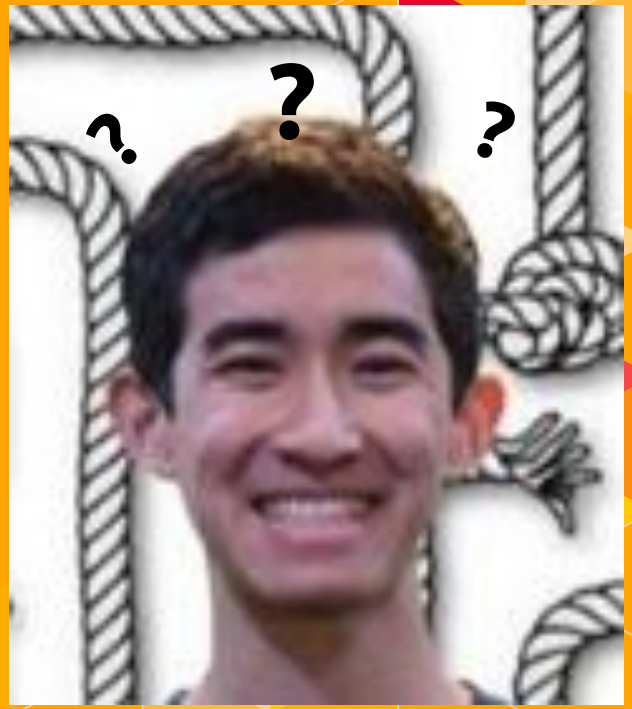
Password Auditing

A bit of review

- /etc/passwd
  - UIDs
  - usernames
  - home directory
  - shell
- /etc/shadow
  - encryption type
  - hashed password
  - password last changed
  - number of days before password can be changed
  - number of days required until password change
  - password change warning threshold

So, remind me

Why are passwords importante?



Well… they are the keys

- Typically used to “lock up” your user account and other privileges/data from intruders
- Only YOU should know it
  - It is unique for you

How to make a password *effectively*

- I can use anything as a password right?
  - Think again
- Don’t make them obvious
  - There are lists, lists, and lists of common passwords out there
  - *cough* *cough* rockyou *cough* *cough*
  - Attackers will bruteforce

PASSWORD LIST USED (MODIFIED "ADOBE TOP 100")

123456	123456789	password	forbes123	12345678	qwerty
1234567	111111	news	123123	1234567890	000000
abc123	1234	forbes1	f0rb3s	azerty	iloveyou
aaaaaa	654321	12345	666666	sunshine	123321
letmein	monkey	asdfgh	password1	shadow	princess
dragon	forbesforbes	daniel	computer	michael	121212
charlie	master	superman	qwertyuiop	112233	asdfasdf
jessica	1q2w3e4r	welcome	1qaz2wsx	987654321	fdsa
753951	chocolate		soccer	tigger	asdasd
thomas	asdfghjkl	internet	michelle	football	123qwe
zxcvbnm	forbes2	7777777	maggie	qazwsx	baseball
jennifer	jordan	abcd1234	trustno1	buster	555555
liverpool	abc	whatever	11111111	102030	123123123
andrea	pepper	nicole	killer	abcdef	hannah
test	alexander	andrew	222222	joshua	freedom
samsung	asdfghj	purple	ginger	123654	matrix
secret	summer	1q2w3e	snoopy1		

Mix it up!

Character length

- At least 10 is ideal

Vary your characters

- Numb3r5
- Upper/Lower case letters
- Speci@! ch@r@cters

Pro tip: make your passwords memorizable

- Make several passwords from a pattern

Pro tip #2: Change your passwords occasionally (30-90 days)

- Hackers will have to redo their bruteforce efforts mwahaha

Changing user passwords

- passwd [user]: begins interactive prompt to change user password

```
user@ubuntu: ~
user@ubuntu:~$ sudo passwd jay
Enter new UNIX password:
```

- passwd -l [user]: locks user account
  - **passwd -l root**: locks root account, we don't want people to login as root

```
user@ubuntu: ~
user@ubuntu:~$ sudo passwd -l root
passwd: password expiry information changed.
```

*Note: run all of the commands mentioned (including those on the previous slide) using sudo because you will probably need privileges

Batch password changes

- chpasswd [user]:[password]: can be used to change many passwords quickly
  - will provide passwords as plaintext
  - e.g. chpasswd bob:s3cUr3p4ssw0rd
  - optional to know: more for use in scripts
    - Makes life easier :)

Now, do you think everyone Will actually make good passwords?

