

Atelier N° :3 Audit des Applications Web

Ahmed Serhrouchni

Objectifs :

Maîtrise des outils d'audit Black Box (audit des sites web côté client)

Utiliser les outils d'audit White Box (audit de code statique côté serveur)

Outils :

Black Box : Acunetix, ZAP










White Box: RIPS, Yasca

Applications Web Vulnérable: DVWA et WebGoat

Audit des sites Web

I. Acunetix : audit XSS sur DVWA :

Le but de cet exercice est de démontrer les différentes étapes à réaliser pour mener une opération d'audit réussie. Au-delà de sa réputation, nous avons choisi d'utiliser l'outil Acunetix uniquement dans le but d'exploiter au mieux les étapes d'audit d'un site web. En effet, Acunetix est une solution d'audit de sites web commerciale. Elle intègre les briques d'audit habituelles (XSS, SQLi, CSRF, Rfi, LFi,...) et un ensemble d'outils qui aident à l'affinement des opérations d'exploration, de test et d'exploitation des données par l'auditeur :

Tools	
 Site Crawler	Maps a web site by tracing all references links and gathering information about every discovered site file (such as scripts with inputs with possible values, file structure)
 Target Finder	Discovers any HTTP/S servers on an IP range, with the option to scan identified servers for vulnerabilities
 Domain Scanner	Identifies new and unlisted subdomains from a higher-level domain, with the option to scan discovered entities for vulnerabilities
 Blind SQL Injector	Automatically exploits SQL injection to extract data from the database used on the server
 HTTP Editor	Enables construction of custom HTTP/S requests for analysis of the server's response
 HTTP Sniffer	HTTP proxy for logging, capturing, and modification of all intercepted HTTP/HTTPS traffic
 HTTP Fuzzer	Tests for vulnerabilities like buffer overflows and input validation by fuzzing request headers and parameters
 Authentication Tester	Tests password strength by performing dictionary-based attacks on basic HTTP, NTLM, or form based authentication methods
 Compare Results	Compares saved scan results and displays differences

I.1 Installation de l'outil Acunetix

Dans le dossier Tools, vous disposez d'un fichier nommé (Accunetix_vulnerabilityscanner.exe).

Installer le programme en mode démo sous Windows.

I.2 Préparation de la cible

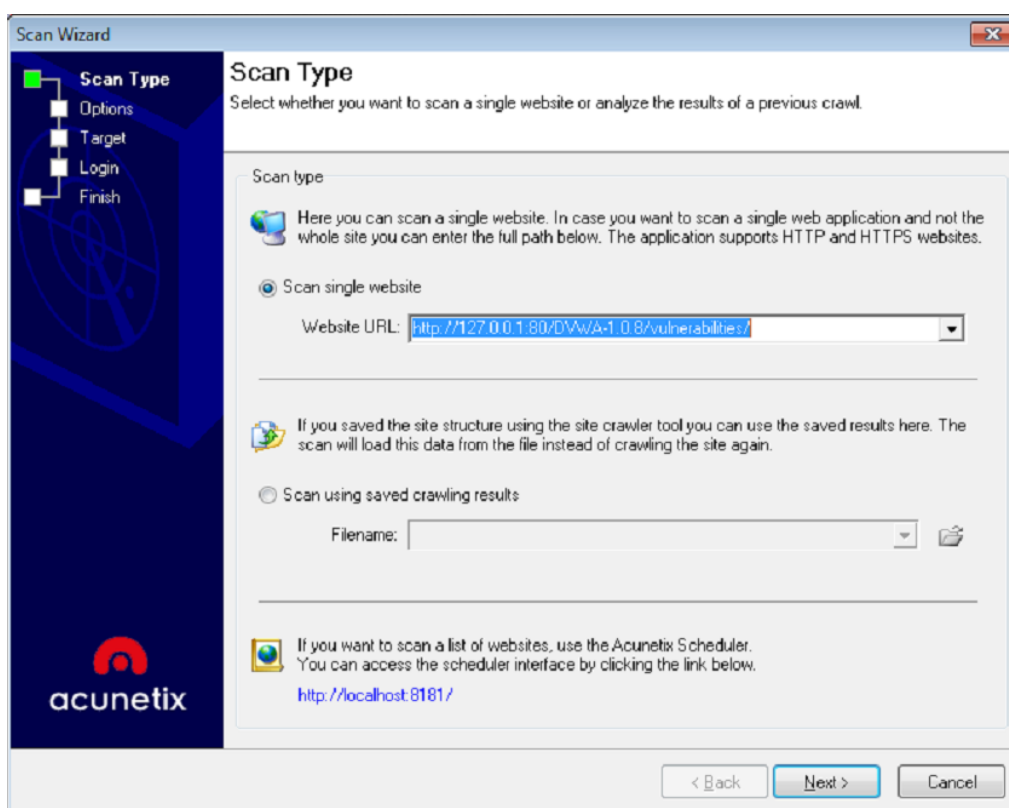
Nous allons utiliser l'application web vulnérable précédemment déployée sur le serveur web (XAMPP).

