Activité 9: Sécurisation des applications Web

Propriétés	Description
Intitulé long	Exploitation d'une plateforme d'apprentissage des vulnérabilités des applications *Web*
Intitulé court	Sécurisation des applications *Web*
Formation concernée	BTS Services Informatiques aux Organisations
Matière	Bloc 3 : Cybersécurité des services informatiques en deuxième année SLAM
Présentation	Ce Côté labo a pour objectif d'exploiter la plateforme d'apprentissage Portswigger.net du groupe *OWASP* (*OpenWeb Application Security Project*) afin de se familiariser avec les principales vulnérabilités des applications *Web*. Chaque activité couvre une problématique spécifique (*SQLi*, *XSS*, *CSRF*…) en référence au top 10 des vulnérabilités décrites par l'*OWASP*. Dans un premier temps, l'étudiant doit comprendre le mécanisme des attaques. Dans un deuxième temps, l'objectif est de réaliser des défis à travers des manipulations pratiques. **Cette neuvième activité** concerne les problématiques liées à l'identification et l'authentification sur u:ne application web. Cette vulnérabilité est classée n°7 dans la classement OWASP 2021.

Compétences	• Protéger les données à caractère personnel ;
	  ○ Identifier les risques liés à la collecte, au traitement, au stockage et à la diffusion de données à caractère personnel.
	• Garantir la disponibilité, l'intégrité et la confidentialité des services informatiques et des données de l'organisation face à des cyberattaques.
	  ○ Caractériser les risques liés à l'utilisation malveillante d'un service informatique ;
	Recenser les conséquences d'une perte de disponibilité, d'intégrité ou de confidentialité.

Savoirs	• Sécurité des applications web : risques, menaces et protocoles.
Prérequis	Administration d'un système *Linux*.
Outils	Une machine Kali Linux disposant d'un accès à internet et du logiciel BurpSuite (disponible sous Windows). Sites officiels : https://www.owasp.org et https://portswigger.net/burp/communitydownload
Mots-clés	OWASP, vulnérabilités, identification, authentification, BurpSuite, sniper.
Durée	Deux heures
Auteur(es)	Patrice Dignan, avec la relecture, les tests et les suggestions de Hervé Le Guern.
Version	v 1.0
Date de publication	01/05/25

I Introduction

L'identification et l'authentification sont des étapes clés de la sécurité dans une application web. **L'identification** consiste à déclarer qui l'on est (souvent avec un identifiant ou une adresse mail), tandis que **l'authentification** permet de prouver cette identité (par exemple en donnant un mot de passe).

Dans une application bien conçue, seules les personnes autorisées doivent accéder à certaines fonctionnalités ou données. Si l'identification ou l'authentification est mal gérée, des utilisateurs malveillants peuvent se faire passer pour d'autres.

C'est pourquoi il est essentiel, lors du développement, d'appliquer les bonnes pratiques de sécurité pour ces mécanismes.

II Présentation du type de vulnérabilité

Dans le classement OWASP Top 10 – 2021, la vulnérabilité "Identification and Authentication Failures" se classe en 7e position. Elle regroupe les failles qui permettent à un attaquant d'outrepasser l'authentification, de forcer des mots de passe ou encore d'accéder à des comptes sans autorisation.

Quelques exemples fréquents :

- CWE-287 : Authentification manquante ou incorrecte.
- CWE-798 : Utilisation de mots de passe codés en dur dans le code.
- CVE-2020-14750 : Faille dans Oracle WebLogic permettant une connexion sans authentification.

Conséquences : sur un site e-commerce, cela peut permettre de consulter des commandes d'autres clients, de modifier des données sensibles ou de détourner des paiements.

III Exemples de codes vulnérables

Quelques exemples de codes vulnérables :

PHP – Authentification simple et vulnérable :

```
if ($_POST['password'] == "admin123") {
    echo "Accès autorisé";
}
```

Mauvaise pratique : mot de passe en clair dans le code, sans gestion d'utilisateurs ni vérification sécurisée.

JavaScript (client-side) – Authentification côté client :

```
if (user === "admin" && password === "1234") {
    window.location = "admin.html";
}
```

L'authentification est visible et modifiable depuis le navigateur. Aucune sécurité côté serveur.

Python (Flask) – Session non protégée :

```
@app.route('/admin')
def admin():
```

```
if session.get('logged_in'):
    return "Bienvenue admin"
```

Risque : si un utilisateur modifie la session sans vérification du rôle ou identifiant, il peut accéder à une page réservée.

IV Les différentes formes d'authentification

Il existe plusieurs formes d'authentification selon le facteur utilisé :

1. Connaissance : mot de passe, code PIN.
2. Possession : téléphone, carte à puce, clé USB sécurisée.
3. Inhérence : empreinte digitale, reconnaissance faciale.
4. Contexte : géolocalisation, horaire habituel, adresse IP.

L'authentification multifacteur (MFA) combine au moins deux de ces facteurs. Exemple : mot de passe + code envoyé par SMS.
L'authentification forte, elle, repose souvent sur des techniques cryptographiques, comme une signature numérique ou un certificat. Elle est plus difficile à contourner.

V Contre-mesures de limitation et d'évitement

Pour éviter les failles d'authentification, plusieurs bonnes pratiques sont recommandées :

- Ne jamais stocker de mots de passe en clair : utiliser des algorithmes comme bcrypt ou Argon2.
- Appliquer une politique de mots de passe forts.
- Protéger les pages critiques avec une authentification serveur, et gérer correctement les sessions.
- Implémenter une limite de tentatives (protection contre le brute force).
- Éviter les messages d'erreur trop explicites ("Mot de passe incorrect" ou plutôt "Utilisateur inexistant").
- Utiliser des bibliothèques de sécurité reconnues pour gérer l'authentification.

