

Université Cheikh Anta Diop
École Supérieure Polytechnique
Département Génie Informatique
Année Universitaire 2023-2024
DIC2 Option : Sécurité des Systèmes d'Information (SSI)
Atelier 02 : Sécurité Locale
Dr Mandicou BA

Mame Mbaye KANE
Mouhamadou Moustapha DIONE

Tâche 1: Casser des mots de passe via un simple script et un dictionnaire maison

1. Créer plusieurs comptes utilisateur avec des ou sans mot de passe

```
mandicou@mandicou-VirtualBox:~$ sudo adduser toto
[sudo] Mot de passe de mandicou :
Ajout de l'utilisateur « toto » ...
Ajout du nouveau groupe « toto » (1001) ...
Ajout du nouvel utilisateur « toto » (1001) avec le groupe « toto » ...
Création du répertoire personnel « /home/toto »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
Les mots de passe ne correspondent pas.
passwd : Erreur de manipulation du jeton d'authentification
Mot de passe non changé
Essayer à nouveau ? [o/N] o
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : le mot de passe a été mis à jour avec succès
Modification des informations relatives à l'utilisateur toto
Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée
    Nom complet []:
    N° de bureau []:
    Téléphone professionnel []:
    Téléphone personnel []:
    Autre []:
Ces informations sont-elles correctes ? [O/n]
```

```
mandicou@mandicou-VirtualBox:~$ sudo adduser toto
[sudo] Mot de passe de mandicou :
Ajout de l'utilisateur « toto » ...
Ajout du nouveau groupe « toto » (1001) ...
Ajout du nouvel utilisateur « toto » (1001) avec le groupe « toto » ...
Création du répertoire personnel « /home/toto »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
Les mots de passe ne correspondent pas.
passwd : Erreur de manipulation du jeton d'authentification
Mot de passe non changé
Essayer à nouveau ? [o/N] o
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : le mot de passe a été mis à jour avec succès
Modification des informations relatives à l'utilisateur toto
Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée
    Nom complet []:
    N° de bureau []:
    Téléphone professionnel []:
    Téléphone personnel []:
    Autre []:
Ces informations sont-elles correctes ? [O/n]
```

2. Créer un dictionnaire

```
mandicou@mandicou-VirtualBox:~$ cat dict.txt
marcher
courir
aller
vent
air
plume
passer
plus
avantage
mer

pluie
dinosaur
mouton
plat
12345
```

3. Créer un script (perl) d'attaque

Le script en python :

```
mandicou@mandicou-VirtualBox:~$ cat password_cracker.py
import crypt

def load_hashes(hash_file):
    """
    Charge les paires username:hash depuis le fichier hash_file.
    """
    hashes = {}
    with open(hash_file, 'r') as file:
        for line in file:
            username, hash_value = line.strip().split(':', 1)
            hashes[username] = hash_value
    print(hashes)
    return hashes

def load_dictionary(dictionary_file):
    """
    Charge les mots du dictionnaire depuis dictionary_file.
    """
    with open(dictionary_file, 'r') as file:
        return [line.strip() for line in file]

def crack_passwords(hashes, dictionary):
    """
    Tente de casser les mots de passe en utilisant le dictionnaire.
    """
    cracked = {}
    for username, hash_value in hashes.items():
        salt = '$'.join(hash_value.split('$')[:3]) # Extrait le sel
        for password in dictionary:
            hashed_password = crypt.crypt(password, salt)
            if hashed_password == hash_value:
                cracked[username] = password
                print(f"[SUCCESS] Username: {username} | Password: {password}")
                break
    return cracked

if __name__ == "__main__":
    hash_file = 'shadow_copy.txt'
    dictionary_file = 'dict.txt'

    hashes = load_hashes(hash_file)
    dictionary = load_dictionary(dictionary_file)

    cracked_passwords = crack_passwords(hashes, dictionary)

    if not cracked_passwords:
        print("No passwords were cracked.")
    else:
        print("\nCracked passwords:")
        for username, password in cracked_passwords.items():
            print(f"Username: {username} | Password: {password}")
```

