d'un utilisateur mal intentionné, voire d'un logiciel malveillant , ou un Malware. Le testeur analyse alors les risques potentiels dus à une mauvaise configuration d'un système, d'un défaut de programmation ou encore d'une vulnérabilité liée à la solution testée. Lors d'un test d'intrusion, le testeur adopte la position de l'attaquant potentiel.*
>
> *Le principal but de cette manœuvre est de trouver des vulnérabilités exploitables en vue de proposer un plan d'actions permettant d'améliorer la sécurité d'un système.*


ble Web App (DVWA) v1.0.7 :: Welcome - Mozilla Firefox
Bookmarks Tools Help
(DVW...
/index.php
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Welcome to Damn Vulnerable Web App!
Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.
WARNING!
Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.
Disclaimer
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.
General Instructions
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.
You have logged in as 'admin'

D.V.W.A se présente sous la forme d'un site WEB vulnérables à exploiter et à attaquer.

On y recense différentes failles (Brute force, CSRF, Injection SQL, XSS, LFI etc..) avec différents niveau de difficultés (Facile – Médium – Difficile).

Idéal pour débuter et s'entrainer.

bWAPP

an extremely buggy web application !

bWAPP (application web buggée), est une application Web libre et open source délibérément insécurisée.

Il aide les amateurs de sécurité, les développeurs et les étudiants à découvrir et à prévenir les vulnérabilités du Web. bWAPP prépare une entreprise à mener des tests de pénétration réussis et des projets de piratage éthique.

Qu'est-ce qui rend bWAPP si unique? Eh bien, il a plus de 100 vulnérabilités web!

Il couvre tous les principaux bugs connus, y compris tous les risques du OWASP Top 10 Project.

OWASP

Open Web Application
Security Project

OWASP
Open Web Application
Security Project

Open Web Application Security Project (OWASP) est une communauté en ligne travaillant sur la sécurité des applications Web.

Sa philosophie est d'être à la fois libre et ouverte à tous.

Elle a pour vocation de publier des recommandations de sécurisation Web et de proposer aux internautes, administrateurs et entreprises des méthodes et outils de référence permettant de contrôler le niveau de sécurisation de ses applications Web.

OWASP
Open Web Application Security Project

On June 12, 2013 the OWASP Top 10 for 2013 was officially released.

OWASP Top 10 2013 - French (PDF)


md5oogle™

MD5 et SHA1

Les mots de passe codés en MD5 sont sécurisés dans le sens ou il n'y a pas de clé à trouver pour les cracker...

Par contre, avec un bon bruteforce (tests de toutes les combinaisons) ou une rainbow table (énorme base de donnée de MD5 et leurs équivalent en clair), il est relativement facile de trouver à quel mot de passe correspond un MD5.

Voici un sites qui vous permettra de mettre un mot de passe en clair sur ce foutu MD5 qui traine dans votre base de données...

hashtoolkit.com


Rainbow table

Rainbow table

> *“ Une rainbow table (littéralement table arc-en-ciel) est, en cryptanalyse, une structure de données créée en 2003 par Philippe Oechslin de l'EPFL pour retrouver un mot de passe à partir de son empreinte.*

RainbowCrack

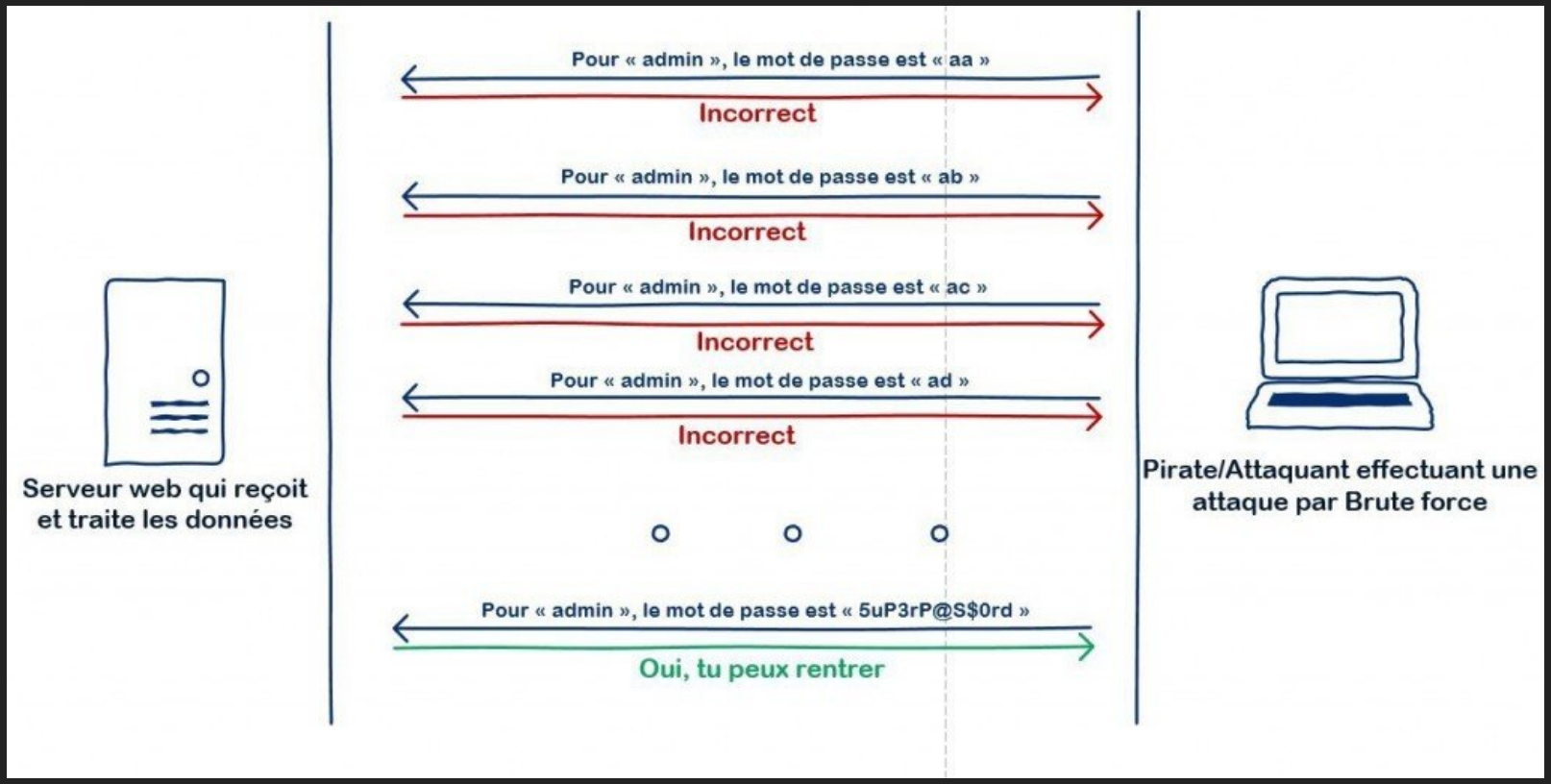

Brute Force

Faille du type Brute Force

> *" L'attaque par force brute est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Il s'agit de tester, une à une, toutes les combinaisons possibles.*

Cette méthode est en général considérée comme la plus simple concevable.

Sécurité : Fonctionnement et impact d'une attaque par brute force



Wfuzz

Wfuzz is a tool designed for bruteforcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts, etc), bruteforce GET and POST parameters for checking different kind of injections (SQL, XSS, LDAP,etc), bruteforce Forms parameters (User/Password), Fuzzing,etc.

```
********************************************************
* Wfuzz  2.0 - The Web Bruteforcer                     *
* Coded by:                                            *
* Christian Martorella (cmartorella@edge-security.com) *
* Xavier Mendez aka Javi (xmendez@edge-security.com)   *
* Carlos del ojo (deepbit@gmail.com)                   *
********************************************************

Target: http://localhost:8888/FUZZ
Payload type: file,wordlist/general/common.txt

Total requests: 950
=====================================================================================================================
ID      Response   Lines      Word         Chars                 Server                                  Redirect   Request
=====================================================================================================================

00023:  C=301      9 L        30 W         327 Ch     Apache/2.0.63 (Un                 http://localhost:8888/Admin/   " - Admin"
00060:  C=301      9 L        30 W         327 Ch     Apache/2.0.63 (Un                 http://localhost:8888/admin/   " - admin"
00265:  C=301      9 L        30 W         326 Ch     Apache/2.0.63 (Un                  http://localhost:8888/demo/   " - demo"
00257:  C=301      9 L        30 W         324 Ch     Apache/2.0.63 (Un                    http://localhost:8888/db/   " - db"
00622:  C=301      9 L        30 W         332 Ch     Apache/2.0.63 (Un           http://localhost:8888/phpMyAdmin/   " - phpMyAdmin"
00942:  C=301      9 L        30 W         325 Ch     Apache/2.0.63 (Un                 http://localhost:8888/xml/   " - xml"
```

DVWA – Brute force – level "low"

```
wfuzz -H Cookie:PHPSESSID=<PHPSESSID> -c -z file,/root/password.txt
"http://localhost/securiteWEB/dvwa/vulnerabilities/brute/?username=admin&password=FUZZ&Login=Login#"
```

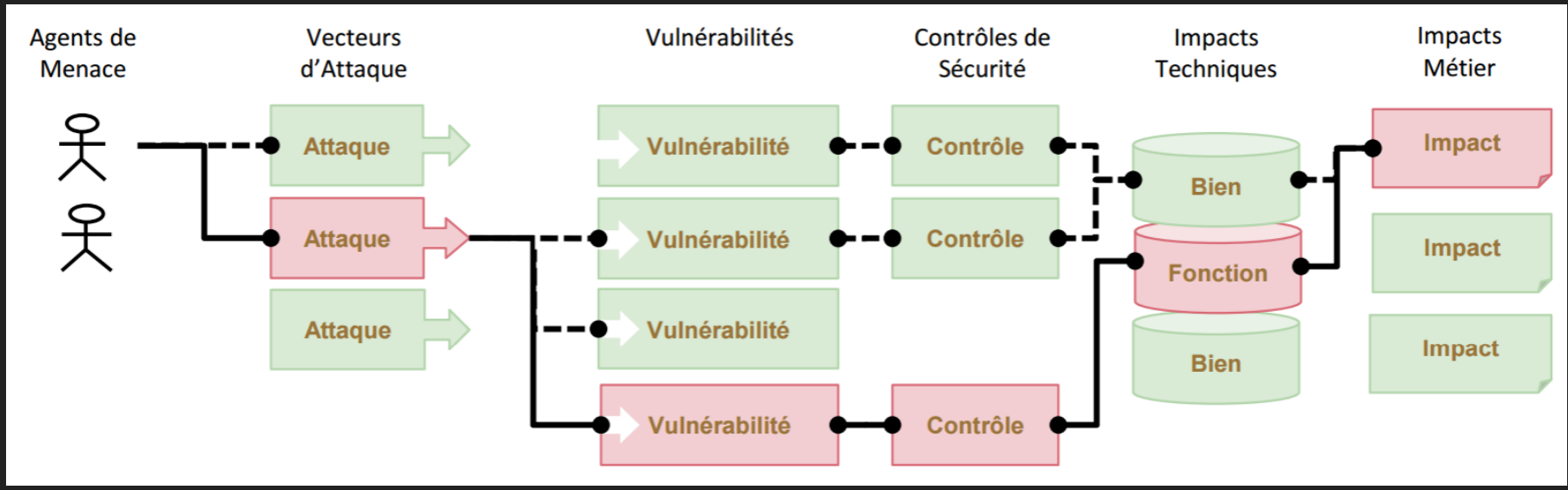
Pour détailler rapidement les options utilisées :

- -H : Permet de spécifier le contenu du Header HTTP des requêtes envoyées, ici on précise donc le contenu de notre Cookie qui nous permettra de passer la page login.php. Le contenu du cookie est bien entendu à adapter
- -c : Permet d'avoir une sortie en couleur, optionnel
- -z : Permet de préciser la source de données, ici, il s'agit donc d'un fichier que l'on précise ensuite : "password.txt"
- Enfin, on spécifie notre URL avec les paramètres GET. Le mot "FUZZ" est celui qui sera remplacé à chaque requête par le contenu de notre fichier source, il s'agit donc ici d'une attaque par dictionnaire

TOP 10 OWASP

Risques de sécurité applicatifs

Quels sont les Risques de Sécurité des Applications?