Assurez-vous que DVWA est en ligne sur le serveur Apache local : <http://localhost/DVWA>

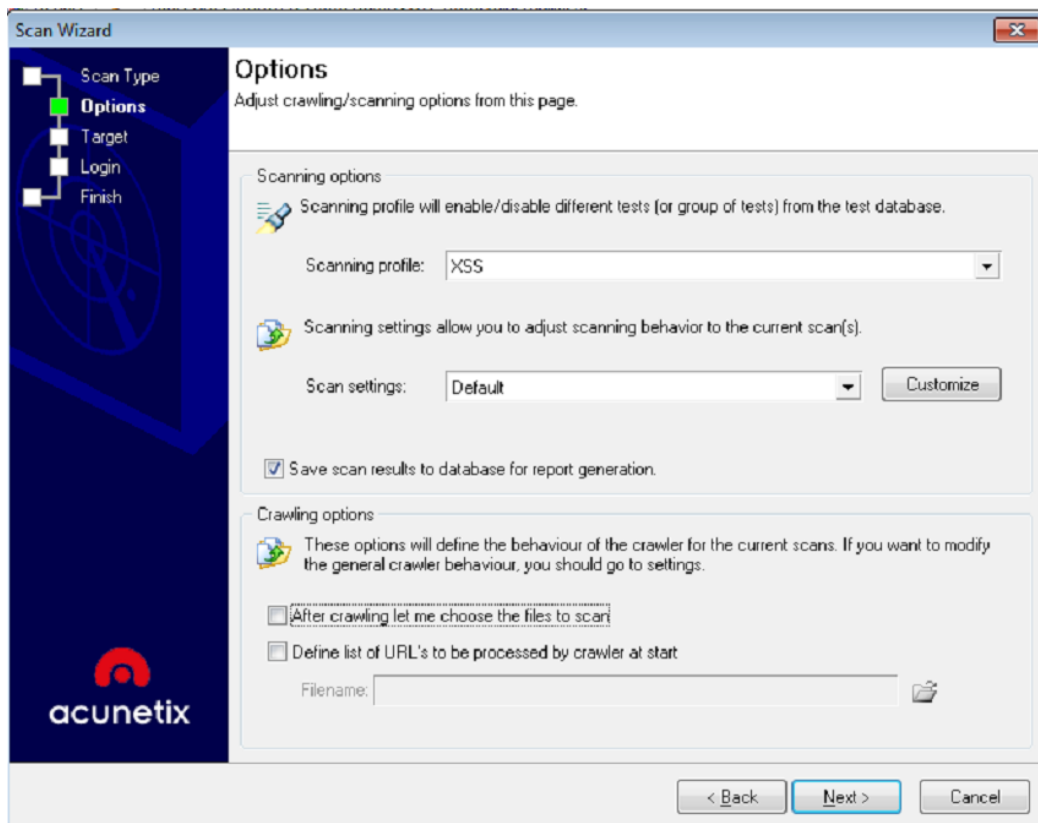
I.3 L'audit du site DVWA :

Pour lancer l'opération d'audit de notre application web (DVWA), Ouvrez l'application Acunetix et procédez comme suit :

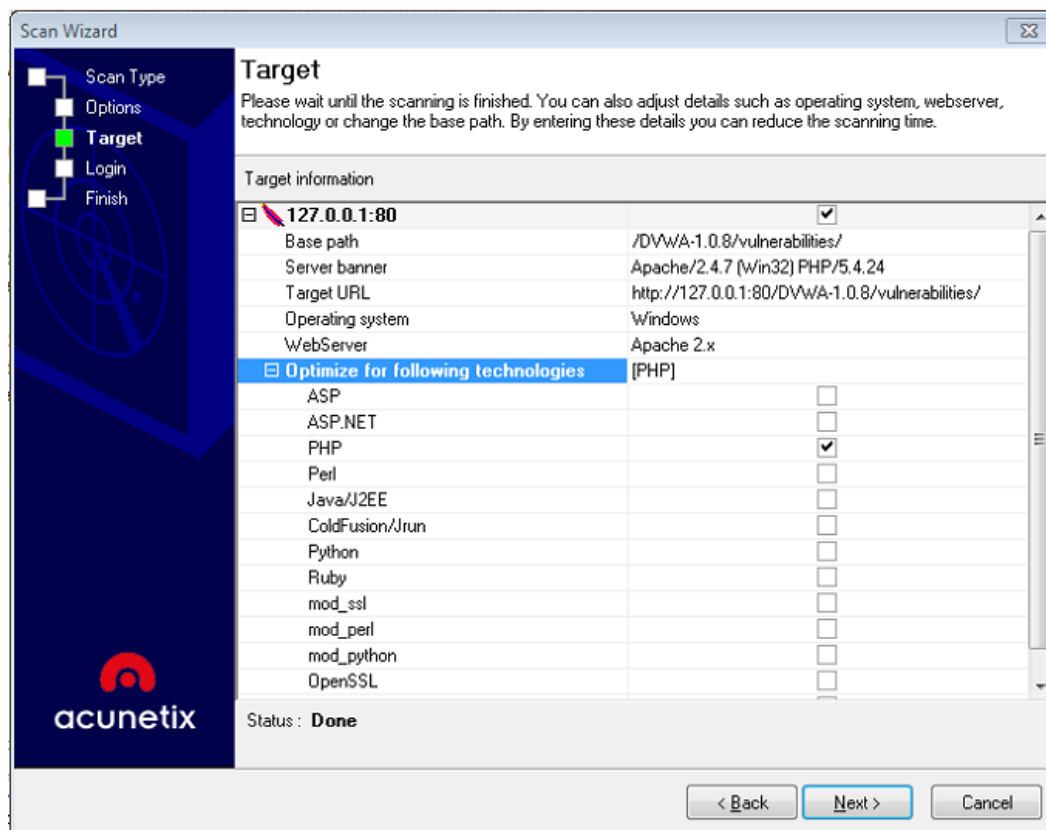
- **File > New > New Website** Scan pour commencer, ou cliquez sur le bouton « **New Scan** »
- Analysez un seul site web « Scan a single website »: Entrez l'URL de la cible, par exemple : <http://127.0.0.1/DVWA/vulnerabilities/>



