POLYTECHNIQUE MONTRÉAL

LE GÉNIE EN PREMIÈRE CLASSE

Travail pratique 1

Présenté à M. Bresteau
INF4420A Sécurité informatique

Fait par :

Étienne Asselin 1773922
Vincent Rodier 1744784
Groupe laboratoire B1-9

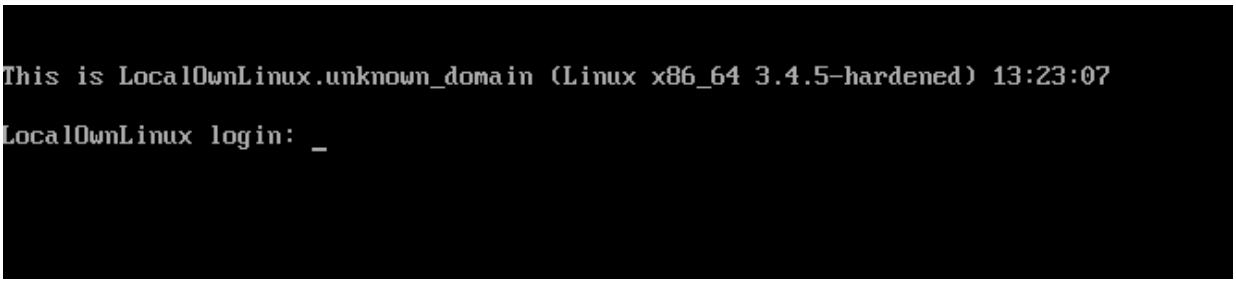
Le vendredi 9 février 2018
École Polytechnique de Montréal

Question 1

Phase de reconnaissance

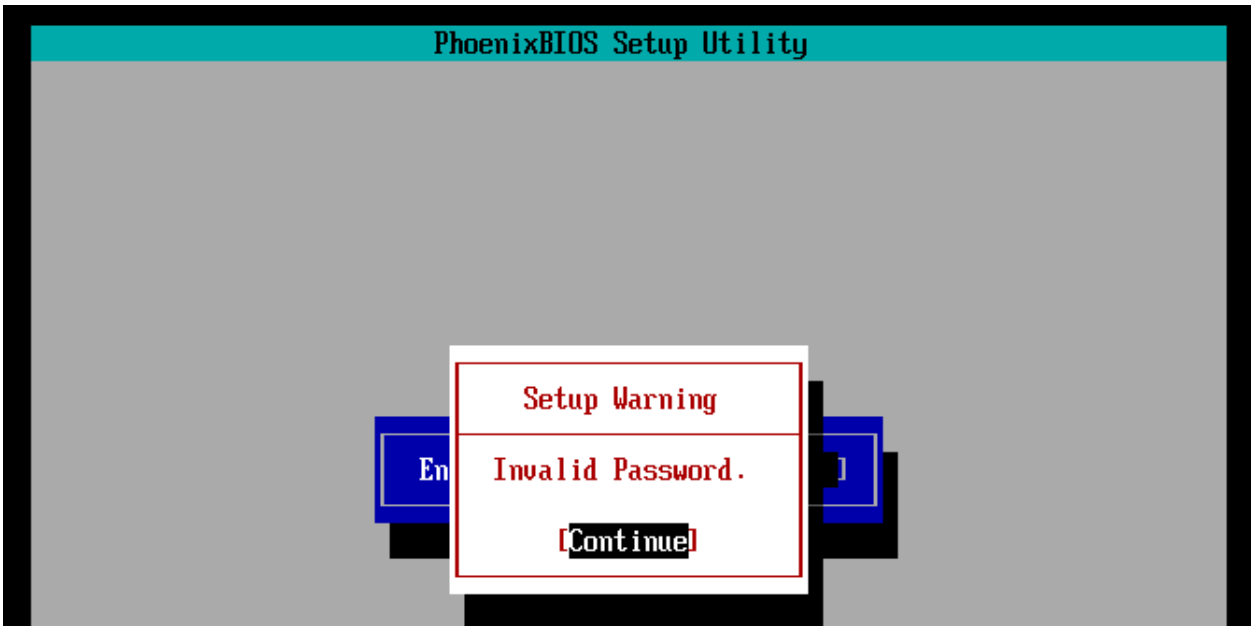
1. Démarrer la machine virtuelle (VM) et essayer de vous connecter à une session. Que constatez-vous ?

Rep : Il est impossible de se connecter sans les informations de connexion.



2. Redémarrez la VM et au démarrage appuyez sur F2 pour rentrer dans le BIOS. Que se passe-t-il ?

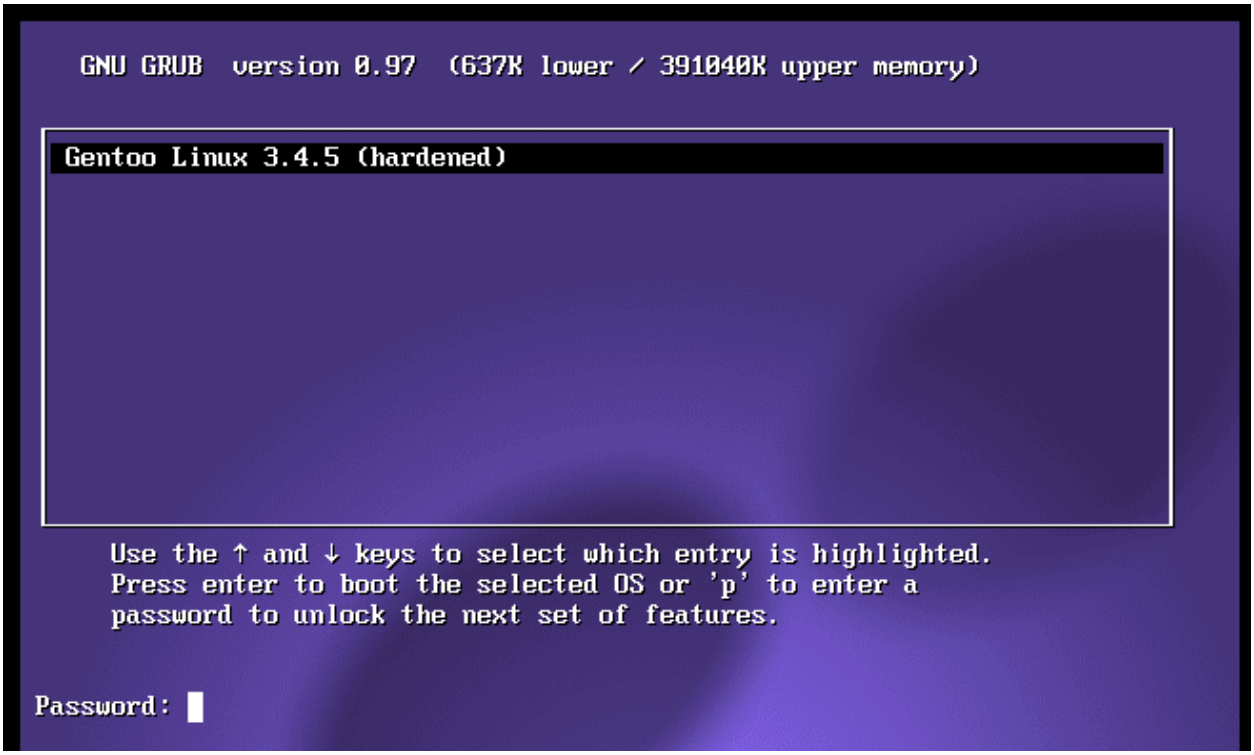
Rep : On doit absolument entrer un mot de passe pour pouvoir accéder au BIOS.