le script en perl :

```
#!/usr/bin/perl
```

```
use strict;
```

```
use warnings;
```

```
use Crypt::Passwd::XS;
```

```
sub load_hashes {
```

```
    my ($hash_file) = @_;
```

```
    my %hashes;
```

```
    open(my $fh, '<', $hash_file) or die "Could not open file '$hash_file' $!";
```

```
    while (my $line = <$fh>) {
```

```
        chomp $line;
```

```
        my ($username, $hash_value) = split(':', $line, 2);
```

```
        $hashes{$username} = $hash_value;
```

```
    }
```

```
    close($fh);
```

```
    return %hashes;
```

```
}
```

```
sub load_dictionary {
```

```
    my ($dictionary_file) = @_;
```

```
    my @dictionary;
```

```
    open(my $fh, '<', $dictionary_file) or die "Could not open file '$dictionary_file' $!";
```

```
    while (my $line = <$fh>) {
```

```
        chomp $line;
```

```
        push @dictionary, $line;
```

```
    }
```

```
    close($fh);
```

```
    return @dictionary;
```

```
}
```

```
sub crack_passwords {
```

```
    my ($hashes, $dictionary) = @_;
```

```
    my %cracked;
```

```
    foreach my $username (keys %$hashes) {
```

```
        my $hash_value = $hashes->{$username};
```

```

my ($id, $salt, $rest) = (split(/\$/, $hash_value))[1..3];
my $full_salt = join('$', '$' . $id, $salt);

foreach my $password (@$dictionary) {
    my $hashed_password = crypt($password, $full_salt);
    if ($hashed_password eq $hash_value) {
        $cracked{$username} = $password;
        print "[SUCCESS] Username: $username | Password: $password\n";
        last;
    }
}
}
return %cracked;
}

my $hash_file = 'shadow_copy.txt';
my $dictionary_file = 'dict.txt';

my %hashes = load_hashes($hash_file);
my @dictionary = load_dictionary($dictionary_file);

my %cracked_passwords = crack_passwords(\%hashes, \@dictionary);

if (!%cracked_passwords) {
    print "No passwords were cracked.\n";
} else {
    print "\nCracked passwords:\n";
    foreach my $username (keys %cracked_passwords) {
        print "Username: $username | Password: $cracked_passwords{$username}\n";
    }
}
}

```

4. Proposer un scénario de lancement de l'attaque

Pour le scénario d'attaque, nous avons écrit un code qui utilise deux fichiers:
dict.txt => qui regroupe une ensemble de mot de passe possible pour nos utilisateurs.
shadow_copy.txt => qui regroupe l'ensemble des utilisateurs sur lesquels nous voulons faire l'attaque et le hash de leur mot de passe correspondant comme dans le format du fichier **/etc/shadow**.

Ainsi pour chaque utilisateur, nous allons haché chaque mot de passe de notre dictionnaire et faire la comparaison entre le haché qui est dans le fichier shadow_copy. Si le hashé n'est pas le même, on en déduit que le mot de passe choisi n'est pas le bon. Par contre, si les hachés correspondent on en déduit que le mot de passe correspondant est celui de l'utilisateur choisi.

Ainsi on se base sur la propriété de **Déterminisme** des fonctions de hachage.


```
mandicou@mandicou-VirtualBox:~$ python3 password_cracker.py
[SUCCESS] Username: toto | Password: passer
[SUCCESS] Username: test | Password: 12345

Cracked passwords:
Username: toto | Password: passer
Username: test | Password: 12345
mandicou@mandicou-VirtualBox:~$ ./password_cracker.pl
[SUCCESS] Username: toto | Password: passer
[SUCCESS] Username: test | Password: 12345

Cracked passwords:
Username: toto | Password: passer
Username: test | Password: 12345
```

Tâche 2: Casser des mots de passe avec John The Ripper

1. Télécharger John à partir de son site officiel (www.openwall.com/john) et l'installer (Le compilateur doit être présent, sinon il faut l'installer)



[Products](#)[Services](#)[Publications](#)[Resources](#)[What's new](#)

John the Ripper password cracker

John the Ripper is an Open Source password security auditing and password recovery tool available for many operating systems. **John the Ripper jumbo** supports types, including for: user passwords of Unix flavors (Linux, *BSD, Solaris, AIX, QNX, etc.), macOS, Windows, "web apps" (e.g., WordPress), groupware (e.g., Novell Groupware), servers (SQL, LDAP, etc.); network traffic captures (Windows network authentication, WiFi WPA-PSK, etc.); encrypted private keys (SSH, GnuPG, cryptocurrency wallets, etc.); macOS .dmg files and "sparse bundles", Windows BitLocker, etc.; archives (ZIP, RAR, 7z), and document files (PDF, Microsoft Office's, etc.) These are just a few of the many more.