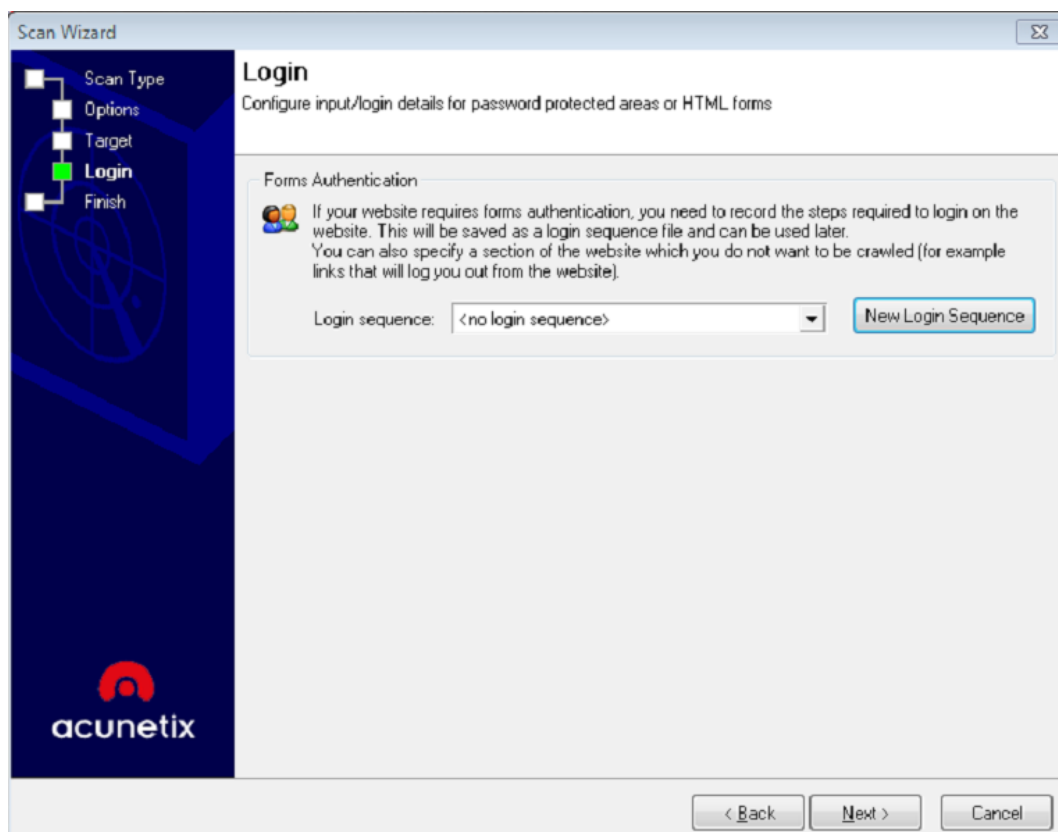
- On peut aussi utiliser les résultats enregistrés pour lancer une analyse au lieu de faire à nouveau la phase de « **Crawling** » du site, ce qui permet de gagner un temps précieux. Cliquer « **Next** » pour continuer.
- Ecran 2 - Choisir le profil d'analyse, les paramètres d'analyses et les options d'exploration « **Crawling Options** »



- Le profil d'analyse détermine les tests à être appliqué sur la cible. Par exemple : nous voulons tester seulement la faille Cross Site Scripting dans cette étape, nous choisissons donc « XSS » comme profil d'analyse.
- Paramètres d'analyse « Scan Settings » : Les paramètres d'analyse déterminent comment les phases d'exploration « Crawling » et de l'analyse « scanning » vont être utilisées pendant l'analyse. Les options d'exploration 'Crawling Options': Permet de choisir les fichiers à analyser au lieu d'analyser tous les fichiers de l'application après la phase d'exploration « Crawling » et de choisir par quelle URL Acunetix commence la phase de « Crawling ». Dans notre cas on prend les paramètres par défaut.
- **Ecran 3** Confirmer la cible et la technologie utilisée: Pour une analyse rapide Acunetix se contente d'analyser les technologies web sélectionnés. Dans notre cas c'est le PHP qui nous intéresse (DVWA est développée en PHP)