3. Appuyez sur Echap pour continuer le boot de la machine. A l'écran de Grub (fond violet), appuyer sur une touche quelconque (sauf Entrée). Cet écran présente les différentes options de boot pour la machine, dans notre cas il n'y a qu'une seule ligne correspond au système Gentoo Linux. Habituellement il est possible d'éditer la ligne de commande correspondante en appuyant sur la touche e.

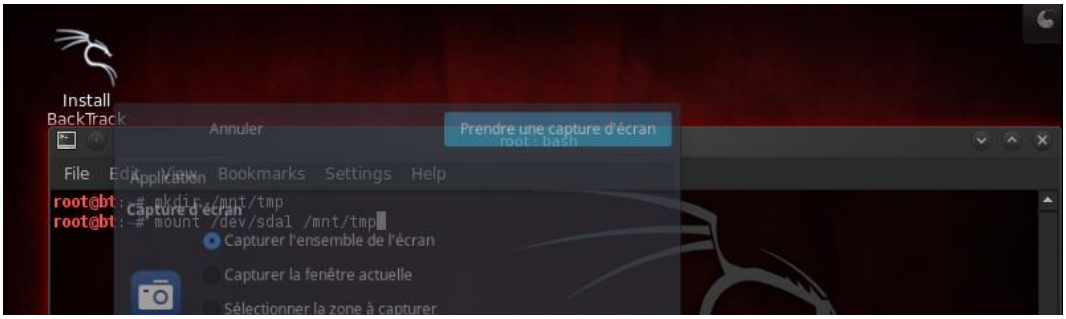
4. Est-ce possible dans notre cas ? Sinon, pourquoi ?

Rep : Dans notre cas, il n'est pas possible d'éditer la ligne de commande. Il préalablement entré un mot de passe afin d'accéder à des options supplémentaires.



Réalisation de l'attaque

Étape 1 à 4 réaliser :

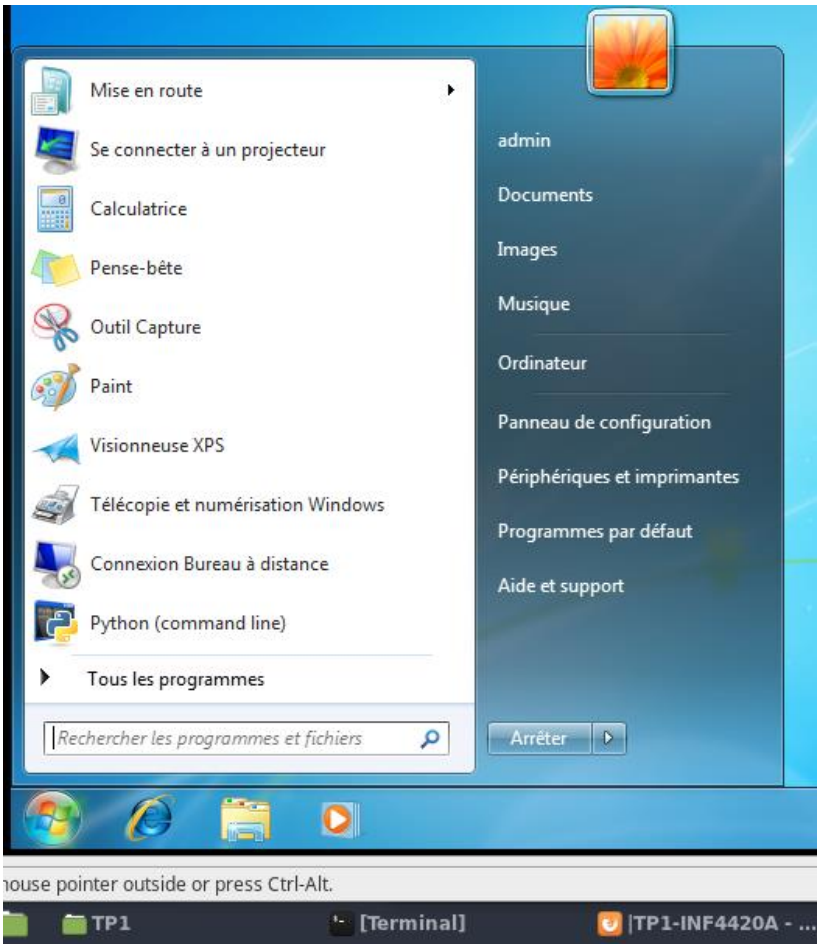


Réalisation des étapes 5 à 7 :

```
This is LocalOwnLinux.unknown_domain (Linux x86_64 3.4.5-hardened) 13:57:39

LocalOwnLinux login: root
Password:
Last login: Tue Oct  9 12:39:22 EDT 2012 on tty1
LocalOwnLinux ~ # _
```

Preuve de la réalisation de l'attaque sur la machine virtuel Windows :



Question 7 Machine LocalOwnWin

5. Expliquez à quoi sert ce fichier SAM

Le fichier SAM correspond à Sécurity Account Manager. C'est un fichier dans lequel est enregistrer les mots de passes des usagers. C'est mots de passes sont hasher et peuvent servent à l'authentification des utilisateurs sur la machine.

Question 2

1. Examinez le fichier /etc/passwd. Contient-il des mots de passe ? Pourquoi? Quelles sont ses permissions d'accès? Pourquoi ?

   Le fichier ne semble pas contenir de mot de passe explicite. Plusieurs personnes ont accès à ce fichier en lecture et il ne serait pas sécuritaire de simplement écrire les mots de passes dans un fichier. L'auteur du fichier à des droits de lecture et d'écriture, tandis que tous les autres utilisateurs ont simplement les droits en lecture.

```
Mdp etc # ls -la passwd
-rw-r--r-- 1 root root 648 Aug 27  2012 passwd
```

```
Mdp etc # cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
news:x:9:13:news:/var/spool/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucp:/bin/false
operator:x:11:0:operator:/root:/bin/bash
portage:x:250:250:portage:/var/tmp/portage:/bin/false
nobody:x:65534:65534:nobody:/var/empty:/bin/false
man:x:13:15:added by portage for man:/usr/share/man:/sbin/nologin
sshd:x:22:22:added by portage for openssh:/var/empty:/sbin/nologin
```

2. Observez ce qui se passe dans les fichiers passwd et shadow. Lequel ou lesquels de ces deux fichiers sont modifiés ? Pourquoi ?
   Les deux fichiers sont modifiés. Les deux fichiers contiennent une ligne supplémentaire avec le nom du nouvel utilisateur. Le fichier shadow semble cependant contenir la valeur hasher du mot de passe de l'utilisateur.