VI Authentification et réglementation

La CNIL et le RGPD imposent de sécuriser l'accès aux données personnelles. Cela concerne directement l'identification et l'authentification.
Selon la CNIL, l'accès à des données sensibles doit être protégé par une authentification forte.

Les développeurs doivent aussi :

- Prévoir un système de journalisation des accès.
- Informer les utilisateurs des traitements de leurs données.
- Permettre la modification et suppression des comptes.

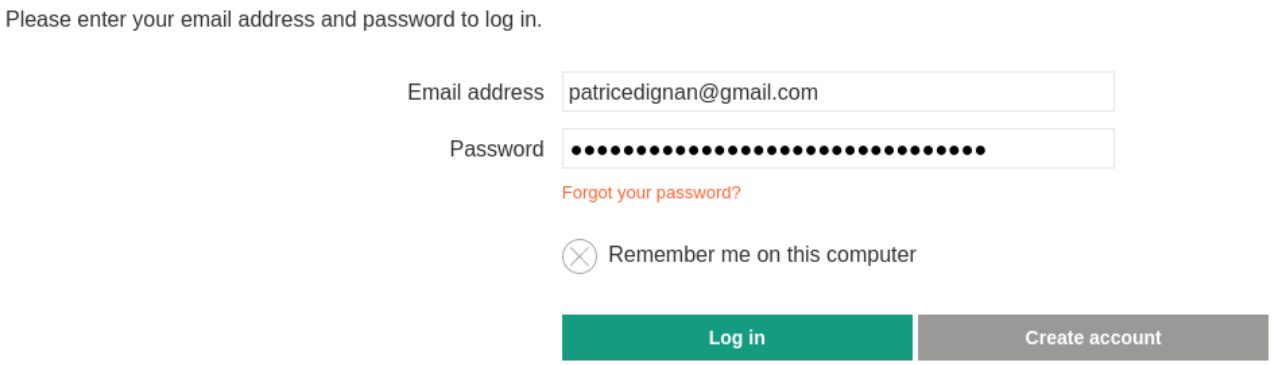
Ne pas respecter ces règles peut entraîner des sanctions importantes (amendes, responsabilité juridique...).

VII Présentation des défis

Les défis permettant d'illustrer la problématique de l'authentification sont extrait du site portswigger.net. PortSwigger est un leader mondial dans la création d'outils logiciels pour les tests de

sécurité des applications Web. Ces défis nécessitent un accès à internet ainsi que le logiciel BurpSuite.

Avant de commencer les labos, il est nécessaire de créer un compte sur le site. La réalisation des défis nécessite l'utilisation du dossier do



Les trois défis proposés permettent d'explorer différentes failles liées à l'identification et à l'authentification :

- **Défi n°1 : Énumération de logins et attaque par force brute**
  L'objectif est d'identifier un identifiant valide, puis de deviner son mot de passe en testant de nombreuses combinaisons.
- **Défi n°2 : Contournement d'une authentification à deux facteurs**
  Ce défi consiste à contourner une authentification renforcée reposant sur l'envoi d'un code de vérification par e-mail.
- **Défi n°3 : Usurpation de compte via la fonctionnalité "mot de passe oublié"**
  Le but est d'exploiter une vulnérabilité dans le mécanisme de réinitialisation du mot de passe pour accéder frauduleusement à un compte utilisateur.

VIII Réalisation des défis

Le lien permettant d'accéder au labo est le suivant :

https://portswigger.net/web-security/authentication/password-based/lab-username-enumeration-via-different-responses

Ce laboratoire présente des vulnérabilités liées à l'énumération des noms d'utilisateurs et à la force brute des mots de passe. Un compte y est accessible à partir d'un identifiant et d'un mot de passe faciles à deviner, tous deux présents dans des listes de mots couramment utilisées (dictionnaires).

Q1 : Effectuez les travaux préparatoires nécessaires pour le défi.

Q2 : Suivez les étapes pour identifier un login valide.

Q3 : Trouvez le mot de passe associé au login identifié pour valider le défi.

Défi n°2 : Contournement d'une authentification à double facteur

Le lien permettant d'accéder au labo est le suivant :

https://portswigger.net/web-security/authentication/multi-factor/lab-2fa-simple-bypass

L'authentification à deux facteurs de ce laboratoire peut être contournée. Vous disposez déjà d'un nom d'utilisateur et d'un mot de passe valides, mais vous n'avez pas accès au code de vérification 2FA de l'utilisateur. Pour résoudre le problème, accédez à la page du compte de Carlos.

Vos identifiants en tant qu'attaquant sont : wiener:peter
Les identifiants de la victime sont : carlos:montoya

Q1 : Réalisez les préparatifs indispensables pour le défi.

Q2 : Mettez en œuvre l'attaque permettant de contourner la saisie d'un code de validation envoyé par email.

Q3 : Analysez ce qui rend cette vulnérabilité possible et indiquez les contre-mesures à adopter.

Défi n°3 : Réinitialisation malveillante d'un mot de passe

Le lien permettant d'accéder au labo est le suivant :

https://portswigger.net/web-security/learning-paths/authentication-vulnerabilities/vulnerabilities-in-other-authentication-mechanisms/authentication/other-mechanisms/lab-password-reset-broken-logic

La fonctionnalité de réinitialisation du mot de passe de ce laboratoire est vulnérable. Pour résoudre le problème, réinitialiser le mot de passe de l'utilisateur Carlos, puis se connecter et accéder à sa page « mon compte ».

Vos identifiants en tant qu'attaquant : wiener/peter
Login de la victime : carlos

Q1 : Effectuez les préparatifs nécessaires pour le défi.

Q2 : Validez le défi d'usurpation d'identité de Carlos en tirant parti de la vulnérabilité du formulaire de réinitialisation du mot de passe.

