

**INF4420A - Sécurité informatique**

**Automne 2018**

**TP  No. [2]**

**Groupe [2]**

**[1972528] – [Manuel Rodriguez]**

**[1748125] – [David Tremblay]**

**Soumis à : Mikaela Stéphanie Ngamboe Mvogo**

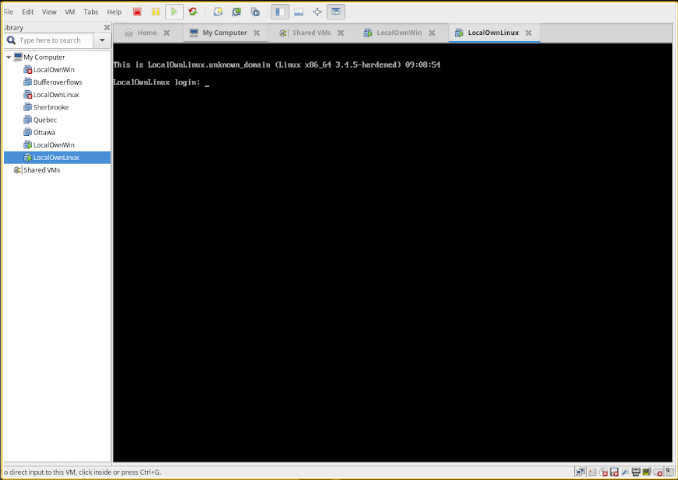
**[12-11-2018]**

**Question 1**

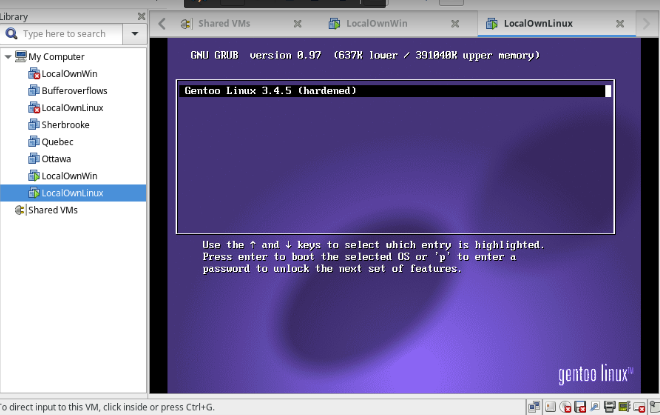
**Machine LocalOwnLinux**

*Phase de reconnaissance*

1- La machine virtuelle nous demande un utilisateur et un mot de passe.



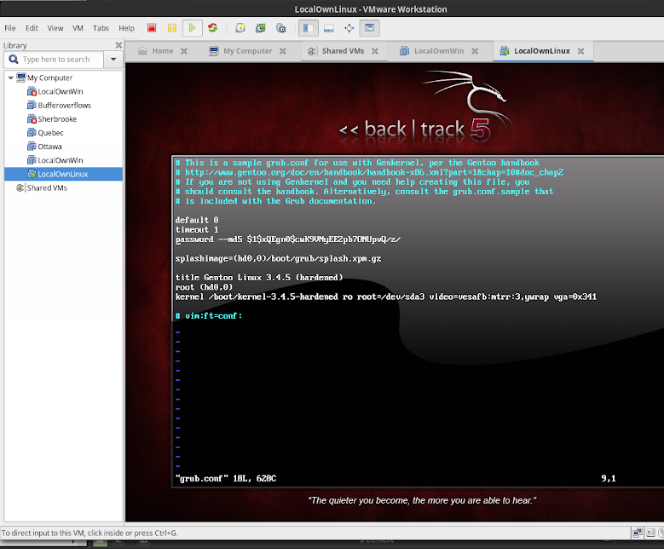
2 & 3- On nous demande d’entrer un mot de passe pour entrer dans le BIOS ou de choisir un OS ou d’entrer un mot de passe pour plus de fonctionnalités dans le grub.

****

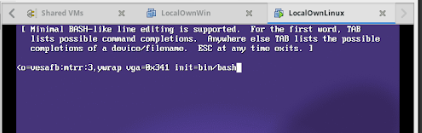
4- On ne peux pas utiliser l’option puisque nous devons entrer un mot de passe pour pouvoir utiliser cette fonctionnalitée.

*Réalisation de l’attaque*

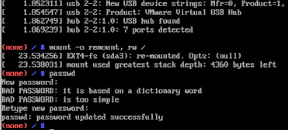
La figure ci-dessous illustre notre accès au fichier grub.conf afin d’y effacer la ligne commençant par “password” en suivant les instructions 1 à 4 de l’énoncé. Nous éditons le fichier grub.conf à l’aide de la commande vim.