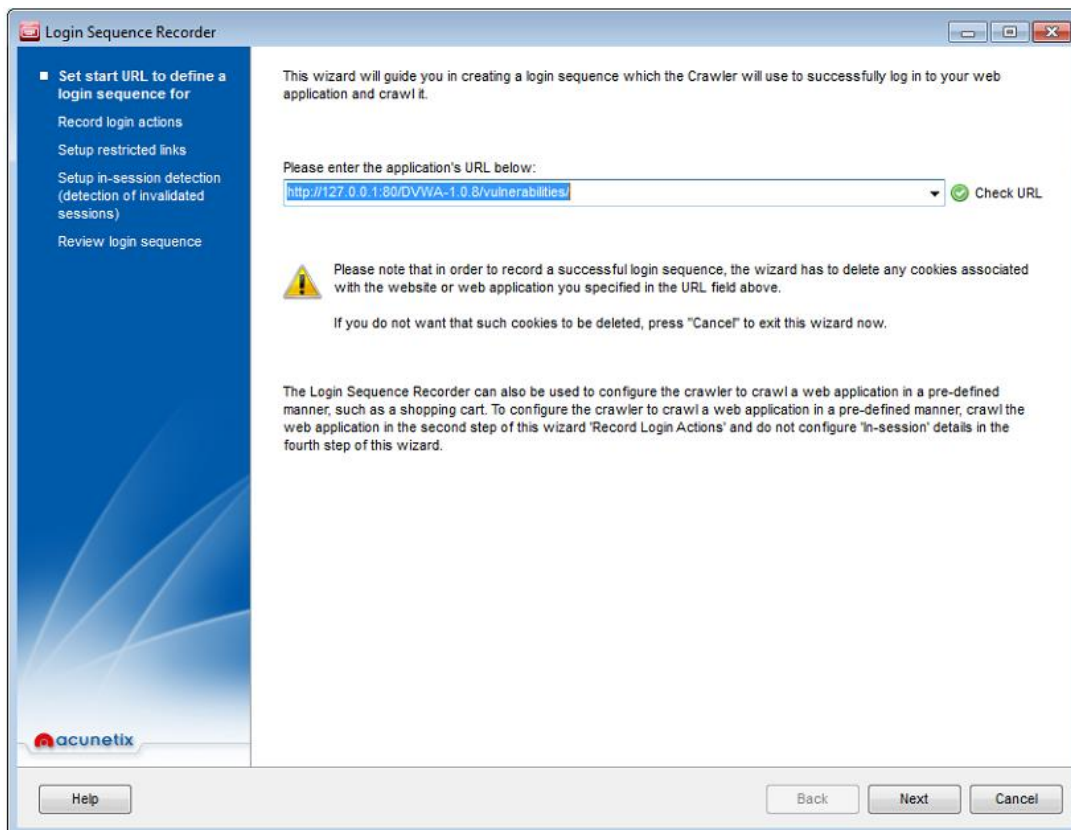


- **Ecran 4** Acunetix détecte automatiquement les demandes d'authentification (Réponse 401) et il propose d'enregistrer une session d'authentification pour explorer toutes les ressources de l'application.



Acunetix peut être configuré pour mettre d'une manière automatique les informations d'authentification dans une 'Web Forms'. Comme ci-dessous : Cliquez 'New Login Séquence' pour enregistrer les informations d'authentification :

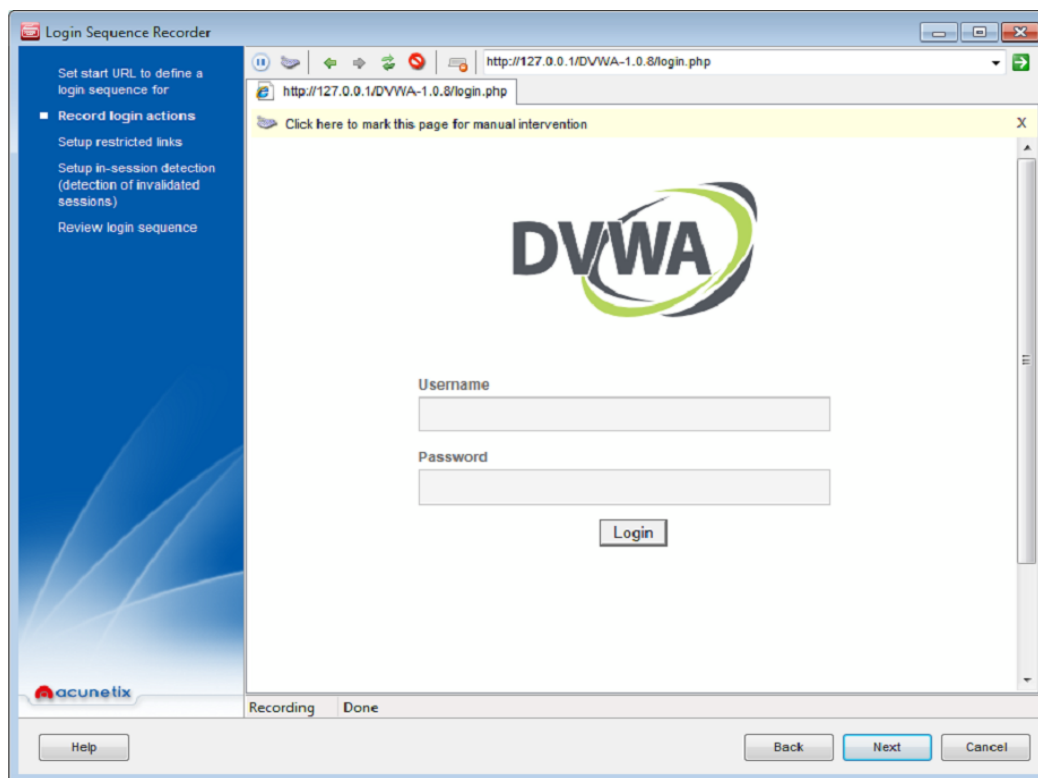
- **Ecran 5** : L'authentification :



Entrez l'URL de l'application web (ou la valider si ok), puis cliquez 'Next'.

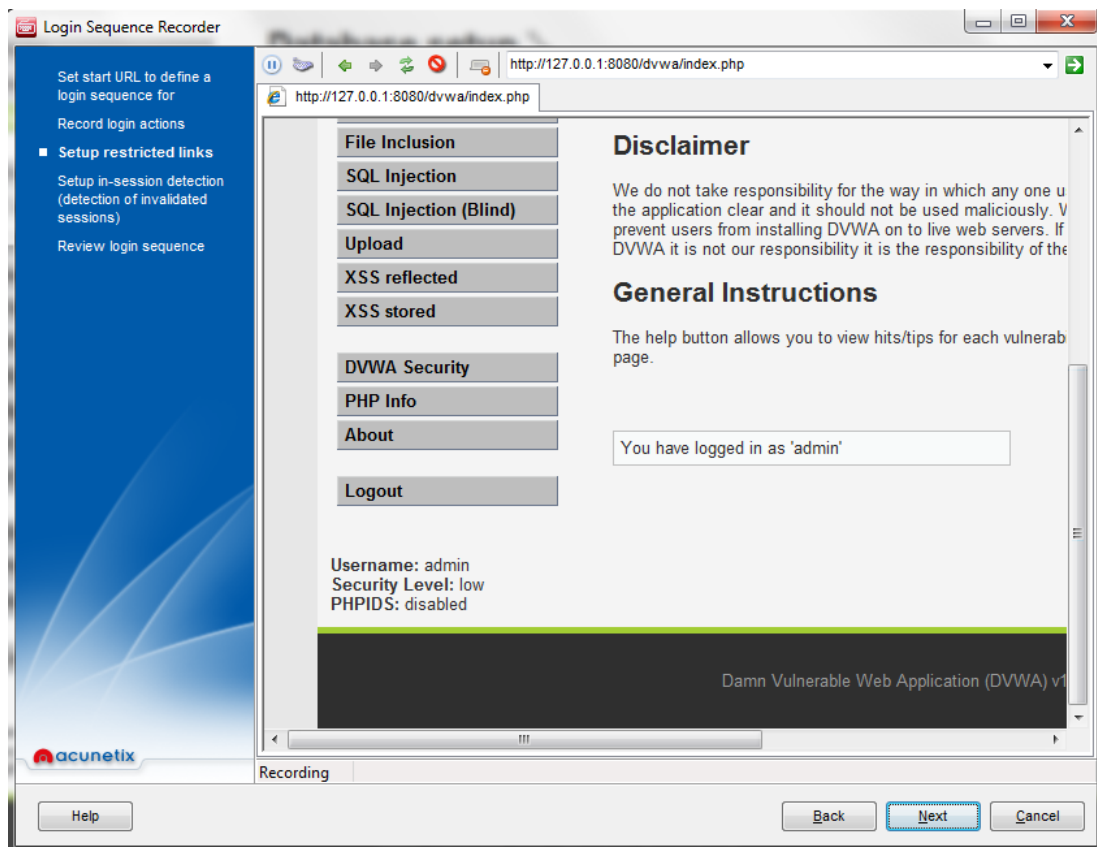
- **Ecran 5** : L'authentification (la suite)