Avant :

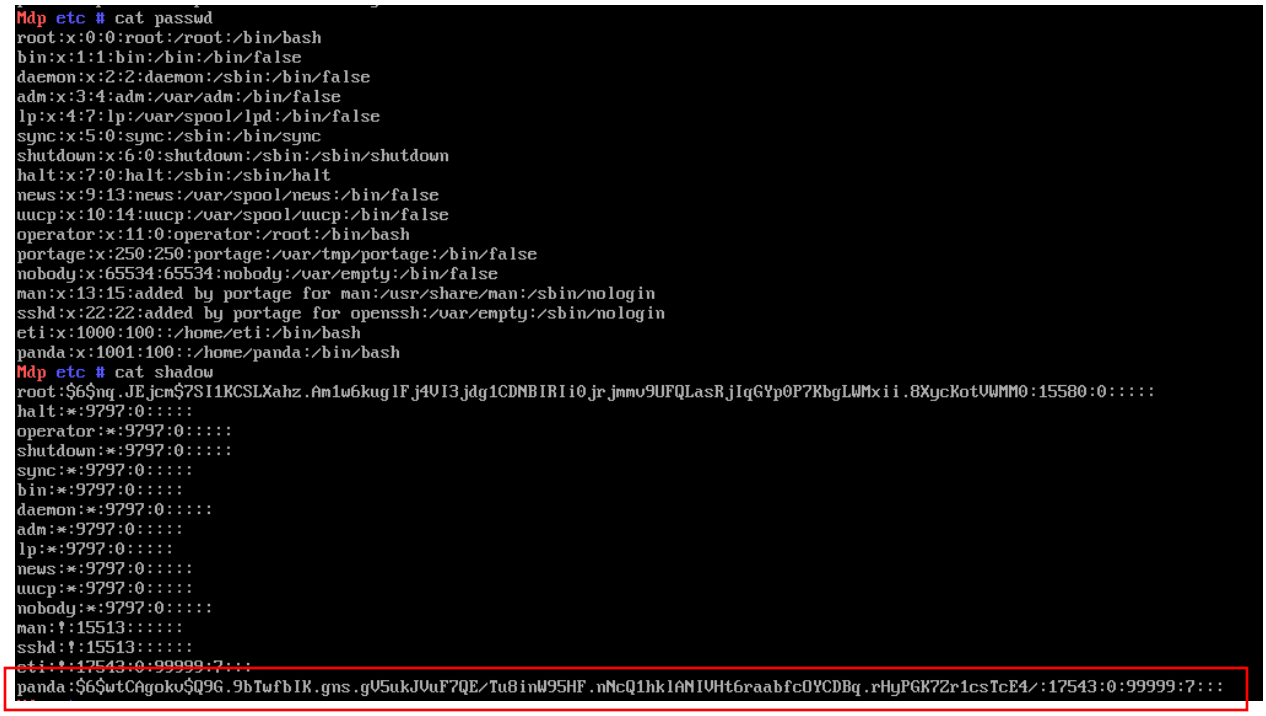
```
Mdp etc # cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
news:x:9:13:news:/var/spool/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucp:/bin/false
operator:x:11:0:operator:/root:/bin/bash
portage:x:250:250:portage:/var/tmp/portage:/bin/false
nobody:x:65534:65534:nobody:/var/empty:/bin/false
man:x:13:15:added by portage for man:/usr/share/man:/sbin/nologin
sshd:x:22:22:added by portage for openssh:/var/empty:/sbin/nologin
Mdp etc # cat sh
shadow   shadow-  shells
Mdp etc # cat shadow
root:$6$nq.JEjcm$7SI1KCSLXahz.Am1w6kuglFj4VI3jdg1CDNBIRIi0jrjmmv9UFQLasRjIqGYp0P7KbgLWMxii.8XycKotVWMM0:15580:0:::::
halt:*:9797:0:::::
operator:*:9797:0:::::
shutdown:*:9797:0:::::
sync:*:9797:0:::::
bin:*:9797:0:::::
daemon:*:9797:0:::::
adm:*:9797:0:::::
lp:*:9797:0:::::
news:*:9797:0:::::
uucp:*:9797:0:::::
nobody:*:9797:0:::::
man:!:15513:::::::
sshd:!:15513:::::::
```

Après :

```
Mdp etc # cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
news:x:9:13:news:/var/spool/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucp:/bin/false
operator:x:11:0:operator:/root:/bin/bash
portage:x:250:250:portage:/var/tmp/portage:/bin/false
nobody:x:65534:65534:nobody:/var/empty:/bin/false
man:x:13:15:added by portage for man:/usr/share/man:/sbin/nologin
sshd:x:22:22:added by portage for openssh:/var/empty:/sbin/nologin
eti:x:1000:100::/home/eti:/bin/bash
panda:x:1001:100::/home/panda:/bin/bash
Mdp etc # cat shadow
root:$6$nq.JEjcm$7SI1KCSLXahz.Am1w6kuglFj4VI3jdg1CDNBIRIi0jrjmmv9UFQLasRjIqGYp0P7KbgLWMxii.8XycKotVWMM0:15580:0:::::
halt:*:9797:0:::::
operator:*:9797:0:::::
shutdown:*:9797:0:::::
sync:*:9797:0:::::
bin:*:9797:0:::::
daemon:*:9797:0:::::
adm:*:9797:0:::::
lp:*:9797:0:::::
news:*:9797:0:::::
uucp:*:9797:0:::::
nobody:*:9797:0:::::
man:!:15513:::::::
sshd:!:15513:::::::
eti:!:17543:0:99999:7:::
panda:$6$iASuMRlv$pzpANyfHjjca9rk3VAhtJqDrzoxH6/DeLKz6U7d64nkevA52fjtdUZbkfjsQehXYcqUt74CU//RGURvzFeCSr1:17543:0:99999:7:::
```

3. Qu'est-ce que vous remarquez dans les fichiers passwd et shadow? Lequel de ces deux fichiers change? Pourquoi ? Où se trouve donc l'information du mot de passe? Quelles sont les permissions du fichier shadow et pourquoi ?

Le fichier passwd n'a pas subi de modification à la suite du changement du mot de passe de l'utilisateur, mais le fichier shadow oui. L'information du mot de passe est contenue dans la longue chaine de caractère suivant le nom de l'utilisateur (mot de passe hasher). Les permissions du fichier shadow sont d'écriture et de lecture pour le créateur du fichier. Les utilisateurs faisant parti d'un certain groupe on accès au fichier shadow en lecture seulement et les autres utilisateurs n'ont simplement aucun droit. Les droits du fichier shadow sont strict puisqu'il contient des informations sensibles de sécurité. Ainsi seulement son créateur (root) à droit d'accéder à se fichier en mode écriture.



4. Changez à nouveau le mot de passe du même utilisateur et donnez-lui le *même* mot de passe. Est-ce que les informations du mot de passe ont changé? Pourquoi?

   Les informations du mot de passe ont changé puisque en "modifiant" le mot de passe, un nombre générer automatiquement que l'on appelle « salt » est regénérer et se salt est ajouté au mot de passe avant d'être hasher, donc même si le mot de passe est identique, le salt ne l'est pas et donc le résultat du hash est différent.

