Aim: Hacking a password.

Theory:

Booting process:

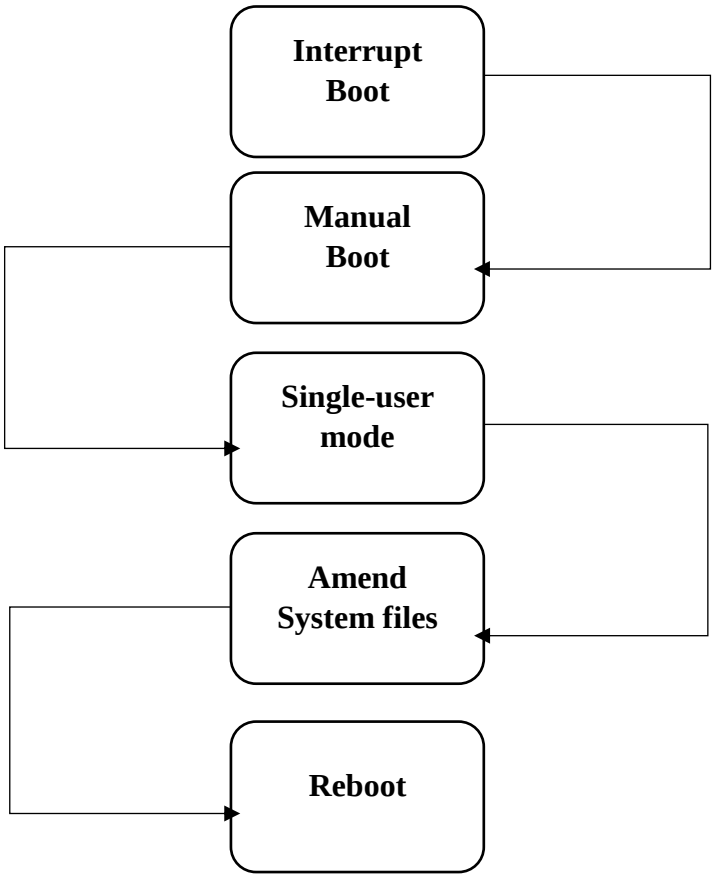
- On power up, BIOS, which stands for Basic Input/Output System, is initialised. It performs some system integrity checks. It searches, loads, and executes the MBR boot loader.
- MBR stands for Master Boot Record. It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda. It contains information about GRUB, it executes the GRUB boot loader.
- GRUB stands for Grand Unified Bootloader. If you have multiple kernel images installed on your system, you can choose which one to be executed.
  GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file. GRUB just loads and executes Kernel and the mentioned kernel images.
- Linux systems can boot in either automatic mode or manual mode. In automatic mode, the system performs the complete boot procedure on its own, without any external assistance.
  In manual mode, the system follows the automatic procedure up to a point but then turns control over to an operator before initialisation scripts have been run. At this point, the computer is in '**single-user-mode**'. Most system processes are not running and other users cannot log in.

Single User mode:

- Single User mode is a mode in which a multiuser operating system boots into a single superuser. It is mainly used for maintenance of multi-user environments.
- A usual installation of Ubuntu does not ask for authentication to enter into single user mode. This enables a user with no privileges to enter this mode. This is the vulnerability we look to exploit to achieve the project's aim.
  Once in this mode, we are in the kernel space implying we have rights to read, write and execute any system files.

Proposed System:

Here we access the single-user mode which lets us amend the passwd file such that we can make changes that are favourable to achieve the objective.



Implementation:

Password file:

- One of the system files that we are interested in is the password file found in the /etc folder, which is named as passwd.
  /etc/passwd file stores essential information, which is required during login, that is user account information. It is a text file which contains a list of system's accounts, giving each user's information.
  This is illustrated below:

  user_name:x:int_uid:int_gid:userid_info:home_directory/user_name:/bin/bash

  Each of these fields separated by colons is listed below in sequence starting from the far left:

  - Username: It is used when the user logs in.
  - Password: An x character indicates that encrypted password is stored in /etc/shadow file.
  - User ID (UID)
  - Group ID (GID)
  - User ID Info: It allows you to add extra information about the users such as the user's full name, phone number, etc.
  - Home Directory
  - Command/shell
- The information about the users in the file, except for the password, aren't encrypted since other system processes need access to this information as and when needed.

Shadow file:

- The etc/shadow file is a system file in which user passwords are stored in encrypted form. In earlier versions of the distro, the encrypted passwords were stored in etc/passwd. To provide another layer of protection the password file contains a reference to this shadow file instead.
- The original password is encrypted by using a randomly generated value or encryption key between 1 and 4096 and a one-way hashing function to arrive at the encoded password that is actually stored.
  The shadow file's hash follows the following syntax:
  Id:salt:encryption

  Each of these fields separated by colons is listed below in sequence starting from the far left:

  ID: Here id denotes the types of hashing algorithms used to encrypt the file. Some of them which are used are DES, MD5, Blowfish, SHA512, etc.

Salt: A salt is a random data that is used as an additional input to a one-way function that hashes data.
Encryption: This refers to the password required by the user to login which is stored in the encrypted format.

- The key is stored with the encoded password. When someone enters a password, their password is then rehashed with the salt value and then compared with the encoded password value. If they match the user is given access to the system.
- The shadow file is accessible to authenticate a login only if the reference 'x' for it is present in the password file. If the reference isn't present, the contents stored in the shadow file cannot be accessed by the system during a user login.

The following steps are to be performed:

1. On powering up the system, on the GRUB loader screen move to the Ubuntu partition of interest and press 'e' to access the booting commands.
2. Navigate down to the text 'linux/boot/vmlinuz…….'
   Change the argument 'ro' to 'rw'.
   - This changes the default read only mode to a read and write mode.

   Remove the text 'quiet splash $vt_handoff'.
   Add the text 'init=/bin/bash'.
   - This argument will initialize the bash shell in the single user mode instead of splashing the booting graphics on screen which it does whilst an automatic boot.
3. Press Ctrl + X to boot.
   - This will initialize the bash shell in the single user mode.
4. From here we can either change a particular user's/root's password using the command: passwd user_name

   OR

   Navigate to the /etc folder and open the passwd file using a text editor. This can be done in the following way: vipw -p
   - vipw is a command to enable safe editing of the etc/passwd and etc/shadow files. It will set the appropriate locks to prevent file corruption.

   In the passwd file navigate down to the user whose password you wish to bypass. Remove the reference (x) to the shadow file in the password field.
5. Reboot the system with the command reboot -f
   - -f does not invoke shutdown and instead performs the actual action that you would expect.
6. The user login page will now not require a password to log in.

Conclusion:

- Realising the vulnerabilities in most of the Ubuntu systems that we use.
- Bypassing the user password is possible because of having access to the single user mode which lets us make changes in the etc/passwd file without any authentication, which otherwise would have been a read only file to all users except sudoers.
- These vulnerabilities can be fixed with some simple measures, but still exist out of the box and need to be addressed by the user.

Future Scope:

- Since the vulnerabilities have been identified, it becomes obvious to find a fix for them.
- Not having authentication to access the single user mode although being a facility to gain access the system when passwords have been forgotten is a vulnerability that can be exploited.
- The future scope would be to still avail this facility to the user while patching this vulnerability.

References:

[1] manpages.ubuntu.com
[2] www.//ifconfig.dk/ubuntu/
[3] www.//en.m.wikipedia.org/wiki/Salt_(cryptography)
[4] www.//security.stackexchange.com
[5] www.youtube.com/user/cgermany77