Risques de sécurité applicatifs

Quel est Mon Risque?

Agent de menace	Vecteurs d'attaque	Prévalence de la vulnérabilité	Détection de la vulnérabilité	Impact Technique	Impact Métier
Spécifique à l'Application	Facile	Trés répandue	Facile	Sévère	Spécifiques à l'Application ou au Métier
	Moyen	Commune	Moyen	Modéré	
	Difficile	Rare	Difficile	Mineur	

Méthodologie d'évaluation des risques OWASP

Article sur la modélisation Menace / Risque

OWASP Top Ten (2013 Edition)

OWASP
The Open Web Application Security Project

- A1: Injection
- A2: Broken Authentication and Session Management
- A3: Cross-Site Scripting (XSS)
- A4: Insecure Direct Object References
- A5: Security Misconfiguration
- A6: Sensitive Data Exposure
- A7: Missing Function Level Access Control
- A8: Cross Site Request Forgery (CSRF)
- A9: Using Known Vulnerable Components
- A10: Unvalidated Redirects and Forwards

Installation de la plateforme de Travail

L'OS

Kali

Téléchargez un logiciel de virtualisation (hyperviseur).

- VirtualBox avec VirtualBox Extension Pack

Téléchargez l'image VirtualBox du système d'exploitation Kali Linux

- Kali Linux VirtualBox Images

Après l'installation :

- Kali propose d'entrer un identifiant, entrez **root**.
- Pour le mot de passe, saisissez **toor**.

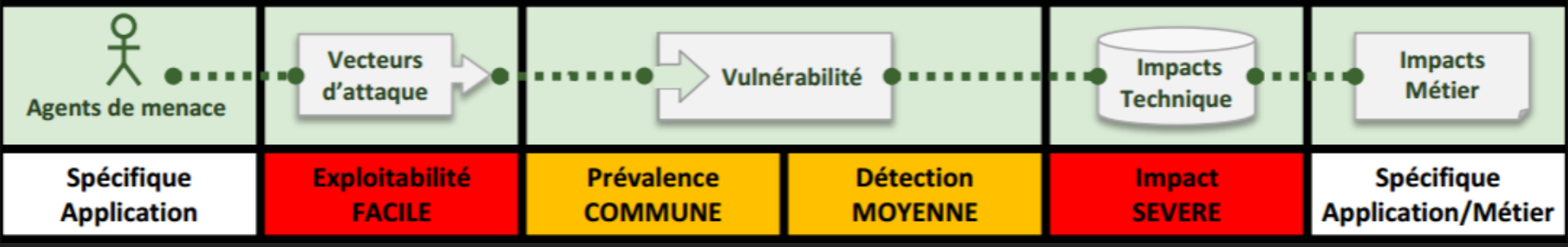
Les outils pour les pentests

Maintenant que la VM Kali Linux est sur notre système, nous devons installer l'application **DVWA** (Damn Vulnerable Web Application) et **bWAPP** afin de tester les vulnérabilités présentées.

Une fois le terminal ouvert, il est nécessaire de cloner un repository GIT travaillé :

- Dans le terminal (shell), pour passer le clavier en mode AZERTY, tapez la commande **setxkbmap fr**.
- Déplacez-vous dans le bon répertoire : tapez la commande **cd /var/www/html**.
- Maintenant il s'agit de cloner DVWA à partir de Github. Pour cela tapez **git clone https://github.com/EditionsENI/securiteWEB.git**
- Une fois le git cloné, lancez Apache et MySQL avec la commande **service apache2 start & service mysql start**.

Injection



Injection

> *" Les fuites de données (leak) via injection de données sont devenues monnaie courante. Verizon avec 1,5 million de données utilisateurs, Yahoo! avec 500 millions de comptes utilisateurs et mots de passe, Sony PSN avec 7 millions de cartes de crédit, Vtech, Visa, Adobe, la liste est longue.*

Injection

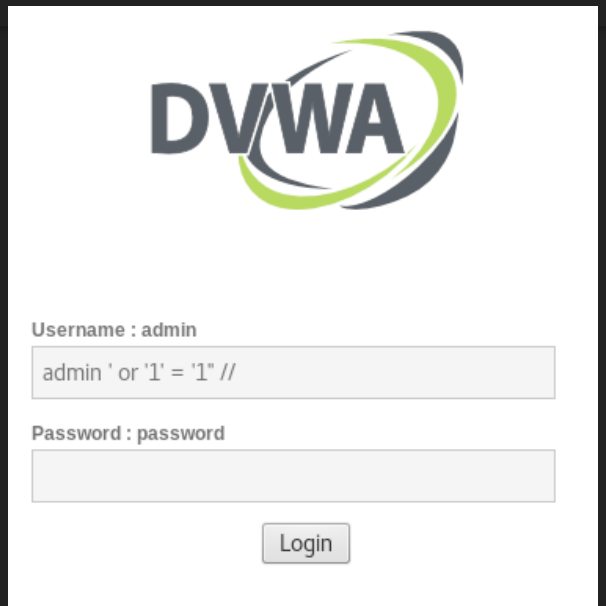
Une faille d'injection, telle l'injection SQL, OS et LDAP, se produit quand une donnée non fiable est envoyée à un interpréteur en tant qu'élément d'une commande ou d'une requête. Les données hostiles de l'attaquant peuvent duper l'interpréteur afin de l'amener à exécuter des commandes fortuites ou accéder à des données non autorisées.

- Injection SQL / NoSQL
- Injection commande OS
- Injection Xpath
- Injection LDAP
- Injection protocole (header HTTP, SMTP, ...)
- PHP (LFI, RFI, ...)
- ...

Faille du type Injection SQL

WIKIPÉDIA
L'encyclopédie libre

> *“ La faille SQLi, abréviation de "SQL Injection", soit "injection SQL" en français, est un groupe de méthodes d'exploitation de faille de sécurité d'une application interagissant avec une base de données. Elle permet d'injecter dans la requête SQL en cours un morceau de requête non prévu par le système et pouvant en compromettre la sécurité.*



Injection SQL

Requête	**SELECT** username, password **FROM** users **WHERE** username = 'data' **AND** password = 'data';
Légitime	**SELECT** username, password **FROM** users **WHERE** username = 'admin' **AND** password = 'password';
Injection	**SELECT** username, password **FROM** users **WHERE** username = 'admin ' or '1' = '1'' //' **AND** password = '';

Les injections SQL sont encore bien trop présentes dans les applications web. Il est pourtant simple de contrer ces attaques. Voici un tableau récapitulatif des différents moyens de défense les plus pragmatiques :

N°	Méthode	Description
1	Framework	Utiliser un framework pour le développement, de préférence avec un ORM (mapping objet-relationnel)
2	Préparer vos requêtes	Désinfecter (*sanitize*) les requêtes envoyées en base de données avec des méthodes telles que : **JAVA** - **PreparedStatement()** **.NET** - **SqlCommand()** ou **OleDbCommand()** **PHP** - utiliser **PDO** avec la méthode **prepare()** et **bindParam()** pour les paramètres contenant des entiers.
3	Réécriture d'URL	L'utilisation de la réécriture d'URL permet de compliquer fortement la tâche au cybercriminel et s'avère être une des protections contre les injections SQL.

Faille du type Injection XPath

> *“ XPath est un langage permettant de rechercher de l’information à l’intérieur d’un document XML (Extensible Markup Language) qui est aujourd’hui un incontournable dans le monde de l’informatique.*

```xml
<?xml version="1.0" encoding="UTF-8"?>
<heroes>
    <hero>
        <id>1</id>
        <login>neo</login>
        <password>trinity</password>
        <secret>Oh why didn’t I took that BLACK pill?</secret>
        <movie>The Matrix</movie>
        <genre>action sci-fi</genre>
    </hero>
    <hero>
        <id>2</id>
        <login>alice</login>
        <password>loveZombies</password>
        <secret>There’s a cure!</secret>
        <movie>Resident Evil</movie>
        <genre>action horror sci-fi</genre>
    </hero>
</heroes>
```

Injection XPath

Une requête simple XPath permettrait d'aller chercher les identifiants d'un utilisateur :

```php
<?php
    $login = $_REQUEST["login"];
    $login = xmli($login);
    $password = $_REQUEST["password"];
    $password = xmli($password);

    // Loads the XML file
    $xml = simplexml_load_file("passwords/heroes.xml");

    // XPath search
    $result = $xml->xpath("/heroes/hero[login='" . $login . "' and password='" . $password . "'
```

N°	Méthode	Description
1	Fonction de remplacement	Utilisez une fonction permettant de trouver les caractères non adaptés (quote et double quote) pour les remplacer par leur équivalent XML (**'**)

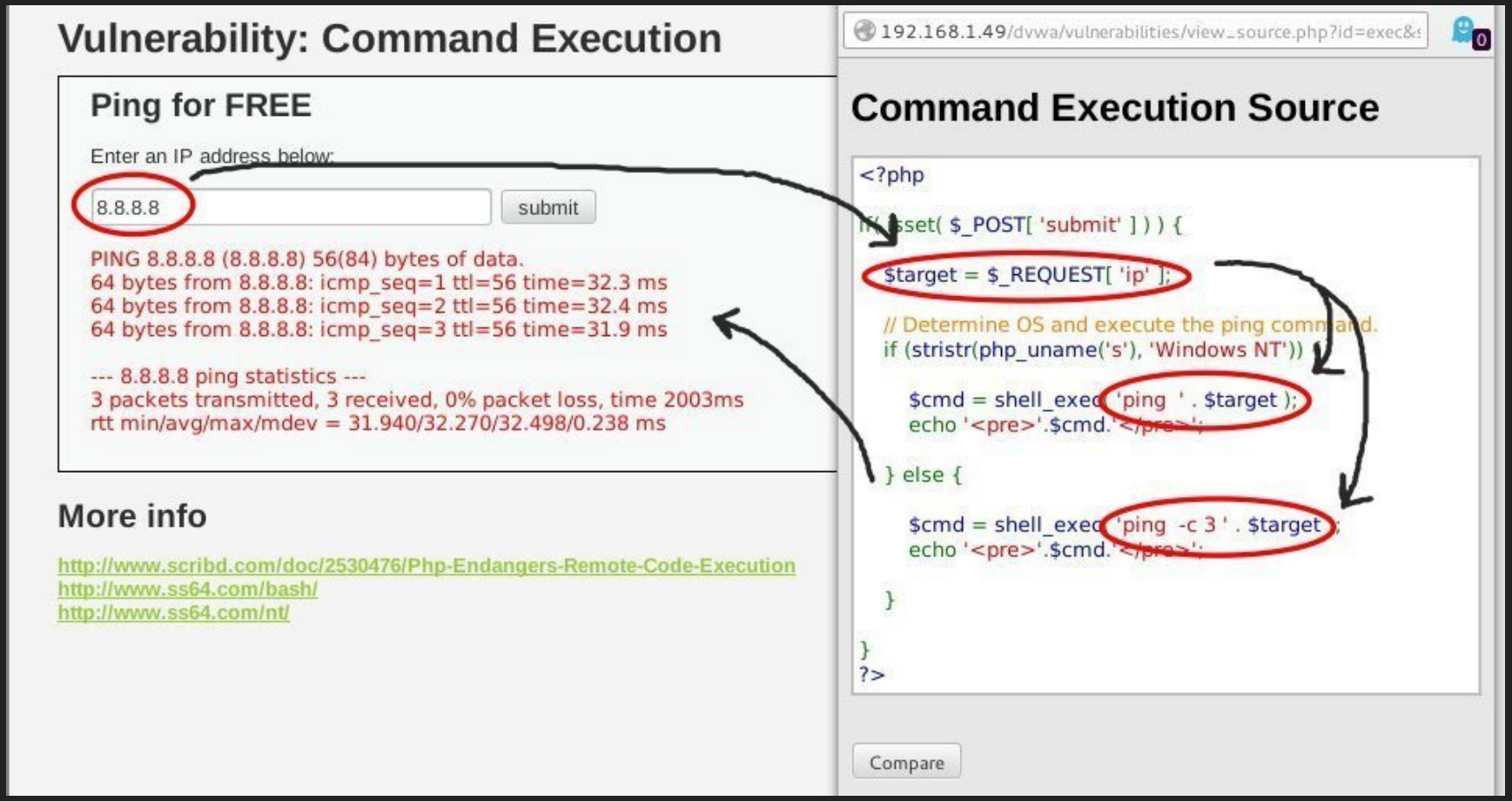
Faille du type Command Execution

Les attaques de type “Command Execution” ou “exécution de commande“, également appelées “remote code execution” pour “exécution de code à distance“.

