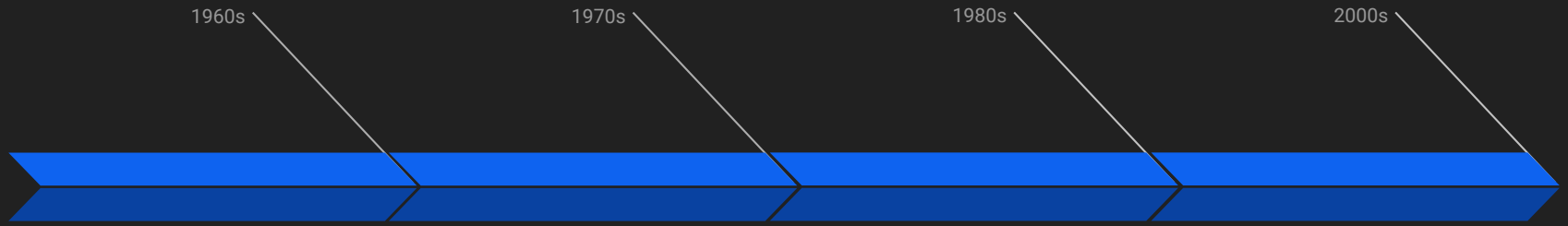
Linux Authentication and PAM

Henry Roeth

What Is Authentication?

- the process of verifying an identity through something someone knows, something they have, or something they are
- goal is to prevent unauthorized access and protect against security threats, such as identity theft, data breaches, and system intrusions

History of Authentication



Password-Based Authentication

users log in with a secret password they create

Unix Crypt (hashing)

Unix operating system introduced a password hashing algorithm known as Unix Crypt

Biometric Authentication

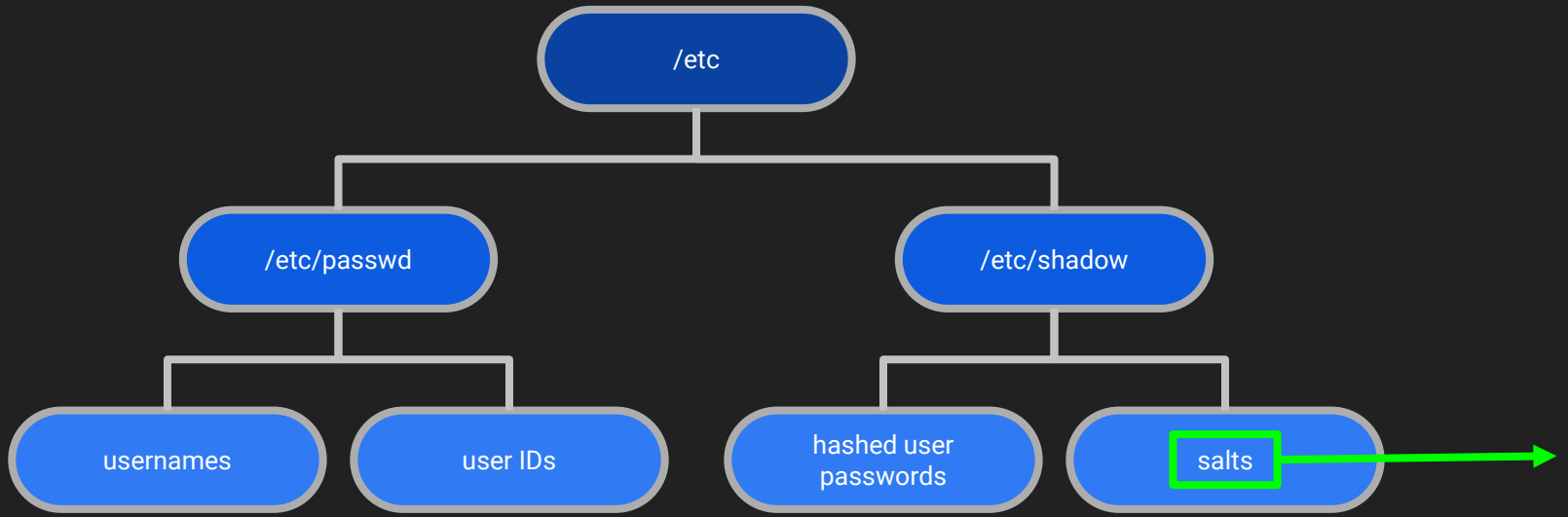
authentication using unique physical traits, like fingerprints or face recognition