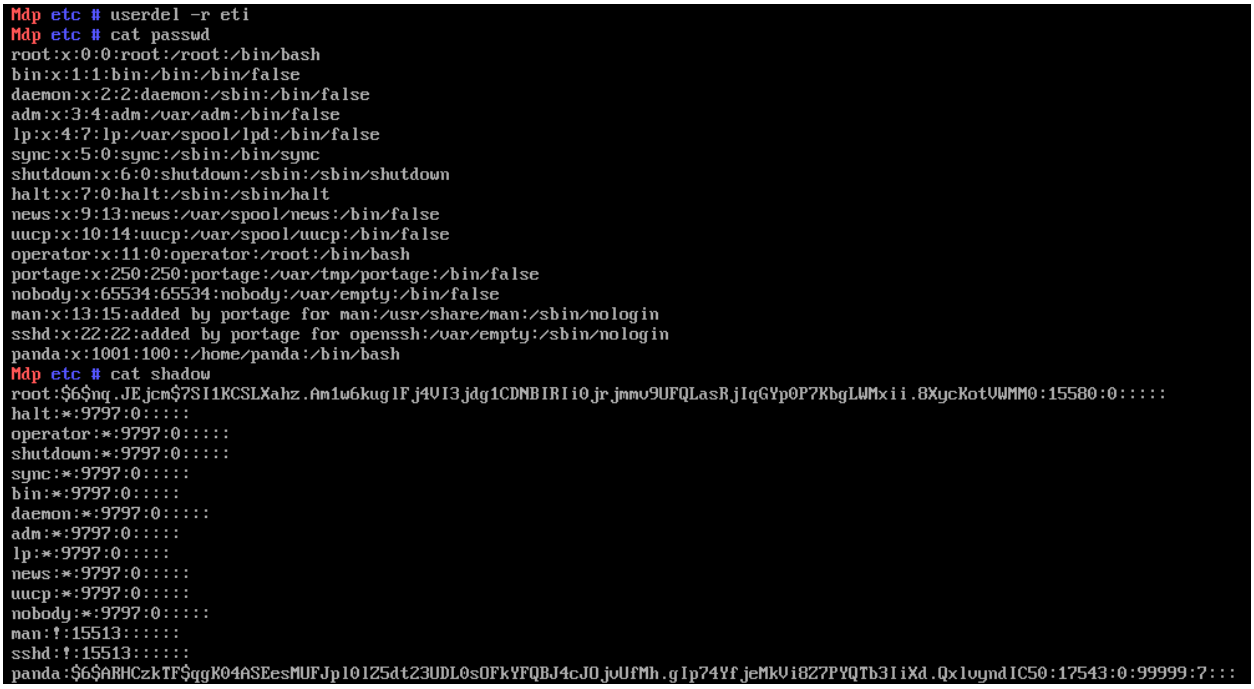
5. Estce que ceci est possible? Expliquez pourquoi. Quel est le problème?

   Oui il est possible de se connecter au deuxième compte avec le mot de passe du premier compte. Cela est possible parce que le hash mis par force au deuxième utilisateur est un hash

valide. Le problème est donc que si on peut modifier le fichier shadow, nous avons seulemtn besoin de connaitre un seul mot de passe

6. Effacez cet utilisateur avec la commande $ userdel –r NOM. Qu'est-ce qui se passe dans passwd et shadow ?

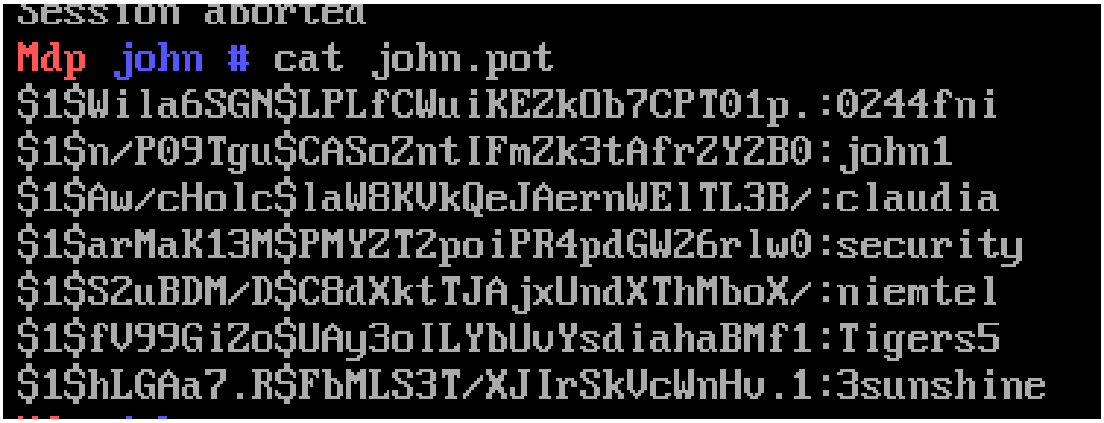
   L'entrée de l'utilisateur est supprimée.



Question 3

a) À la suite de près de 5 minutes d'exécution de john sur le fichier password1 nous avons répertorier 4 mots de passe. Le même exercice a été effectué avec le fichier password2 et 3 mots de passe ont été trouvés.

Fichier	Utilisateur	Mot de passe
Password1	Inf4420	0244fni
Password1	john	john1
Password1	david	claudia
Password1	admin	security
Password2	lola	niemtel
Password2	andre	Tigers5
Password2	morning	3sunshine



Les mots de passe dont assez faible car ils ne sont pas long en moyenne 7 caractères. Certains mots de passe comportent que des minuscules. Les mots de passe ne sont pas formé aléatoirement, il s'agit de mots du dictionnaire. Finalement, les chiffres sont seulement placé aux extrémités des mots.

b)

Voici la formule de l'entropie qui donne la complexité en nombre de bits

$$entropie = \log_2(alphabet^{longueur})$$

En prenant comme hypothèse que la longueur maximale du mot de passe est de 8 caractères voici la complexité selon les alphabètes :

Alphabète	**Formule**	**Entropie (bits)**
[a-zA-Z]	$8 * \log_2(52)$	45.6
[a-zA-Z0-9]	$8 * \log_2(62)$	47.6
Table ascii	$8 * \log_2(256^*)$	64

*Il s'agit d'une valeur théorique car il y a des caractères qui sont impossibles de taper sur le clavier. Nous utilisons 256 pour démontrer que plus le nombre est grand plus l'entropie augmente.

c)

En analysant le tableau ci-haut, il est facile de remarquer que plus l'alphabète est diversifié, plus l'entropie maximale augmente.

d)

En regardant les résultats de John the ripper, on s'aperçoit que les mots de passe n'ont pas un dictionnaire élargi, qu'ils sont courts et qu'ils ne sont pas formé de façon aléatoire.

Voici 3 critères qui augmente l'entropie d'un mot de passe :