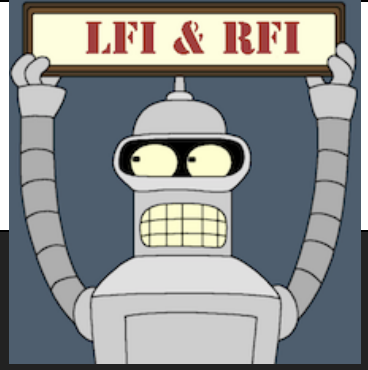
En effet, c’est tout le principe de ce type d’attaque qui consiste à exécuter du code (PHP ou bash par exemple) tout en étant dans la position d’un client web : à distance.

Les fonctions eval(), include() et shell_exec() sont souvent citées parmi les fonctions à bannir car l’injection de code PHP dans une fonction include() permet d’exploiter une vulnérabilité appelée RFI (RFI File Inclusion)

DVWA – Command Injection – level "low"



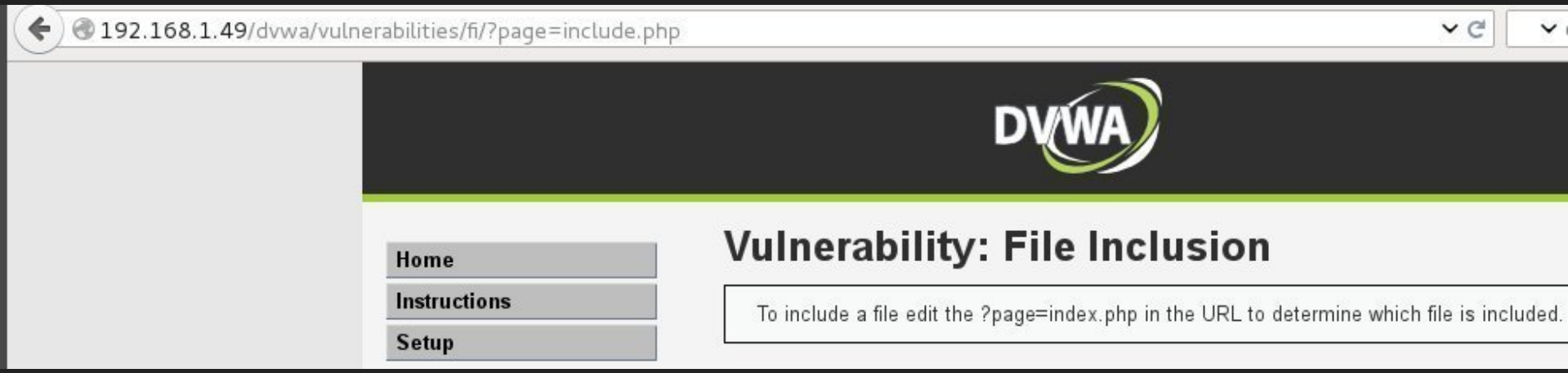
Failles de type File Inclusion



Local File Inclusion (LFI) ou Remote File Inclusion (RFI) est un type de vulnérabilité trouvé le plus souvent sur des sites web. Il permet à un attaquant d'inclure un fichier distant, généralement par le biais d'un script sur le serveur web. La vulnérabilité est due à l'utilisation de l'entrée fournie par l'utilisateur sans validation adéquate. Elle peut conduire à :

- L'exécution de code sur le serveur web
- L'exécution de code sur le côté client comme le JavaScript qui peut conduire à d'autres attaques comme Cross-site scripting (XSS)
- Déni de service (DoS)
- Le vol de données / Manipulation

DVWA – File Inclusion – level "low"

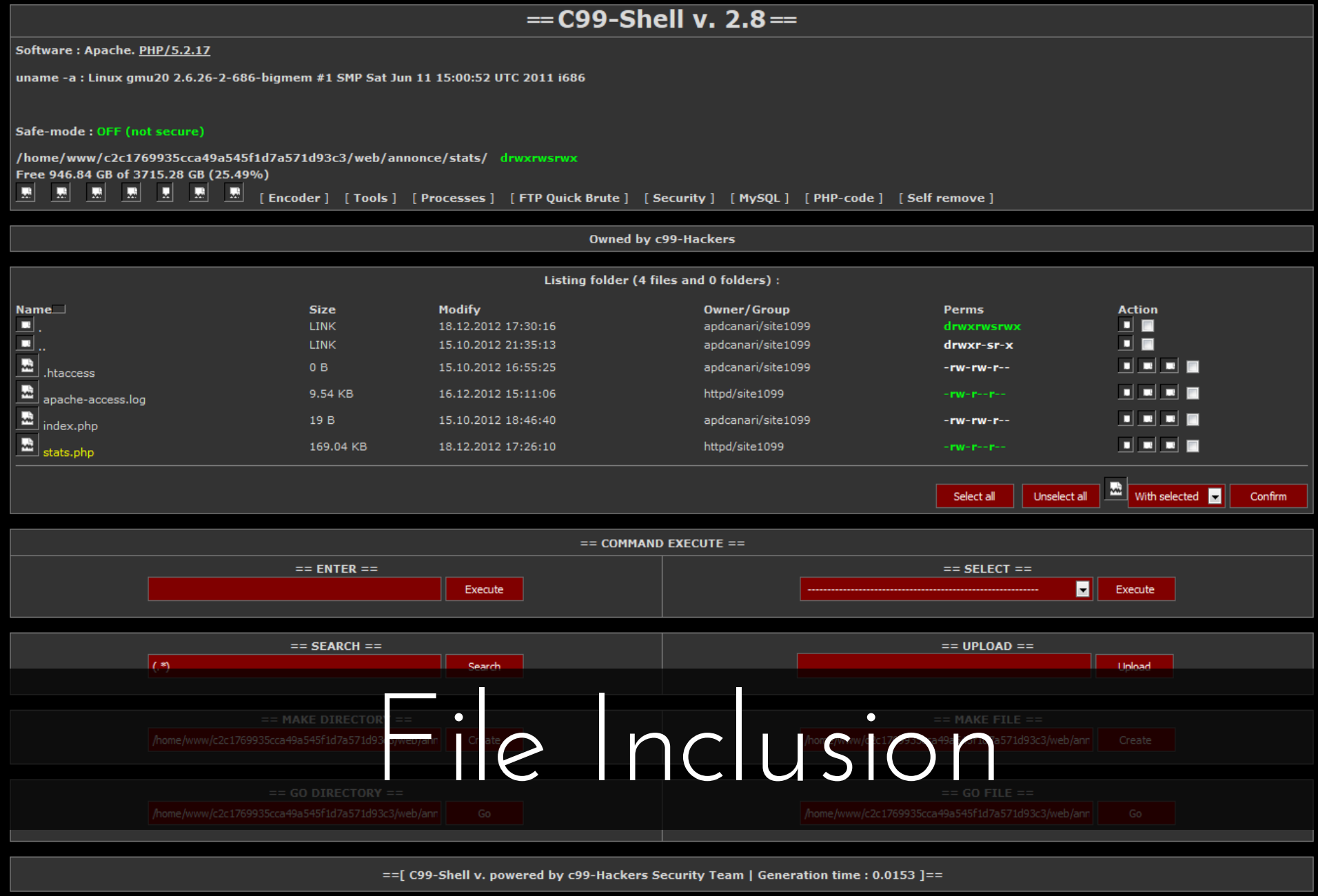


Ici, on peut donc essayer de demander à la page de charger un fichier qui se situe hors de l'application web :

http://localhost/securiteWEB/dvwa/vulnerabilities/fi/?page=../../../../../../../../etc/passwd

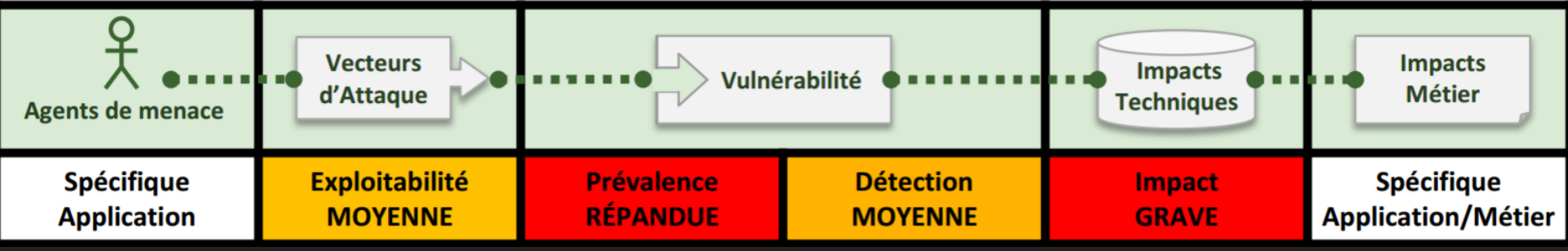
Failles de type File Inclusion

N°	Méthode	Description
1	Bonne utilisation des fonctions d'inclusion de code	Les fonctions d'inclusion de code doivent être utilisées à bon escient et ne jamais inclure des entrées utilisateurs.
2	Bannir la fonction **eval()**	La fonction **eval()** ne doit jamais être à portée d'une entrée utilisateur.



File Inclusion

Violation de Gestion d'authentification et de Session



Violation de Gestion d'Authentification et de Session

" La violation de session et d'authentification permet à un cybercriminel de capturer ou contourner les méthodes d'authentification utilisées par une application web.

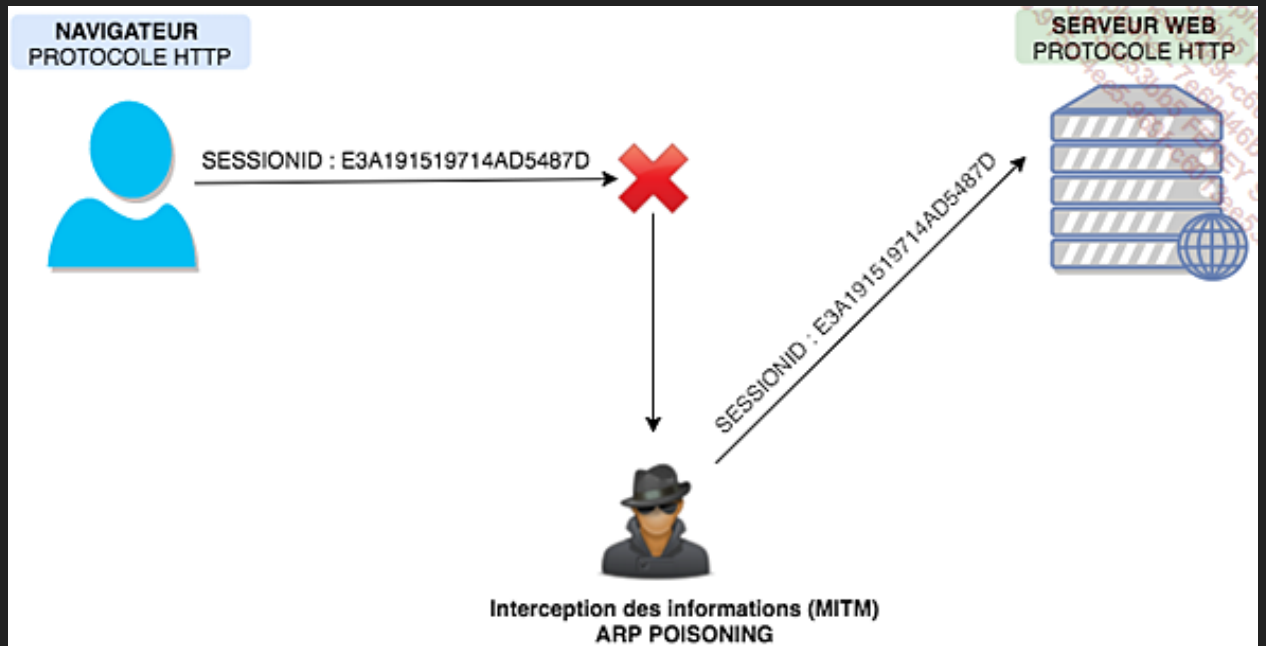
Les attaques de ce type sont dues à des défauts de conception lors de la création de l'application ou au manque de chiffrement réseau ou des mots de passe.