Multi-Factor Authentication

users need two or more ways to prove who they are (e.g., password paired with a security code sent to email)

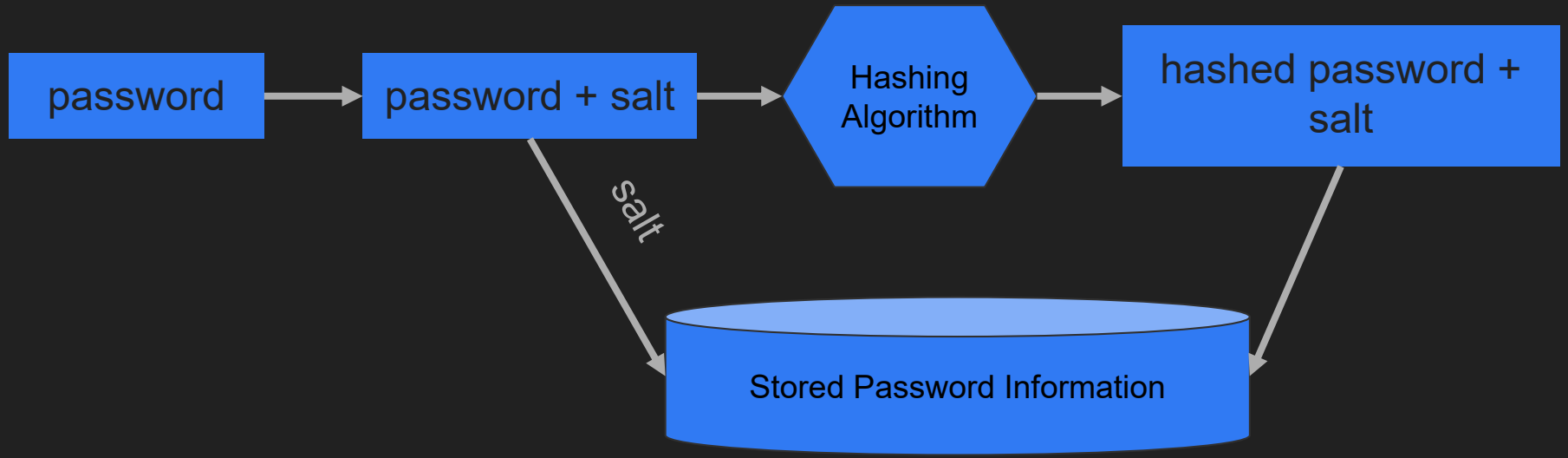
Authentication in Linux

- security information stored in the /etc/shadow and /etc/passwd files
- /etc/shadow file added for extra security as only the root user has inherent access