1. Diversité du dictionnaire.
2. Longueur du mot de passe.
3. Mot de passe aléatoire.

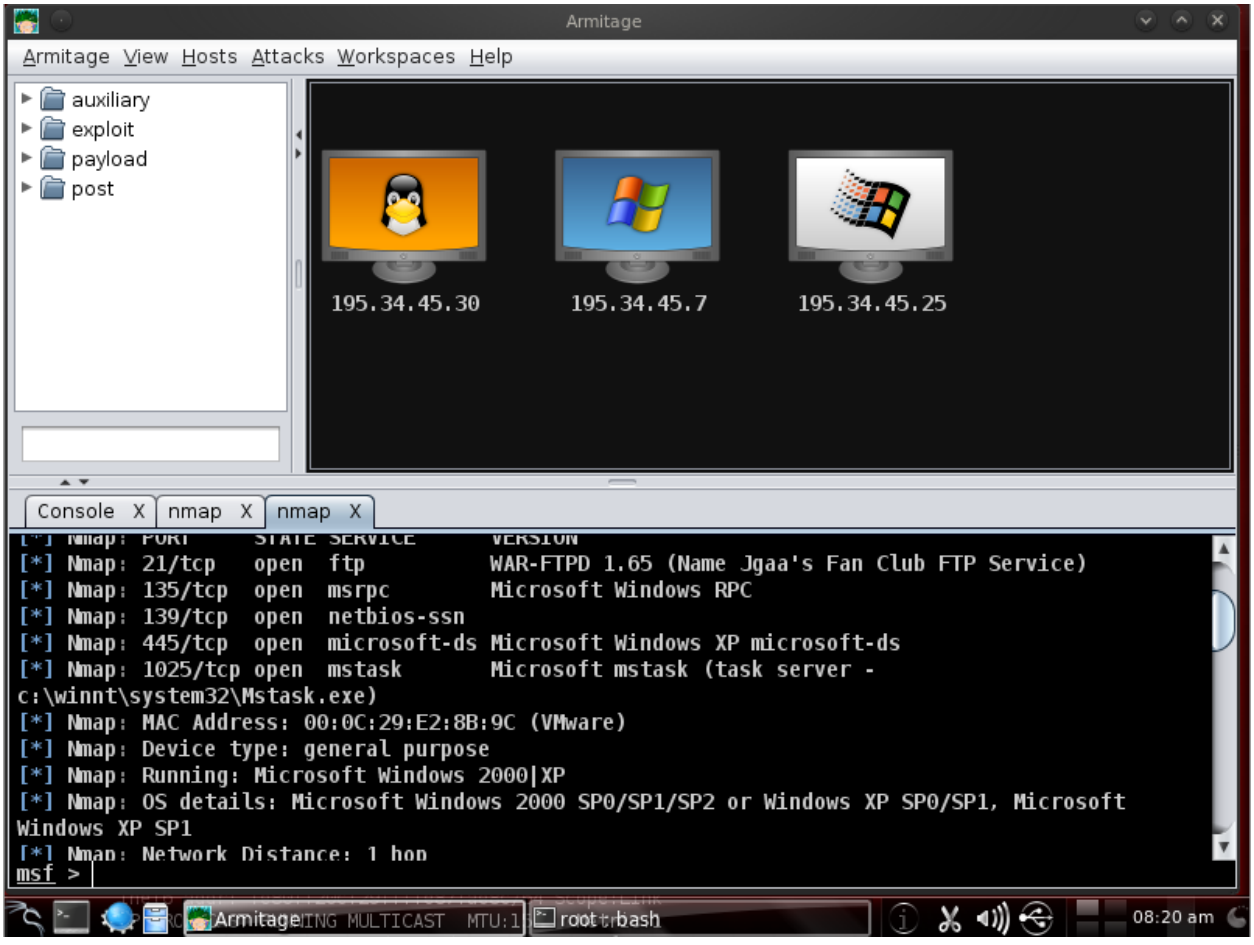
Question 4

Phase de reconnaissance

1. Expliquez en quoi la modification effectuée en 2. ne serait pas nécessaire si vous étiez dans votre sous-sol chez votre mère en train de vouloir hacker les serveurs Québec, Ottawa et Sherbrooke ?

   Dans le sous-sol de notre mère, il y a un routeur qui s'occuperait de changer notre adresse IP privé en adresse IP public.

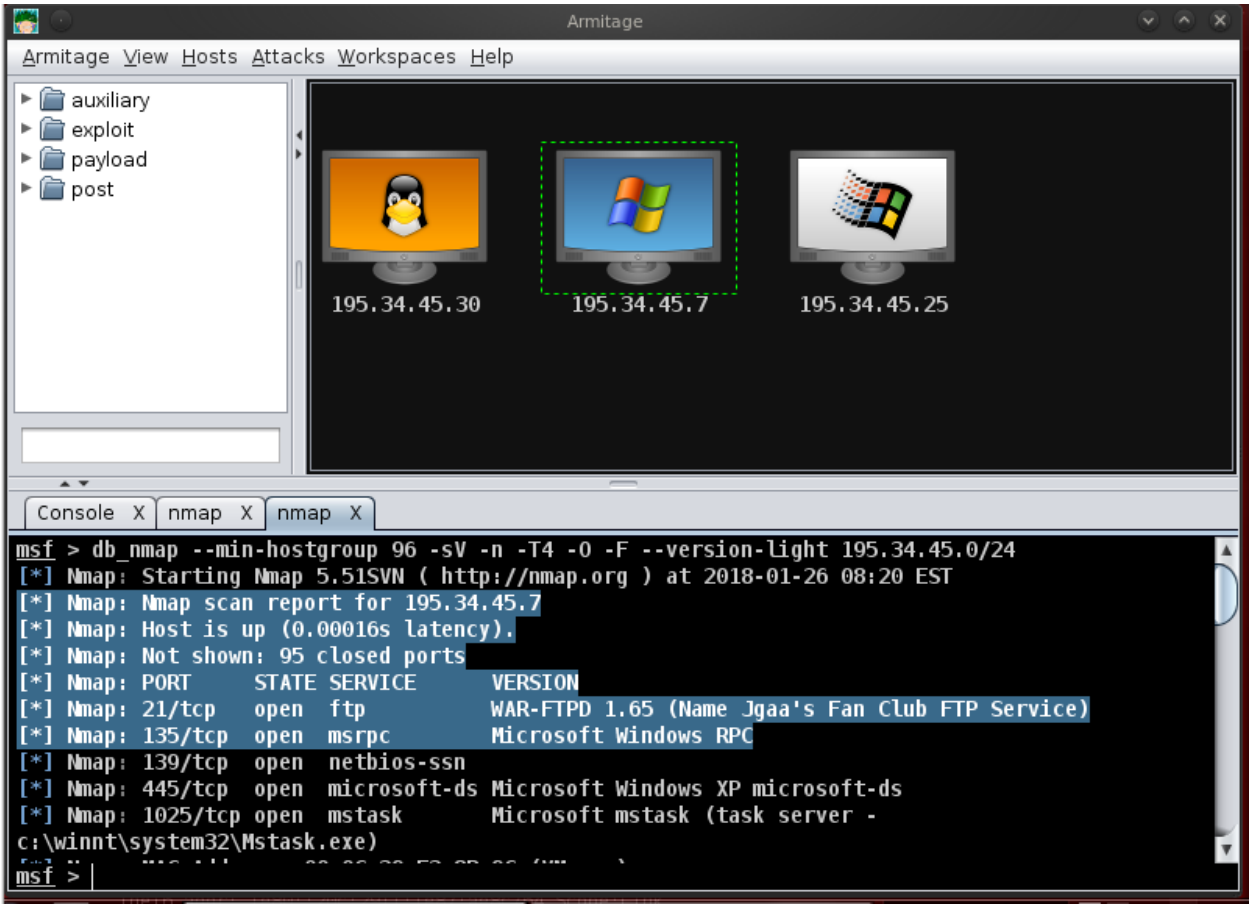
2. Le scan à trouver 3 machines différentes avec les huit derniers bits d'adresse 30, 7 et 25. nMap permet de détecter tous les ports ouverts dans la plage d'adresse à scanner. Une fois un port détecter, le programme affiche les machines visibles.



Exploitation de failles de sécurité connues

1. Dans le cas du système installé sur les machines de ce TP, nous avons remarqué que, sur une d'elle, le service rpcdcom écoute par défaut sur le port 135. Quelle est cette machine ?

Le scan à trouver le port 135 ouvert sur la machine avec l'adresse IP 195.34.45.7. C'est une machine qui roule le système d'exploitation Windows 2000 et qui correspond à la VM Québec.



2. Utilisez l'exploit ms03_026_dcom sur cette machine. Cet exploit profite d'une faille de sécurité dans le service rpcdcom. Quel est le résultat ? Quelles sont les nouvelles possibilités avec cette machine cible ?

