#### **Université FERHAT ABBAS SETIF I**

Faculté des sciences Département d'informatique Sécurité des Applications Web 1<sup>ère</sup> année Master GL, IDTW

# TP n°3

## Objectif

L'objectif de ce TP est :

- de détecter les vulnérabilités les plus importantes pouvant être rencontrées au niveau d'un application web vulnérable Damn Vulnerable Web Application (DVWA)
- d'apprendre à exploiter ces vulnérabilités pour mener des attaques sur l'application web
- de comprendre comment ces attaques peuvent être empêchées
- d'autres vulnérabilités et d'autres attaques seront étudiées dans le TP n°4

### Outils utilisés

Les outils que nous utiliserons dans ce TP sont :

- une machine hôte (machine physique) disposant d'un navigateur web récent (Firefox)
- une machine virtuelle Metasploitable2 disposant de l'application web Damn Vulnerable Web Application (DVWA)

### Damn Vulnerable Web Application (DVWA)

Damn Vulnerable Web Application (DVWA) est une application à but pédagogique permettant de s'initier à la sécurité des applications web. L'application se présente sous la forme d'un formulaire de connexion demandant à l'utilisateur d'entrer ses identifiants (login et mot de passe).

DVWA		
Username		
admin		
Password	Login	
• • • • • • • •		
Damn Vulnerable Web Applicatio	on (DVWA) is a RandomStorm OpenSou	urce project

Une fois la connexion effectuée, la page principale de l'application se compose :

- d'un menu vertical (A) à gauche présentant chacune des vulnérabilités dont souffre l'application et conduisant vers la page web permettant de tester cette vulnérabilité,
- d'un espace central d'affichage (B),
- et d'informations de configuration (C), en bas à gauche, tels que le login de l'utilisateur courant (**Username**) ainsi que le niveau de sécurité courant (**Security Level**) qui peut prendre un des trois valeurs suivantes : **low**, **medium** ou **high**

DVWA				
Home Instructions Setup Menu (A) Brute Force Command Execution CSRF File Inclusion SQL Injection SQL Injection (Blind) Upload XSS reflected XSS stored DVWA Security PHP Info About Logout Username: admin	Welcome to Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment. WARNING! Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing. Disclaimer We do not take responsibility for the way in which any one uses this application. We have made the purposes of prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the very in which any one uses this application. We have made the purposes of prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it. General Instructions You have logged in as 'admin'			
PHPIDS: disabled				
	Damn Vulnerable Web Application (DVWA) v1.0.7			

Le changement du niveau de sécurité courant s'effectue en cliquant sur l'item de menu **DVWA Security**, en choisissant dans la liste déroulante le niveau souhaité, puis en cliquant sur le bouton **submit.** 

Home	DVWA Security 🆗
Instructions Setup	Script Security
Brute Force Command Execution CSRF File Inclusion	Security Level is currently <b>medium</b> . You can set the security level to low, medium or high. The security level changes the vulnerability level of DVWA. medium
SQL Injection SQL Injection (Blind) Upload XSS reflected XSS stored	<b>PHPIDS</b> v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications. You can enable PHPIDS across this site for the duration of your session.
DVWA Security PHP Info About	PHPIDS is currently <b>disabled</b> . [enable PHPIDS] [Simulate attack] - [View IDS log]

Pour créer ou réinitialiser la base de données de l'application dans son été initial, il suffit de cliquer sur l'item de menu Setup puis sur le bouton **Create / Reset Database**.

Home	Database setup 🦒	
Instructions	Click on the 'Create / Reset Database' button below to create or reset your database. If you get an error make	
Setup	sure you have the correct user credentials in /config/config.inc.php	
	If the database already exists, it will be cleared and the data will be reset.	
Brute Force		
Command Execution	Backend Database: MySQL	
CSRF		
File Inclusion	Create / Reset Database	
SQL Injection		
SQL Injection (Blind)		
Upload		

## Travail à réaliser

- 1. Se connecter sur la machine **Metasploitable2** en tant que l'utilisateur **msfadmin** ayant pour mot de passe **msfadmin** (il faut taper **,sfqd,in** en cas d'utilisation d'un clavier AZERTY).
- 2. Si le clavier utilisé est en AZERTY, il faut taper la commande sudo loadkeys fr
- 3. Pour connaître l'adresse IP de la machine **Metasploitable2** il faut taper la commande **ifconfig**
- 4. Sur la machine hôte (physique), lancer le navigateur Firefox et taper l'adresse IP de la machine **Metasploitable2** dans la barre d'adresse
- 5. Une liste d'applications web s'affiche. Il faut alors choisir l'application **DVWA**.



