Week 3 - Supplementary Materials

Markdown | PDF | MS Word DOCX | Libre ODT | HTML

Permissions Calculator

chmod is the command used on Unix-like / GNU Linux systems to modify filesystem permissions. These permissions are stored in octal, in three columns from 0-7 for "Owner", "Group", and "Public". 000 equals no permission, 777 equals full permission for everyone to read, write, and execute the file or directory. The combinations in between can be calculated via addition or the calculator linked below.

- CHMOD Calculator

Filesystem permissions, root user, and sudo

- Use `ls -l` to see detailed permissions and ownership of files and directories. See more options for `ls` with `ls --help`
- The `chmod` command changes permissions:
    - `chmod 400 file.txt` gives read permission to the owner of `file.txt`
    - `chmod 644 file.txt` gives read+write permission (6) to the owner of `file.txt` and gives read permission (4) to the group and public.
    - Use the `-R` option to apply permissions recursively to a directory (i.e. to the directory and all of its contents): `chmod 777 -R ~/Desktop` gives read+write+execute permissions to everyone for `/home/kali/Desktop` and its contents.
    - See more options for `chmod` with `chmod --help`
    - Remember, write permission is permission to **delete** a file or directory as well.
- Each file or directory has an "owner" and a "group". Usually, a user also has a group with the same name (e.g., `kali:kali`). If we had groups for our class on a filesystem, each student might be a member of `students` and your instructors might be in a `teachers` group.
- Run the command `groups` to list all groups associated with your user. To see groups for other users, type the username afterword such as `groups root`
- The `chown` command changes ownership by users and groups:
    - `chown kali file.txt` gives ownership of `file.txt` to user `kali`
    - `chown kali:root file.txt` gives ownership of `file.txt` to user `kali` and assigns the group `root` to the file.
    - Use the `-R` option to apply ownership recursively to a directory (i.e. to the directory and all of its contents): `chown -R kali:kali ~/Desktop` gives ownership of the `/home/kali/Desktop` directory and its contents to user `kali` and assigns the group `kali` to the directory and its contents.
    - See more options for `chown` with `chown --help`
- Hidden files in the filesystem start with the `.` character such as `.hidden-file` To view all files including hidden files, use `ls -la`
- Activating the administrative superuser `root` is dangerous, but it may be necessary for certain tasks (and **definitely** something an attacker would like to do). To activate the `root` user, type `sudo passwd -l root`

- Then, you can set a password for the administrative superuser `root` user using `sudo passwd root`.
- These commands assume you have `sudo` permissions, which is the default in Kali Linux for your user `kali`. `sudo` allows you to temporarily act as the administrative superuser `root` and times out (depending on system settings). This protection means that you're only **escalating privileges** for a short time for a specific task or tasks, limiting the potential time-window for an attack against your system that does serious damage. It's also protection by the operating system **from you** - if you were instead running with `root` privileges all the time, it's more likely your (inevitable) mistakes would destroy the system.
- If you do want to switch into the `root` account, you can use `su`. This switch user command can also be used to switch into other user accounts such as `su sean` or `su scott`. To leave the session for that user, type `exit` which will bring you back to being the default user `kali`.

Cracking Passwords with Dictionary Attacks

These are commands we used in the livestream session for Week 3 using the "John the Ripper" program.

User account information is stored in two special text files:

- `/etc/passwd`
- `/etc/shadow`

To view these files try:

- `cat passwd` or `cat passwd | less`
- `cat shadow` or `cat shadow | less`

The `passwd` file contains information and settings for each user account. That includes many "users" that only exist to run system services or "daemons" e.g. printers, boot and init processes, audio services, and so on. This segmentation of the permissions for these services as a "pseudo-user" or system account named specifically for its purpose makes administering the system more comprehensible while also providing some basic security protections. For our purposes, the `kali` and `root` user are all that matter in this `passwd` file.

Likewise, the `shadow` file stores user settings **and hashes of passwords** for those users. Again, we only care about the `kali` and `root` entries in this file.

John the Ripper or the command `john` combines brute force attacks against a password hash with password dictionaries. It is a time versus memory tradeoff because many of these dictionaries or "wordlists" can get large. However, brute forcing can be near-impossible in many circumstances unless a password is extremely weak and/or short.

- To combine the `/etc/passwd` and `/etc/shadow` files for `john` to process, run:

```
sudo unshadow /etc/passwd /etc/shadow > unshadow.txt
```

This will create an `unshadow.txt` file in the folder you are currently in.

- Now, we try to crack the password(s) in this file using one of the wordlists in `/usr/share/wordlists` called `john.lst`

```
sudo john --format=crypt --wordlist=/usr/share/wordlists/john.lst
unshadow.txt
```

THAT'S IT! After letting `john` run, you should see the cracked passwords for these users.

If you set your password for the `kali` or `root` user to something weak and/or short (such as "kali" or "password") you will see the cracking takes approx. two minutes or less.

NOTE: If you are running a system that does not have the `john.lst` wordlist, please download it here](https://raw.githubusercontent.com/lawfareblog/hacking-cybersecurity/main/week03/wordlists/john.lst), for example by using Firefox inside of Kali Linux or using wget:

```
wget https://raw.githubusercontent.com/lawfareblog/hacking-cybersecurity/main/week03/wordlists/john.lst
```

Then change the path for the `--wordlist` option for the `john` command. If `john.lst` is in the current directory, for example:

```
sudo john --format=crypt --wordlist=john.lst unshadow.txt
```

...or use the `mv` command to move the file to `/usr/share/wordlists/`:

```
sudo mv john.lst /usr/share/wordlists/
```

...and then run:

```
sudo john --format=crypt --wordlist=/usr/share/wordlists/john.lst unshadow.txt
```

Further Reading

Common passwords and wordlists

Beware, some of the entries in these lists are English swear words NSFW

- https://hashtoolkit.com/common-passwords/
- https://en.wikipedia.org/wiki/Wikipedia:10,000_most_common_passwords
- https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-10000.txt
- https://www.geeksforgeeks.org/how-to-extract-rockyou-txt-gz-file-in-kali-linux/
- https://github.com/ohmybahgosh/RockYou2021.txt

Password Tips

Here are some password strength, storage, and retention strategies.

- **Passphrases:** Can be a favorite song lyric, movie quote, or joke. Add numbers and special characters to increase the "keyspace": `Cecil+Harambe4ever`
- **Strength:** General guidelines | Diceware method
- EFF diceware wordlists
- **KeePass** (Windows/Linux) or **KeePassX** (Mac): Organize your passwords in an encrypted database.
- **Browser Password Manager:** In Firefox | In Chromium
- **(2FA/MFA):** One-time Pass (OTP) authenticator apps for Android or iOS | Physical key tokens Nitrokey or YubiKey

NOTE: Using a cloud-based password manager puts trust in the organization or company that runs that cloud. As with all of these recommendations, YMMV. Our word is not gospel - ultimately, you need to decide who to trust.

Hashes & Message Digests

- https://www.youtube.com/watch?v=b4b8ktEV4Bg
- https://en.wikipedia.org/wiki/Cryptographic_hash_function
- https://en.wikipedia.org/wiki/Cryptographic_hash_function#/media/File:Cryptographic_Hash_Function.svg