Q3 : Examinez les facteurs qui rendent cette vulnérabilité possible et proposez des contre-mesures appropriées.

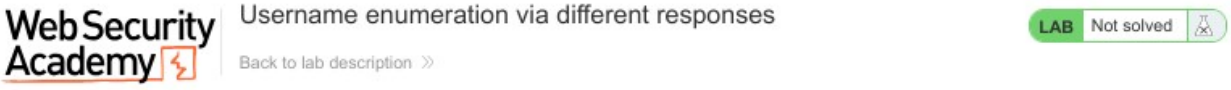
Dossier documentaire

Défi n°1 : Énumération des utilisateurs via différentes réponses et force brute du mot de passe

Le lien permettant d'accéder au labo est le suivant :

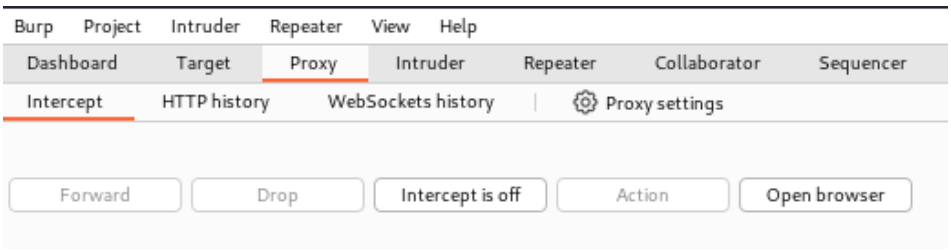
https://portswigger.net/web-security/authentication/password-based/lab-username-enumeration-via-different-responses

Ce laboratoire présente des vulnérabilités liées à l'énumération des noms d'utilisateurs et à la force brute des mots de passe. Un compte y est accessible à partir d'un identifiant et d'un mot de passe faciles à deviner, tous deux présents dans des listes de mots couramment utilisées (dictionnaires).



Travaux préparatoires :

1- Depuis une machine Kali connectée à Internet, lancez **BurpSuite**. Configurez ensuite le navigateur pour qu'il utilise un **proxy local** : adresse **localhost** et **port 8080**. Dans BurpSuite, rendez-vous dans l'onglet **Proxy** et désactivez le mode **Intercept** (mettre sur *off*).



2- Téléchargez ensuite les deux fichiers dictionnaires contenant les **identifiants** et **mots de passe** à tester.
Pour cela, utilisez un **éditeur de texte** depuis la machine Kali afin de créer ou modifier ces fichiers.

Lab: Username enumeration via different responses

APPRENTICE

LAB | Not solved

This lab is vulnerable to username enumeration and password brute-force attacks. It has an account with a predictable username and password, which can be found in the following wordlists:

- Candidate usernames
- Candidate passwords

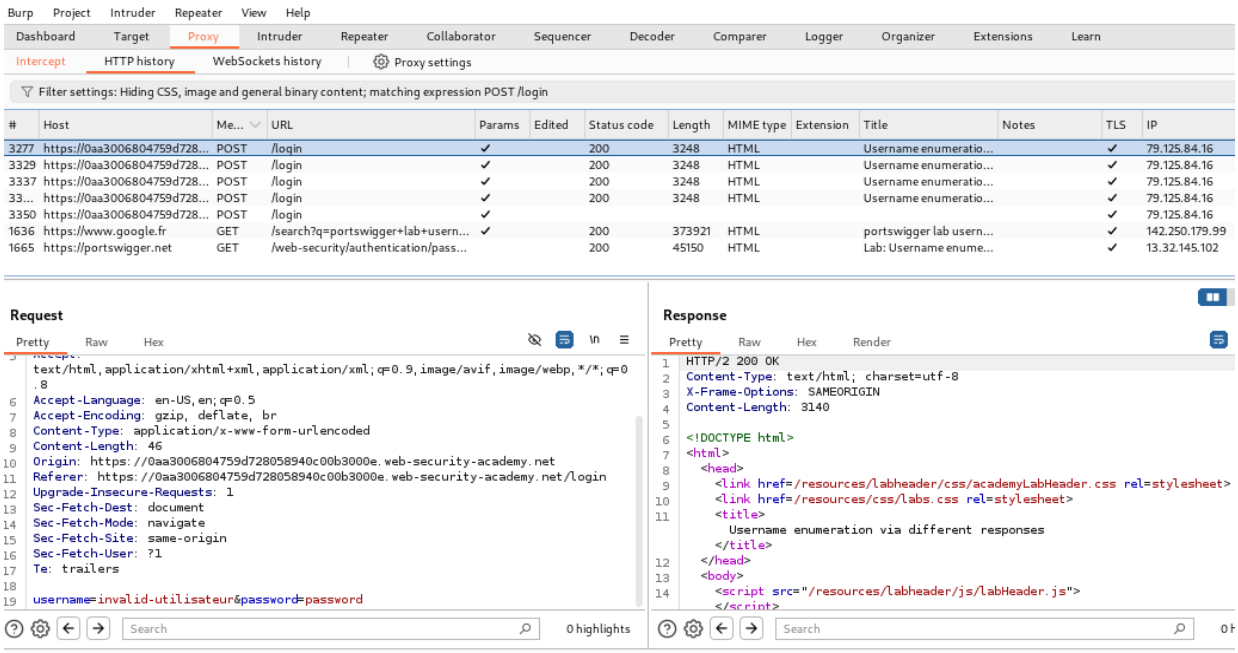
To solve the lab, enumerate a valid username, brute-force this user's password, then access their account page.

Identification d'un login valide :

1- Sur la page d'accueil du défi, cliquez sur le lien **« My account »** situé en haut à droite de l'écran.