**

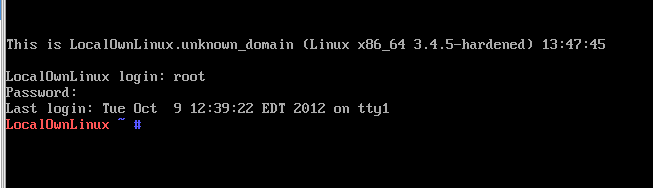
5- On ajoute notre ligne de code à l’endroit indiqué.



6- Nous avons changé le mot de passe pour “hacker”.

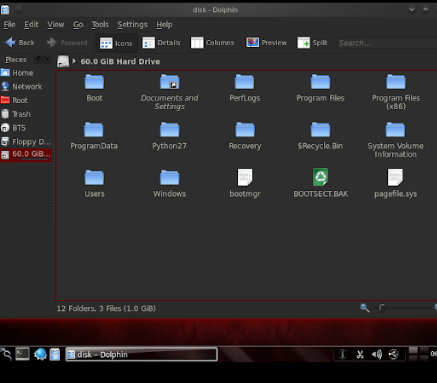


7- Nous avons réussi à nous connecter avec notre nouveau mot de passe.

****

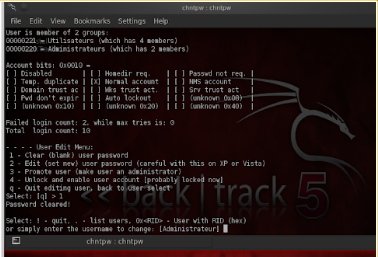
**Machine LocalOwnWindows**

4- On monte la partition en ouvrant l’explorateur de fichiers.

****

7- Le fichier SAM(Security Account Manager) est la base de données des comptes locaux et contient les mot de passes locaux de la machine virtuelle Windows.

8- Nous avons effacé le mot de passe de admin à l’aide de l’option 1.

****

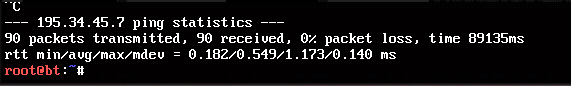
9- Nous pouvons maintenant entrer dans la session admin puisqu’elle ne contient plus de mot de passe. Nous avons ajouté un fichier texte nommé “HACKER” sur le bureau.



**Question 2**

*Phase de reconnaissance*

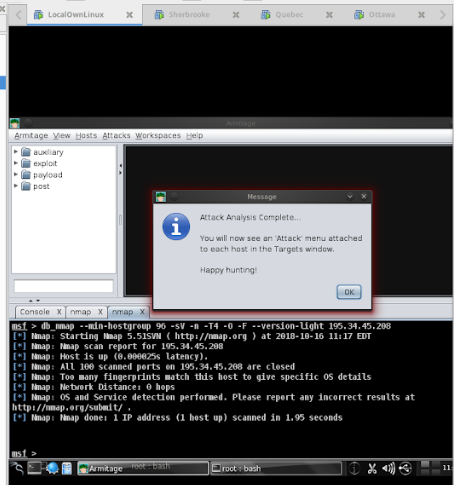
2&3- Nous avons testé notre connexion avec la commande ping et tous les paquets ont été correctement transmis.



4- Nous n’aurions pas eu à effectuer cette manipulation puisque nous aurions eu accès à un routeur qui permet à des sous-réseaux de communiquer ensemble.

9- Nous obtenons les 3 machines démarrées précédemment (Quebec, Ottawa, Sherbrook). Nmap est un scanner de ports libres qui détecte les ports libres et indique des informations sur le système d’exploitation distant.

10- On fait une recherche des exploits applicables.

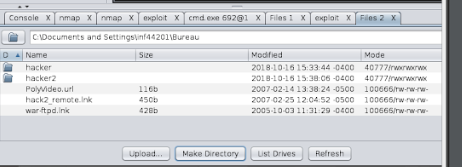
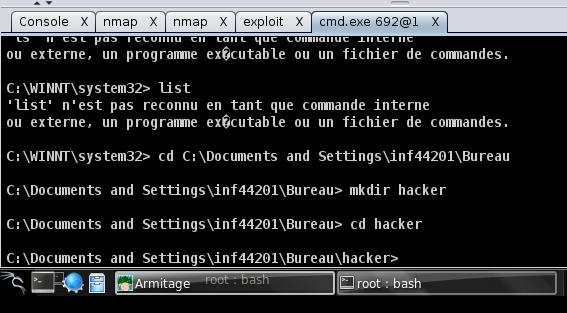