Salts

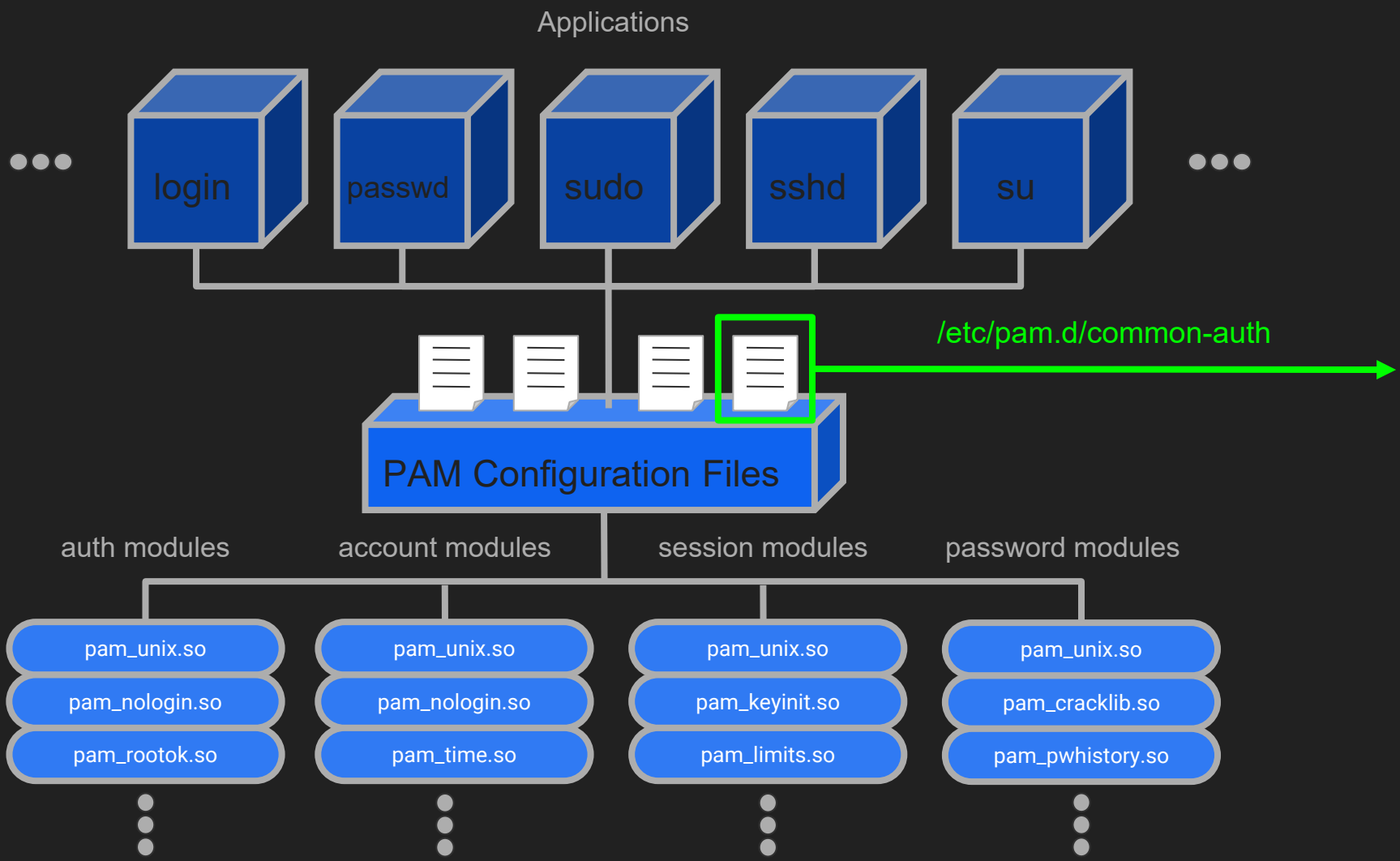
- random data that is added to a password before it is hashed
- makes the hashed output unique even if the input is the same as another
- prevents attackers from using precomputed hash tables (rainbow tables)



What is PAM?

- PAM (Pluggable Authentication Module) mechanism integrates various low-level authentication schemes into a high-level API, allowing programs that require some form of authentication to be developed independently from the desired authentication scheme

like a multitool


Applications
login
passwd
sudo
sshd
su
/etc/pam.d/common-auth
PAM Configuration Files
auth modules
account modules
session modules
password modules
pam_unix.so
pam_nologin.so
pam_rootok.so
pam_unix.so
pam_nologin.so
pam_time.so
pam_unix.so
pam_keyinit.so
pam_limits.so
pam_unix.so
pam_cracklib.so
pam_pwhistory.so

```
#
# /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=2 default=ignore]  pam_unix.so nullok
auth    [success=1 default=ignore]  pam_sss.so use_first_pass
# 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)
auth    optional            pam_cap.so
# end of pam-auth-update config
```