Les risques d'une telle attaque sont importants car elle touche directement à la protection des données des utilisateurs et à leur vie privée mais aussi aux administrateurs des entreprises avec le risque que des cybercriminels accèdent à des comptes non autorisés.

Vol de session (session hijacking)

La méthode de l'homme du milieu (MITM), est très efficace pour le vol de sessions.

Toute connexion HTTP sans système de chiffrement laisse passer en clair les identifiants et mots de passe ainsi que les identifiants de session demandés par le serveur lors de chaque requête HTTP par le client, ce qui laisse les applications web sans connexion HTTPS sensibles au vol de session sur les réseaux locaux, privés et les hotspots (café, aéroport, gare, etc.).



Vol de session (session hijacking)

N°	Méthode	Description
1	Ajout HTTPS	Un moyen efficace est l'utilisation **exclusive** du protocole HTTPS sur un serveur web. Le protocole se démocratise et la plupart des hébergeurs le proposent. La Linux Foundation propose des certificats reconnus et gratuits, disponibles à cette adresse : https://letsencrypt.org/.
2	Flag secure cookie	Activer sur le serveur web la fonctionnalité **flag secure cookie** déjà introduite dans le chapitre précédent. Cette fonctionnalité permet de transmettre les identifiants de session au serveur seulement sous un protocole sécurisé (HTTPS), ce qui permet de réduire les attaques de l'homme du milieu.

Une personne célèbre telle que Mark Zuckerberg a vu ses comptes Twitter et Pinterest piratés en 2016 car son mot de passe était "Dadada".

cupp.py - Common User Passwords Profiler

À l'aide du script **cupp.py**, nous allons générer des mots de passe

cupp.py

Faiblesses des mots de passe

N°	Méthode	Description
1	Mot de passe complexe	Lors de la création d'un compte d'utilisateur, il est nécessaire de forcer l'utilisateur à utiliser un mot de passe complexe (majuscules, minuscules, chiffres, caractères complexes). Cette vérification doit s'effectuer côté client et serveur. Voici un plugin jQuery très bien réalisé pour la partie client : http://jquerycards.com/forms/inputs/strength-js/
2	Jeton	La création d'un jeton sur la page d'authentification permet de rendre la tâche plus compliquée pour les cybercriminels. Même s'il est possible d'intercepter, cette méthode représente déjà un premier rempart.
3	Captcha	Suivant les exigences en matière de sécurité, si le token n'est pas suffisant, le Captcha reste un moyen efficace pour les attaques par dictionnaire ou *brute force*. Tous les captchas ne se valent pas, Google propose un service de Captcha (reCAPTCHA) simple et efficace. Pour plus d'informations sur reCAPTCHA : https://www.google.com/recaptcha/intro/index.html
4	Temporisateur et blocage	Une solution assez peu répandue mais très efficace est de mettre un temporisateur sur les tentatives d'authentification trop importantes et de bloquer le compte utilisateur lorsqu'une attaque par dictionnaire ou *brute force* est détectée. Cette solution reste quand même lourde à mettre en place et n'est pas très agréable pour les utilisateurs lors d'un oubli de mot de passe par exemple.

Mot de passe non protégé en base de données

C’est le cas avec les mots de passe stockés pour une application (SGBD, SGBD-R, XML, etc.) : aucun mot de passe ne doit y être conservé en clair car si une personne malveillante y accède, aucune mesure ne pourra remédier à cette fuite de données.

Ordinairement, les mots de passe sont hachés avant l’insertion en base de données, ce qui est une bonne démarche, à condition d’utiliser les bonnes méthodes de hachage et du salage.

Un cybercriminel ayant volé une base de données pourrait facilement retranscrire le mot de passe "password" avec une table arc-en-ciel.

Mot de passe non protégé en base de données

N°	Méthode	Description
1	Hash + Salt	Une méthode pour l'insertion de mots de passe salés et hachés en base de données consiste à concaténer le mot de passe et la clé dynamique. Chaque langage récent propose des méthodes de hachage et salage dynamique.

Exemple avec PHP 5 pour un salage dynamique :

```
// Salage dynamique et hachage
<?php
$options = [
    'cost' => 11,
    'salt' => mcrypt_create_iv(22, MCRYPT_DEV_URANDOM),
];
echo password_hash("rasmuslerdorf", PASSWORD_BCRYPT, $options)."\n";
?>

// Vérification du hachage
<?php
$hash = '$2y$07$BCryptRequires22Chrcte/VlQH0piJtjXl.0t1XkA8pw9dMXTpOq';

if (password_verify('rasmuslerdorf', $hash)) {
    echo 'Password is valid!';
} else {
    echo 'Invalid password.';
}
?>
```

Faiblesses dans la conception des sessions

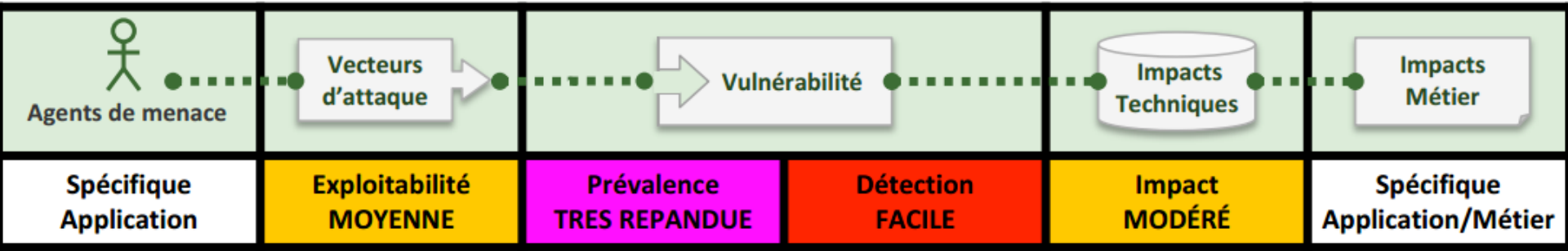
Les erreurs concernant la "gestion des sessions" pouvant être observées sur les applications web communément conçues *from scratch*, sans framework. Les erreurs les plus communes sont

- Une mauvaise gestion du temps de validité (*timeout*) d'une session, ce qui augmente le risque de vol de sessions. Chaque application web doit déterminer le temps valable d'une session utilisateur.
- L'utilisation dans les URL (méthode GET) des identifiants de session.
- L'identifiant de sessions est trop petit, 128 bits requis au minimum.
- Le flag HTTPOnly est absent (vu dans le chapitre précédent : Panorama de la sécurité web - La sécurité des navigateurs et serveurs web).

Faiblesses dans la conception des sessions

N°	Méthode	Description
1	*Timeout*	Le temps donné au *timeout* de session doit être défini par la politique de sécurité ou le développeur. Il est bien de connaître le temps moyen de fréquentation d'un utilisateur sur l'application et de ce fait, de gérer le temps d'inactivité de celui-ci afin de quantifier le nombre de minutes avant expiration de la session.

Cross-Site Scripting



Cross-Site Scripting (XSS)

" La troisième place du classement OWASP TOP 10 est attribuée au Cross-Site Scripting qui a pour particularité d'être le seul risque du TOP 10 ayant comme indice de prévalence "très répandu". Effectivement, d'après une étude de l'éditeur Egdescan, 52 % des applications web seraient sujettes au XSS en 2015.

Malgré les protections sur les navigateurs, les pare-feu applicatifs (WAF) et les outils d'analyses de code, les XSS restent une vraie source de menaces pour les utilisateurs.

Failles de type XSS

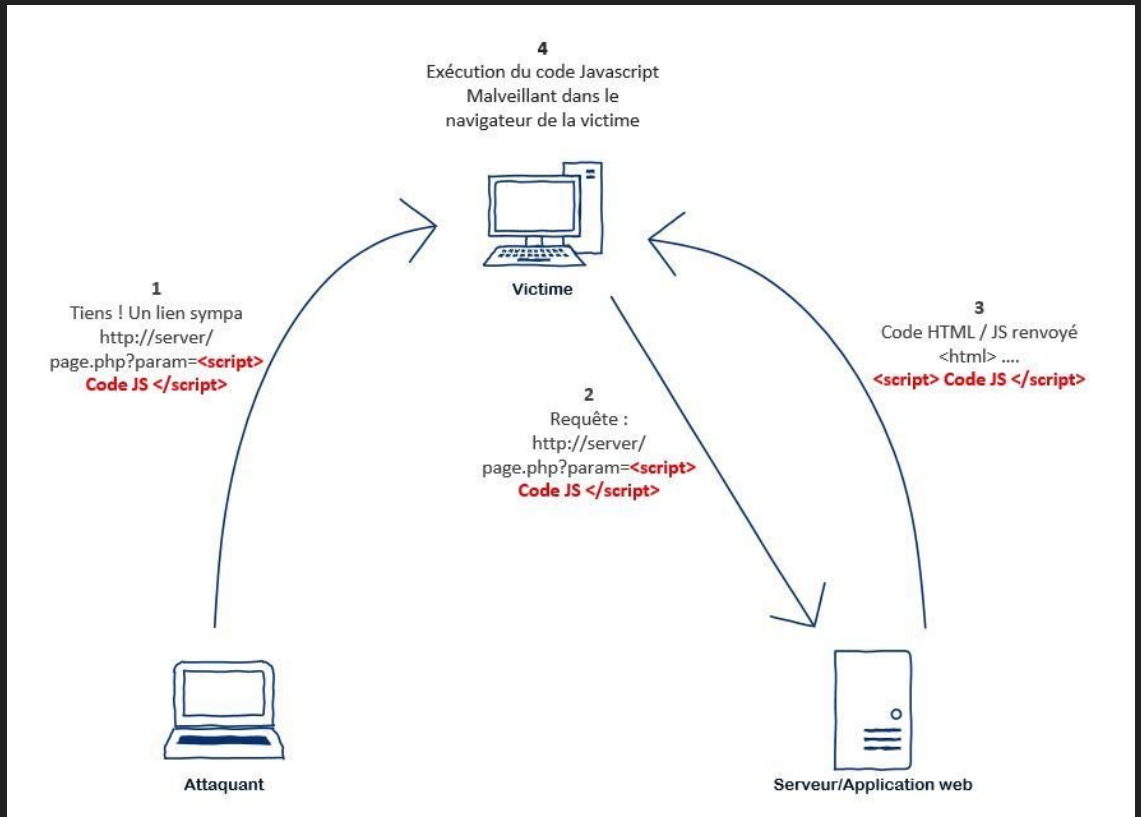
WIKIPÉDIA
L'encyclopédie libre

> *“ Le cross-site scripting (abrégé XSS), est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, permettant ainsi de provoquer des actions sur les navigateurs web visitant la page.*
> *Les possibilités des XSS sont très larges puisque l'attaquant peut utiliser tous les langages pris en charge par le navigateur (JavaScript, Java, Flash...) et de nouvelles possibilités sont régulièrement découvertes notamment avec l'arrivée de nouvelles technologies comme HTML5.*

Il est toutefois à noter qu’il en existe deux types, les failles XSS réfléchies (comme un miroir) et les failles XSS stockées.