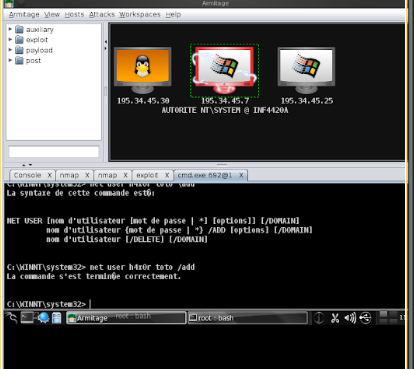
*Exploitation de failles de sécurité connues*

1- Nous allons réaliser l’attaque sur la machine dont l’adresse IP est 195.34.45.7.

2- Nous avons accès à une nouvelle option “meterpreter 1” lorsque qu’on exécute un clic droit sur la machine. Cette option permet de réaliser plusieurs nouvelles actions sur la machine dont entre autres ouvrir une fenêtre de commande.

3- Nous avons ouvert une fenêtre de commande et nous avons été sur le bureau avec la commande “cd”. Puis, nous avons créer un répertoire avec la commande “mkdir”. Finalement, nous avons utilisé la commande “net user h4xx0r toto /add” pour ajouter l’utilisateur.

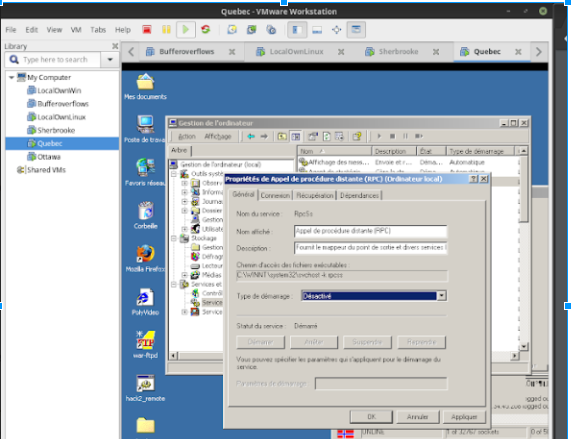




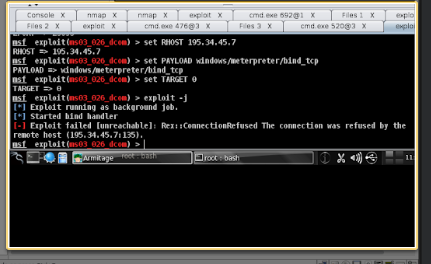
4- Nous avons exploité la faille dans WarFTPD à l’aide de l’attaque waFTPD\_165\_pass. Comme pour l’attaque précédente, ceci débloque des nouvelles actions possibles sur la machine distantes. Cette fois-ci, nous avons créé un répertoire à l’aide du bouton “Make Directory” présent dans la fenêtre de l’explorateur de fichier ouvert grâce aux options dans “meterinterpreter 3”.

5- Les procédures pour l’attaque sur la nouvelle machine est très similaire à la première attaque utilisée sur la machine précédente. Nous avons lancé l’attaque ms\_04\_007\_killbill qui permet d’empêcher la machine distante de traiter l’authentification des utilisateurs correctement. Une fois l’attaque réalisée, nous avons ajouté un dossier sur la machine avec la commande “mkdir” dans une fenêtre de commandes.

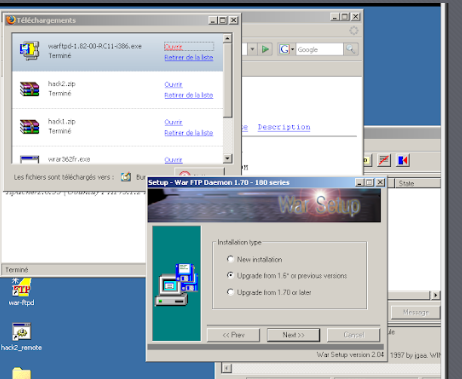
6- Nous avons arrêter le processus rpcdcom sur la machine virtuelle Quebec.