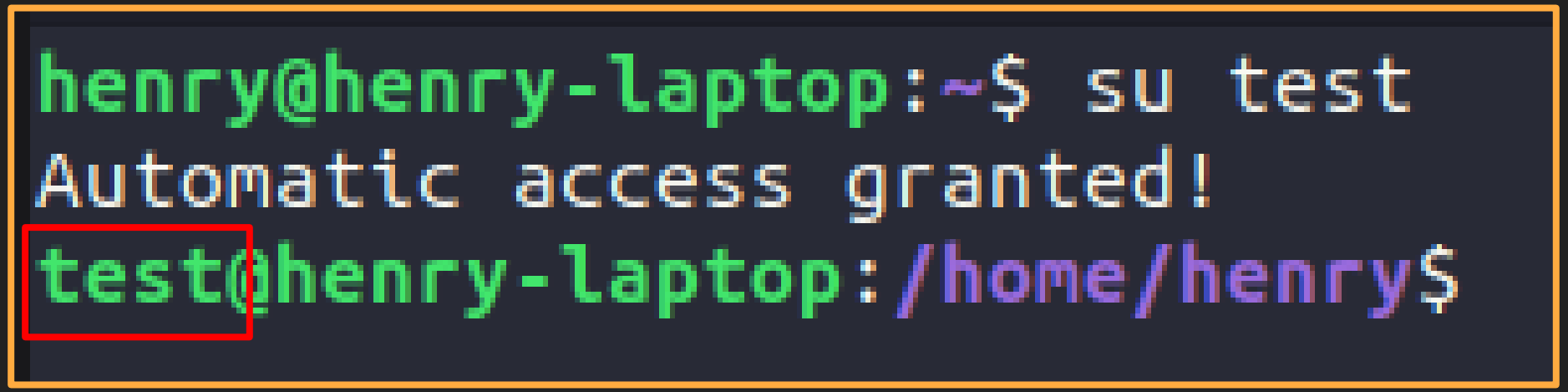
Custom Bypass Module

```
# /etc/pam.d/su

# Allow root to su without passwords
auth    sufficient pam_rootok.so

# Includes a custom authentication module granting automatic access
auth    required   pam_always_allow.so
```

