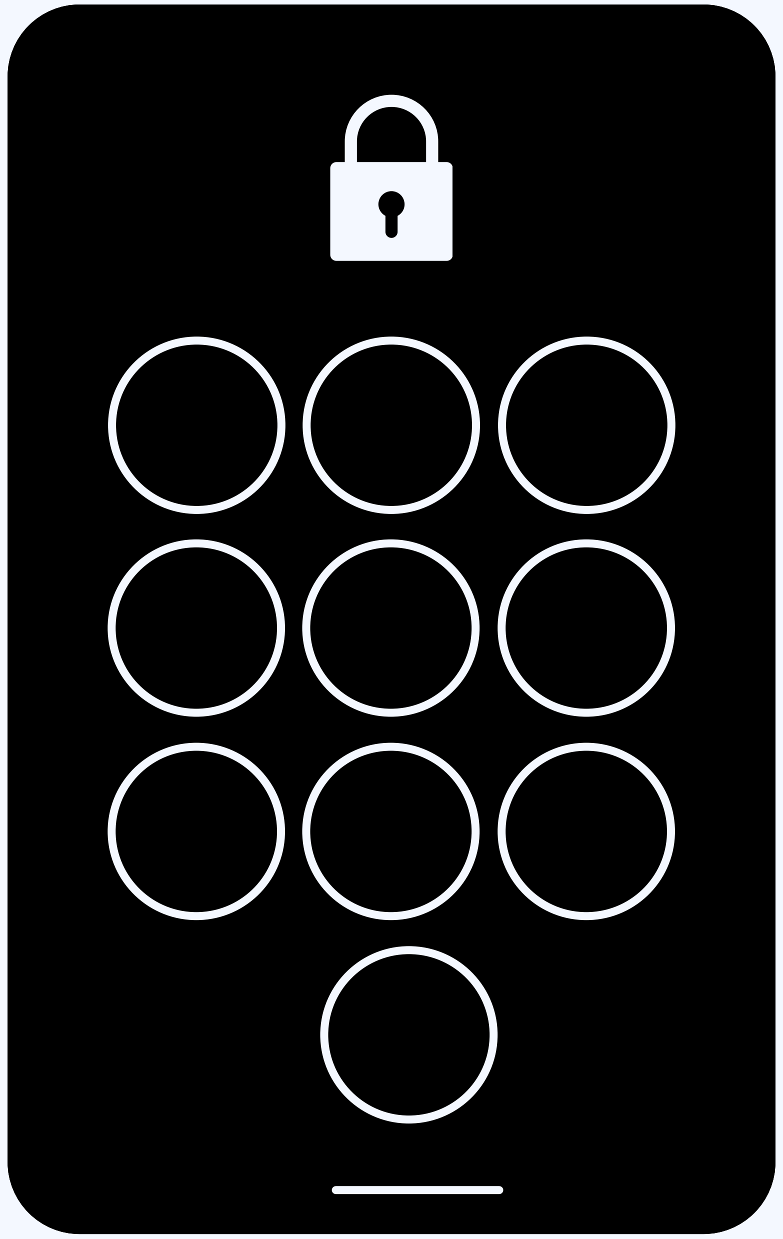
Cyber Protect Data Threat Security Attack Firewall Malware

DATA INTEGRITY



HANNAH EMAD 2205123
MARIAM MOSTAFA 2205084
NADA MOHAMED 2205173

INTRODUCTION

What is a MAC?

- MAC = Message Authentication Code
- Ensures:
  - Integrity: Detects unauthorized changes
  - Authenticity: Verifies the sender identity
  - Replay Protection: Blocks reuse via timestamps/nonces

WHY ARE MACS CRITICAL?

Real-World Risks if MACs Fail

1. Length Extension Attack: Grants admin privileges
2. Weak Hash Functions (MD5): Allow forgery
3. Timing Attacks: Reveal key bits via delay
4. Collision Attacks: Two different messages = same MAC

LENGTH EXTENSION ATTACK

Vulnerability

- When MAC = hash(secret || message)
- Hash functions: MD5, SHA-1, SHA-256 (Merkle-Damgard based)
- Attacker doesn't know the secret key but guesses its length