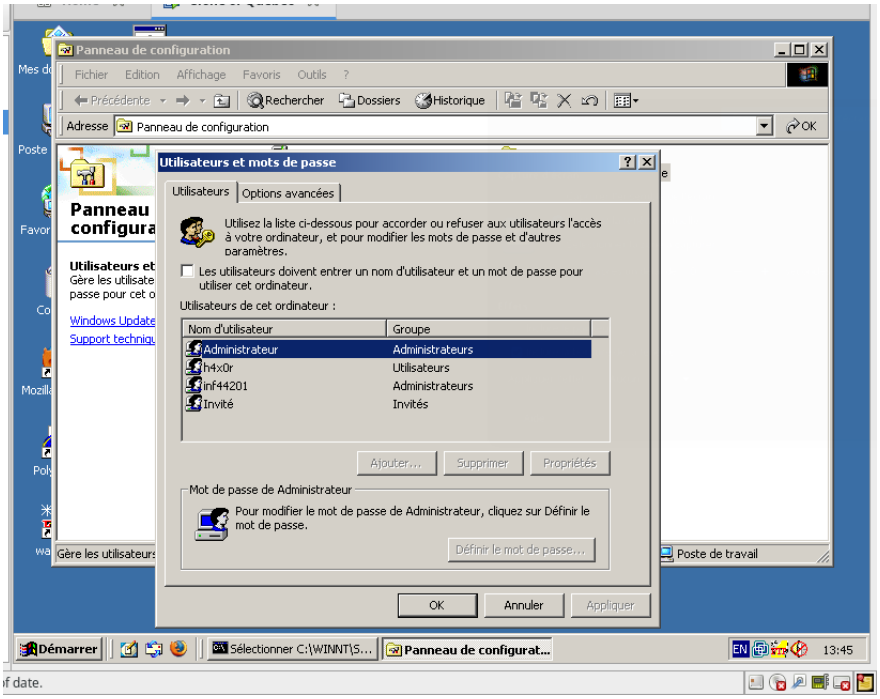
   La machine est devenue rouge dans l'interface graphique. Il est maintenant possible d'utiliser un explorateur de fichier et de parcourir les fichiers de la victime entre autres. Il est aussi possible d'avoir un invité de commande. Une nouvelle option est disponible qui se nomme meterpreter1 qui permet d'avoir accès à beaucoup d'option sur l'ordinateur de la victime.

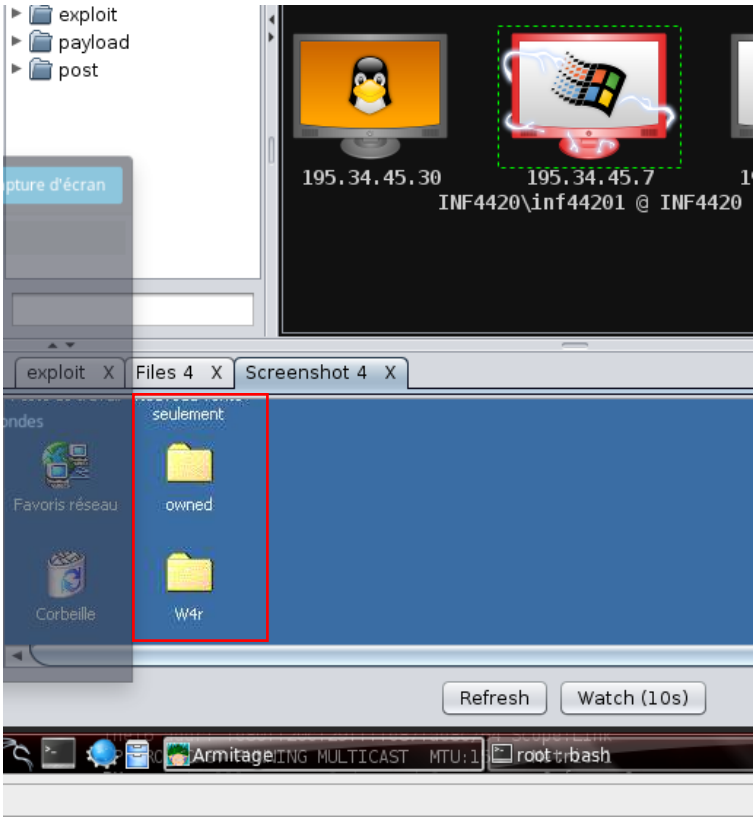
3. Grâce à l'exploit précédent ajoutez un utilisateur "h4x0r" avec le mot de passe "toto" sur la machine cible et créez un répertoire "owned" sur le bureau de l'utilisateur inf44201 (C:\Documents and Settings\inf44201\Bureau). Vous devez donner un listing des commandes que vous avez utilisées, une explication de votre démarche et une preuve des résultats (captures d'écran).

   Afin de créer un nouvel utilisateur, nous avons ouvert un invité de commande et entré la ligne :

net user h4x0r toto /ADD. Cela à créer un nouvel utilisateur nommé h4x0r avec le mot de passe toto. Pour créer un dossier sur le bureau, nous avons ouvert un explorateur de fichier et simplement ajouter le dossier sur le bureau.

```
C:\WINNT\system32> net user h4x0r toto /ADD
La commande s'est termin@e correctement.
```