Failles de type Reflected XSS

La particularité des failles XSS réfléchies est qu'elles ne sont pas permanentes, cela signifie que le code malveillant que l'attaquant souhaite exécuter sur les navigateurs des victimes n'est pas stocké dans l'application web mais est propre à la requête/réponse entre le serveur web et la victime.



DVWA – Reflected XSS – level "low"

Par exemple afin de mettre en gras une certaine partie de la chaîne via les balises "strong" : john pas John

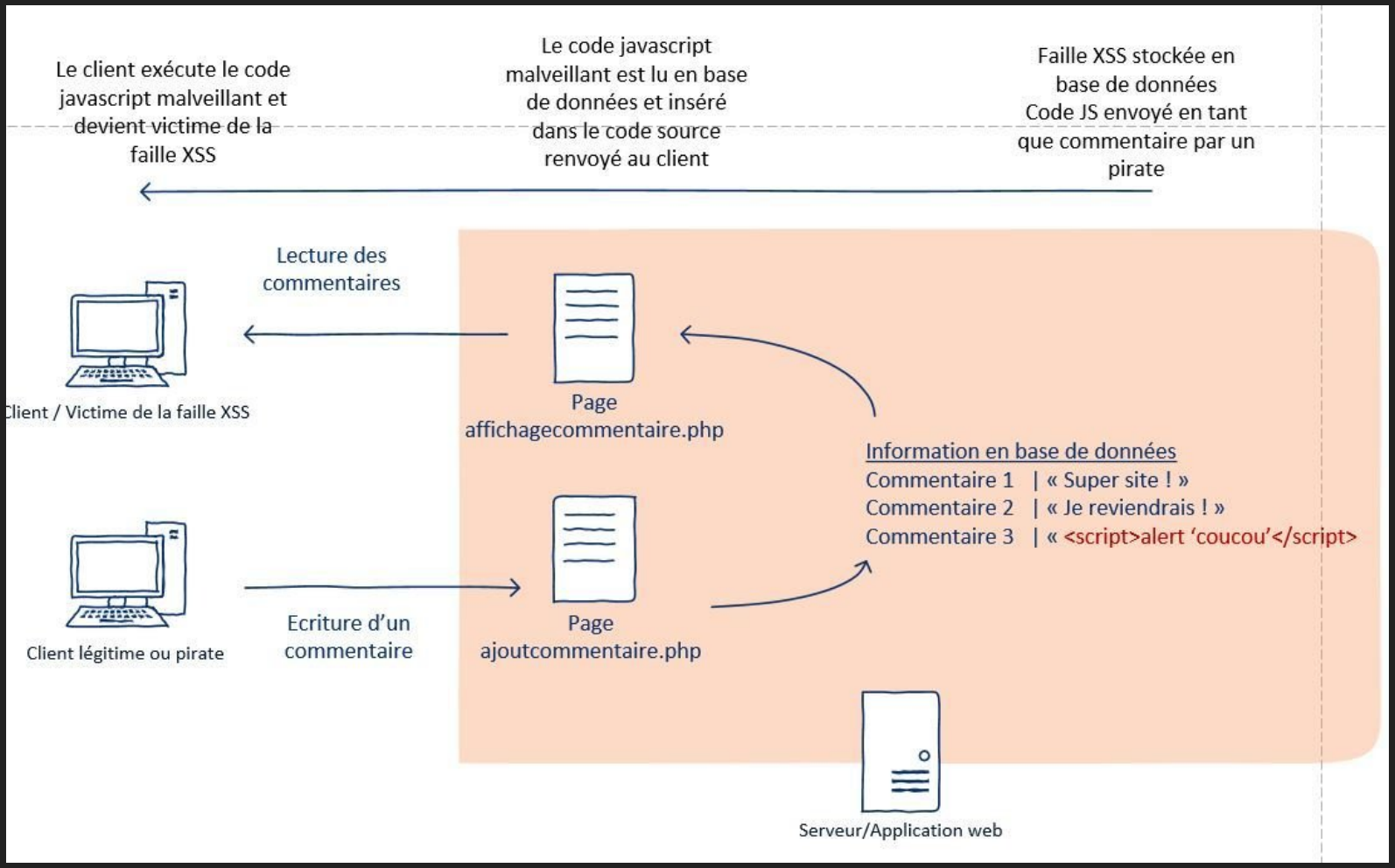
Failles de type Stored XSS

Les failles de type XSS stockées possèdent un principe d'exploitation qui diffère légèrement des XSS réfléchies.

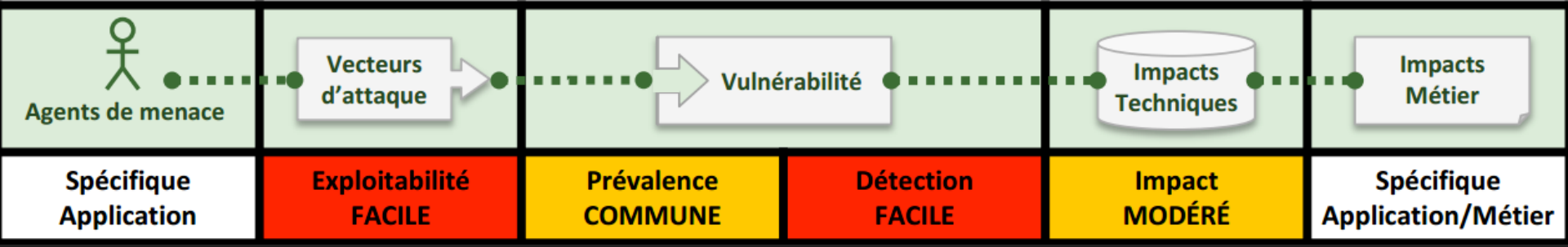
En effet, ici, le code JavaScript sera stocké dans l'application web, le plus souvent en base de données.

Cette faille est également appelée "Faille du livre d'or", les commentaires ou livres d'or permettent en effet à des utilisateurs sans permissions spécifiques de stocker des informations dans la base de données du serveur, en temps normal des commentaires tout à fait standards.

Schema


Le client exécute le code javascript malveillant et devient victime de la faille XSS
Le code javascript malveillant est lu en base de données et inséré dans le code source renvoyé au client
Faille XSS stockée en base de données
Code JS envoyé en tant que commentaire par un pirate
Lecture des commentaires
Client / Victime de la faille XSS
Page affichagecommentaire.php
Information en base de données
Commentaire 1 | « Super site ! »
Commentaire 2 | « Je reviendrais ! »
Commentaire 3 | « <script>alert 'coucou'</script>
Client légitime ou pirate
Ecriture d'un commentaire
Page ajoutcommentaire.php
Serveur/Application web

Références directes non sécurisées à un objet



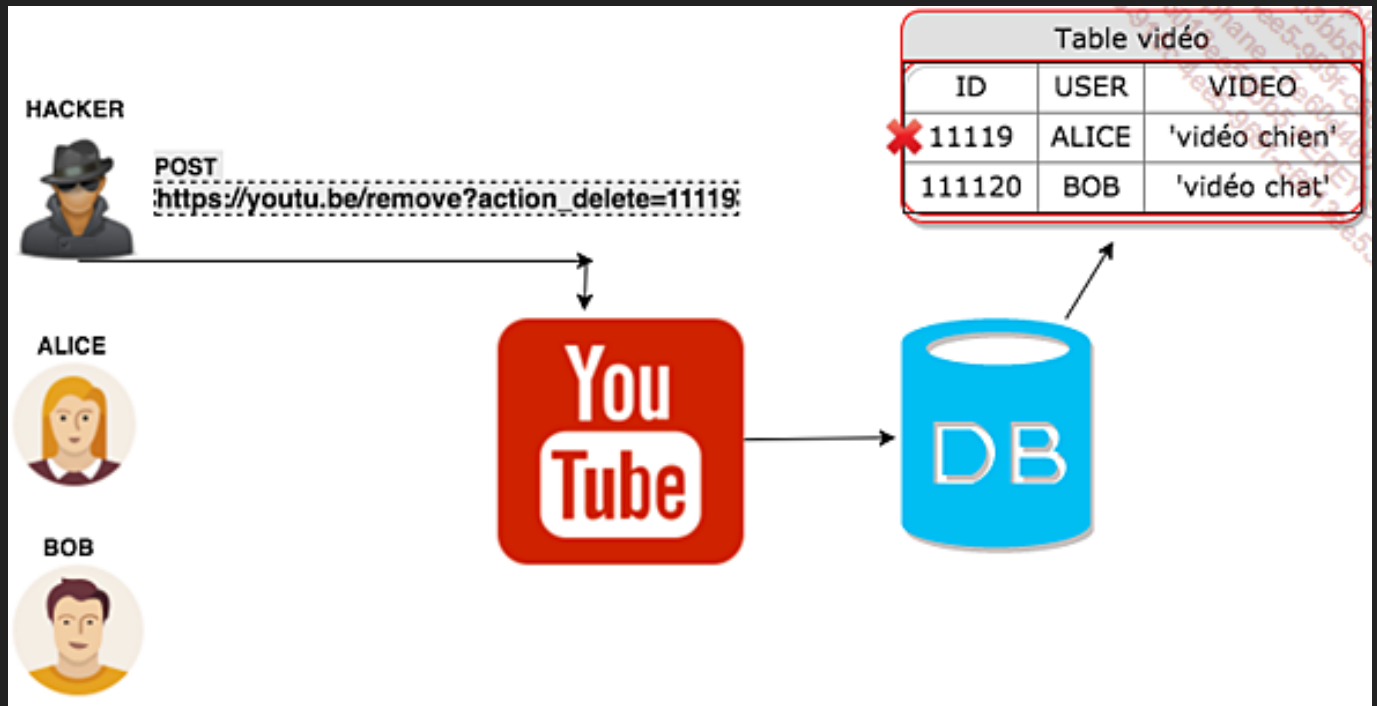
Références directes non sécurisées à un objet

" Les références directes non sécurisées à un objet sont des vulnérabilités s'attaquant à la mauvaise gestion des droits et d'identification dans une application.

Chaque site web contenant des données dynamiques comme un back-office pour l'administration en arrière-plan d'une application, doit être pourvu de rôles d'utilisateurs, d'administrateurs ou de modérateurs suivant les besoins de l'application. Si une référence à un objet n'est pas contrôlée à chaque instanciation de celui-ci, un cybercriminel pourrait donc obtenir et modifier des informations auxquelles il ne devrait pas avoir accès.

Références directes non sécurisées à un objet

Une vulnérabilité du type Références directes non sécurisées a été découverte sur YouTube par un chercheur en sécurité russe nommé Kamil Hismatullin en 2015. Celui-ci pouvait détruire n'importe quelle vidéo en modifiant l'identifiant de la vidéo lors de la demande de suppression sur sa page YouTube.



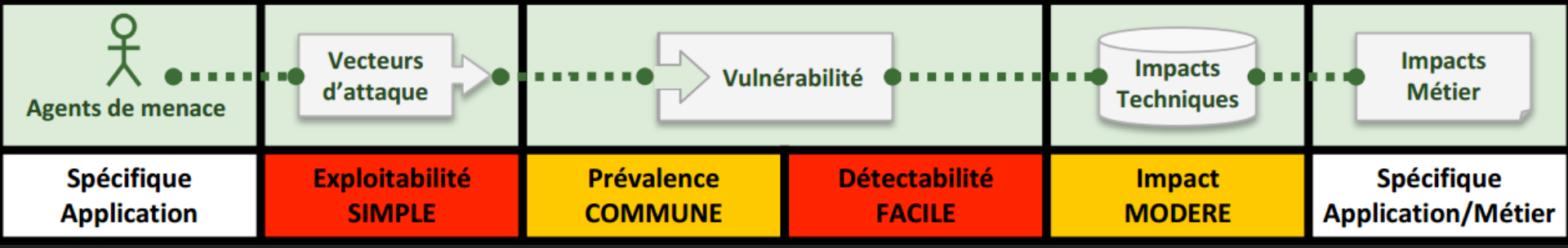
Entrées directes cachées et non contrôlées

En raison des références directes, non sécurisées, qui lui sont associées, dans les formulaires, les champs de type *hidden* (caché) sont des éléments souvent vulnérables.