/etc/sudoers

The sudoers file controls how sudo and elevated privileges function on the system.

- check for NOPASSWD and !authenticate
  - remove if it exists
  - you can refer to a default sudoers file to see if anything is out of place

/etc/sudoers

Here is the sudoers syntax:

username hostlist = (userlist) commandlist

- e.g. root ALL=(ALL) ALL
  - root user on <u>all</u> hosts on <u>all</u> users can execute <u>all</u> commands

/etc/sudoers

```
user@ubuntu: ~
  GNU nano 2.5.3                    File: /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

/etc/login.defs

Configuration file for system login restrictions.

- PASS_MIN_DAYS 30
  - user cannot change password for 30 days after changing password
- PASS_MAX_DAYS 90
  - user must change password after 90 days
- PASS_WARN_DAYS 7
  - warns user to change threshold 7 days before

chage

Changes user password expiry settings.

- chage -M [days] [user]: sets maximum password age for user
- chage -m [days] [user]: sets minimum password age for user
- chage -W [days] [user]: sets warning threshold for user

PAM

PAM (Pluggable Authentication Module) controls password authentication for many applications.

- all configurations are in /etc/pam.d/
- **sudo apt install libpam-cracklib**: installs a PAM module to make sure our passwords are secured

common-auth

The common-auth module controls how authentication is handled.

Located at /etc/pam.d/common-auth

common-password

The common-password module controls how passwords are set, managed, and restricted.

Located at /etc/pam.d/common-password

common-password

password requisite pam_cracklib.so retry=3 minlen=8 difok=3 reject_username minclass=3 maxrepeat=2 dcredit=-1 ucredit=-1 lcredit=-1 ocredit=-1 gecoscheck enforce_for_root

- retry=3: allow for three password attempts
- minlen=8: minimum password length of 8
- difok=3: at least 3 characters must be different from previous password
- reject_username: prevent password from being same as username
- minclass=3: at least 3 types of characters needed
- maxrepeat=2: at most 2 repeated characters
- dcredit=-1: 1 digit required
- ucredit=-1: 1 uppercase required
- lcredit=-1: 1 lowercase required
- ocredit=-1: 1 other character required (like a symbol)
  *negative=required, positive=recommended for the different credits
- gecoscheck: prevent extra fields (e.g. full name, address) from being used as password
- enforce_for_root: apply same restrictions to root password

common-password

```
user@ubuntu: ~
  GNU nano 2.5.3                File: /etc/pam.d/common-password

#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords.  The default is pam_unix.

# Explanation of pam_unix options:
#
# The "sha512" option enables salted SHA512 passwords.  Without this option,
# the default is Unix crypt.  Prior releases used the option "md5".
#
# The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
# login.defs.
#
# See the pam_unix manpage for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password        [success=1 default=ignore]      pam_unix.so obscure sha512
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so
# end of pam-auth-update config
```

common-auth

auth optional pam_tally.so deny=5 unlock_time=900 onerr=fail audit even_deny_root_account silent

- deny=5: deny user after 5 login attempts
- unlock_time=900: locks user out for 900 seconds if all login attempts are used up
- onerr=fail: return a fail code if an error happens
- audit: logs user if login attempts exceeded
- even_deny_root_account: rules apply to root, locks root out if attempts exceeded
- silent: don’t print extra information

common-auth

```
user@ubuntu: ~
  GNU nano 2.5.3                    File: /etc/pam.d/common-auth

#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
auth    [success=1 default=ignore]      pam_unix.so nullok_secure
# here's the fallback if no module succeeds
auth    requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

/etc/sudoers

- DO NOT EDIT THIS FILE DIRECTLY
- sudo visudo: allows you to edit this file indirectly, checks for syntax errors before saving changes to actual file
  - prevents you from messing up and destroying sudo privileges