5ATTACK - 2024-2025

Projet - Résultats

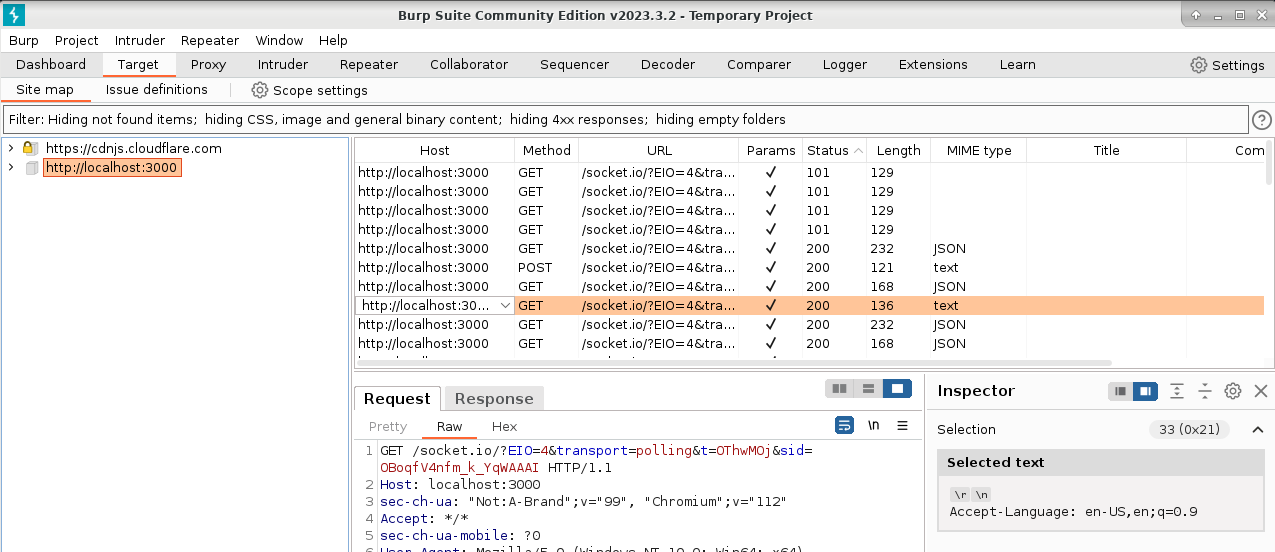
1. Avertissement

Rappels :

* Vous ne devez pas utiliser d’outils de test d’intrusion sur les moyens informatiques fournis par l’école ou sur ceux des entreprises pour qui vous travaillez. Plus généralement, vous ne devez pas utiliser ce type d’outil sans l’autorisation explicite des propriétaires des moyens que vous utilisez ou que vous ciblez.
* Il est recommandé d’effectuer tous vos travaux dans un environnement virtualisé dédié, non privilégié (i.e. sans droits d’administration sur la machine hôte), et de préférence avec un utilisateur banalisé (i.e. ni ‘root’ sur Linux, ni ‘Administrateur’ ou équivalent[[1]](#footnote-1) sur Windows).

1. Parcours
   1. Objectif 1 : mise en place de l’environnement d’expérimentation

* Installer un environnement containerisé (privilégier l’utilisation de Podman) ou virtualisé[[2]](#footnote-2) (privilégier l’utilisation de [VirtualBox](https://www.virtualbox.org/)), sur Windows ou sur Linux, avec :
* un environnement OWASP Juice Shop, utilisé comme cible d’attaque :
* télécharger le container Juice Shop : « podman pull bkimminich/juice-shop » ;
* démarrer l’application : « podman run --rm -p 3000:3000 bkimminich/juice-shop »;
* lire toute les pages de la section « [Hacking Preparations](https://help.owasp-juice.shop/part1/) », et en particulier la page « [Walking the happy path](https://help.owasp-juice.shop/part1/happy-path.html) » ;
* créer un compte (local) avec une adresse mail et un mot de passe **parfaitement anonymes** (utilisés pour des besoins de test uniquement ; ne pas utiliser de références personnelles, de comptes existants ou de mot de passe personnels) ;
* de l’outillage d’attaque et d’investigation (vous pouvez utiliser vos résultats de l’exercice 1) :
* installer « Burp Suite Community Edition » (qui contient entre autres choses un proxy Web permettant d’intercepter et au besoin modifier les requêtes et les réponses échangées par un client et un serveur Web ; d’autres outils sont disponibles dans la suite, dont un répéteur HTTP[[3]](#footnote-3) et un scanner) : [Installing Burp Suite Professional / Community Edition - PortSwigger](https://portswigger.net/burp/documentation/desktop/getting-started/download-and-install) ;
* lancer Burp Suite avec un projet temporaire et les paramètres par défaut ;
* sélectionner l’onglet « Dashboard » et vérifier que le proxy est à l’écoute du port 8080 ;
* sélectionner l’onglet « Target » et ouvrir le navigateur intégré de Burp Suite ; depuis le navigateur de Burp Suite, accéder à Juice Shop : <http://localhost:3000> ; consulter les premières URL collectées par Burp Suite pour confirmer que le log des échanges est activé.