2- Dans BurpSuite, activez le **mode Proxy** (Intercept : *on*). Sur la page de connexion du laboratoire, entrez un **identifiant** et un **mot de passe incorrects**, puis cliquez sur le bouton **« Log In »** pour valider.

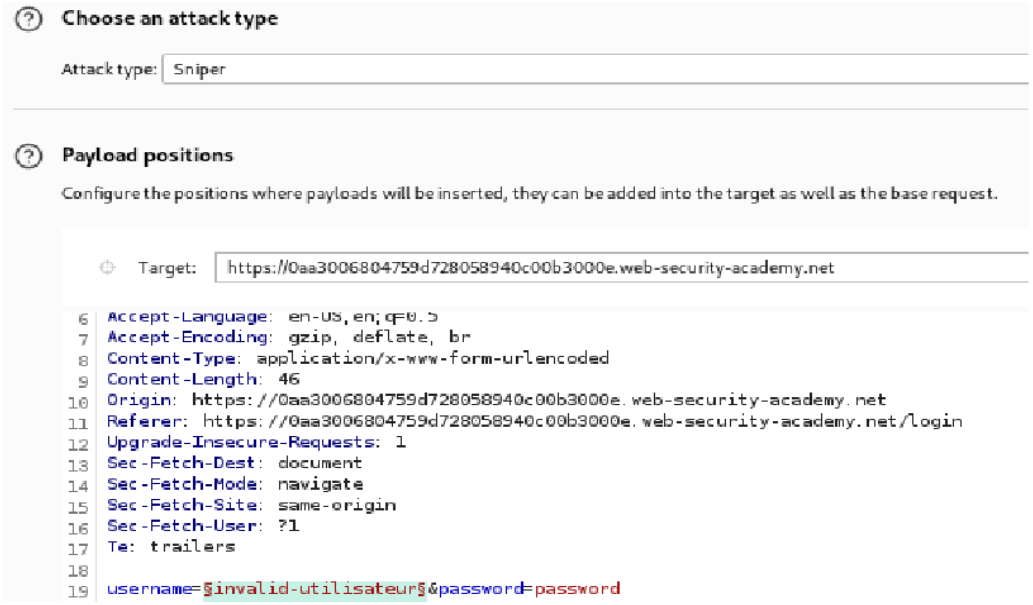
3- Une fois le bouton **« Log In »** cliqué, retournez dans BurpSuite et validez la requête interceptée en cliquant sur **« Forward »**. Ensuite, ouvrez l'onglet **HTTP history**, puis filtrez les requêtes affichées en recherchant la chaîne **POST/login**. Les détails de la requête envoyée et de la réponse reçue s'affichent dans la partie inférieure de la fenêtre.



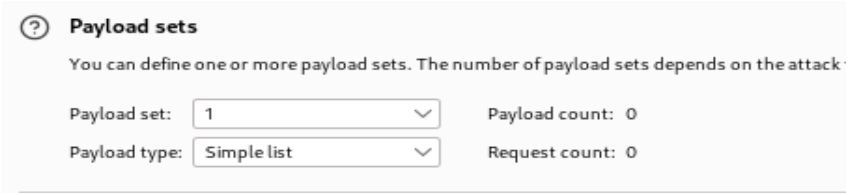
4- Dans la partie **Request** (à gauche), repérez la ligne contenant la valeur saisie pour le champ **username** (par exemple invalid-utilisateur).
Sélectionnez cette valeur avec la souris, faites un **clic droit**, puis choisissez **« Send to Intruder »** pour l'envoyer à l'outil Burp Intruder.

5- Rendez-vous dans l'onglet **Intruder** de BurpSuite.

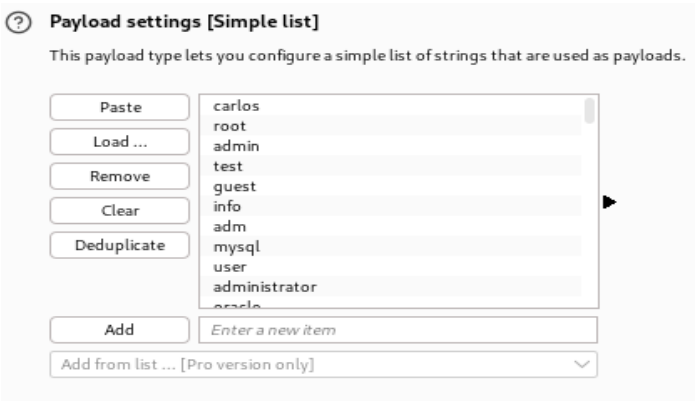
Vous y verrez que la valeur du champ **username** est désormais encadrée par les symboles **§**, ce qui indique qu'elle sera utilisée comme **variable de test** lors de l'attaque.



Vérifiez que le mode **Sniper** est bien activé avec l'option **Simple list**.



6- Dans l'onglet **Payloads**, chargez la liste de logins créée précédemment. Une fois la liste ajoutée, cliquez sur « S**tart attack** » pour lancer l'attaque.



L'attaque prend un certains temps. Ne pas hésiter à faire une pause en prenant un café.