Références directes non sécurisées à un objet

N°	Méthode	Description
1	Contrôle des références	Afin de se protéger contre ce genre d'attaques, il est nécessaire de mettre des contrôles stricts sur les références utilisateur. Un pattern permettant de vérifier si l'utilisateur est bien le propriétaire de l'objet peut s'avérer très utile.
2	Champs cachés	Protéger les références avec la technique 1 présentée sur la colonne ci-dessus pour les champs cachés, même ceux non affichés dans le DOM. Les techniques *Data biding* ou *Mass assignment vulnerability*ont la fonction d'envoyer des requêtes avec des paramètres aléatoires essayant de trouver des champs cachés et de compromettre l'application.
3	Framework	Les frameworks actuels, bien utilisés, comportent une gestion des rôles sur les applications efficaces. Encore une fois, ne pas hésiter à se servir de ce type d'outils pour développer.


Mauvaise configuration Sécurité



Mauvaise configuration Sécurité

" La mauvaise configuration des éléments de sécurité arrive en cinquième position du classement OWASP TOP 10.

Le périmètre d'une mauvaise configuration de sécurité peut dépasser le périmètre d'une application. La mise à jour des serveurs web, des librairies, des comptes administrateur inactifs, l'affichage des messages d'erreurs... sont des vulnérabilités associées à ce risque. Le principe des moindres privilèges, la sécurité par défaut, la défense en profondeur et la réduction des surfaces d'attaque aident à se protéger de ces risques et seront étudiés dans le prochain chapitre en détail.

Scénarios

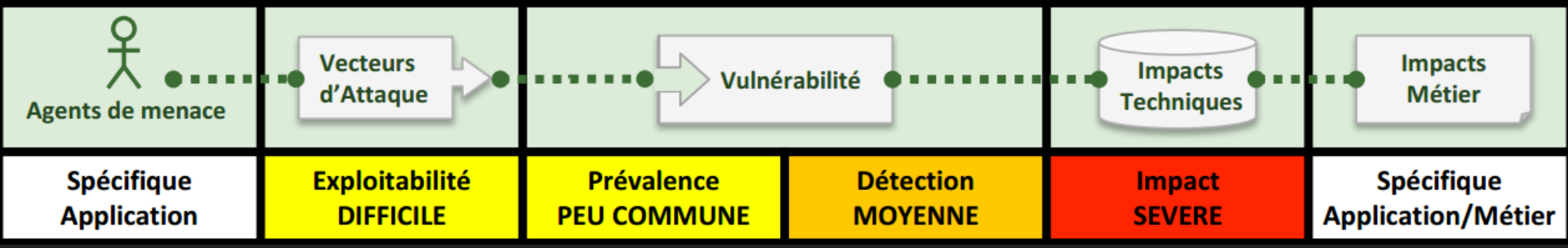
- Le cybercriminel provoque une erreur sur l'application, ce qui lui permet d'obtenir des informations importantes pour corrompre l'application.
- Le serveur hébergeant l'application a des utilisateurs et mots de passe par défaut sur les services SQL, SSH, FTP, Apache, IIS, etc.
- Une application faite avec un framework est "poussée" en production mais la partie préproduction avec des identifiants par défaut est aussi en production.
- Le serveur web liste les répertoires de type *index of* et permet à un attaquant de télécharger des fichiers et d'obtenir des informations.
- Il manque des mises à jour de sécurité sur les serveurs web et de stockage de données.
- Le serveur web autorise les requêtes HTTP de type TRACE, ce qui permet à l'attaquant d'exploiter une vulnérabilité XST (*Cross-Site Tracing*).
- Le serveur de base de données utilise le même utilisateur pour toutes les bases de données, ce qui permet à un attaquant lors d'une injection de pivoter entre les bases.
- Mauvaise configuration du fichier Robots.txt permettant de récupérer des informations sur l'architecture de l'application.

Solutions

Une bonne configuration est avant tout de l'expérience, le suivi des bonnes pratiques recommandées par les éditeurs et du bon sens.

- Recommandations ASVS : https://www.owasp.org/index.php/ASVS, chapitre 12.
- Cheatsheet OWASP sur la configuration sécurisée : https://www.owasp.org/index.php/Insecure_Configuration_Management
- Hardening pour serveur Apache : https://geekflare.com/apache-web-server-hardening-security/

Exposition de données sensibles



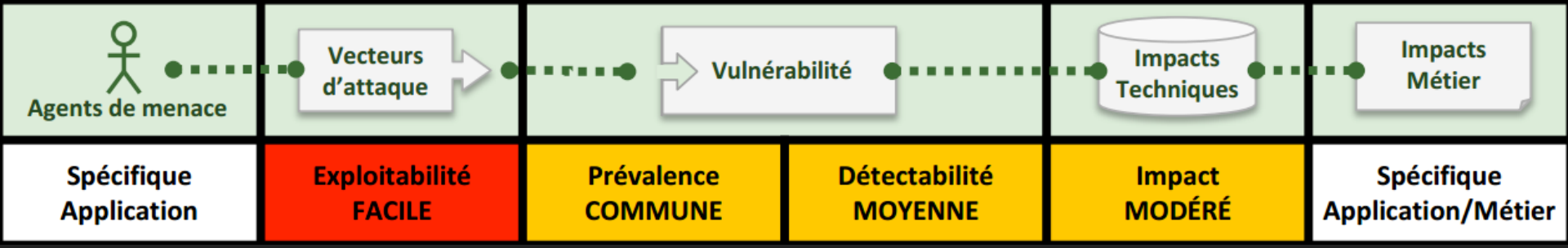
Exposition de données sensibles

> *" L'exposition de données sensibles concerne surtout les attaques cryptographiques, voire le manque de chiffrement sur les réseaux et données stockées.*

Scénarios

- Utilisation de protocoles de transport de données sécurisées faibles (SSL V2).
- Confusion entre encodage et chiffrement, par exemple, des mots de passe encodés en base64 dans la base de données.
- Aucun salage mis en place pour les mots de passe.
- Algorithme de hachage dépassé (MD5, SHA1-0).
- Utilisation de faibles clés de chiffrement.
- Pas de désactivation de l'attribut autocomplete dans les formulaires collectant des données sensibles.
- Stockage sur le serveur web de données sensibles sans rapport avec l'application.
- Utilisation de données sensibles de l'utilisateur dans les stockages localStorage, SessionStorage et indexedDB HTML5.
- Utilisation de caches dans les en-têtes HTTP pour le transfert de données sensibles.

Manque de contrôle d'accès au niveau fonctionnel



Manque de contrôle d'accès au niveau fonctionnel

" Le manque de contrôle d'accès est un risque lié à des vulnérabilités retrouvées dans la gestion des droits d'une application, des défauts de conception ou l'utilisation de fonctions appropriées à l'intérieur du code. L'attaquant peut alors accéder à des fonctionnalités non autorisées, voire compromettre le site, avec des attaques du type déni de service (DoS, DDoS).

Local/remote file inclusion

Ce code laisse la possibilité à un utilisateur d'envoyer au serveur web une référence à une page demandée sur le serveur, tel un menu.

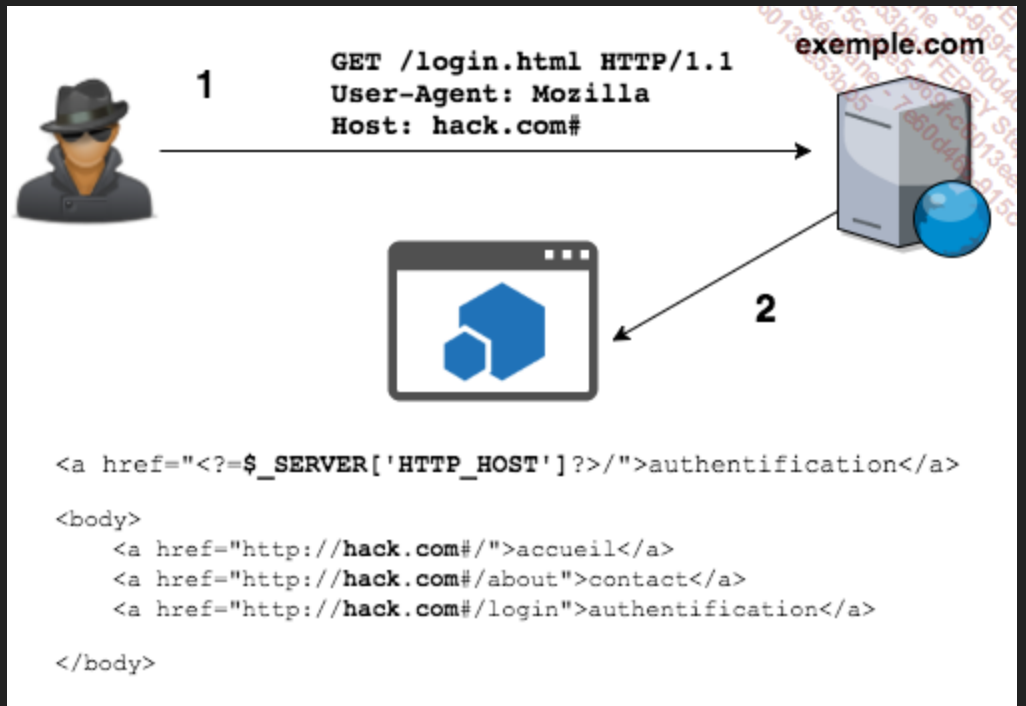
La faiblesse dans ce type de conception de code est qu'un cybercriminel pourrait modifier la requête envoyée au serveur avec une demande autre que l'item d'un menu, par exemple un fichier en local (LFI) ou sur un autre serveur (RFI).

Local/remote file inclusion

N°	Méthode	Description
1	Entrées utilisateur	Valider les entrées utilisateur et ne pas utiliser celles-ci dans des fonctions d'inclusions.
2	Désactiver des directives	Désactiver les directives permet l'utilisation d'inclusion d'URL. Exemple en PHP : **allow_url_open**, **allow_url_include** (php.ini).

Host Header Attack

Un attaquant peut exploiter et manipuler les en-têtes HTTP afin d'usurper ou modifier les valeurs utilisées par la suite dans le code. C'est le cas du Web cache poisoning, qui a pour effet d'envoyer une requête HTTP forgée, altérée, au serveur afin de s'emparer de celui-ci et de modifier le comportement de l'application.



Host Header Attack

N°	Méthode	Description
1	Bannir HTTP-HOST	Il est préférable d'utiliser les variables du type SERVER_NAME à HTTP-HOST car celui-ci utilise sa propre configuration de domaine et non pas celle proposée par l'utilisateur.