* Retranscrire ici une copie d’écran montrant la page d’accueil de Juice Shop dans votre VM :

<A compléter>

* Retranscrire ici une copie d’écran montrant une instance de Burp Suite en fonctionnement, avec une première capture de trafic vers Juice Shop :

<A compléter>

* Prendre connaissance des principales fonctionnalités de Burp Suite, et en particulier les onglets :
* « Proxy » : interception / modification (optionnelle) / transmission[[4]](#footnote-4) de requête ; utiliser une requête déjà capturée pour effectuer quelques tests ;
* « Decoder » : essayer de décoder une chaîne de caractères encodée en Base64 (p.ex. « Qm9uam91ciBsJ0VTVElBTSAh ») ;
* « Logger » : équivalent de l’onglet « Target », mais toutes cibles confondues, et avec une présentation simultanée de la requête et de la réponse ;
* « Intruder » : génération de requêtes à partir d’une requête de base et enchaînement d’attaques.
* Prendre connaissance des principales fonctionnalités offertes par le site Web à attaquer en créant un compte, en parcourant le catalogue, en commandant des items, etc.
  1. Objectif 2 : découvrir le lien qui mène au « score board »

L’implémentation de Juice Shop est instrumentée pour mettre à jour un « tableau des scores » reflétant les progrès de l’attaquant dans ses travaux. Ce premier exercice consiste à trouver le lien permettant de retrouver le lien conduisant à ce tableau des scores.

Conseils :

* Analyser le code source de la page principale de Juice Shop (et en particulier le code JavaScript).

Retranscrire ici une copie d’écran montrant le tableau de scores récupéré ; décrire / commenter l’attaque :

<A compléter>

* 1. Objectif 3 : Injection

Il s’agit ici de réaliser une injection SQL et d’obtenir une escalade de privilèges jusqu’au niveau administrateur, en faisant quelques hypothèses sur l’architecture de la solution.

Conseils :

* Essayer de provoquer une erreur d’authentification.
* Utiliser Burp Suite pour analyser le trafic d’authentification.
* Utliser les outils de Burp Suite (en particulier l’onglet « Repeater ») pour modifier et soumettre des requêtes préalablement capturées et observer les réponses de l’application.
* Trouver une injection permettant d’obtenir un accès à la base des données d’authentification (intéressez-vous alors à la première entrée de la base).
* Tenter une authentification avec les données récupérées.

Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

* 1. Objectif 4 : Broken Authentication

Il s’agit cette fois-ci de prendre l’identité d’un utilisateur (Jim) en utilisant le mécanisme de réinitialisation de mot de passe.

Conseils :

* Créer un utilisateur et demander la réinitialisation de son mot de passe. Analyser les commandes et les réponses correspondantes. Analyser en particulier ce qui se passe lorsque la réponse à la question posée n’est pas la bonne.
* Réaliser l’attaque avec l’adresse email de Jim comme « login ».
* Pour retrouver la réponse à la question posée, utiliser une technique OSINT (Open Source Intelligence) basée sur les informations disponibles sur Jim, ou une attaque dictionnaire (notamment : une liste de prénoms).
* Utiliser l’onglet « Intruder » pour modifier / générer des requêtes avec le module « Sniper » ciblant le champ dédié à la réponse de sécurité.

Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

* 1. Objectif 5 : Sensitive Data Exposure

On se donne pour objectif de récupérer un fichier sensible (acquisitions.md) en analysant et utilisant les URLs utilisées par le site.

Conseils :

* Rechercher des informations (URLs, contenu de fichiers de configuration) dans les pages accessibles sans authentification.