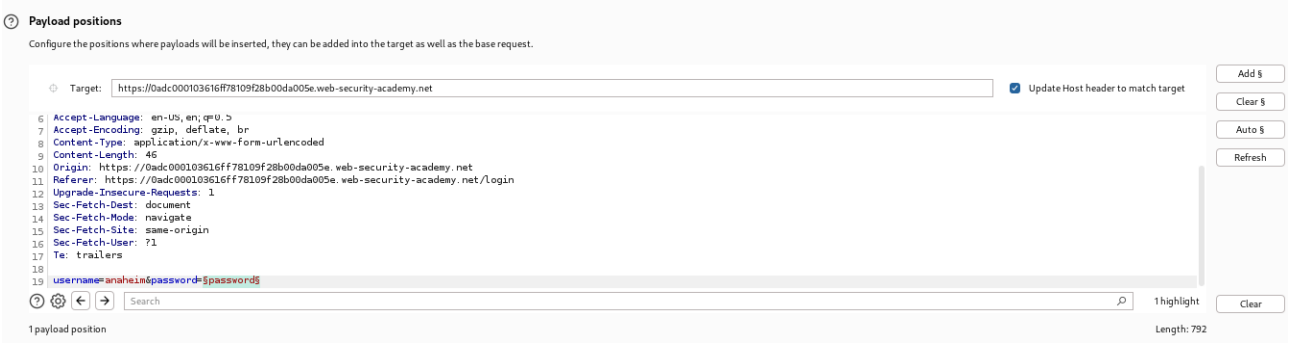
7 - Une fois l'attaque terminée, examinez la colonne **Length**, qui indique la longueur de la réponse reçue pour chaque tentative. Vous constaterez qu'un des identifiants testés génère une réponse plus longue que les autres, accompagnée d'un message différent. Alors que la majorité des requêtes

affichent « **Invalid username** », celle-ci retourne « **Incorrect password** ». Cela signifie que nous avons identifié un login valide. Il reste maintenant à trouver le mot de passe associé à ce compte.

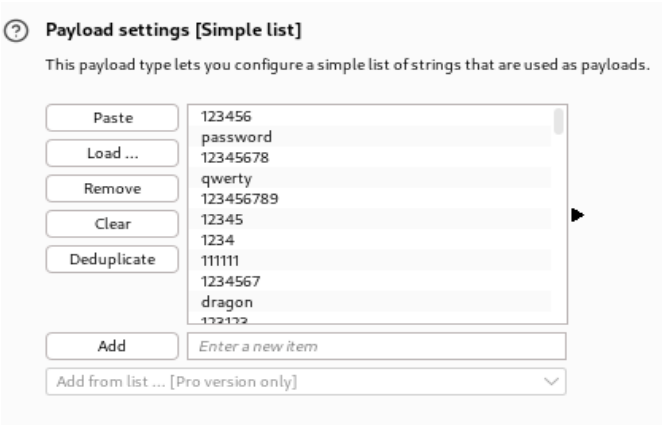
Force brute du mot de passe du login valide :

1 - Fermez la fenêtre de l'attaque précédente, puis retournez dans l'onglet Positions de **Burp Intruder**. Cliquez sur le bouton « **Clear** » pour réinitialiser les marqueurs. Remplacez ensuite l'ancien identifiant par celui que vous avez identifié comme valide. Puis, sélectionnez avec la souris la valeur du champ mot de passe, et cliquez sur « **Add** » pour définir cette partie comme variable de test, en restant en mode Sniper.

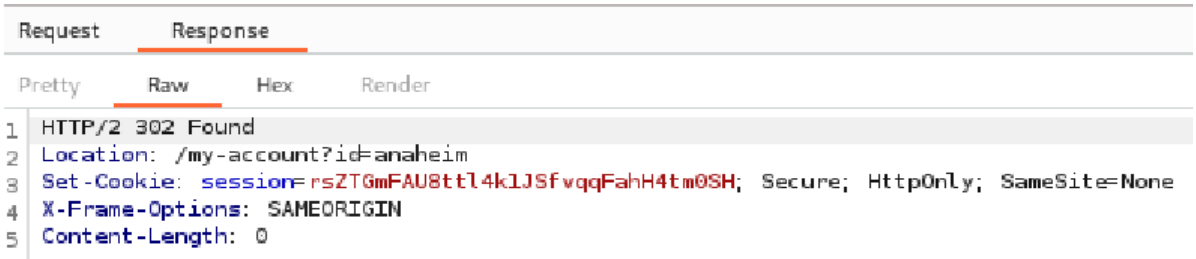
username=identified-user&password=§invalid-password§



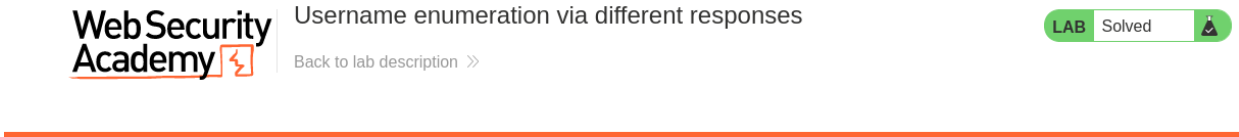
2- Dans l'onglet **Payloads**, effacez la liste des identifiants précédemment utilisée. Remplacez-la par la liste des mots de passe que vous avez préparée, puis cliquez sur « **Start attack** » pour lancer l'attaque.



3- La ligne de réponse correspondant au **code de retour 302** indique le bon mot de passe.



4- S'authentifier avec le login et le mot de passe trouvé pour valider le défi.



Défi n°2 : Contournement d'une authentification à double facteur

Le lien permettant d'accéder au labo est le suivant :

https://portswigger.net/web-security/authentication/multi-factor/lab-2fa-simple-bypass

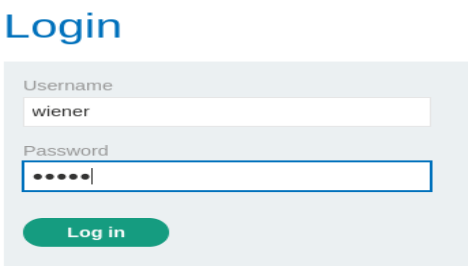
L'authentification à deux facteurs de ce laboratoire peut être contournée. Vous disposez déjà d'un nom d'utilisateur et d'un mot de passe valides, mais vous n'avez pas accès au code de vérification 2FA de l'utilisateur. Pour résoudre le défi, il faut accéder à la page du compte de Carlos.