Follow @Openwall on Twitter for new release announcements and other news

John the Ripper is free and Open Source software, distributed primarily in source code form. If you would rather use a commercial product, please consider [John the Ripper Pro](#) primarily in the form of "native" packages for the target operating systems and in general is meant to be easier to install and use while delivering optimal performance.

Proceed to **John the Ripper Pro** homepage for your OS:

- [John the Ripper Pro for Linux](#)
- [John the Ripper Pro for macOS](#)
- On Windows, consider [Hash Suite](#) (developed by a contributor to John the Ripper)
- On Android, consider [Hash Suite Droid](#)

Download the latest John the Ripper jumbo release ([release notes](#)) or development snapshot:

- 1.9.0-jumbo-1 sources in [tar.xz](#), 33 MB (signature) or [tar.gz](#), 43 MB (signature)
- 1.9.0-jumbo-1 64-bit Windows binaries in [7z](#), 22 MB (signature) or [zip](#), 63 MB (signature)
- 1.9.0-jumbo-1 32-bit Windows binaries in [7z](#), 21 MB (signature) or [zip](#), 61 MB (signature)
- Development source code in [GitHub repository](#) (download as [tar.gz](#) or [zip](#))

(a) Télécharger le logiciel et le démarrer.

```
moustapha@moustapha-virtual-machine:~$ wget https://www.openwall.com/john/k/john-1.9.0-jumbo-1.tar.gz
--2024-06-04 02:17:56-- https://www.openwall.com/john/k/john-1.9.0-jumbo-1.tar.gz
Résolution de www.openwall.com (www.openwall.com)... 193.110.157.242
Connexion à www.openwall.com (www.openwall.com) [193.110.157.242]:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 44303366 (42M) [application/octet-stream]
Enregistre : «john-1.9.0-jumbo-1.tar.gz»

john-1.9.0-jumbo-1. 100%[=====>] 42,25M 5,10MB/s ds 40s

2024-06-04 02:18:36 (1,06 MB/s) - «john-1.9.0-jumbo-1.tar.gz» enregistré [44303366/44303366]

moustapha@moustapha-virtual-machine:~$ ls
''$'\004'      compilation_example.o      Musique
a.out          docker-compose.yml        overflow.c
bonarien.c    Documents                 plone
bonarien.o    'echo \'                  Public
bonarien.s    Images                    Téléchargements
Bureau        john-1.9.0-jumbo-1.tar.gz  Vidéos
compilation_example.c  Modèles

moustapha@moustapha-virtual-machine:~$ tar -xvf john-1.9.0-jumbo-1.tar.gz
john-1.9.0-jumbo-1/.ci/Dockerfile
john-1.9.0-jumbo-1/.ci/disable_formats.sh
john-1.9.0-jumbo-1/.circleci/circle-ci.sh
```

On l'a installé et dézippé

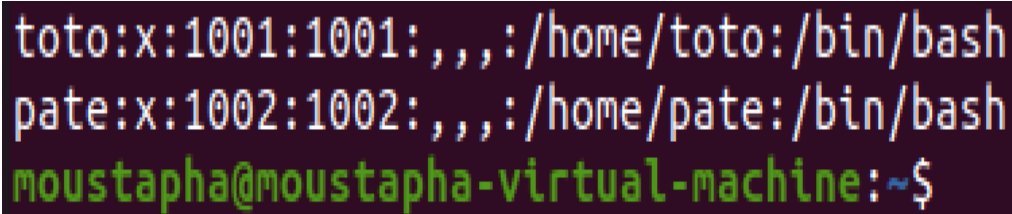
(b) Patcher le logiciel pour inclure la gestion de sha512.

Y a de patch pour cette version

(c) Compiler le logiciel.

fait.