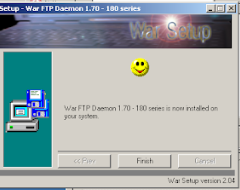


On peut voir dans la figure suivante que l’attaque ms\_03\_026 exploitant le processus rpcdcom ne marche plus puisque ce processus est maintenant désactivé sur la machine distante. (the connection was refused by the remote host)

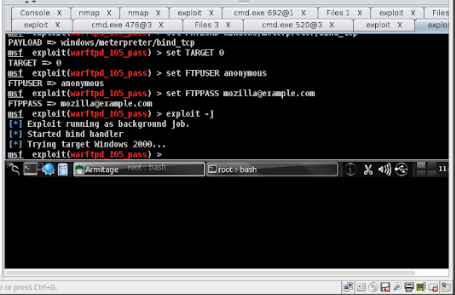


7- Nous avons installé la dernière version de warFTPD.



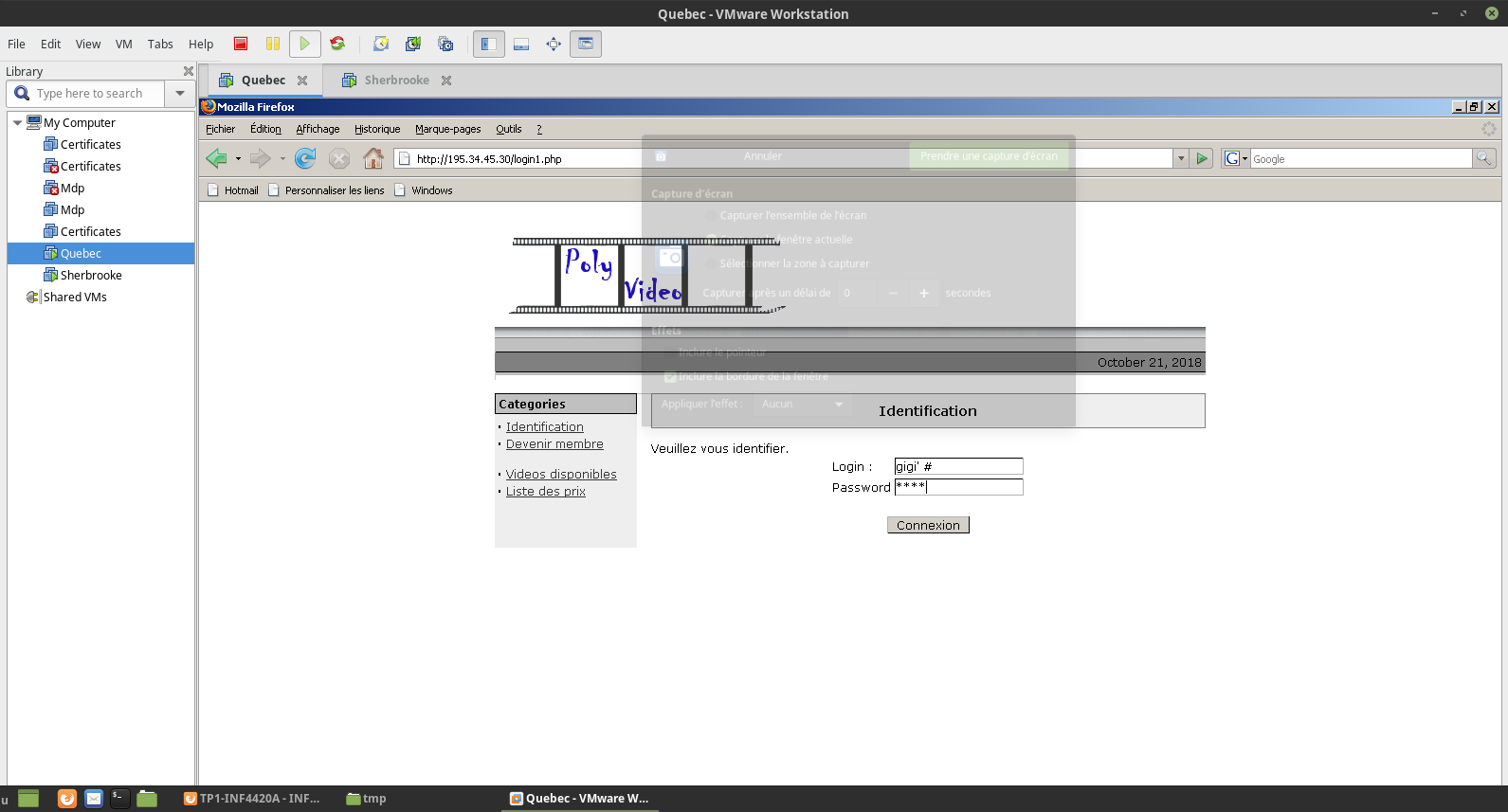


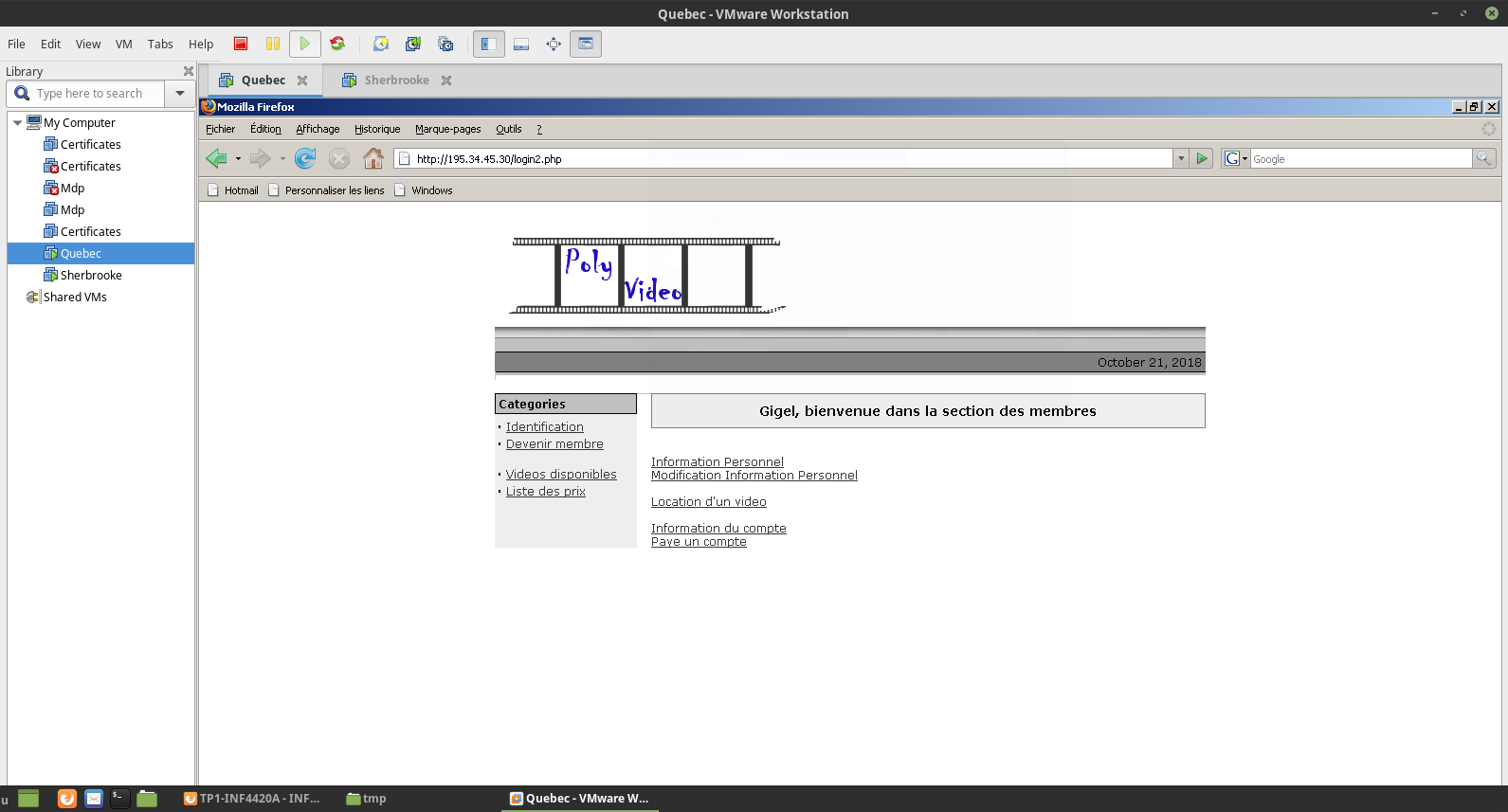
Nous avons ensuite essayé de réaliser l’attaque. Nous n’obtenons pas de message d’erreur mais l’attaquant tente de se connecter à la machine Quebec sans jamais y parvenir.