Vos identifiants en tant qu'attaquant sont : wiener:peter

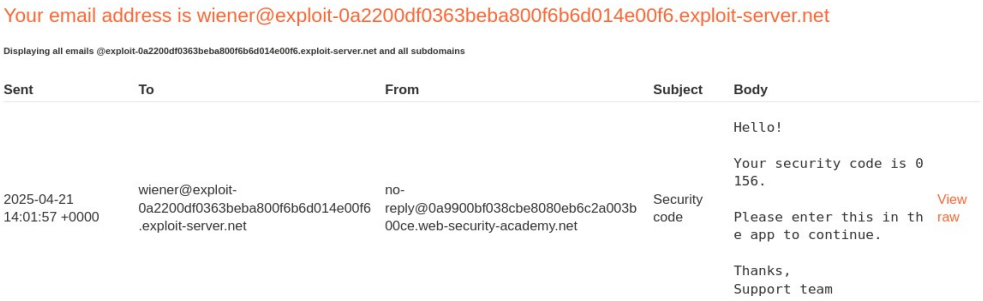
Les identifiants de la victime sont : carlos:montoya

Travaux préparatoires :

1- Connectez-vous avec le compte attaquant dont l'identifiant est **wiener**, puis lancez BurpSuite en veillant à ce que le mode Intercept soit désactivé (**Intercept : off**).



2- Ouvrez le lien vers votre **boîte mail** (Email client) pour consulter le **code de vérification** envoyé après la saisie de votre mot de passe, dans le cadre de la **double authentification**.

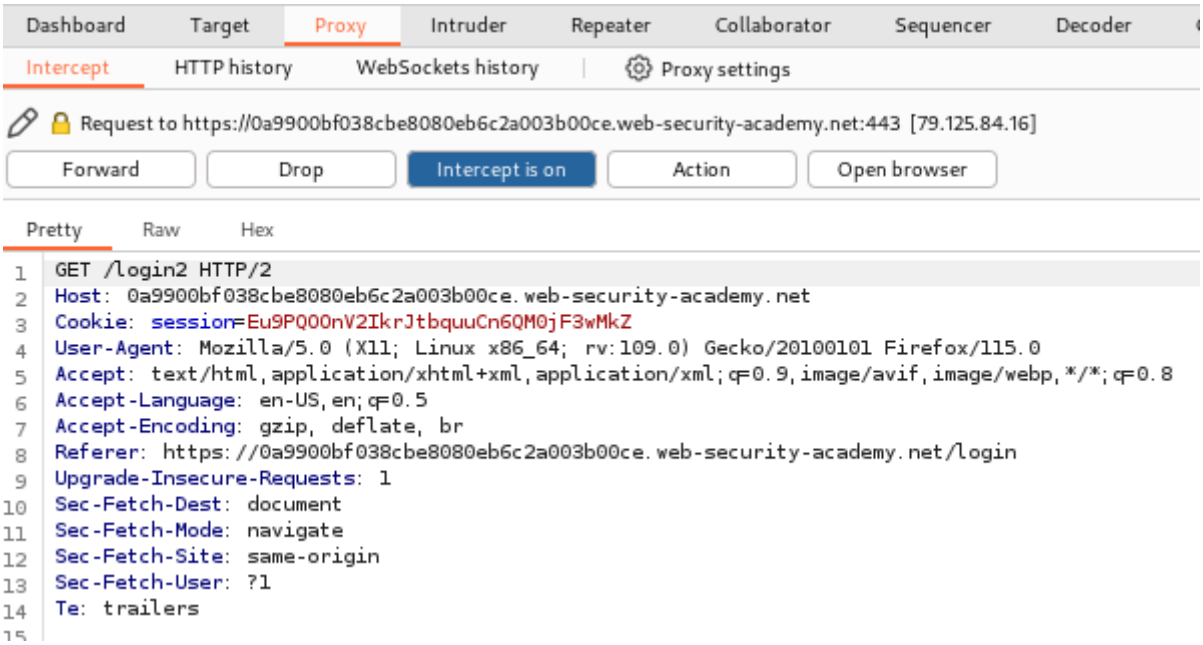


3 - Copiez le code reçu par mail et utilisez-le pour finaliser l'authentification. En analysant la requête dans BurpSuite, on observe que la validation du code modifie la fin de l'URL, qui devient alors **/my-account** en cas de succès. Il peut donc être pertinent de **tester manuellement une modification de l'URL**, en remplaçant sa fin par **/my-account**, après avoir validé la première étape de l'authentification (identifiant et mot de passe) sur le **compte de la victime**.

4- Déconnectez-vous du **compte wiener** en vous déauthentifiant depuis l'interface du site.

Réalisation de l'attaque :

1 - Activez le mode Intercept dans BurpSuite (**Intercept : on**). Connectez-vous ensuite avec le compte de Carlos, puis cliquez sur le bouton **Forward** dans BurpSuite pour laisser passer la requête.



2- Cliquer sur le bouton **Drop** pour annuler la requête. Un message d'erreur s'affiche (ERROR : request was dropped by user) :

3- Dans l'URL de la transaction, remplacez **/login2** par **/my-account**, puis appuyez sur ENTRÉE pour exécuter la requête avec l'URL modifiée. Ensuite, dans BurpSuite, désactivez l'interception et revenez à la page d'authentification. Le défi est alors réalisé.

