Credential Collection and Cracking

AAA

- Authentication
  - Verifying you are who you say you are
- Authorization
  - You have permission to do what you want to do
- Accountability
  - Maintaining an account (log) of what happened

Authentication

- You are who you say you are
- Can be configured to use multiple factors
  - Something you know (password, pin, phrase,etc)
  - Something you have (CAC, YubiKey, badge, etc)
  - Something you are (fingerprint, voice, other biometrics, etc)

Authentication

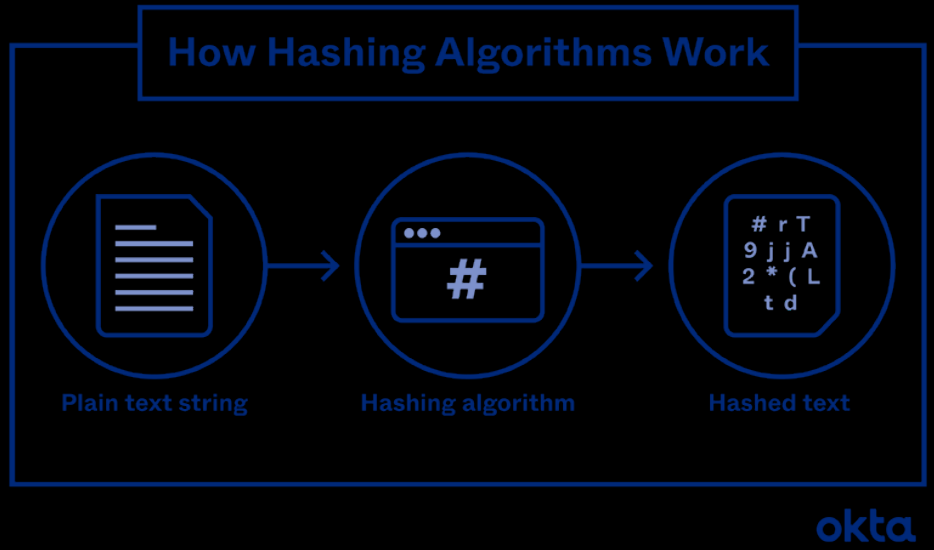
https://www.nist.gov/image/cia-triad

Stored Passwords

- Most operating systems will store hashed versions of local accounts.
- Where does Linux store password hashes?
- Windows Stores password hashes in the SAM database; LSASS is the process that manages password hashes in active memory

Hashing Vs Encrypting

- Encryption can be reversed if you have the proper key.
- Hashing is a one way process **NOT** intended to be reversed to plaintext



https://www.okta.com/identity-101/hashing-algorithms

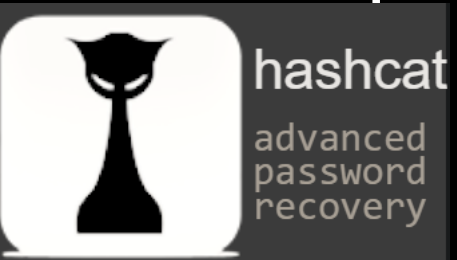
Some Popular Hashing Algorithms

- **MD-5**. This is one of the first algorithms to gain widespread approval. It was designed in 1991, and at the time, it was considered remarkably secure. Since then, hackers have discovered how to decode the algorithm, and they can do so in seconds. Most experts feel it's not safe for widespread use since it is so easy to tear apart.
- **SHA**. Algorithms in the SHA family are considered slightly more secure. The first versions were developed by the United States government, but other programmers have built on the original frameworks and made later variations more stringent and harder to break. In general, the bigger the number after the letters "SHA," the more recent the release and the more complex the program. For example, SHA-3 includes sources of randomness in the code, which makes it much more difficult to crack than those that came before. It became a standard hashing algorithm in 2015 for that reason.