**Question 3**

*SQL*

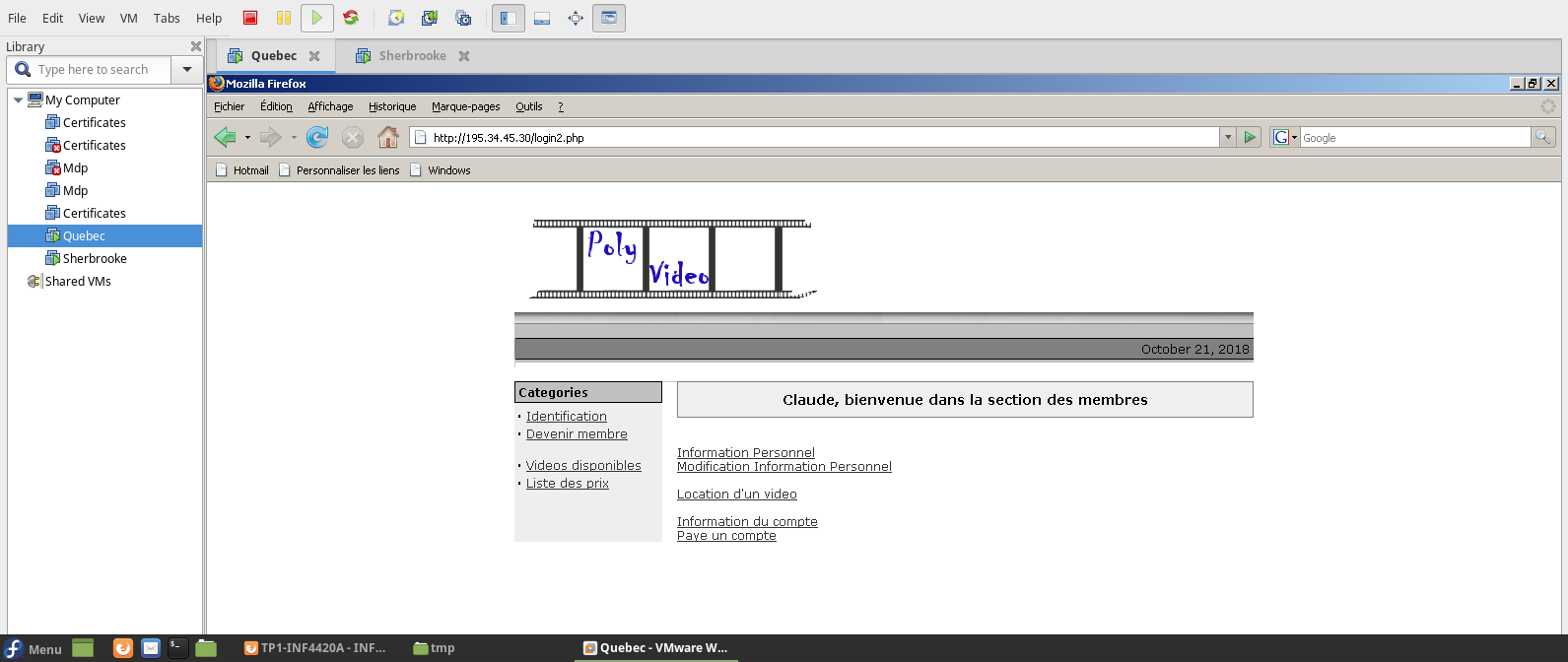
On peut se connecter au site web avec le compte d’utilisateur ‘gigi’ en utilisant le code suivant: ***gigi’ #***. De cette manière, nous laissons en dehors de la requête SQL la vérification du mot de passe avec le caractère #. Résultat: 



2. Si on ne connaît pas le nom d’aucun compte on doit utiliser une vérification qui sera toujours vrai. Comme ceci: **‘ or ‘’=**’. Comme ça, la requête SQL sera:

SELECT mem\_code FROM membres WHERE mem\_login = ‘’ or ‘’=’’ and mem\_pwd = ‘’ or = ‘’;

Résultat:



3. Pour l'attaque 1 nous profitons du fait que le code n'utilise pas la méthode PREPARE, et donc nous avons un accès direct au code PHP. Par le symbole # nous pouvons commenter tout ce qui vient après et ceci dans la même ligne.

Pour l’attaque 2 nous profitons de la même chose, mais cette fois nous ne savons pas quel utilisateur attaquer, donc nous introduisons une condition qui sera toujours vraie pour retourner le premier résultat de la table ‘membres’.

