Unit #3

Authentication

- **Non-human identities:** workloads, services, machines
    - These can be considered the majority of users in organizations.
    - They have more privileged accounts than humans.
    - Certificates or keys are used for these identities rather than traditional passwords.

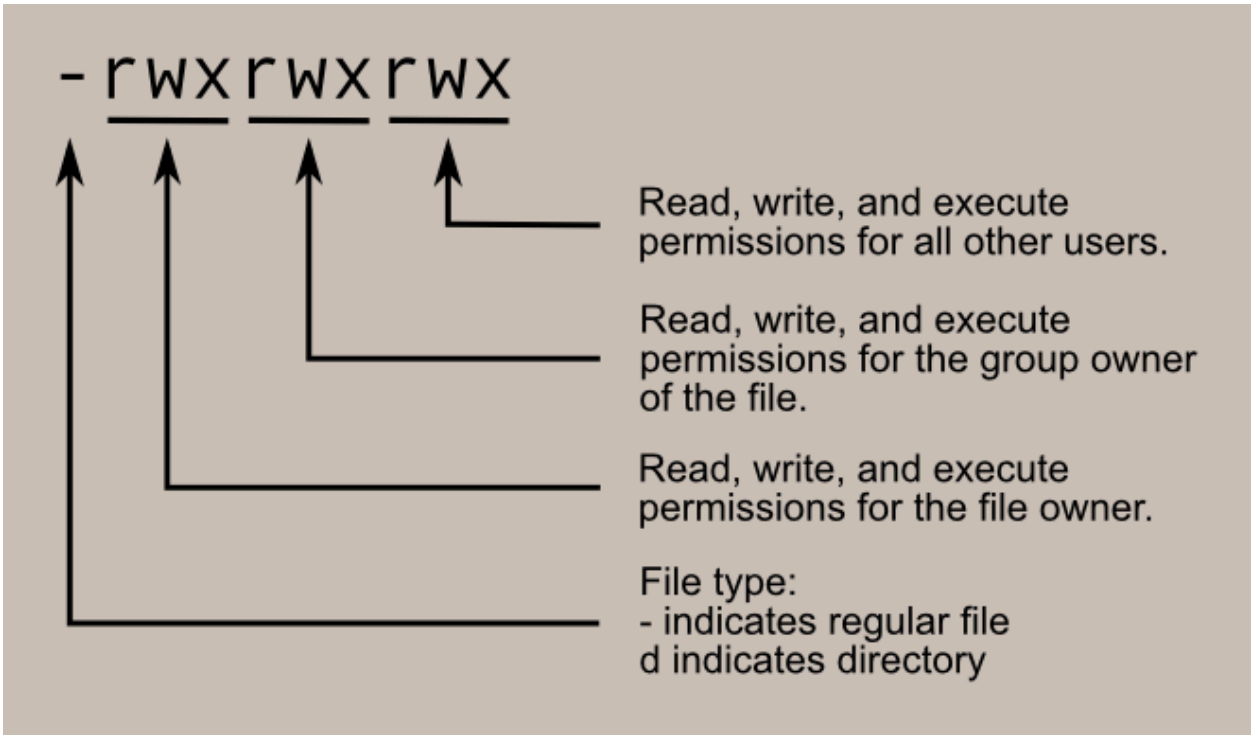
Securing device identities	
Public Key Infrastructure	● Issuance of digital certificates to provide unique digital identities for users.
On-device code generation	● Code generation apps ensure that only authorized users can access resources.
Mutual authentication	● Two sides of a communication channel verify each other's identity.
Zero trust	● Don't grant access to resources until the device verifies its identity

Three ways to authenticate	
Knows	● A password, pin, answer to a secret question, etc. ● **Vulnerabilities:** forgetting passwords, weak passwords, reusing passwords, discoverable.
Has	● ID, magnetic card, token, etc. ● **Vulnerabilities:** loss, duplication.
Is	● Biometrics (fingerprints, facial recognition, etc.) ● **Vulnerabilities:** error rates can be high, high costs, privacy concerns, and some people can have similar features.

- Every file and directory has an **owner** and **group**.
- There are **sets of permissions** for the owner, group, and world (all users that can log into the system).

Types of permissions	
Read	● A user can see the contents.

Write	• A user can modify the content.
Execute	• A user can run a file.



How passwords work	
Signup	1. Encrypt the user's password. 2. Stores encrypted string (hash) with the user's record in the database.
Login	1. The user submits a username and password on a login page. 2. The attempted password is encrypted. 3. The new hash is verified against the stored hash.