Dans cette étape accédez à la page d'authentification et saisissez les informations pour vous authentifier. Une fois vous êtes authentifié cliquez sur 'Next'.



- **Ecran 6 : L'authentification (la suite) Setup Restricted Links**

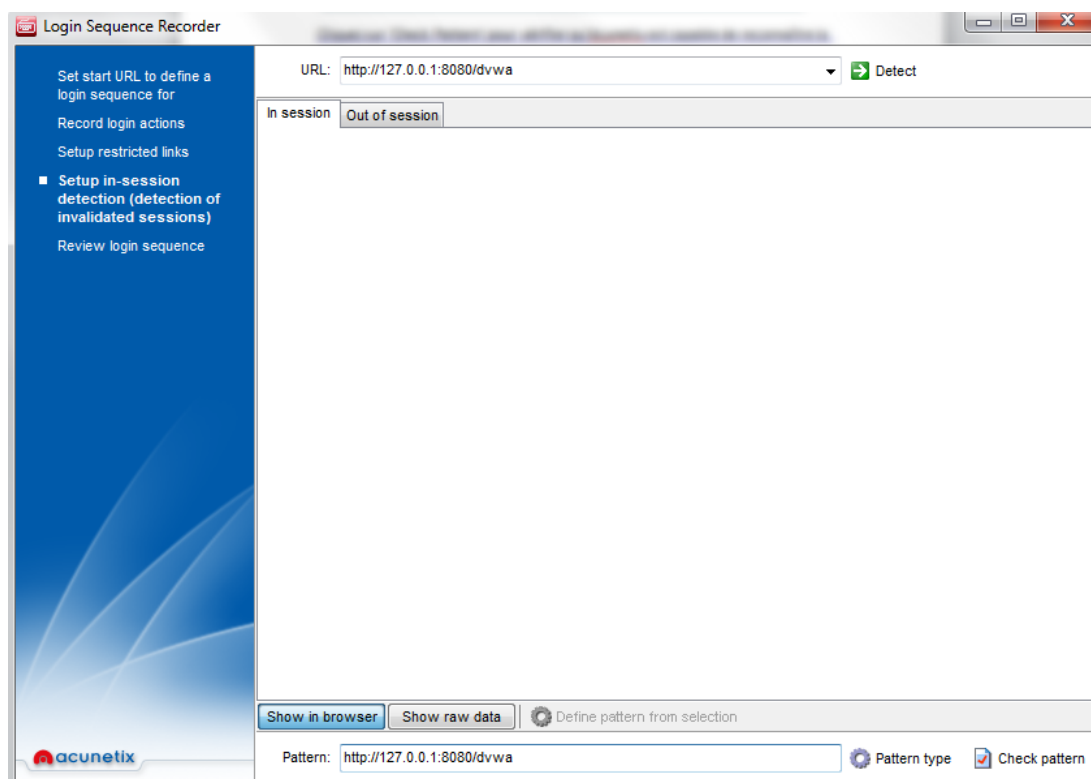
A cette étape il faut indiquer à Acunetix d'ignorer le bouton de déconnexion 'Logout' lors de la phase d'exploration « Crawling » et ainsi éviter une fermeture brutale de la session, ce qui évite de tronquer la phase de « crawling ». Si le bouton de déconnexion 'Logout' n'est pas dans la même page, cliquez sur le bouton Pause, accédez à la page où se trouve le bouton de déconnexion. On ignore et on continue avec Next.