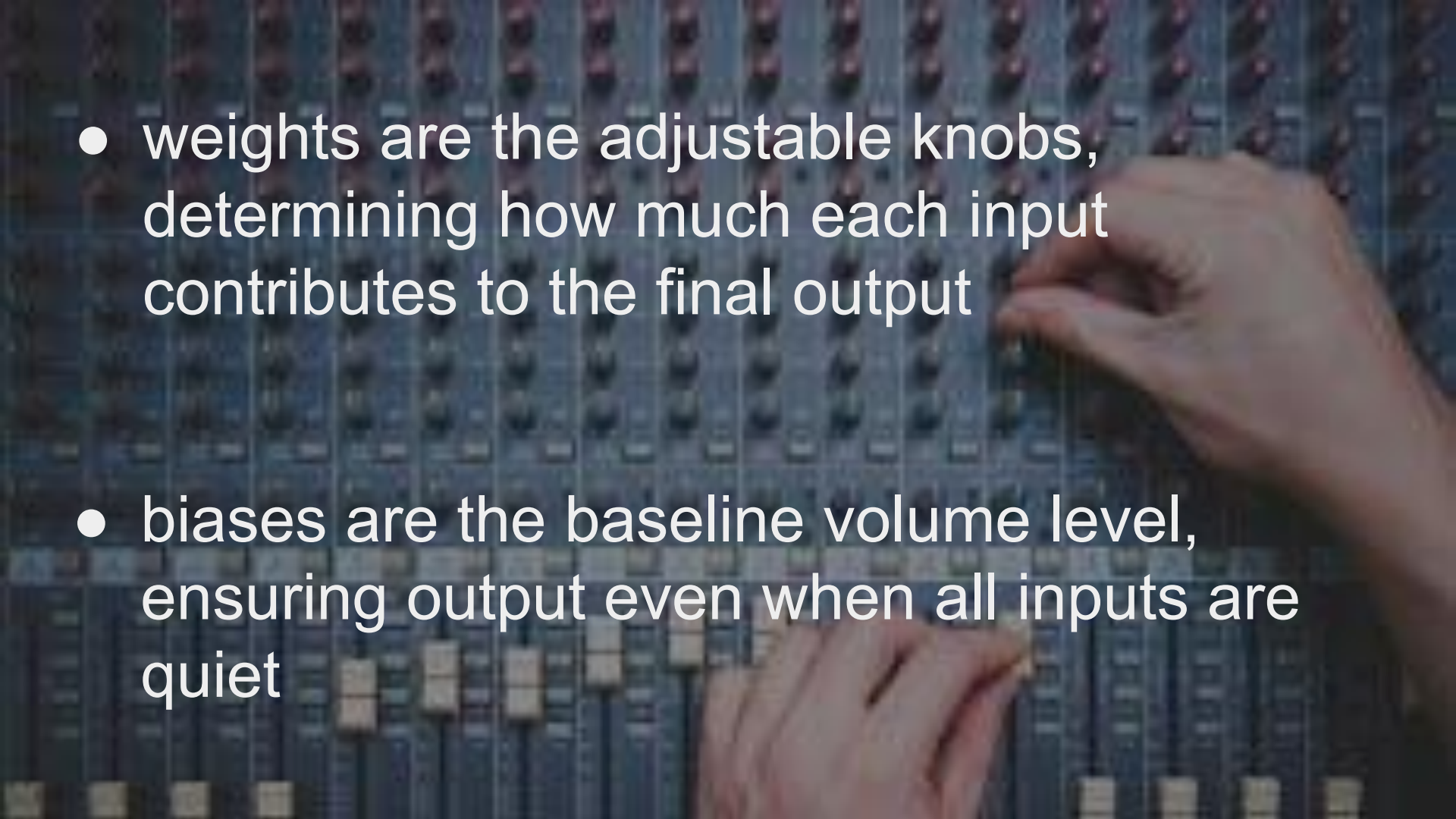
implementation


henry@henry-laptop:~$ su test
Automatic access granted!
test@henry-laptop:/home/henry$

What Is Machine Learning?

- a subset of artificial intelligence that enables systems to learn from data and make predictions or decisions without being explicitly programmed
- models are trained with “weights” and “biases”
- weights are like parameters that adjust the contribution of different inputs
- biases act as additional values that allow the model to account for situations where all inputs are zero or have no effect

think of a sound mixer…

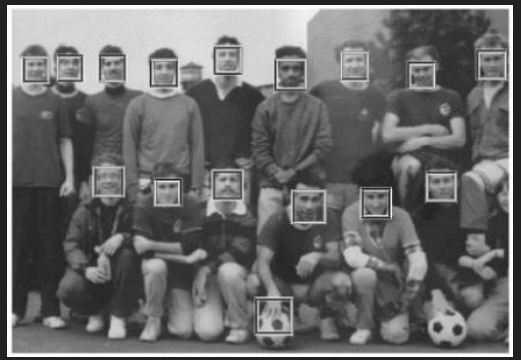


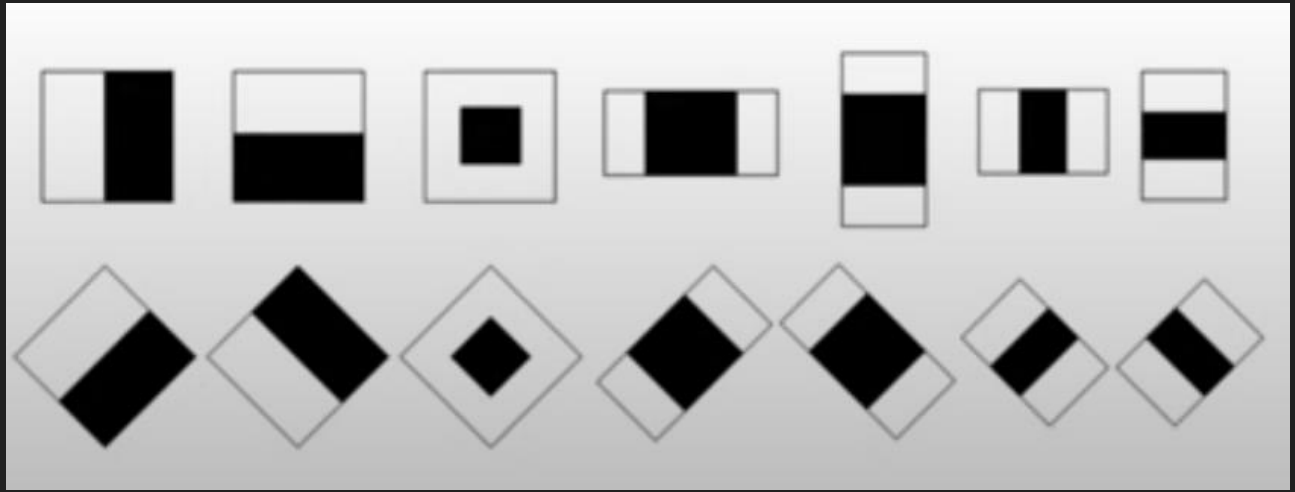
- weights are the adjustable knobs, determining how much each input contributes to the final output