4. On doit seulement utiliser le méthode de SQL PREPARE.

*extract($\_POST);*

*$req = $bd -> prepare("SELECT mem\_code FROM membres WHERE mem\_login = ? and mem\_pwd = ?");*

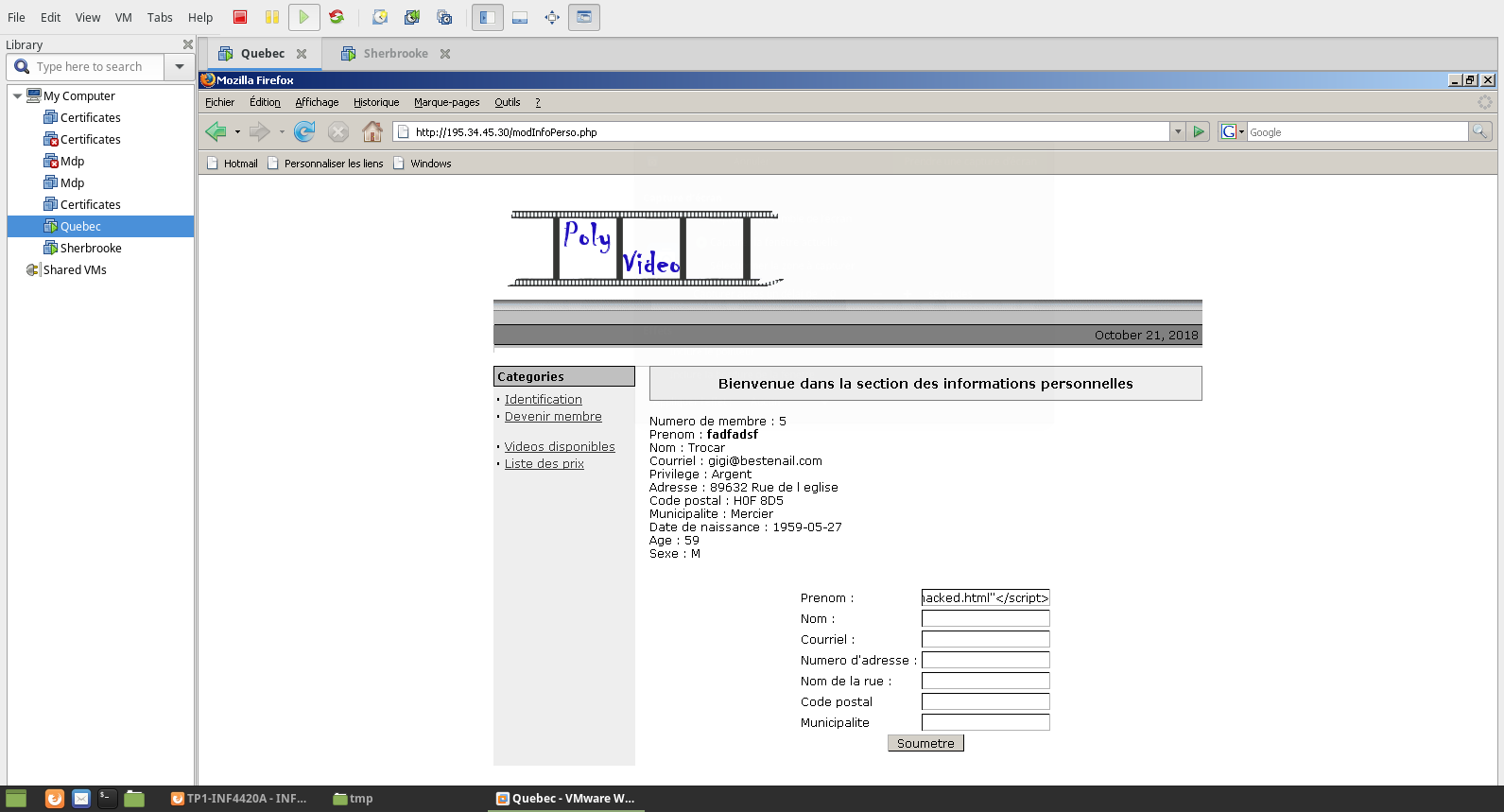
*$req -> execute(array($login, $password));*

*$result = mysql\_query($req) or die ("Error : the SQL request".$req."is not valid: ".mysql\_error()); list($mem\_code) = mysql\_fetch\_array($result);*

*if (empty($mem\_code)){ //verifier que la requete a retourne une reponse positive*

*XSS*

1. Comme les noms d’utilisateurs sont affichés sur la page principale, on peut simplement changer le prénom de quelqu’un pour le code Javascript. Comme ça:



2. On a utilisé la possibilité d'introduire des caractères HTML dans la requête SQL. Pour corriger la faille, on peut simplement utiliser la fonction htmlspecialchars() de PHP.