- **Ecran 7 : L'authentification (la suite) Setup In-Session**

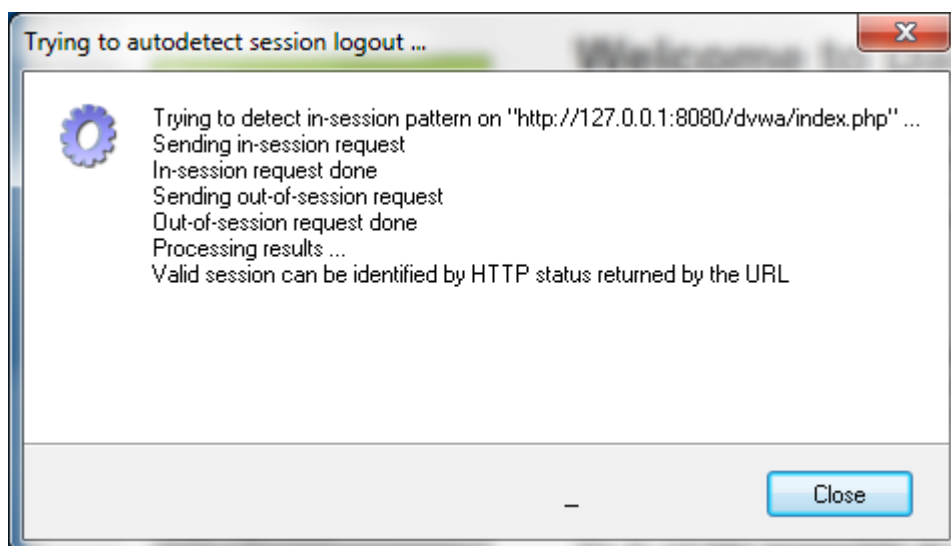
Maintenant il faut spécifier les pages de détection 'In Session' ou 'Out of Session', pour permettre à Acunetix de détecter si la session est toujours valide ou pas. Si, pour une raison quelconque, la session expire pendant la phase d'exploration 'Crawling' Acunetix va automatiquement se connecter à nouveau. Il faut saisir un URL et cliquer sur '**Detect**' pour permettre à Acunetix de détecter la page valide par exemple: <http://127.0.0.1:8080/dvwa/index.php>

Cela va permettre à Accunetix de reconnaître une session-In (valide)



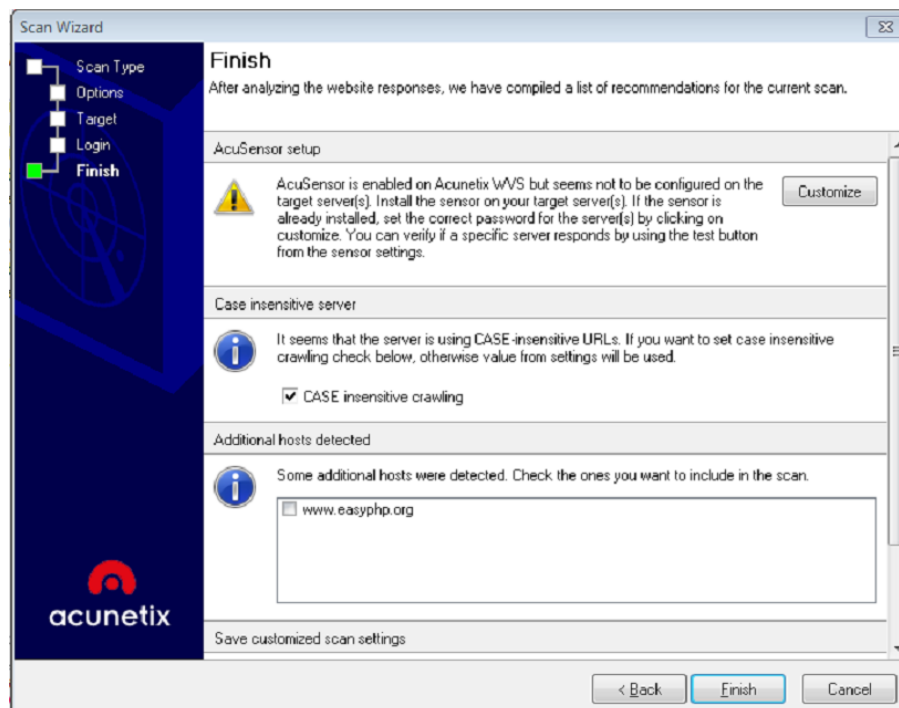
- **Ecran 8 : L'authentification (la suite) Setup In-Session**

On cliquant sur le bouton check pattern on pourra recevoir le pop-up suivant :



Cliquez 'Next' une fois cette étape fini. Vous pouvez voir la séquence enregistrée. Cliquez sur 'Finish' pour finir cette étape.

- **Ecran 9 – Finaliser les options d’audit :**



- Analysez les résultats de cet audit.

II. ZAP audit LFI/RFI sur DVWA :

ZAP (Zed Attack Proxy) est un outil Open source qui appartient à la liste des projets adoptés par l'OWASP.

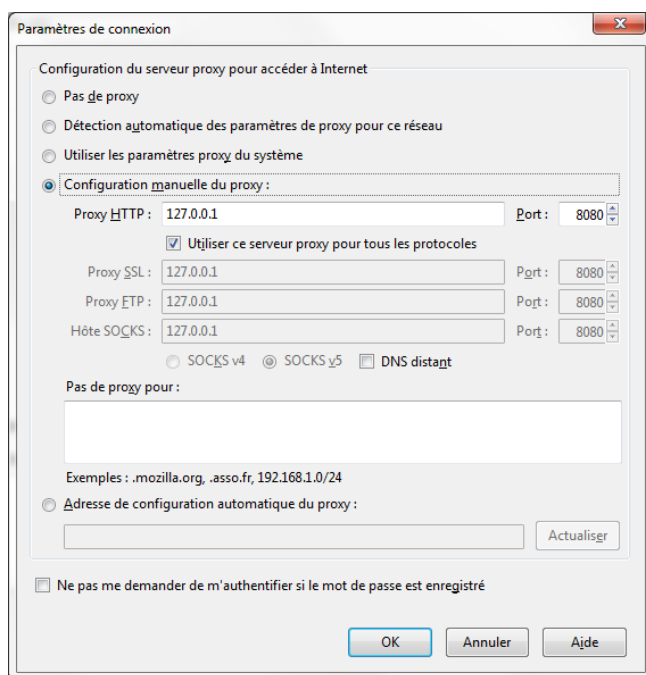
L'intérêt derrière l'utilisation de cet outil, est de présenter une autre manière d'auditer un site web en interceptant le flux http au moyen d'un Proxy. Donc pas besoin d'entrer les identifiants de sécurité (à priori déjà fait par l'utilisateur).

L'exercice consiste à mettre en coupure un Proxy (ZAP) entre l'application web (du côté serveur) et le navigateur web (côté client). L'audit se fera d'une manière semi-automatique en guidant le proxy vers une cible et en lançant l'audit à partir de la racine de cette cible.