2. Créer des comptes utilisateur avec de mauvais mots de passe ou sans mot de passe (cf. tâche précédente).

A terminal window with a dark purple background. It shows two commands being entered to create users: 'toto:x:1001:1001:,,,:/home/toto:/bin/bash' and 'pate:x:1002:1002:,,,:/home/pate:/bin/bash'. The prompt 'moustapha@moustapha-virtual-machine:~\$' is visible at the bottom.

```
toto:x:1001:1001:,,,:/home/toto:/bin/bash
pate:x:1002:1002:,,,:/home/pate:/bin/bash
moustapha@moustapha-virtual-machine:~$
```

3. Créer le fichier des mots de passe.


```
moustapha@moustapha-virtual-machine:~$ sudo unshadow /etc/passwd /etc/shadow >
motdepasse.txt
moustapha@moustapha-virtual-machine:~$
```

on voit les deux utilisateurs et leurs mots de passe hachés tout à fait en bas

```
moustapha@moustapha-virtual-machine:~$ john motdepasse.txt
Created directory: /home/moustapha/.john
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:32 1% 2/3 0g/s 155.2p/s 313.1c/s 313.1C/s ncc1701d..1022
```

fait

(b) Créer son propre dictionnaire.

```
GNU nano 4.8                                dictionnaire
passer123
bonjour
password
sys
mame
```

(c) Mener l'attaque

```
moustapha@moustapha-virtual-machine:~$ john -wordlist=dictionnaire motdepasse.txt
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
passer123      (toto)
passer123      (pate)
2g 0:00:00:00 100% 25.00g/s 62.50p/s 187.5c/s 187.5C/s passer123..mame
Use the "--show" option to display all of the cracked passwords reliably
Session completed
moustapha@moustapha-virtual-machine:~$
```

6. Mener une attaque exhaustive (temps de recherche infini !).

L'attaque exhaustive, également connue sous le nom de **brute-force**, est une méthode utilisée pour tenter toutes les combinaisons possibles de mots de passe jusqu'à ce que le bon soit trouvé.

```
moustapha@moustapha-virtual-machine:~$ john --incremental motdepasse.txt
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:33 0g/s 335.5p/s 335.5c/s 335.5C/s 033205..083224
```

7. Mener Attaque complète

Une attaque complète avec John the Ripper est une combinaison d'attaques par dictionnaire et par **règles**.

Les **règles** permettent d'appliquer des transformations aux mots de passe du dictionnaire pour générer des variations. Par exemple, les règles peuvent inverser les mots, ajouter des chiffres à la fin, remplacer des lettres par des caractères spéciaux, etc.

```

moustapha@moustapha-virtual-machine:~$ john --wordlist=dictionnaire --rules motdepasse.txt
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 100% 0g/s 344.8p/s 344.8c/s 344.8C/s Bonjoured..Maming
Session completed
moustapha@moustapha-virtual-machine:~$ john --show
Password files required, but none specified
moustapha@moustapha-virtual-machine:~$ john --show motdepasse.txt
toto:passer123:1001:1001:,,,:/home/toto:/bin/bash
pate:passer123:1002:1002:,,,:/home/pate:/bin/bash

2 password hashes cracked, 1 left
moustapha@moustapha-virtual-machine:~$ █

```

8. Mesurer la rapidité de John

```

moustapha@moustapha-virtual-machine:~$ john --test
Benchmarking: descrypt, traditional crypt(3) [DES 128/128 SSE2-16]... DONE
Many salts: 4437K c/s real, 4518K c/s virtual
Only one salt: 4366K c/s real, 4401K c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES 128/128 SSE2-16]... DONE
Many salts: 149529 c/s real, 150735 c/s virtual
Only one salt: 148864 c/s real, 150064 c/s virtual

Benchmarking: md5crypt [MD5 32/64 X2]... DONE
Raw: 14293 c/s real, 14350 c/s virtual

Benchmarking: bcrypt ("S2a$05", 32 iterations) [Blowfish 32/64 X2]... ^CWait...
Session aborted
moustapha@moustapha-virtual-machine:~$ █

```

Tâche 3 : La sécurité de connexion, le password aping

1. Créer un compte guest et on lui affecter un mot de passe

```
1 guest:guest:::don't have an account:/home/guest:/bin/bash
mandicou@mandicou-VirtualBox:~$ sudo newusers users1.txt
```