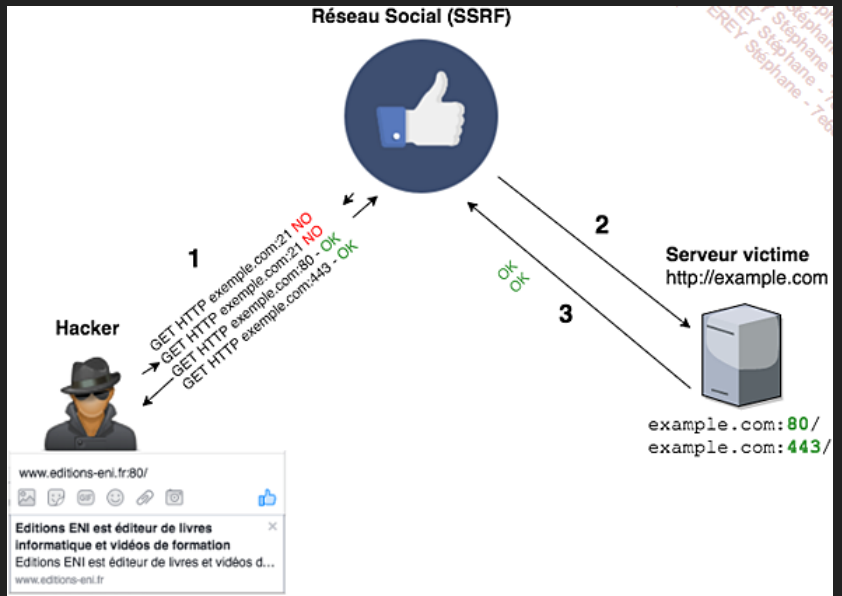
User-agent spoofing

Sur le même principe que les attaques Host Header, le *User-agent spoofing* permet de modifier, forger, une requête HTTP afin de passer outre des règles de redirection ou même utiliser des vulnérabilités XSS. Un *User-agent* est tout simplement l'identité du navigateur, envoyée à chaque requête HTTP lors d'une navigation sur un serveur web.

Cela permet de rediriger l'utilisateur vers un site mobile par exemple si celui-ci est un mobinaute. Les *User-agents* sont propres au navigateur, il est donc très simple de *spoofer*, d'usurper, cette identité à l'aide d'extensions, plugins ou logiciels

Server Side Request Forgery

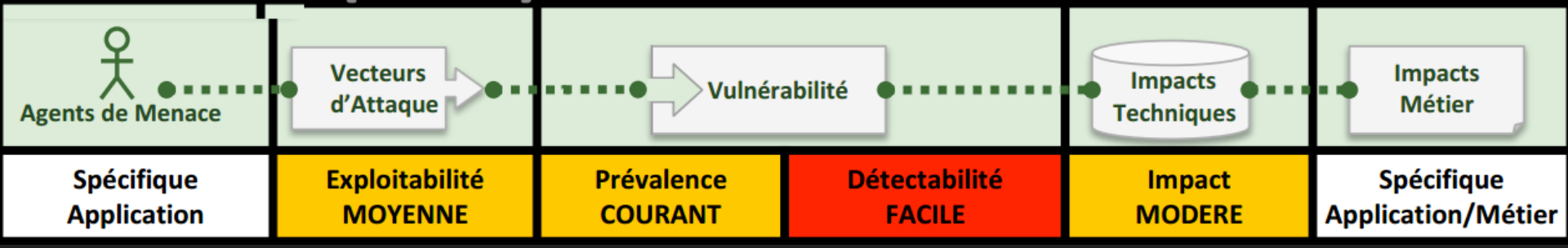
Les attaques de type *Server Side Request Forgery* ont pour finalité de faire exécuter une requête par le serveur pour servir le cybercriminel, généralement pour un scan de ports. Les techniques de *fingerprint*, de recherche d'information sur des serveurs et des infrastructures sont primordiales lors d'un test de pénétration. Des outils tels que Nmap et Netcat supportent le scan de ports sur des machines (client/serveur), le but étant d'énumérer les services disponibles afin de pouvoir exploiter des vulnérabilités liées à ceux-ci.



Host Header Attack

N°	Méthode	Description
1	Retour d'erreurs	Contrôler au maximum les retours d'erreurs (try and catch).
2	Désactiver les protocoles non nécessaires	À l'aide d'une *whitelist*, autoriser les protocoles nécessaires (HTTP, HTTPS) et interdire implicitement les protocoles non nécessaires. Ceci peut se faire sur le serveur web ou dans le code lui-même.

CSRF



Falsification de requête intersite (CSRF)

> *" Les vulnérabilités de type Cross Site Request Forgery peuvent s'apparenter aux vulnérabilités du type SSRF vu précédemment mais côté client. De plus en plus présentes sur le Web, elles permettent de forcer l'utilisateur à envoyer une requête HTTP à son insu afin de changer son mot de passe, acheter sur un site marchand ou le déconnecter d'une application.*

Faille du type CSRF

“ En sécurité informatique, le Cross-Site Request Forgery, abrégé CSRF (parfois prononcé sea-surfing en anglais) ou XSRF, est un type de vulnérabilité des services d'authentification web.

L’objet de cette attaque est de transmettre à un utilisateur authentifié une requête HTTP falsifiée qui pointe sur une action interne au site, afin qu'il l'exécute sans en avoir conscience et en utilisant ses propres droits. L’utilisateur devient donc complice d’une attaque sans même s'en rendre compte. L'attaque étant actionnée par l'utilisateur, un grand nombre de systèmes d'authentification sont contournés.

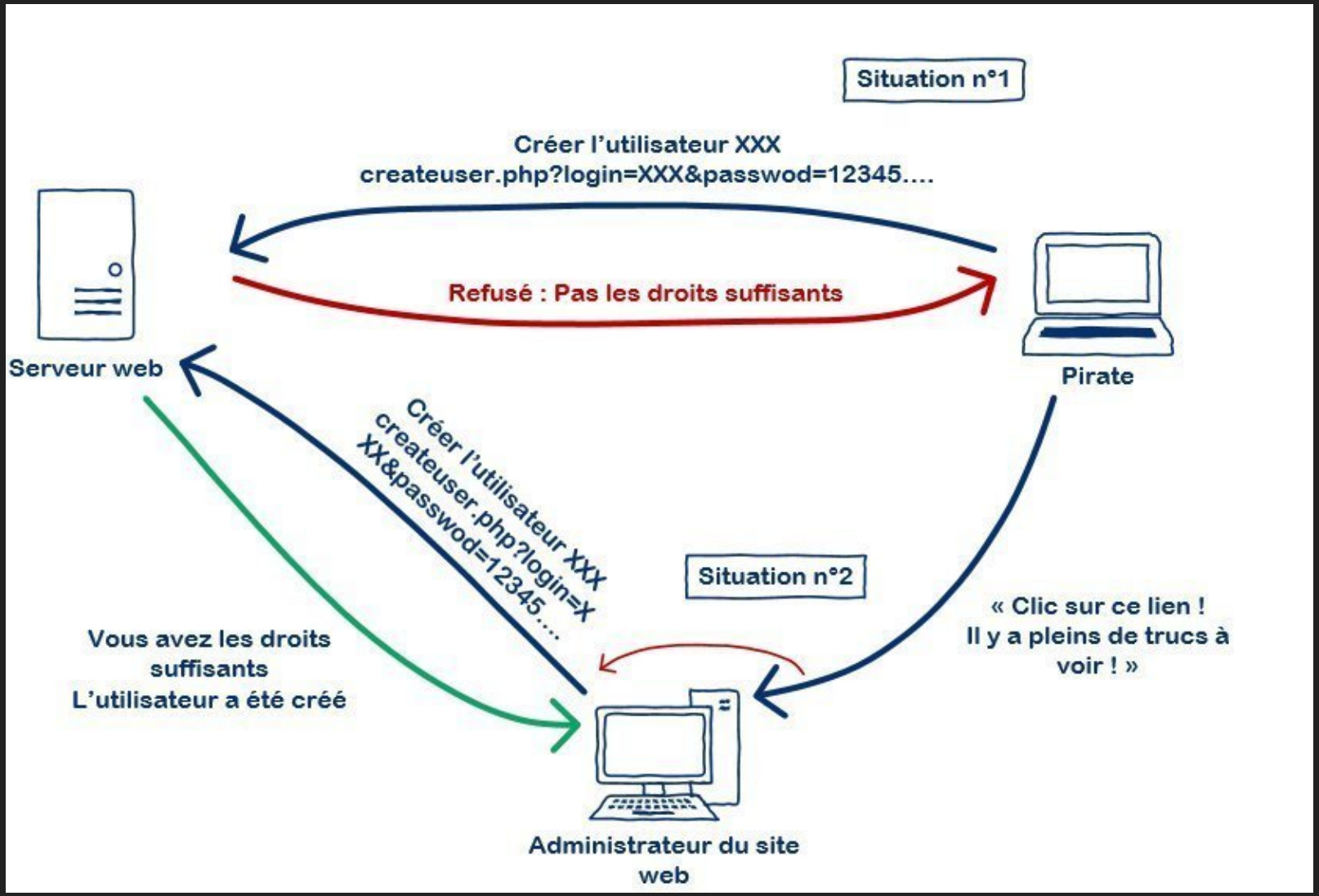
Illustration avec Alice et Bob

Supposons que Bob soit l'administrateur d'un forum et qu'il soit connecté à celui-ci par un système de sessions. Alice est un membre de ce même forum, elle veut supprimer un des messages du forum. Comme elle n'a pas les droits nécessaires avec son compte, elle utilise celui de Bob grâce à une attaque de type CSRF.

Illustration avec Alice et Bob

- Alice arrive à connaitre le lien qui permet de supprimer le message en question.
- Alice envoie un message à Bob contenant une pseudo-image à afficher (qui est en fait un script). L'URL de l'image est le lien vers le script permettant de supprimer le message désiré.
- Bob lit le message d'Alice, son navigateur tente de récupérer le contenu de l'image. En faisant cela, le navigateur actionne le lien et supprime le message, il récupère une page web comme contenu pour l'image. Ne reconnaissant pas le type d'image associé, il n'affiche pas d'image et Bob ne sait pas qu'Alice vient de lui faire supprimer un message contre son gré.

Schéma


Situation n°1
Créer l'utilisateur XXX
createuser.php?login=XXX&passwod=12345....
Refusé : Pas les droits suffisants
Serveur web
Pirate
Créer l'utilisateur XXX
createuser.php?login=X
XX&passwod=12345....
Situation n°2
« Clic sur ce lien !
Il y a pleins de trucs à
voir ! »
Vous avez les droits
suffisants
L'utilisateur a été créé
Administrateur du site
web

DVWA - CSRF - level "low"

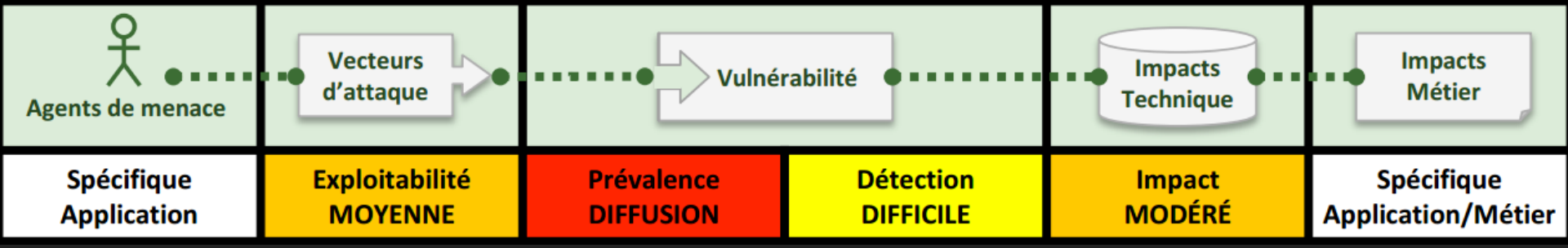
```
<a href=http://192.168.1.49/dvwa/vulnerabilities/csrf/?
password_new=weakpass&password_conf=weakpass&Change=Change#> Voir ce lien pour plus
d'information </a>
```

```
<img src=http://192.168.1.49/dvwa/vulnerabilities/csrf/?
password_new=weakpass&password_conf=weakpass&Change=Change#>
```

Falsification de requête intersite (CSRF)

N°	Méthode	Description
1	Jeton CSRF	Un des meilleurs moyens pour sécuriser les CSRF est l'insertion dans les formulaires d'un jeton unique attribué au tout début de la navigation utilisateur. De ce fait, si une page extérieure essaye de soumettre un formulaire, elle devra d'abord passer par la page d'accueil ou par une autre page de l'application, respectant ainsi le chaînage d'une navigation sur un site web.
2	Captcha	Un Captcha peut s'avérer efficace pour contrer les CSRF, en particulier les Captchas Google, dont l'efficacité est à la hauteur de leur facilité d'installation.
3	Framework	Les frameworks modernes automatisent et initialisent les jetons CSRF dans les formulaires des applications.