https://www.okta.com/identity-101/hashing-algorithms/

Compromising Linux

- Brute Forcing passwords either generating them character by character or using a dictionary like RockYou
- Cracking Hashes with a tool like John the Ripper, Hashcat, or crackstation.net
- Compromising SSH keys



https://hashcat.net/hashcat/

https://crackstation.net/hashing-security.htm

/etc/passwd and /etc/shadow

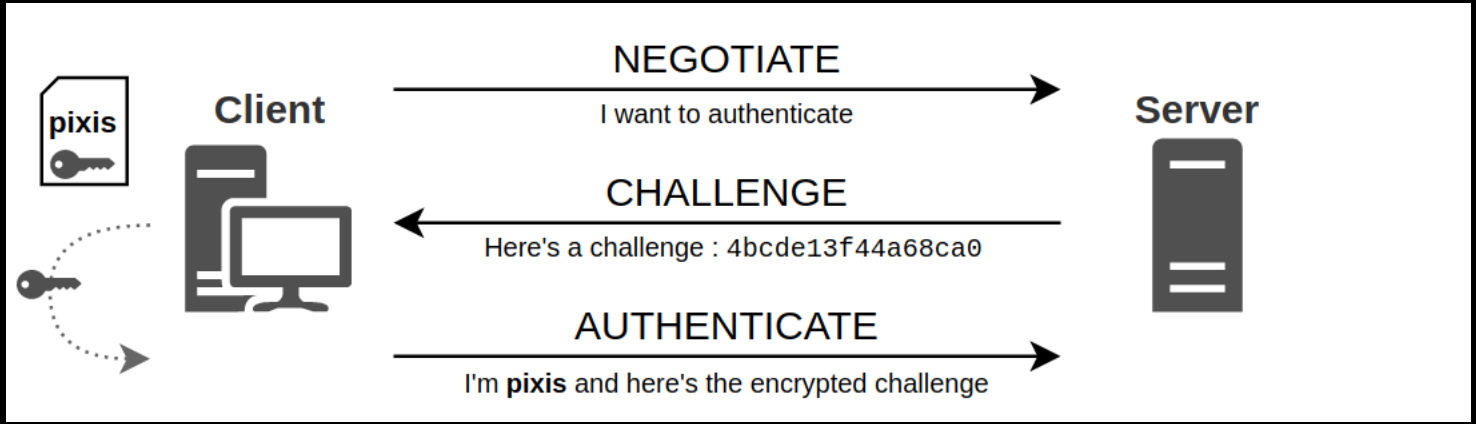
- Back in the 90s /etc/passwd was a globally readable file used to store all user information including username, groups, and importantly password hashes. Many programs needed information about the user running them and would access the /etc/passwd file to get it.
- /etc/shadow was created to split off the password hashes and protect them using file permissions
- Why not just change permissions on the existing /etc/passwd file?

John the Ripper

- John the Ripper is an Open Source password security auditing and password recovery tool available for many operating systems.