2. Visualiser le mot de passe de l'utilisateur.

- (a) En étant connecté sous un compte utilisateur.

```
mandicou@mandicou-VirtualBox:~$ su guest
Mot de passe :
guest@mandicou-VirtualBox:/home/mandicou$ cat /etc/shadow
cat: /etc/shadow: Permission non accordée
```

- (b) En étant connecté sous le compte de l'administrateur

```
mandicou@mandicou-VirtualBox:~$ su guest
Mot de passe :
guest@mandicou-VirtualBox:/home/mandicou$ cat /etc/shadow
cat: /etc/shadow: Permission non accordée
```

3. Changer la période de validité du mot de passe (on la limite a 30 jours)

```
mandicou@mandicou-VirtualBox:~$ sudo chage -M 30 guest
```

4. Limiter la durée de vie d'un compte puis éliminer cette limite.

```
mandicou@mandicou-VirtualBox:~$ sudo chage -E 2024-12-31 guest
mandicou@mandicou-VirtualBox:~$ sudo chage -l guest
Dernier changement de mot de passe           : suw 03, 2024
Fin de validité du mot de passe              : sul 03, 2024
Mot de passe désactivé                       : jamais
Fin de validité du compte                    : des 31, 2024
Nombre minimum de jours entre les changements de mot de passe : 0
Nombre maximum de jours entre les changements de mot de passe : 30
Nombre de jours d'avertissement avant la fin de validité du mot de passe : 7
mandicou@mandicou-VirtualBox:~$ sudo chage -E -1 guest
mandicou@mandicou-VirtualBox:~$ sudo chage -l guest
Dernier changement de mot de passe           : suw 03, 2024
Fin de validité du mot de passe              : sul 03, 2024
Mot de passe désactivé                       : jamais
Fin de validité du compte                    : jamais
Nombre minimum de jours entre les changements de mot de passe : 0
Nombre maximum de jours entre les changements de mot de passe : 30
Nombre de jours d'avertissement avant la fin de validité du mot de passe : 7
```

5. Empêcher l'utilisateur de modifier son mot de passe:

```
mandicou@mandicou-VirtualBox:~$ sudo passwd -l guest
passwd : expiration du mot de passe modifiée.
mandicou@mandicou-VirtualBox:~$ su guest
Mot de passe :
su: Échec de l'authentification
```

6. Changer le mot de passe d'un utilisateur:

```
mandicou@mandicou-VirtualBox:~$ sudo passwd guest
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : le mot de passe a été mis à jour avec succès
```

7. Verrouiller un compte, afficher son état, déverrouiller le compte:

```
mandicou@mandicou-VirtualBox:~$ sudo passwd -l guest
passwd : expiration du mot de passe modifiée.
mandicou@mandicou-VirtualBox:~$ sudo passwd -S guest
guest L 06/03/2024 0 30 7 -1
mandicou@mandicou-VirtualBox:~$ sudo passwd -u guest
passwd : expiration du mot de passe modifiée.
mandicou@mandicou-VirtualBox:~$ sudo passwd -S guest
guest P 06/03/2024 0 30 7 -1
```

8. Supprimer le mot de passe de l'utilisateur. Essayer de se connecter. Remettre un mot de passe :

```
mandicou@mandicou-VirtualBox:~$ sudo passwd -d guest
passwd : expiration du mot de passe modifiée.
mandicou@mandicou-VirtualBox:~$ su guest
guest@mandicou-VirtualBox:/home/mandicou$ su mandicou
Mot de passe :
mandicou@mandicou-VirtualBox:~$ sudo passwd guest
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : le mot de passe a été mis à jour avec succès
```

9. Visualiser les dernières connexions qui ont réussi/échoué:

```
mandicou@mandicou-VirtualBox:~$ grep 'Accepted' /var/log/auth.log
mandicou@mandicou-VirtualBox:~$ grep 'Failed' /var/log/auth.log
Jun  2 00:55:22 mandicou-VirtualBox dbus-daemon[565]: [system] Failed to activate service 'org.b
luez': timed out (service_start_timeout=25000ms)
Jun  2 00:55:56 mandicou-VirtualBox dbus-daemon[565]: [system] Failed to activate service 'org.b
luez': timed out (service_start_timeout=25000ms)
```