Défi n°3 : Réinitialisation malveillante d'un mot de passe

Le lien permettant d'accéder au labo est le suivant :

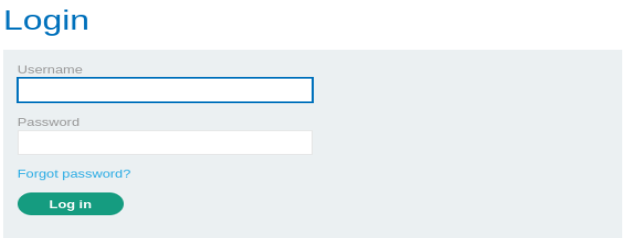
https://portswigger.net/web-security/learning-paths/authentication-vulnerabilities/vulnerabilities-in-other-authentication-mechanisms/authentication/other-mechanisms/lab-password-reset-broken-logic

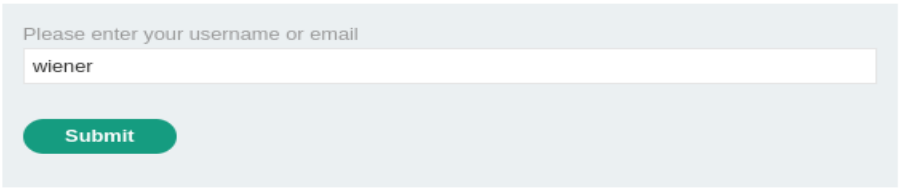
La fonctionnalité de réinitialisation du mot de passe de ce laboratoire est vulnérable. Pour résoudre le défi, il va falloir réinitialiser le mot de passe de l'utilisateur Carlos, puis se connecter et accéder à sa page via le lien intitulé « mon compte ».

Vos identifiants en tant qu'attaquant : wiener/peter
Login de la victime : carlos

Travaux préparatoires :

1- Accédez à la page d'accueil du défi, puis cliquez sur le lien **My account** en haut à droite. Une fois que la page d'authentification s'affiche, cliquez sur le lien **Forgot password** et saisissez votre identifiant d'attaquant dans le formulaire qui apparaît, sans valider.

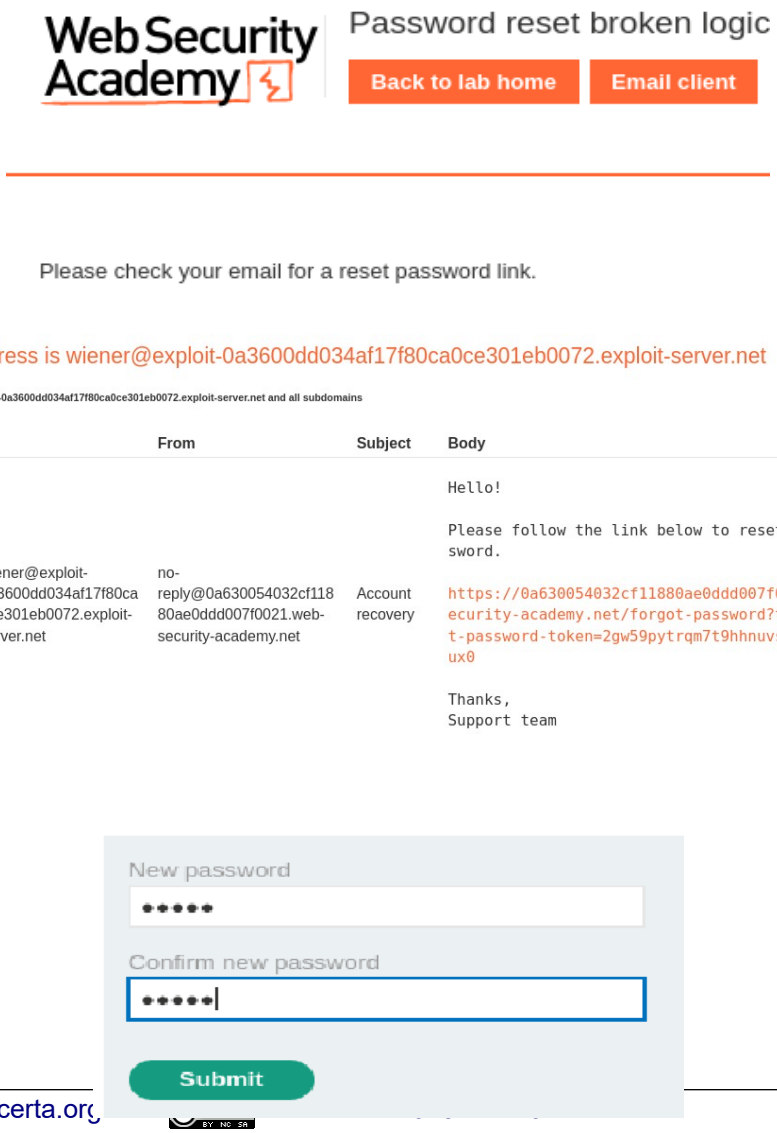




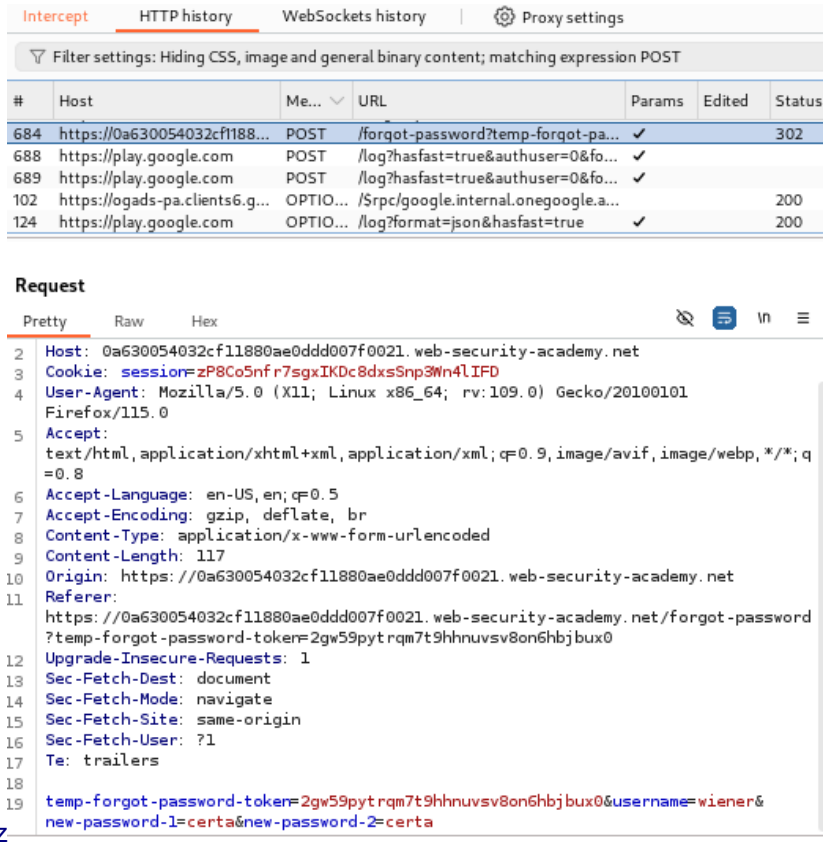
2- Activez le proxy de BurpSuite (**Intercept : on**'), puis confirmez en cliquant sur **Submit**.