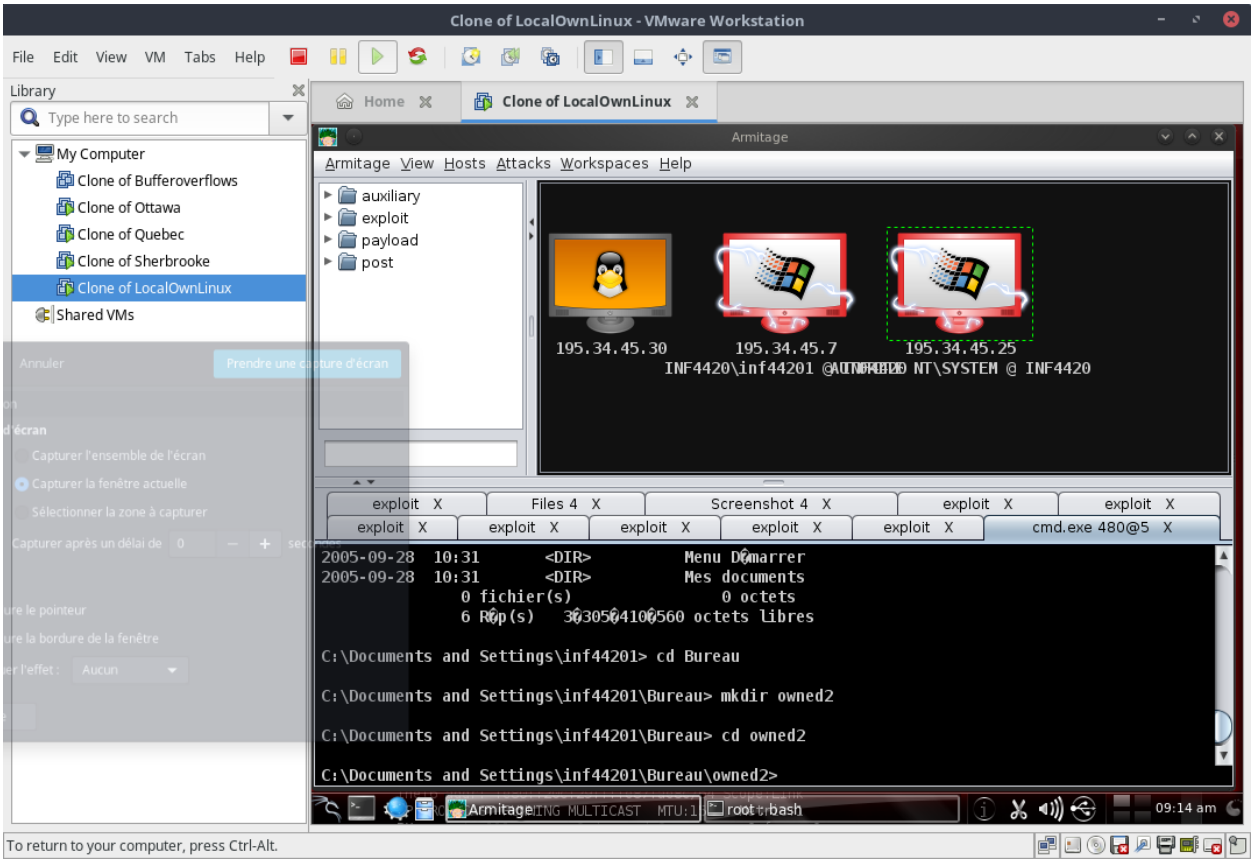




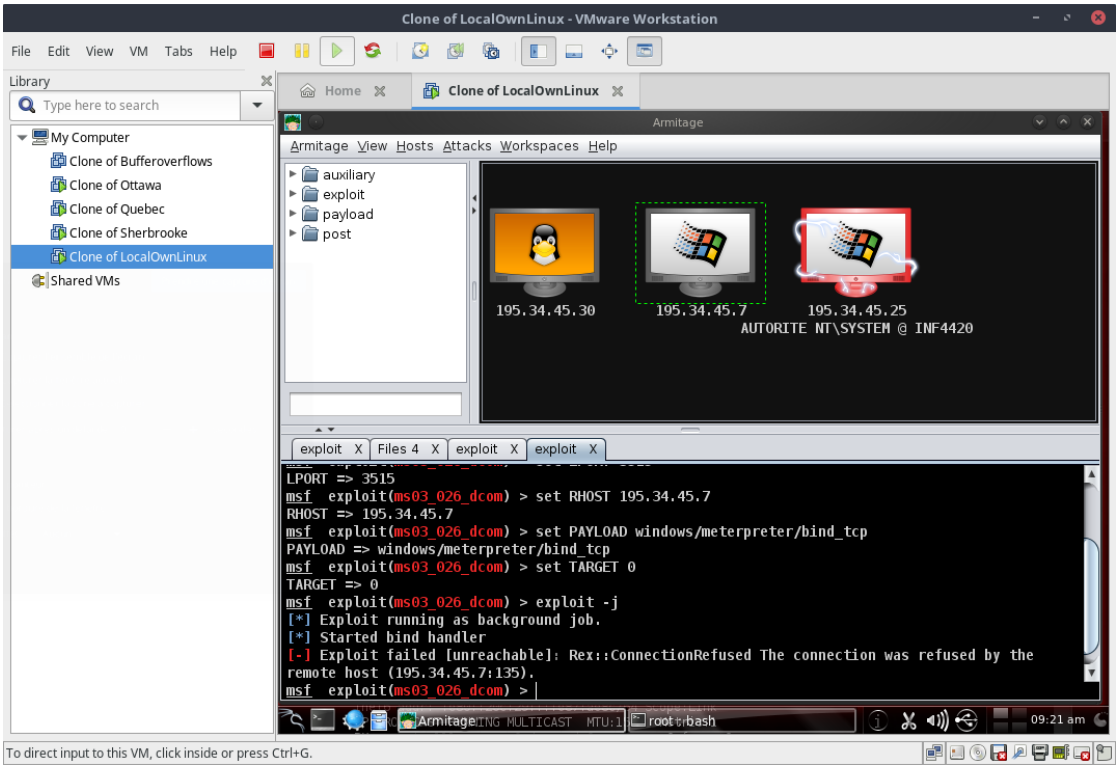
4. Une autre faille a aussi été détectée sur la même machine. Cette faille concerne le service WarFTPD et elle exploite le débordement de tampon dans la commande PASS de la version 1.65 de cette application. Comme vous le savez, WarFTPD est un serveur FTP utilisant par défaut le port 21. Trouvez cet exploit et utilisez-le afin de créer un répertoire sur la machine cible. Pour qu'il s'affiche il faut changer le « Exploit Rank » à « Poor » dans le menu « Armitage » et relancer la recherche des exploits.

   Comme pour l'attaque de la question précédente, nous avons tout simplement ouvert un explorateur de fichier et nous avons créé un fichier nommé W4R sur le bureau. La figure de la question précédente sert aussi de preuve de notre réussite à cette question.

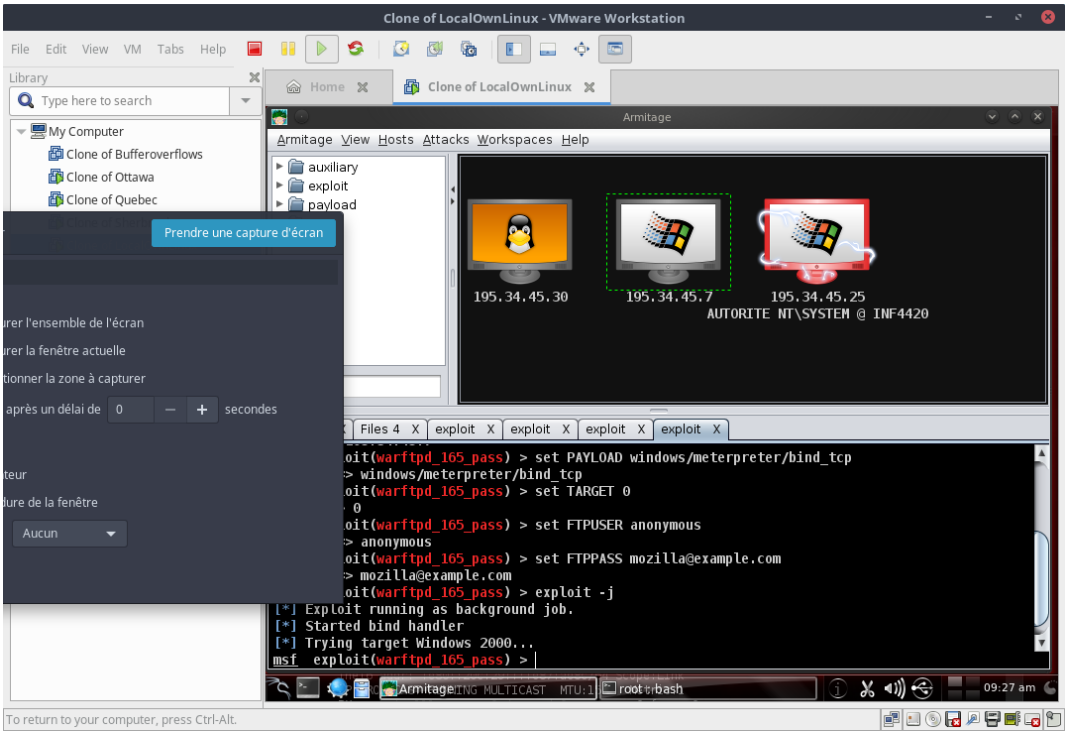
5. Pour cette attaque, nous avons utilisé l'exploit ms08_067_netapi sous l'onglet smb.
   List de commande :
   cd ../..
   cd "Documents and settings/inf44201/Bureau"
   mkdir owned2