10. Visualiser pour chaque compte, la dernière connexion

```
mandicou@mandicou-VirtualBox:~$ sudo lastlog | grep guest
```

11. Visualiser les valeurs par défaut du password aging :

```
GNU nano 4.8 /etc/login.defs
#
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
```

12. Générer des mots de passe aléatoires :

on a utilisé **pwgen** avec les options qui nous permettent de générer un mot de passe aléatoire de 12 caractères :

```
mandicou@mandicou-VirtualBox:~$ pwgen 12 1
Uwae1kaekoex
```

13. Automatiser les changements de mot de passe :

```
#!/bin/bash
```

```
Nouveau_mdp=$(pwgen 12 1)
```

```
echo "guest:$Nouveau_mdp" | sudo chpasswd
```

```
echo "New password for guest is $Nouveau_mdp"
```

Tâche 4 : Les droits d'endossement

1. On crée un mini-shell :

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>

int main() {
    int status;
    char cmd[256], arg[256];
    for(;;){
        printf("==> Command ?");
        fgets(cmd, sizeof(cmd), stdin);
        // Remove the trailing newline from cmd
        cmd[strcspn(cmd, "\n")] = '\0';
        printf("---> Argument ?");
        fgets(arg, sizeof(arg), stdin);
        // Remove the trailing newline from arg
        arg[strcspn(arg, "\n")] = '\0';
        if(fork() == 0){
            execlp(cmd, cmd, arg, NULL);
        } else {
            wait(&status);
        }
    }
}
```

2. Donnons les droit SUID à ce mini-shell et le rendre accessible aux utilisateurs :

```
mandicou@mandicou-VirtualBox:~$ gcc -o msh msh.c
mandicou@mandicou-VirtualBox:~$ sudo chmod +s msh
mandicou@mandicou-VirtualBox:~$ sudo chmod u+x msh
mandicou@mandicou-VirtualBox:~$ ls -l msh*
-rwsrwsr-x 1 mandicou mandicou 16952 suw  4 16:06 msh
```

3. Faisons qu'un utilisateur active le mini-shell et procède les prérogatives de root grâce à lui :

```
mandicou@mandicou-VirtualBox:~$ su guest
Mot de passe :
guest@mandicou-VirtualBox:/home/mandicou$ ./msh
==> Command ?sudo touch test
---> Argument ?
```

4. Recherchons les executables possédant des droits d'endossement :

```
mandicou@mandicou-VirtualBox:~$ sudo find / -type f -perm /4000 -o -perm /2000 -print
find: '/proc/20230/task/20230/fd/6': Aucun fichier ou dossier de ce type
find: '/proc/20230/task/20230/fdinfo/6': Aucun fichier ou dossier de ce type
find: '/proc/20230/fd/5': Aucun fichier ou dossier de ce type
find: '/proc/20230/fdinfo/5': Aucun fichier ou dossier de ce type
find: '/run/user/1000/doc': Permission non accordée
find: '/run/user/1000/gvfs': Permission non accordée
/run/log/journal
/var/crash
/var/metrics
/var/local
/var/log/journal
/var/log/journal/2d4d4dcbbdc476c9333f9ab459ba7fc
/var/mail
/snap/core20/2264/usr/bin/chage
/snap/core20/2264/usr/bin/expiry
/snap/core20/2264/usr/bin/ssh-agent
/snap/core20/2264/usr/bin/wall
/snap/core20/2264/usr/sbin/pam_extrausers_chkpwd
/snap/core20/2264/usr/sbin/unix_chkpwd
/snap/core20/2264/var/mail
/snap/core20/2318/usr/bin/chage
/snap/core20/2318/usr/bin/expiry
/snap/core20/2318/usr/bin/ssh-agent
/snap/core20/2318/usr/sbin/pam_extrausers_chkpwd
/snap/core20/2318/usr/sbin/unix_chkpwd
/snap/core20/2318/var/mail
/usr/sbin/pam_extrausers_chkpwd
/usr/sbin/unix_chkpwd
/usr/libexec/camel-lock-helper-1.2
/usr/share/ppd/custom
/usr/bin/bsd-write
/usr/bin/crontab
/usr/bin/expiry
/usr/bin/chage
/usr/bin/ssh-agent
/usr/local/share/fonts
/usr/local/lib/python3.8
/usr/local/lib/python3.8/dist-packages
/etc/chatscripts
/etc/ppp/peers
```


Tâche 5 : Utiliser sudo

1. Créons un compte :