Réalisation de l'attaque :

1- Suivez la procédure de réinitialisation du mot de passe en le définissant à la valeur souhaitée. Confirmez chaque étape en cliquant sur le bouton **Forward** de Burp.

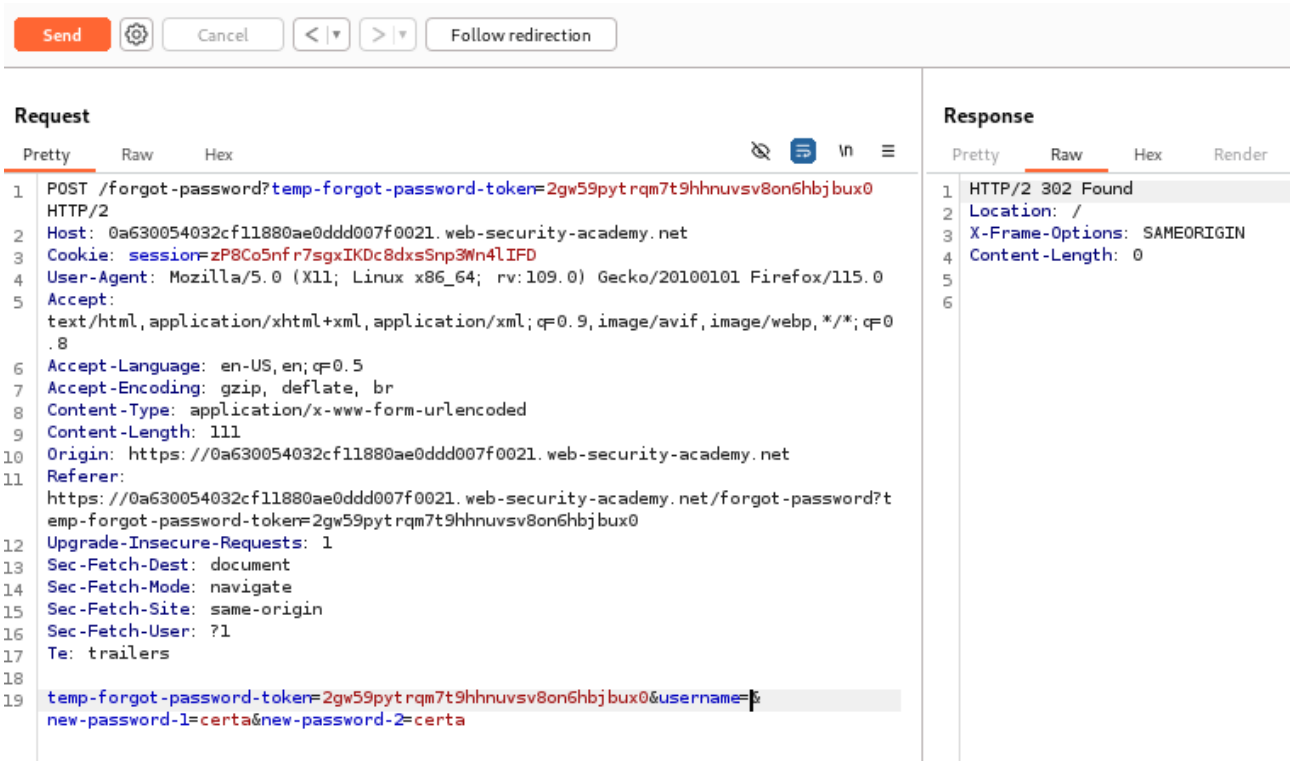


2- Une fois la procédure de réinitialisation du mot de passe terminée, BurpSuite a enregistré toutes les requêtes envoyées au serveur, ce qui permet de les analyser. Dans Burp, accédez à Proxy > HTTP history pour examiner ces requêtes. Recherchez une requête POST associée au changement de mot de passe, incluant le nouveau mot de passe que vous avez saisi.



3- Sélectionnez cette requête puis l'envoyer au répéteur de Burp Suite.

4- Sur la page du répéteur, laissez la variable **username** vide, puis comparez le résultat avec celui obtenu lorsque la variable contient une valeur. On remarque que le comportement de la page de retour reste identique. Cela indique que le serveur ne vérifie pas le contenu de la variable transmise dans la requête POST.



5- Répétez les étapes permettant à l'attaquant de réinitialiser son mot de passe en capturant les requêtes avec Burp Suite. Ensuite, envoyez la requête POST de réinitialisation du mot de passe vers le répéteur et remplacez le contenu de la variable **username** par celui de la victime. Modifiiez la requpete pour réinitialiser le mot de passe de la victime, puis connectez-vous à son compte. Le défi est alors réalisé.