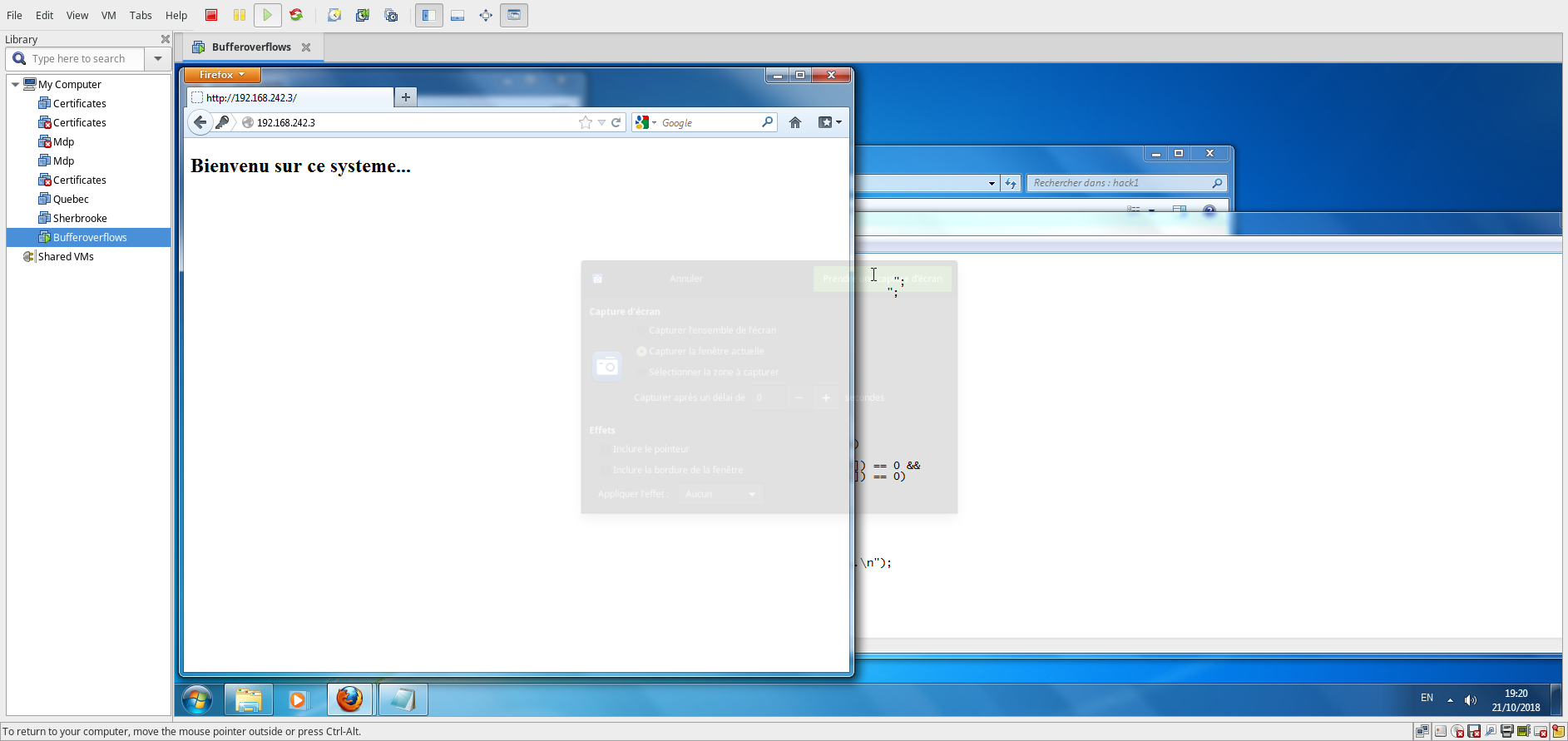
**Question 4**

1. La séquence exacte de caractères serait:

**Utilisateur**: 1234567890123456789 1234567890123456789 1234567890123456789 1234567890123456789

**Mot de passe**: 1234567890123456789

Comment l’utilisateur est un char[20], les 20 premiers caractères seront pour *user\_name*, les prochains 20 pour *password* et après on commence à écrire dans la table, sur root/98765 donc on doit écraser ces champs pour le nom d'utilisateur et mot de passe inventé.



2. Pour enlever ce problème de sécurité il faudrait commencer à utiliser la fonction fgets, dans laquelle on peut introduire la quantité maximum de char qu’on veut accepter.

**Question 5**

1. Avec l’aide d’Ollydbg on peut trouver l’appel à la fonction logon, qui est responsable de nous laisser nous connecter. L’adresse est:



Maintenant, on peut utiliser la vulnérabilité de scanf, contenu dans la fonction d’affichage pour avoir accès au stack et insérer l’adresse d’appel du logon.

2. Comment la vulnérabilité est dans scanf, on doit utiliser une autre fonction plus sûre comme fgets, qui nous permettra de déterminer la taille maximum du string introduit.