```
mandicou@mandicou-VirtualBox:~$ cat users2.txt
p LibreOffice Writer don't have an account:/home/pierre:/bin/bash
mandicou@mandicou-VirtualBox:~$ sudo newusers users2.txt
[sudo] Mot de passe de mandicou :
```

2. Modifions la configuration de sorte que pierre peut créer des comptes utilisateurs :

```
mandicou@mandicou-VirtualBox:~$ sudo usermod -aG sudo pierre
[sudo] Mot de passe de mandicou :
```

3. Connectons nous sous le compte de pierre et créons des users :

```
mandicou@mandicou-VirtualBox:/home/pierre$ su pierre
Mot de passe :
pierre@mandicou-VirtualBox:~$ pwd
/home/pierre
pierre@mandicou-VirtualBox:~$ nano user.txt
pierre@mandicou-VirtualBox:~$ sudo newusers user.txt
pierre@mandicou-VirtualBox:~$ cat user.txt
modou:modou:::don't have an account:/home/modou:/bin/bash
diery:diery:::mouse:/home/diery:/bin/bash
pierre@mandicou-VirtualBox:~$
```

4. Proposons une configuration plus réaliste que ce qui a été fait précédemment :
5. Créons des admins sur la base de la configuration précédente:
6. Testons :

Tâche 6: La gestion des utilisateurs et des droits : cas d'un serveur web

1. Modifier la configuration d'Apache de sorte que l'application Apache s'exécute sous le compte utilis cochise et sous le compte groupe indien. L'arborescence /reserve contiendra notre site Web

Créons d'abord le groupe « indien » et l'utilisateur « cochise » puis l'arborescence /reserve

```
moustapha@moustapha-virtual-machine:~$ sudo addgroup indien
Ajout du groupe « indien » (GID 1004)...
Fait.
moustapha@moustapha-virtual-machine:~$ sudo adduser --home /reserve --ingroup indien cochise
Ajout de l'utilisateur « cochise » ...
Ajout du nouvel utilisateur « cochise » (1003) avec le groupe « indien » ...
Création du répertoire personnel « /reserve »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : le mot de passe a été mis à jour avec succès
Modification des informations relatives à l'utilisateur cochise
Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée
    Nom complet []:
    N° de bureau []:
    Téléphone professionnel []:
    Téléphone personnel []:
    Autre []:
Ces informations sont-elles correctes ? [0/n]
moustapha@moustapha-virtual-machine:~$
```

Éditons le fichier de configuration principal d'Apache : **sudo nano**
/etc/apache2/apache2.conf

```
GNU nano 4.8 /etc/apache2/apache2.conf Modifié
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5

# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
User cochise
Group indien
```

On ajoute ceci à la fin du fichier

```
ServerName localhost
<Directory /reserve>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
DocumentRoot /reserve
```

2. Créer les comptes

```
moustapha@moustapha-virtual-machine:~$ sudo useradd -m -G indien mandicou
moustapha@moustapha-virtual-machine:~$ sudo useradd -m -G indien mandicou2
moustapha@moustapha-virtual-machine:~$
```

3. Créer un site

```
moustapha@moustapha-virtual-machine:/etc/apache2/sites-available$ echo "<html><body><h1>welcome to our site </h1></body></html>" | s
udo tee /reserve/index.html
<html><body><h1>welcome to our site </h1></body></html>
moustapha@moustapha-virtual-machine:/etc/apache2/sites-available$
```

4. Vérifier la syntaxe et activer le serveur

```
moustapha@moustapha-virtual-machine:~$ sudo apachectl configtest
Syntax OK
moustapha@moustapha-virtual-machine:~$ sudo systemctl restart apache2
moustapha@moustapha-virtual-machine:~$
```

5. Faire en sorte que les concepteurs de pages travaillent (ils créent des pages).