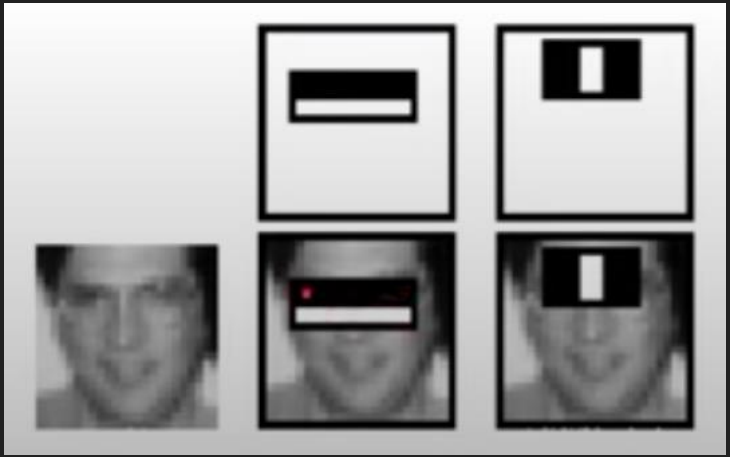
- biases are the baseline volume level, ensuring output even when all inputs are quiet

Haar Cascade Classifier

- named after Alfred Haar, created by Viola and Jones in 2001
- detects simple rectangular patterns that capture contrast differences in the image, such as edges or texture variations
- trained with a large dataset of positive (containing object of interest) and negative (not containing object of interest) images
- algorithm learns to distinguish between positives and negatives