Clone of LocalOwnLinux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Clone of Bufferoverflows
Clone of Ottawa
Clone of Quebec
Clone of Sherbrooke
Clone of LocalOwnLinux
Shared VMs
Home
Clone of LocalOwnLinux
Armitage
Armitage View Hosts Attacks Workspaces Help
auxiliary
exploit
payload
post
195.34.45.30
195.34.45.7
195.34.45.25
INF4420\inf44201 @
NT\SYSTEM @ INF4420
exploit X
Files 4 X
Screenshot 4 X
exploit X
exploit X
exploit X
exploit X
exploit X
exploit X
exploit X
cmd.exe 480@5 X
2005-09-28 10:31 <DIR> Menu D@marrer
2005-09-28 10:31 <DIR> Mes documents
0 fichier(s) 0 octets
6 R@p(s) 3@305@410@560 octets libres
C:\Documents and Settings\inf44201> cd Bureau
C:\Documents and Settings\inf44201\Bureau> mkdir owned2
C:\Documents and Settings\inf44201\Bureau> cd owned2
C:\Documents and Settings\inf44201\Bureau\owned2>
Armitage
root: bash
09:14 am
To return to your computer, press Ctrl-Alt.

6. Vérifier que l'exploit n'est plus réalisable (capture d'écran).



7. Redémarrez la machine et vérifier que l'exploit n'est plus réalisable (capture d'écran).



Question 5

Injection de SQL (SQLi)

1) Proposez une façon de vous loguer sur le site web avec le compte : *gigi*

   Pour se connecter avec le compte *gigi* il suffit d'entrer ses informations dans les champs suivant :

Champs	Valeurs entrées
user	gigi
password	' OR '1

   Ainsi requête donne le résultat :
   *select mem_code from MEMBRES where mem_login = '**gigi**' and mem_pwd = '' **OR '1**'*

   La requête SQL est toujours vrai grâce à la condition OR *'1'.* Donc le mot de passe n'est jamais pris en compte pour se connecter.

2) Proposez une façon pour passer à travers de la partie identification du site en supposant que vous ne connaissez aucun nom de compte

   Pour se connecter sans connaitre les identifiants il suffit de rentrer une condition encore toujours vraie comme par exemple :

Champs	Valeurs entrées
user	' OR '1' OR 'true
password	x

   *select mem_code from MEMBRES where mem_login = '' **OR '1**' **OR 'true**' and mem_pwd = '**x**'*

   Il s'agit du même principe que la question précédente.

3) Quelles failles dans le code avez-vous utilisées pour l'attaque 1 ? Et pour l'attaque 2 ?

   Les variables en PHP ne sont pas traitées et sont utilisées intégralement dans la requête. Donc, il suffit de mettre un guillemet et on peut commencer à insérer d'autres conditions et d'autres actions.

4) Corrigez les failles dans le code que l'admin de PolyVideo vous a envoyé et mettez le code corrigé dans le rapport.

   Il faut nettoyer les variables avec la méthode mysql_real_escape_string() qui ajoute les caractère « \ » devant les caractères suivant : NULL, \x00, \n, \r, \, ', " et \x1a

```
extract($_POST);

$Secure_login = mysql_real_escape_string($login);

$Secure_pass = mysql_real_escape_string($pass);

$req = "select mem_code from MEMBRES where mem_login = '$Secure_login' and
mem_pwd = '$Secure_pass'";

$result = mysql_query($req) or

die ("Error : the SQL request".$req."is not valid: ".mysql_error());

list($mem_code) = mysql_fetch_array($result);
```

Cross Site Scripting (XSS)

1) Comment avez-vous effectué l'attaque ?

   À la suite de la connexion nous avons tenté de modifier nos informations dans la section « Modification Information Personnel ». Dans le champs nom nous avons entré cette ligne de commande :

```
<script type="text/javascript">document.location.href="http://195.34.45.30/hacked.html"</script>
```

2) Quelles ont été les failles que vous avez utilisées ? Comment les corriger ?

   La faille que nous avons utilisée c'est que dans les champs POST du html de la section « Modification Information Personnel » permettent d'insérer du code JavaScript. Une fois la page chargée le code JavaScript s'exécute. Une façon de corriger le problème de XSS est d'utiliser une méthode qui se nomme « HTML Escaping ». Cette technique convertie les caractères potentiellement dangereux dans leur abréviation. Par exemple :

Caractère	Abréviation
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&#x27;
/	&#x2F;

   Ainsi le rendu HTML sera exactement ce que l'usager a entré mais le JavaScript ne sera pas exécuté.

Question 6

1) Donnez la séquence exacte de caractères à entrer. Expliquez brièvement comment votre « hack » fonctionne.

   Avec exactement 60 caractères aléatoires dans le champ username l'entré au site est toujours garantie. Il en est de même avec 100, 140 et 180 caractères aléatoires. Notre hack fonctionne en écrasant la valeur du nom d'utilisateur par une valeur arbitraire et en écrasant la valeur du mot de passe par le caractère de fin de ligne. Ainsi lorsque l'on se connecte avec par exemple 60 caractères, le nouveau nom d'utilisateur "root" est remplacé par les 20 derniers caractères entrés et le mot de passe est remplacé par "\0" qui est ajouté lorsque l'on tente de se connecter. Comme le montre le tableau suivant, les 40 premiers caractères servent à écraser les tableaux user_name et password.

   Voici comment est représenté la mémoire du programme.

Emplacement mémoire	Données
[0-19]	user_name[20]
[20-39]	password[20]
[40-59]	users[0][0] ("root")
[60-79]	users[0][1] ("98765")
[80-99]	users[1][0] ("moi")
[100-119]	users[1][1] ("allo")
[120-139]	users[2][0] ("abc")
[140-159]	users[2][1] ("motdepasse")
[160-179]	users[3][0] ("")
[180-199]	users[3][1] ("")

2) Que faudrait-il changer dans le programme pour enlever ce problème de sécurité?

   Premièrement en mettant un nombre de caractère maximum dans le champ username et password. Il serait aussi possible d'ajouter une variable canarie. Par exemple, en ajoutant une chaine de caractère directement après la déclaration du password et en vérifiant l'intégrité de cette variable avant chaque tentative de connexion.

Question 7

Notre hack exploite une vulnérabilité dans la fonction afficher. Lorsque le programme utilise la méthode scanf, tous les caractères dans le buffer d'entré sont pris en compte et il est donc possible de faire un buffer overflow. Il faut commencer par inscrire 20 caractères aléatoires afin de remplir la mémoire allouer au tableau pour le nom du fichier et ensuite inscrire un autre 8 caractères aléatoires afin de combler l'espace restante entre notre programme et l'endroit où est enregistrer la valeur du registre EIP. Les 4 prochains caractères qui sont inscrit correspondent donc à l'adresse de la prochaine instruction à exécuter (dans notre cas : 004010E0). On commence par entrer l'adresse par 0xE0 qui correspond au caractère « ù » dans le code ascii.

L'utilisation de la méthode fgets en spécifiant le bon nombre de caractère à prendre en entré à la place de la méthode scanf serait une façon de régler le problème de sécurité.