<u>TWiki</u>
<u>phpMyAdmin</u>
<u>Mutillidae</u>
<u>DVWA</u>
WebDAV

- 6. Se connecter à l'application DVWA en utilisant le login **admin** et le mot de passe **password**.
- 7. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **low**.
- 8. Cliquer sur l'item de menu **XSS reflected**. Le formulaire affiché est vulnérable aux attaques XSS. Afficher le code source du script de traitement en appuyant sur le bouton **source**. Réaliser une attaque XSS réfléchie en injectant dans le champ un script JavaScript permettant d'afficher une fenêtre d'alerte.

- 9. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **medium**.
- 10. Retourner sur la page **XSS reflected**. Le script de traitement dispose maintenant d'une défense contre les injections de JavaScript. Modifier l'attaque précédente pour pouvoir à nouveau injecter le script JavaScript.
- 11. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **high**.
- 12. Retourner sur la page **XSS reflected**. Afficher le code source du script de traitement.. Le script de traitement dispose maintenant d'une défense encore plus efficace contre les injections de balises. Est-il encore possible d'injecter un script JavaScript ou n'importe quelle balise HTML ?
- 13. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **low**.
- 14. Cliquer sur l'item de menu **XSS stored**. Le formulaire affiché est vulnérable aux attaques XSS stockées. Afficher le code source du script de traitement en appuyant sur le bouton **source**. Réaliser une attaque XSS stockée en injectant dans le champ un script JavaScript permettant d'afficher une fenêtre d'alerte.
- 15. Se déconnecter de l'application et se reconnecter avec le login **smithy** et le mot de passe **password** et aller sur la page XSS stored pour vérifier l'effet de l'attaque XSS.
- 16. Se déconnecter et se reconnecter en tant que l'utilisateur **admin**.
- 17. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **medium**.
- 18. Retourner sur la page **XSS stored**. Le script de traitement dispose maintenant d'une défense contre les injections de JavaScript. Modifier l'attaque précédente pour pouvoir à nouveau injecter le script JavaScript.
- 19. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **high**.
- 20. Retourner sur la page **XSS stored**. Afficher le code source du script de traitement.. Le script de traitement dispose maintenant d'une défense encore plus efficace contre les injections de balises. Est-il encore possible d'injecter un script JavaScript ?
- 21. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **low**.
- 22. Cliquer sur l'item de menu **Command Execution**. Le formulaire affiché permet d'exécuter un **ping** sur une adresse IP quelconque. Ce script est vulnérable à l'injection de commandes Linux. Afficher le code source du script de traitement en appuyant sur le bouton **source**. Réaliser une attaque en saisissant dans le champs IP une adresse IP (ex : 192.168.56.1) et en injectant une commande (**ls -l**).
- 23. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **medium**.
- 24. Retourner sur la page **Command Execution**. Le script de traitement dispose maintenant d'une défense. Modifier l'attaque précédente pour pouvoir à nouveau injecter la commande **ls**.
- 25. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **high**.
- 26. Retourner sur la page **Command Execution**. Afficher le code source du script de traitement. Le script de traitement dispose maintenant d'une défense encore plus efficace contre les injections. Peut-on saisir dans ce champ autre chose qu'une adresse IP valide ?
- 27. Cliquer sur l'item de menu **DVWA Security** et positionner le niveau de sécurité sur **low**.

28. Cliquer sur l'item de menu **File Inclusion**. L'URL de la page affichée permet d'inclure n'importe quel fichier dont l'URL ou le chemin relatif est spécifié par le paramètre page (.../dvwa/vulnerabilities/fi/?page=**URL ou chemin relatif**). Ce script est vulnérable car il permet d'inclure n'importe quel script et de le faire exécuter par le serveur ou de faire afficher n'importe quel fichier accessible sur le serveur. Afficher le contenu du fichier /**etc/passwd** du serveur en donnant son chemin relatif.