https://www.openwall.com/john/

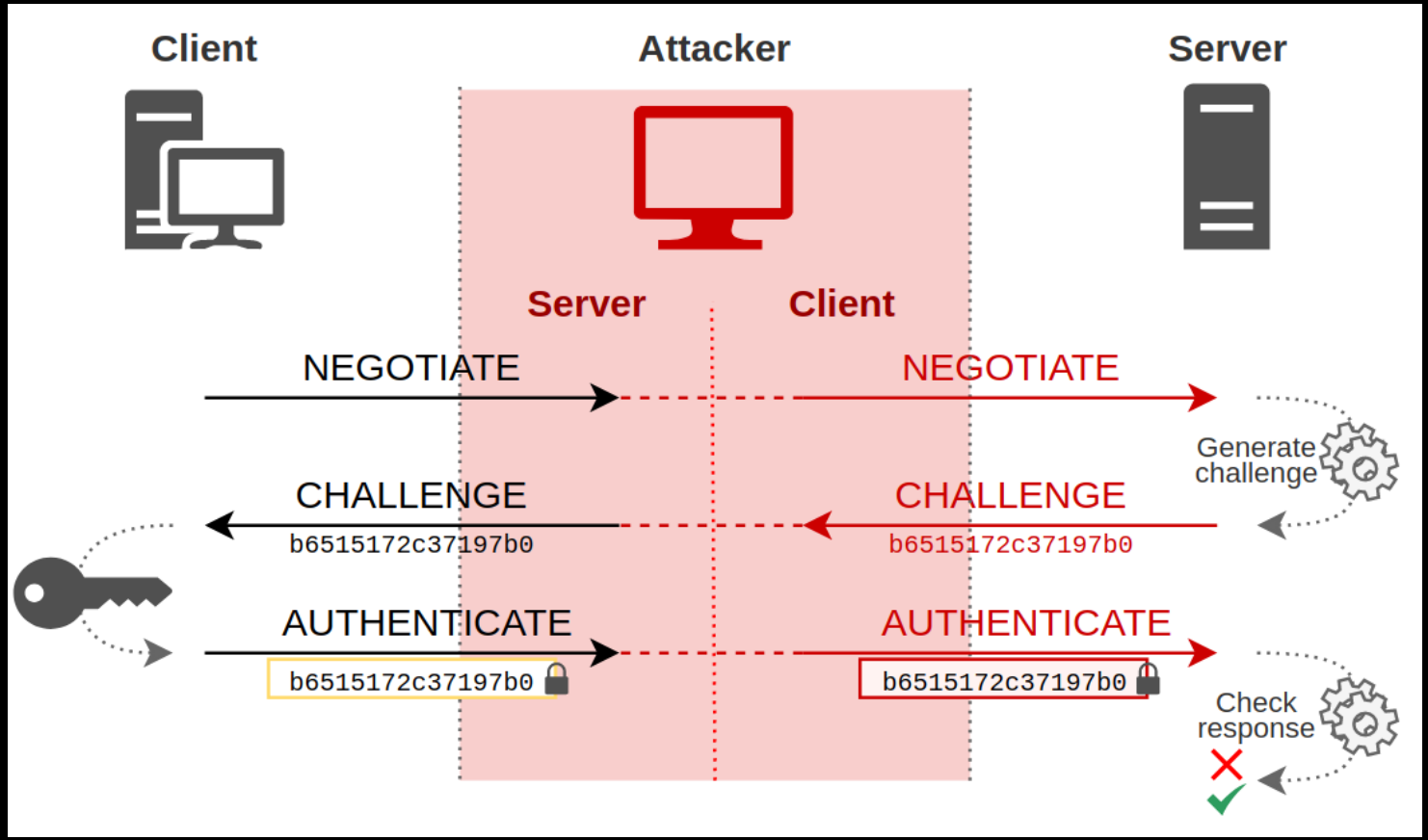
NT LAN Manager 1.0 (NTLMV1)



1. First the client tells the server that it wants to authenticate.
2. The server then responds with a challenge which is nothing more than a random sequence of characters.
3. The client encrypts this challenge with its secret, and sends the result back to the server. This is its response.
4. (Not Shown here) Confirmation or Denial. The server decides to or not to authenticate the client.

https://en.hackndo.com/ntlm-relay/

NT LAN Manager 1.0 (NTLMV1)



https://en.hackndo.com/ntlm-relay/

SAM

- Contains the password hashes of local users
- When attempting to dump the SAM database with Meterpreter you need to use an x64 payload
- You MUST have SYSTEM-level permissions for hashdump to be successful

lsass.exe

- This process contains password hashes for users currently logged into the Windows system
- Exists in active memory
- These users may be local accounts or domain logins

mimikatz

- Well known to extract plaintexts passwords, hash, PIN code and kerberos tickets from memory.
- mimikatz can also perform pass-the-hash, pass-the-ticket, build Golden tickets, play with certificates or private keys, vault, ... maybe make coffee?

https://github.com/gentilkiwi/mimikatz/wiki