Exploitation de vulnérabilités connues

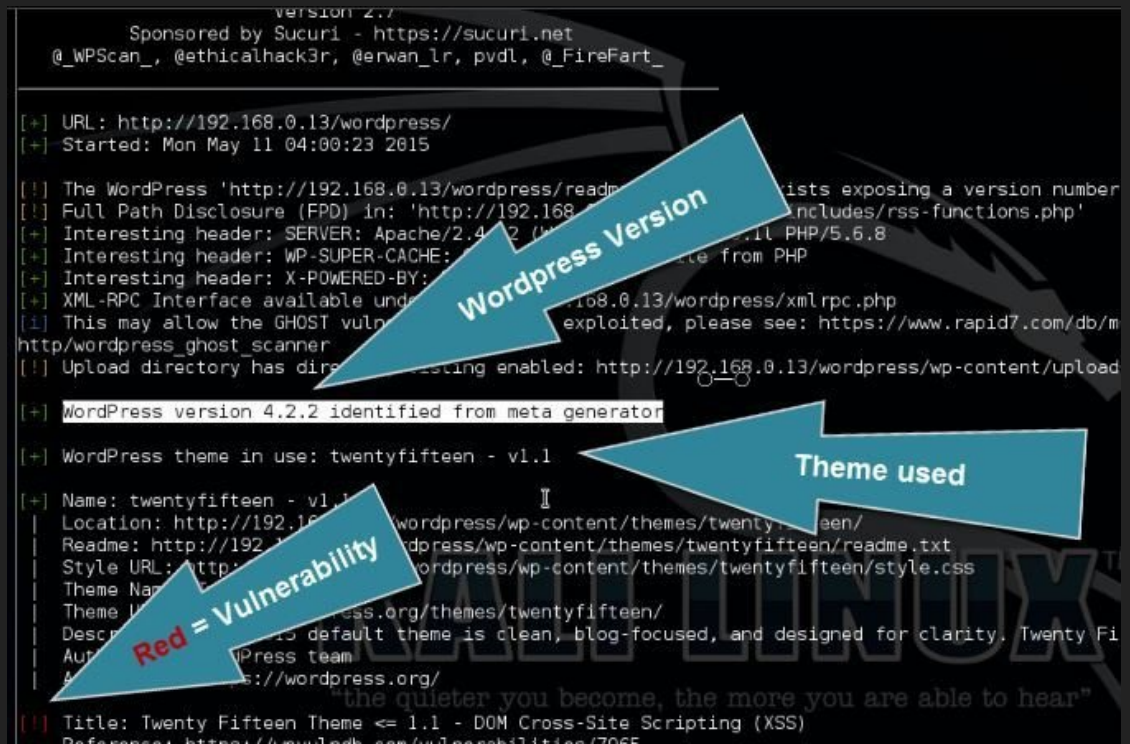


Utilisation de composants avec des vulnérabilités connues

> *" Le risque d'exploitation de vulnérabilités connues ne s'adresse pas seulement au périmètre du code mais à toute l'infrastructure d'une application comme les services web, les frameworks, les systèmes d'exploitation... Chaque année, de nouvelles vulnérabilités avec un impact sur des millions de systèmes sont dévoilées. Une des dernières fut Heartbleed, qui a impacté 17 % des serveurs web dits sécurisés par le protocole TLS, dont Amazon Web Services, GitHub, Reddit, etc.*

Certains outils sont très pratiques pour trouver des vulnérabilités connues sur les applications ou les serveurs. WPScan aide à trouver des vulnérabilités sur le CMS WordPress qui, je le rappelle, est le CMS le plus utilisé au monde.

Les vulnérabilités WordPress sont très répandues en raison de sa popularité.



Nikto

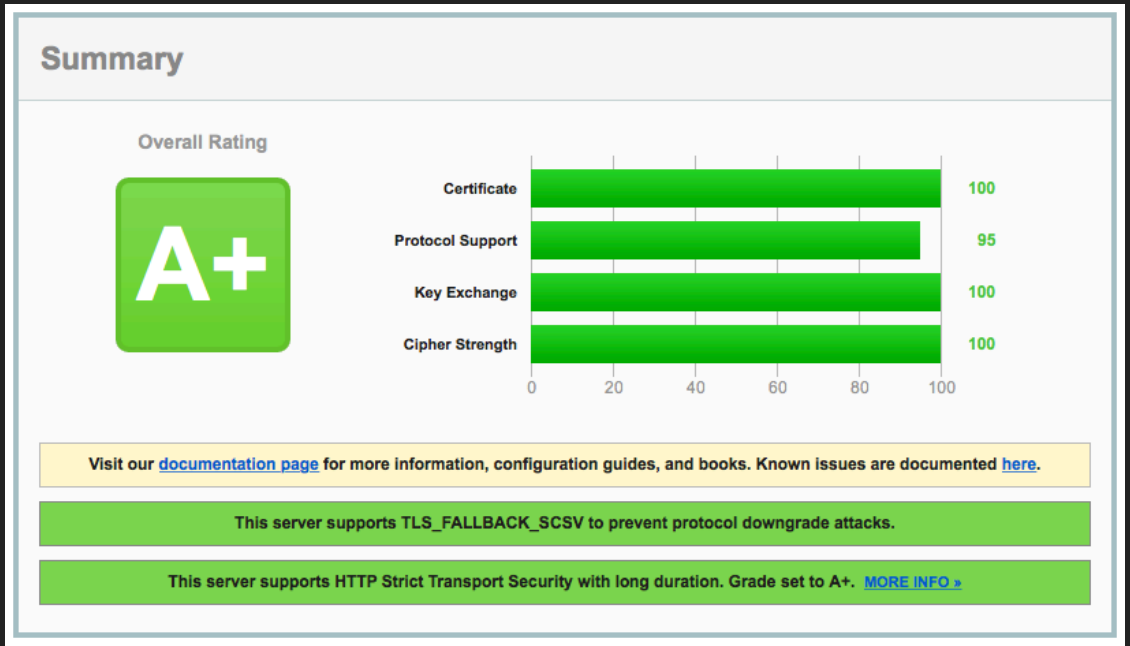
Parmi les scripts incontournables, Nikto a une bonne valeur ajoutée car il fait partie de la famille des scans de vulnérabilités ou *spiders*, le but étant d'alléger le travail du pentester (auditeur technique) ou bien d'utiliser cet outil quand on n'a pas les connaissances nécessaires pour trouver des vulnérabilités sur une application.

Pour aller plus loin dans le scan de vulnérabilités et la recherche de vulnérabilités connues, OpenVAS est un outil open source et un fork de la société Nessus dont la réputation a été construite autour de leur logiciel de scan de vulnérabilités.

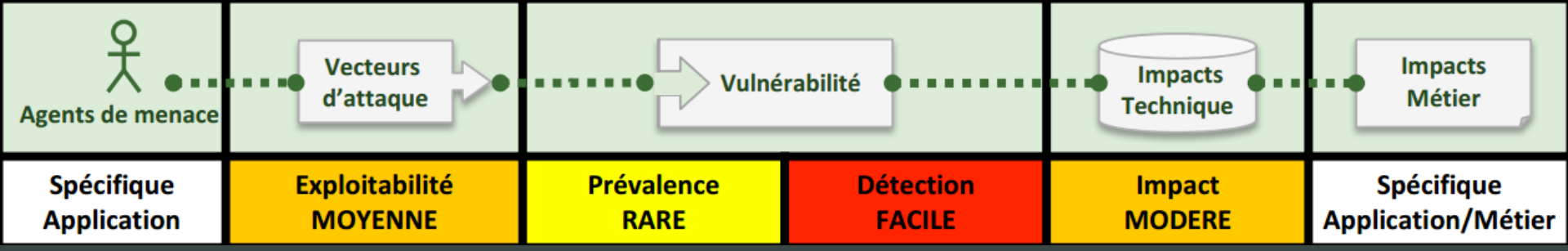
Exploitation de vulnérabilités connues

N°	Méthode	Description
1	Mise à jour	Il est primordial de bien gérer les mises à jour sur les bibliothèques, les plugins, les serveurs et les systèmes d'exploitation autour de votre application. La plupart des CMS proposent des plugins ou options pour la mise à jour automatique du cœur de l'application. Des notifications sont envoyées par e-mail quand un plugin n'est plus à jour. Veillez à bien surveiller celles-ci.
2	Veille en sécurité	S'inscrire à des flux RSS comme celui d'exploit-db.com (https://www.exploit-db.com/rss.xml) ou à une mailing list comme celle de http://www.securityfocus.com/ permet d'avoir un rapport journalier sur les vulnérabilités dévoilées et ainsi, de vérifier si l'on est concerné par celle-ci.

L'éditeur Qualys a développé et mis à disposition une solution en ligne dont le but est de vérifier la sécurité des chiffrements SSL/TLS (HTTPS) d'un serveur web ou d'un navigateur. Le chiffrement étant partagé entre ces deux éléments, Qualys permet de ce fait de faire la vérification de bout en bout de la chaîne de chiffrement entre l'utilisateur et le service web.



Redirections et Renvois Non Validés



Redirections et renvois non validés

> *" Les redirections et les renvois non validés sont une vulnérabilité profitant d'une faiblesse dans le code et dont l'objectif est de rediriger l'utilisateur sur une page malveillante ou administrative du site web. Il est courant de voir des redirections d'URL interne sur la plupart des applications utilisant un modèle MVC (Model View Controller).*

Redirections et renvois non validés

N°	Méthode	Description
1	Pas de redirection	Éviter au maximum les redirections en dur dans une application. Si cela n'est pas possible, utiliser une *whitelist* avec des identifiants pour chaque item.
2	Utiliser un framework	La plupart des frameworks actuels utilisent bien les redirections.

Conclusion

Top 10 OWASP

RISQUES	Agents de Menace	Vecteurs d'attaque	Exploitabilité	Prévalence	Vulnérabilité	Détection	Impacts Techniques	Impact	Impacts Métier
A1-Injection	Spécifique à l'Application	FACILE		COMMUNE	MOYENNEMENT		SÉVÈRE		Spécifique à l'Application
A2-Auth et Sess	Spécifique à l'Application	MOYENNE		RÉPANDUE	MOYENNEMENT		SÉVÈRE		Spécifique à l'Application
A3-XSS	Spécifique à l'Application	MOYENNE		TRÈS RÉPANDUE	FACILEMENT		MODÉRÉ		Spécifique à l'Application
A4-Réf non sécu.	Spécifique à l'Application	FACILE		COMMUNE	FACILEMENT		MODÉRÉ		Spécifique à l'Application
A5-Config	Spécifique à l'Application	FACILE		COMMUNE	FACILEMENT		MODÉRÉ		Spécifique à l'Application
A6-Données	Spécifique à l'Application	DIFFICILE		RARE	MOYENNEMENT		SÉVÈRE		Spécifique à l'Application
A7-ACL Fonc.	Spécifique à l'Application	FACILE		COMMUNE	MOYENNEMENT		MODÉRÉ		Spécifique à l'Application
A8-CSRF	Spécifique à l'Application	MOYENNE		COMMUNE	FACILEMENT		MODÉRÉ		Spécifique à l'Application
A9-Composants	Spécifique à l'Application	MOYENNE		RÉPANDUE	DIFFICILEMENT		MODÉRÉ		Spécifique à l'Application
A10-Redirection	Spécifique à l'Application	MOYENNE		RARE	FACILEMENT		MODÉRÉ		Spécifique à l'Application


Source

INFORMATION-SECURITY

- DVWA (1/4) : Brute Force et Command Execution
- DVWA (2/4) : CSRF et File Inclusion
- DVWA (3/4) : SQL Injection et SQL Injection Blind
- DVWA (4/4) : File Upload, Reflected XSS et Stored XSS

Sécurité informatique sur le Web

Apprenez à sécuriser vos applications

Auteur

Jérôme THÉMÉE

Caractéristiques

- 340 pages