Remarques :

* Eviter d’utiliser des outils d’attaque en force brute, tels que DirBuster ou GoBuster, qui ont trop d’effet de bord sur le site Web et pourraient (sur un cas réel) conduire à une détection prématurée et à un bannissement.

Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

* 1. Objectif 6 : XML External Entities (XXE)

On se donne pour objectif de récupérer un fichier sensible, notamment « /etc/passwd » en utilisant une attaque XXE.

Conseils :

* Construire un fichier XML ciblant le fichier recherché.
* Repérer dans la page principale un point d’entrée permettant de transmettre le fichier XML forgé précédemment.

Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

* 1. Objectif 7 : Improper Input Validation

On souhaite désormais manipuler le contenu d’une commande pour « obtenir plus en payant moins ». Il faut pour cela comprendre comment est construit un panier de courses et comment sont vérifiées les données permettant de construire ce panier.

Conseils :

* Analyser les requêtes utilisées pour passer des commandes ; comprendre comment est construit un panier de courses et comment sont vérifiées les données permettant de construire ce panier.
* Essayer de modifier le paramètre utilisé pour définir un panier et le valider.

Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

* 1. Objectif 8 : Broken Access Control

On souhaite afficher le panier d’autre utilisateur en utilisant une faiblesse du système de contrôle d’accès qui restreint ce que chaque utilisateur (authentifié ou non) peut faire et voir sur le site Web.

Conseils :

* Analyser la page qui permet d’afficher le contenu d’un panier.
* Repérer et analyser l’utilisation de l‘entête http ‘If-None-Match’.
* Forger une requête permettant d’afficher le panier d’un utilisateur connu.

Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

* 1. Objectif 9 : Security Misconfiguration

On souhaite cette fois-ci utiliser une optionqui n’est plus censée être accessible, typiquement l’envoi d’un document au format XML, alors que le site ne souhaite recevoir que des documents PDF ou ZIP.

Conseils :

* Vous connecter.
* Sélectionner l’option permettant de se plaindre auprès du site.
* Analyser la fonction qui permet d’envoyer une facture. En principe, elle ne permet d’envoyer que des PDF ou des ZIP.
* Consulter le script « main.js » : quel autre format de fichier permet-il d’envoyer ?
* Essayer d’envoyer un fichier autre qu’un PDF ou un ZIP. Si cela fonctionne, vous devriez avoir le message suivant dans la console JavaScript de votre navigateur : « B2B customer complaints via file upload have been deprecated for security reasons! » .

Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

1. Objectifs bonus
   1. Objectif 10 : Cross Site Scripting (XSS)

Il s’agit ici d’implémenter une attaque « reflected XSS » : l’attaque est injectée via le navigateur, traitée par le serveur, puis retournée au client pour attaquer celui-ci.

Conseils :

* Vous connecter.
* Effectuer quelques achats.
* Consulter l’historique des achats.
* Consulter l’icône de livraison (un petit camion) pour n’importe quelle commande ; cela vous permet d’accéder au statut de la commande.
* Quelle information affichée à l’écran se trouve aussi dans l’URL ?
* Faire l’hypothèse que l’URL est construite par le navigateur, qu’une partie de son contenu est extraite de l’URL par le serveur pour construire une nouvelle page qui est retournée et affichée par le navigateur.
* Sur cette base, essayer de construire une URL qui contient un bout de script (p.ex. «  <iframe src="javascript:alert(`xss`)"> ») qui va être extrait puis retourné au navigateur pour y être exécuté.
* Retranscrire ici une copie d’écran du nouvel état du tableau des scores et décrire / commenter l’attaque :

<A compléter>

Fin de document

1. Attention : beaucoup de compte Windows sont créés à l’installation avec des droits ‘Administrateur’. Il faut créer un compte local (i.e. sans authentification Microsoft) qui ne dispose d’aucun droit d’administration particulier (i.e. il ne fait pas partie du groupe des administrateurs de la machine). [↑](#footnote-ref-1)
2. L’utilisation d’un environnement virtualisé est recommandé pour isoler les effets de bord des travaux d’expérimentation proposés et éviter de « polluer » un environnement de travail existant. [↑](#footnote-ref-2)
3. Une version simplifiée est fournie dans la version “Community Edition”. [↑](#footnote-ref-3)
4. “Forward”. [↑](#footnote-ref-4)