```
moustapha@moustapha-virtual-machine:~$ sudo chown -R cochise:indien /reserve
moustapha@moustapha-virtual-machine:~$ sudo chmod -R 775 /reserve
moustapha@moustapha-virtual-machine:~$ sudo usermod -aG indien cochise
moustapha@moustapha-virtual-machine:~$ sudo usermod -aG indien mandicou
moustapha@moustapha-virtual-machine:~$ sudo usermod -aG indien mandicou2
moustapha@moustapha-virtual-machine:~$
```

6. Faire en sorte les pages sont protégées contre les modifications non autorisées.

```
sudo chmod -R 755 /reserve
```

7. Faire en sorte qu'un concepteur crée un document inaccessible aux autres.

```
moustapha@moustapha-virtual-machine:~$ sudo -u cochise touch /reserve/inaccessible.html
moustapha@moustapha-virtual-machine:~$ sudo -u cochise chmod 700 /reserve/inaccessible.html
moustapha@moustapha-virtual-machine:~$
```

8. Faire une copie avec préservation.

```
moustapha@moustapha-virtual-machine:~$ sudo cp -p /reserve/index.html /reserve/index_copy.html
moustapha@moustapha-virtual-machine:~$
```

9. Faire en sorte qu'un non-concepteur n'a pas accès au site Web.

```
GNU nano 4.8                                000-default.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

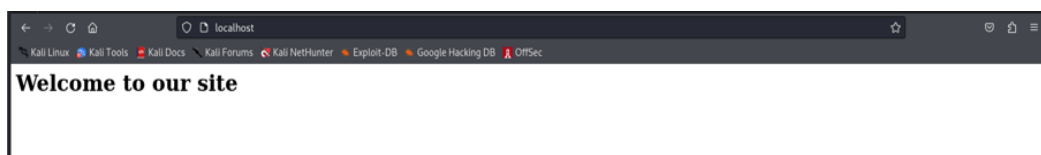
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
<Directory "/reserve">
AllowOverride All
Require group indien
</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

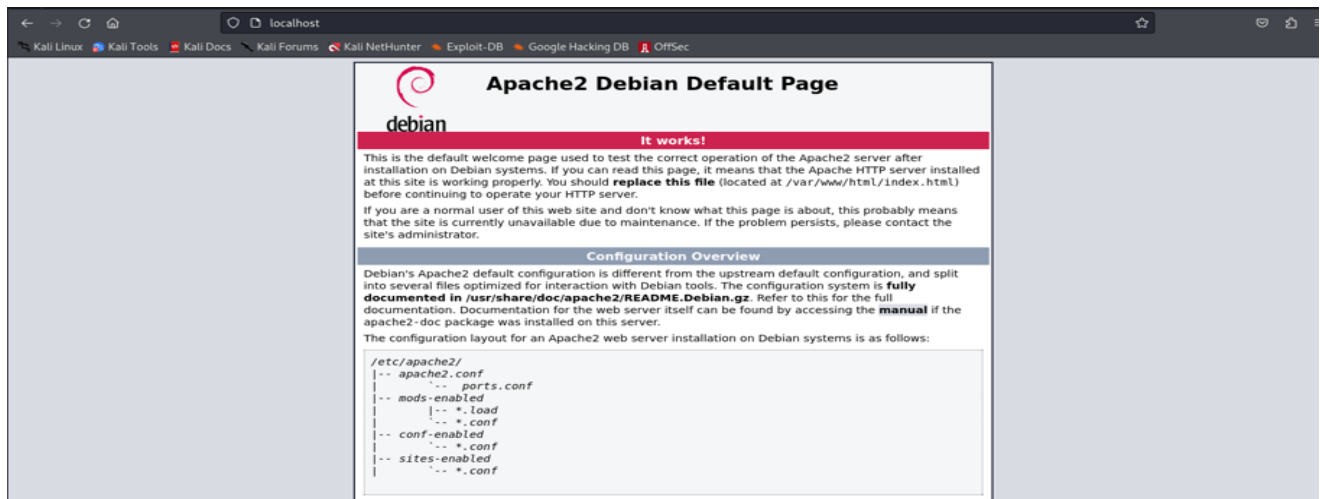
10. Accéder aux pages en client/serveur



11. Remettre la configuration d'origine.

On redonne les droits de /reserve à www-data

```
# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
User www-data
Group www-data
```



NB: A chaque modification j'ai redémarré le serveur apache2 avec la commande **sudo systemctl restart apache2**