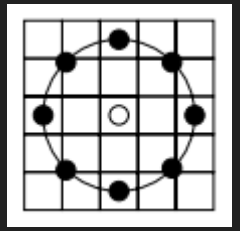


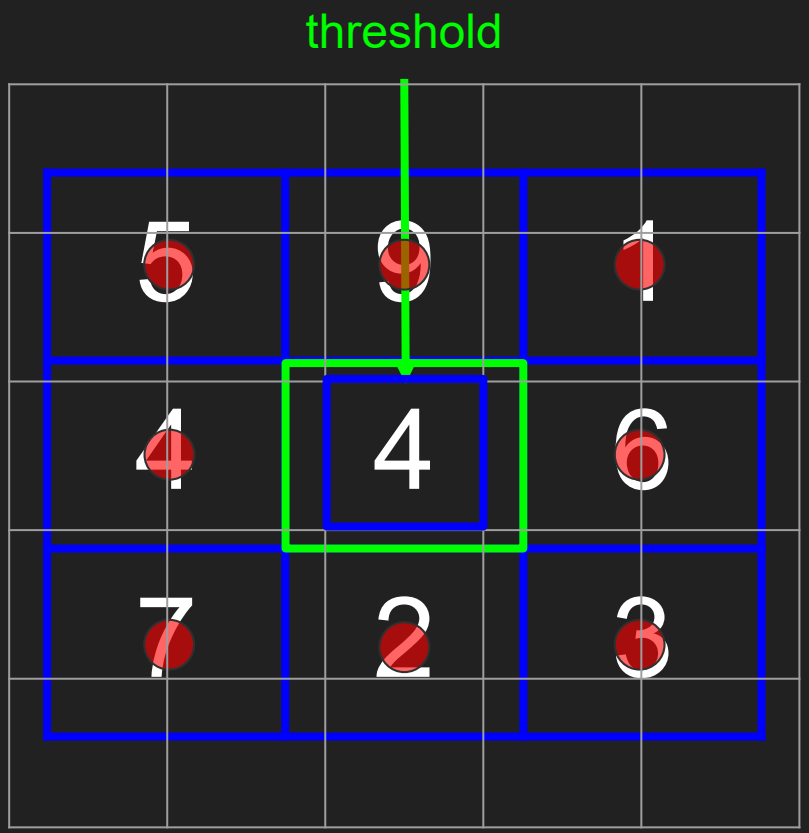


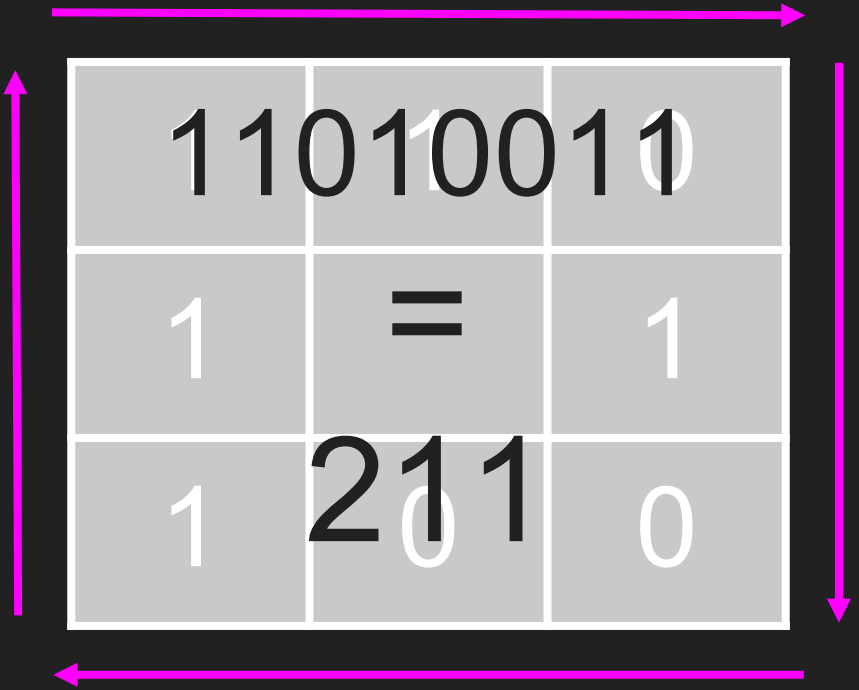


Local Binary Patterns Histograms (LBPH)

- feature-based face recognition method using texture classification
- LBPs encode local texture patterns by comparing each pixel with its neighbors
- binary values are calculated about a center pixel (threshold)
  - values >= threshold are assigned a binary value of 1
  - values < threshold are assigned a binary value of 0
- these “blocks” (9x9 pixel area) yield a decimal value using the binary values
- edges and features can be extracted using these values and analyzed in histograms showing the frequency of specific block values (0-255)

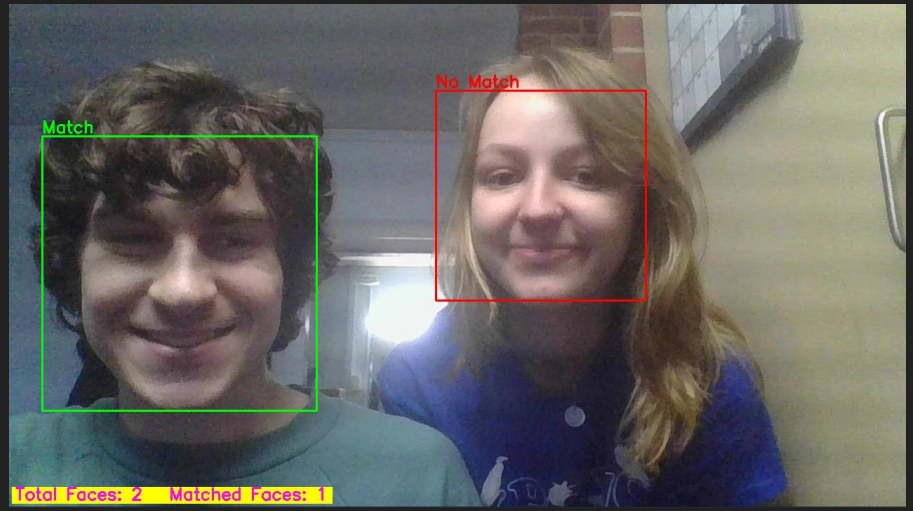



threshold
5
9
1
4
4
6
7
2
3


11010011
1
=
1
1
211
0

Results

- face recognition performed pretty well with minimal training data
- able to differentiate matches from non-matches
- working face recognition logic, but runtime security error causing PAM to default to a failure
- logic performed with a pre-captured image yielded a success
- camera usage causing failure




```
henry@henry-laptop:~$ su test
Please hold still. Recording will begin in 1 seconds.
Recording started. Please hold still.
Average confidence: 16.7201
SUCCESS!
Password:
su: Authentication failure
```

```
henry@henry-laptop:~$ su test
SUCCESS!
test@henry-laptop:/home/henry$
```

LIVE DEMO!

Improvements

- train LBPH model with more gallery images of varying distances from camera, shadows, head positions, etc.
- train with data on multiple individuals to identify more than one person
- ensure there are no runtime errors arising when using the camera

References

Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns.

Geisshirt, K. (2007). Pluggable Authentication Modules: The Definitive Guide to PAM for Linux Sysadmins and C Developers. Packt Publishing, Birmingham, UK.

Viola, P. and Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 1:I–511.

Questions?