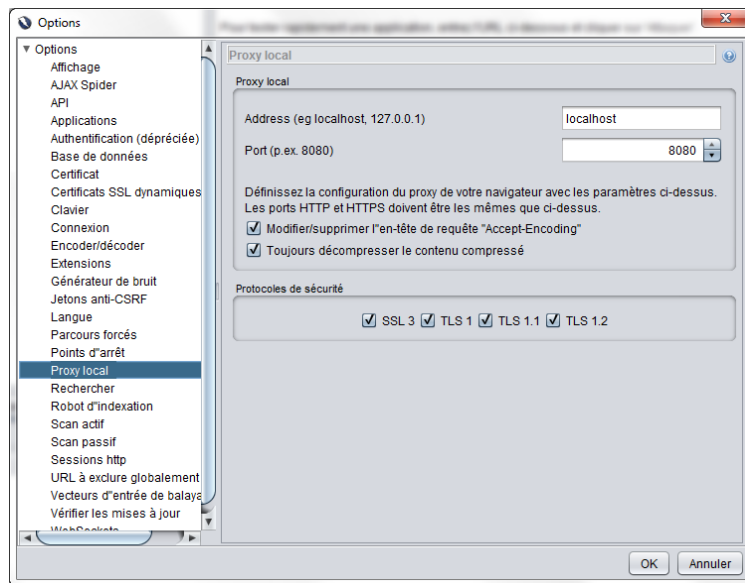
Pour utiliser cet outil, vous disposez d'un package d'installation dans le dossier tools (ZAP_2.3.1_Windows.exe). Installez et lancez le programme sous Windows.

II.1 Configuration du client web (Firefox)

Dans : Option → avancé → Réseau → Paramètres → Configuration manuelle du Proxy

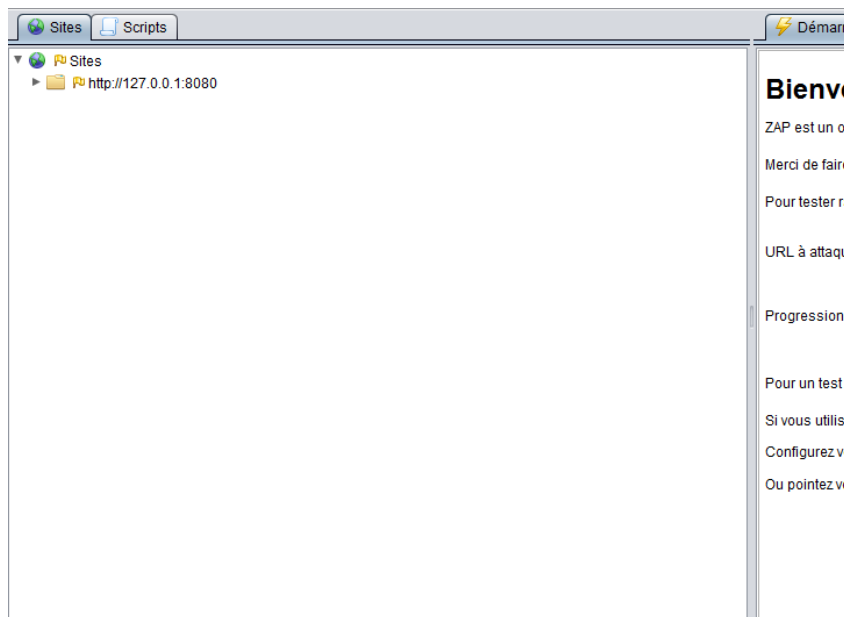


Si le port 8080 de ZAP est utilisé par une autre application (serveur web) on peut le changer dans : Outils → Option → Proxy local