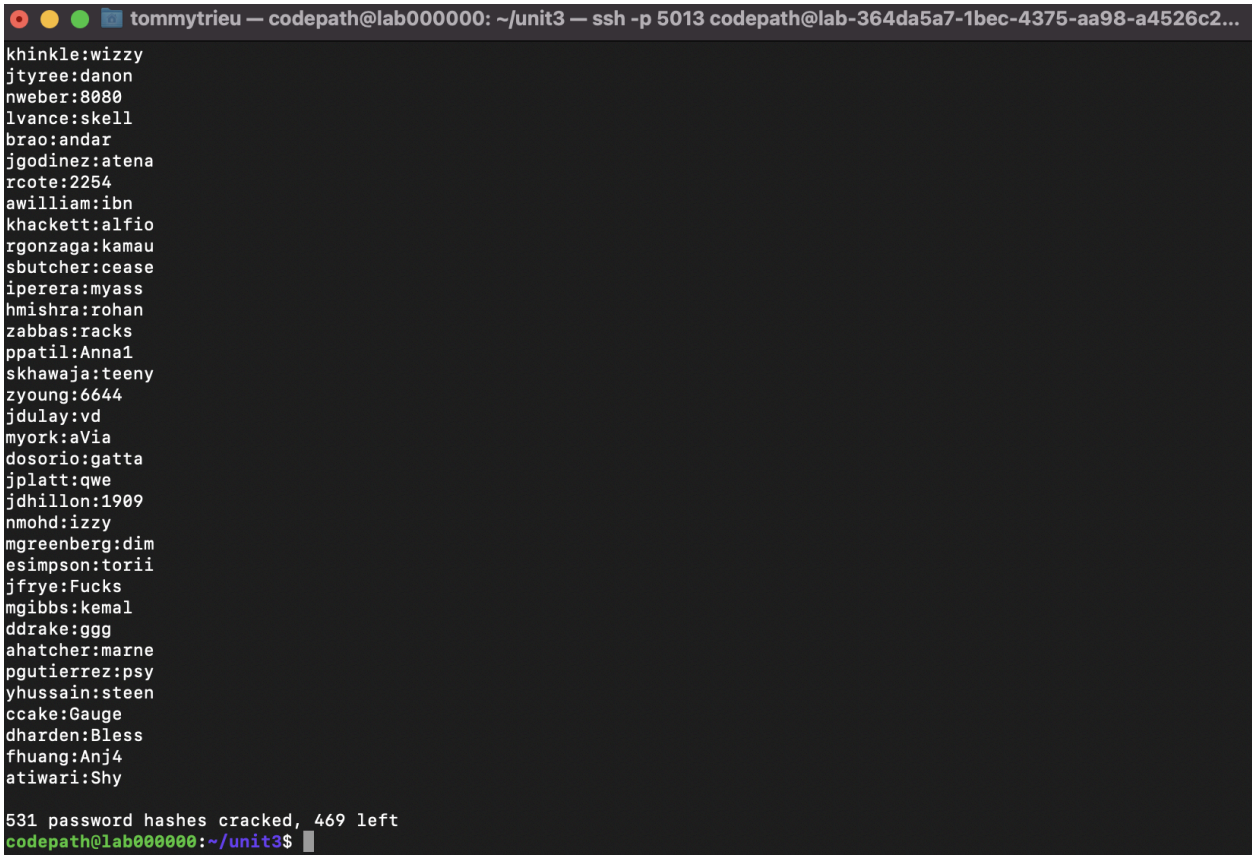
Lab

John the Ripper cracking methods	
Single crack mode	• It uses the user's information such as login names, full name fields, directory name fields, etc. stored in the GECOS field to guess user passwords.
Wordlist mode	• It uses a wordlist of passwords and tries every password in it. In this lab, the wordlist `lower.lst` was used. • Mangling rules can be applied to modify the passwords in the

	list.
Incremental mode	• Tries all possible character combinations as passwords, essentially like a brute force. • Educated guesses about the construct of passwords can be used. Using `-mask` can be used to look for a common pattern.

- crackA.txt passwords: (used single crack mode to reveal all passwords in this file) [`john --single crackA.txt`]
    - bulbasaur:kantograss
    - squirtle:waterSquirtle
    - charmander:charizard22
- crackB.txt
    - jim:paper [`john --wordlist=lower.lst crackB.txt`]
    - pam:tEaPoT [`john --wordlist=lower.lst crackB.txt --rules=l33t`]
    - dwight:b33t [`john --wordlist=lower.lst crackB.txt --rules=shifttoggle`]
- crackC.txt
    - pinball:496821 [`john --incremental=digits --min-length=4 --max-length=6 crackC.txt`]
    - pacman:8Bit [`john --mask=?d?u?l?l crackC.txt`]
    - frogger:bugs7! [`john --mask=?l?l?l?l?d! crackC.txt`
- challengeCrack.txt
    - pupper: bacon! [`john --single crackChallenge.txt`]
    - Birb: birdseed [`john --wordlist=lower.lst crackChallenge.txt`]
    - Kitty: pr3d4t0ry [`john --wordlist=lower.lst crackChallenge.txt --rules=l33t`]

Project


tommytrieu — codepath@lab000000: ~/unit3 — ssh -p 5013 codepath@lab-364da5a7-1bec-4375-aa98-a4526c2...
khinkle:wizzy
jtyree:danon
nweber:8080
lvance:skell
brao:andar
jgodinez:atena
rcote:2254
awilliam:ibn
khackett:alfio
rgonzaga:kamau
sbutcher:cease
iperera:myass
hmishra:rohan
zabbas:racks
ppatil:Anna1
skhawaja:teeny
zyoung:6644
jdulay:vd
myork:aVia
dosorio:gatta
jplatt:qwe
jdhillon:1909
nmohd:izzy
mgreenberg:dim
esimpson:torii
jfrye:Fucks
mgibbs:kemal
ddrake:ggg
ahatcher:marne
pgutierrez:psy
yhussain:steen
ccake:Gauge
dharden:Bless
fhuang:Anj4
atiwari:Shy
531 password hashes cracked, 469 left
codepath@lab000000:~/unit3$