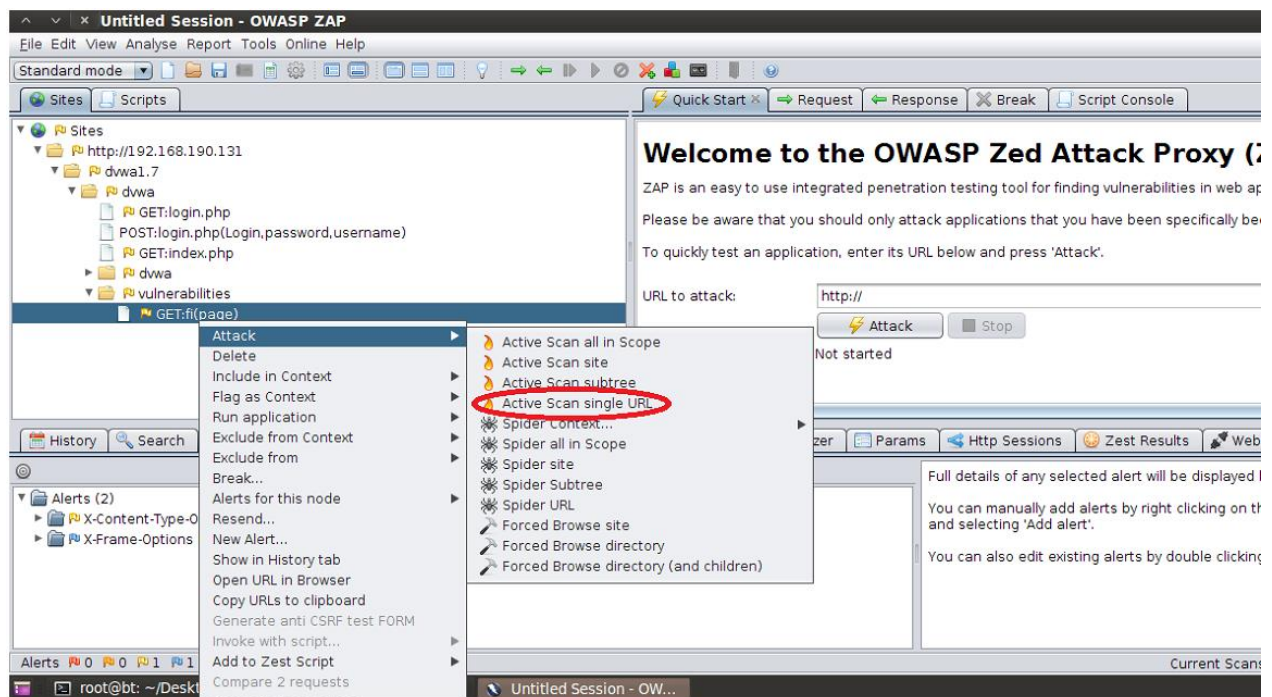


Détection des vulnérabilités LFI et RFI dans ZAP

Test avec niveau faible de DVWA Security : Après avoir paramétré ZAP en tant que proxy et le navigateur en tant que client qui passe par ce proxy ZAP, on accède notre application web via le navigateur (Login -> paramétrer DVWA Security=LOW). ZAP va alors automatiquement enregistrer les trafics (Voir le diagramme ce dessous)



Au niveau du navigateur web maintenant (Firefox), on navigue sur la page cible **file inclusion** (<http://127.0.0.1:8080/dvwa/dvwa/vulnerabilities/fi/?page=include.php>), qui est vulnérable aux attaques LFI et RFI, au même moment ce trafic est capturé par ZAP. On parcourt alors, la fenêtre 'site' de ZAP, pour localiser notre page cible. Une fois trouvée, on lance une attaque ciblée que sur cette page :



- Vous remarquez que l'outil lance un ensemble de requêtes d'attaques visibles dans la fenêtre 'scan actif'. Vous pouvez trouver les résultats dans l'onglet 'Alertes'.
- Lancez l'audit du SQLi (attention : il faut spécifier au moins un paramètre injectable)

Audit de code statique des applications Web

RIPS : Une application Web qui permet l'audit statique (coté serveur) du code PHP uniquement des applications web.

Yasca : une application exécutable qui permet d'auditer du code C, Java, PHP, JSP, ASP...etc. Elle utilise uniquement des plug-ins externes pour scanner chaque langage.

II.1 Auditer DVWA avec RIPS

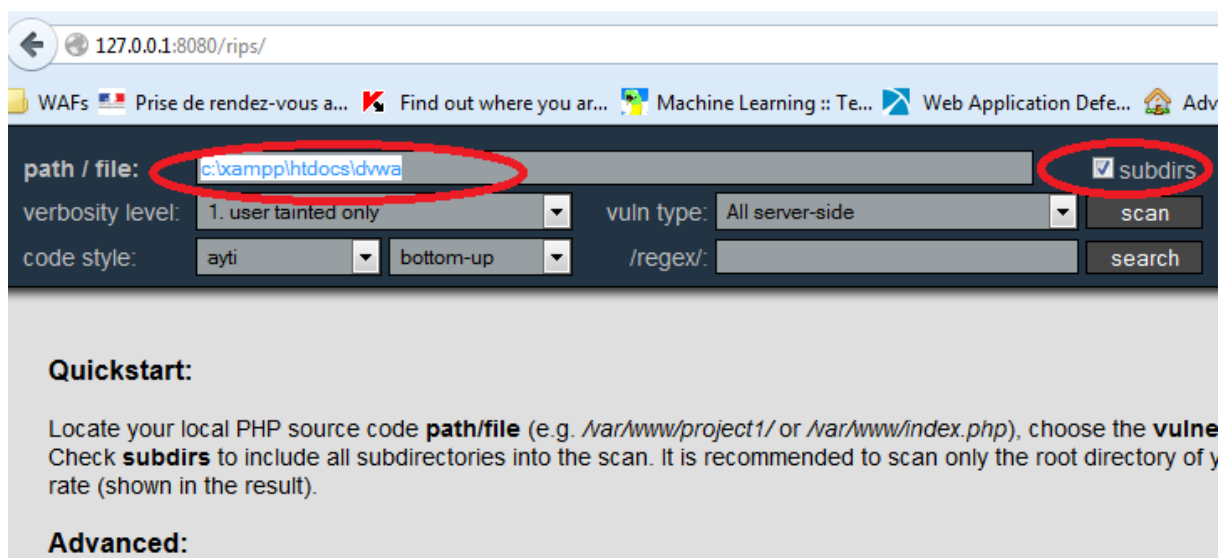
On dispose d'un dossier nommé rips dans le dossier tools

Usage :

Copier le dossier RIPS dans le dossier c:/xampp/htdocs de Xampp précédemment installé

Lancer l'application dans le navigateur Web : <http://localhost:8080/rips>

Donnez le chemin d'accès vers la DVWA dans « **Path / file :** », cochez « **subdir** » et cliquez sur scan.



RIPS inspecte la plupart de vulnérabilités dans le code source grâce notamment à la modélisation en machines à états des différentes fonctions et structures de l'application. Il est ainsi capable de reconnaître une entrée de l'application, d'analyser sa structure de données et la fonction de traitement de cette entrée. Le rapport générée par cet outil est très instructif et contient même des recommandations quant l'usage des bonnes pratiques en termes de codage.

II.1 Auditer WebGoat avec YASCA

Pour utiliser Yasca, on copie le contenu du dossier yasca (de tools) vers la racine du disque c.

Pour faciliter la manipulation, on copie aussi le dossier webgoat (de tools) dans le dossier yasca sur le disque c (c:\yasca\Webgoat).

Pour lancer le scan du code statique de Webgoat (Java) on ouvre un terminal Windows :

```
C:  
Cd yasca  
C:\yasca\yasca webgoat
```

Le résultat de l'audit sera généré dans un dossier sur le bureau Windows portant le nom « yasca »