ATTACK REQUIREMENTS

Prerequisites

1. Uses MD5 or SHA-1
2. MAC structure: hash(secret || message)
3. Attacker can guess secret key length (bytes)

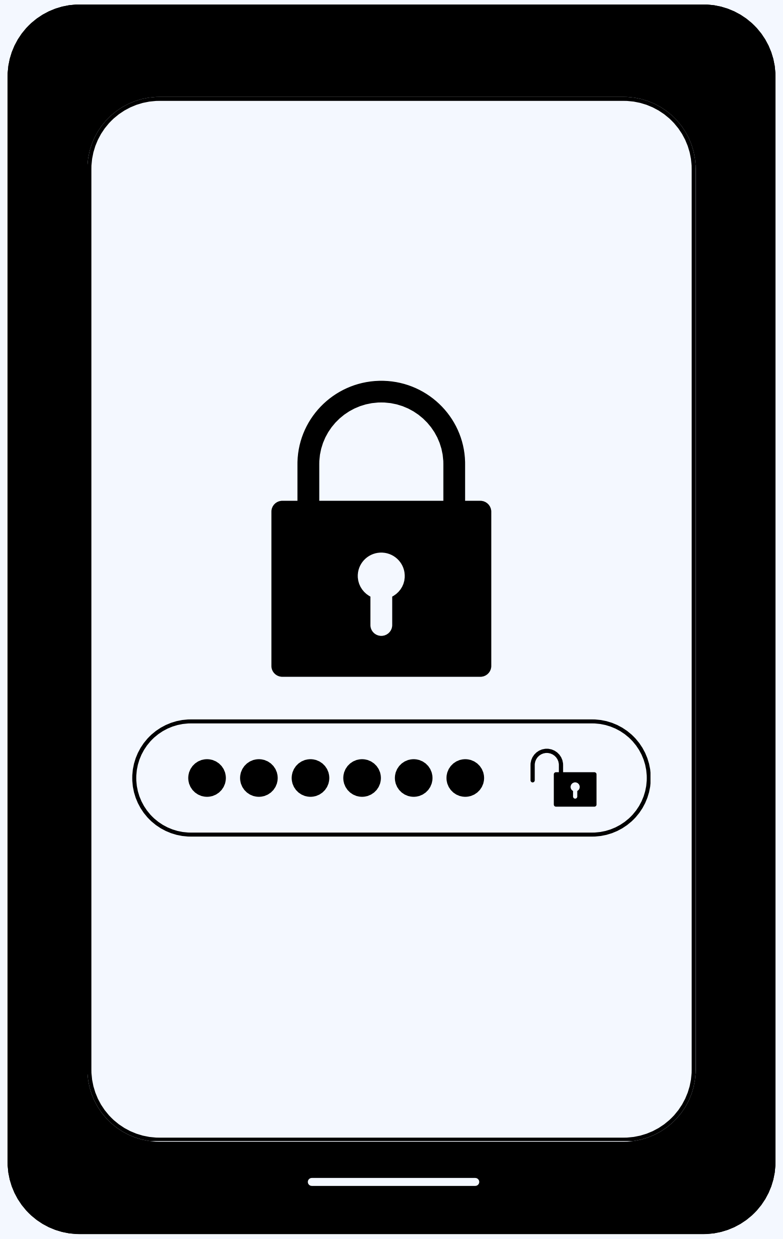
Attacker has

1. Original Message: amount=100
2. Original MAC
3. New Data: &admin=true

Cyber Protect Data Threat Security Attack Firewall Malware

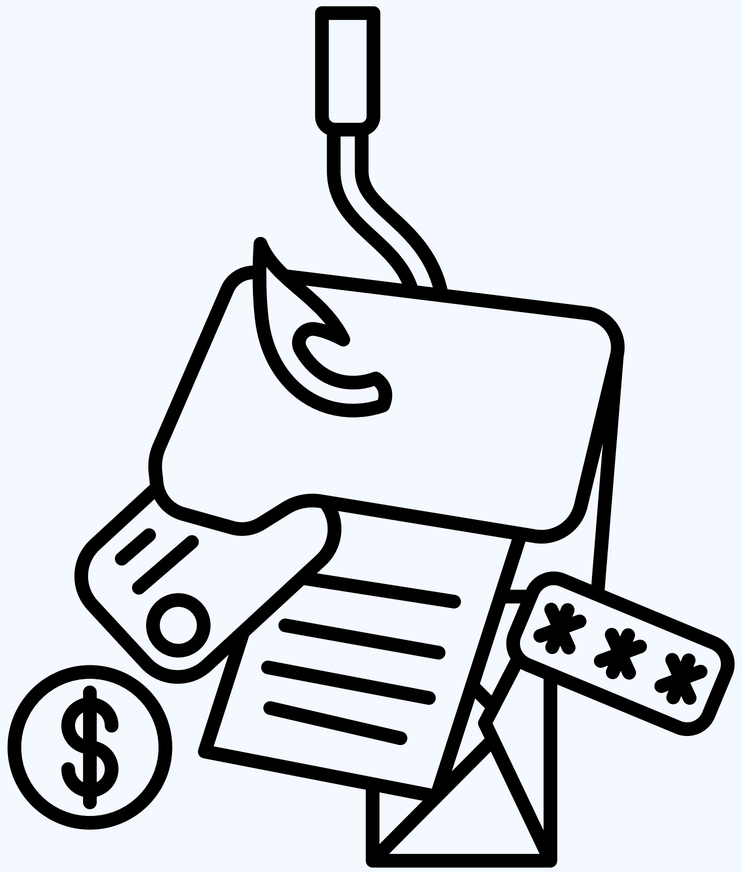
ATTACK MECHANISM

Let's Take a Break



STEPS
ATTACK MECHANISM

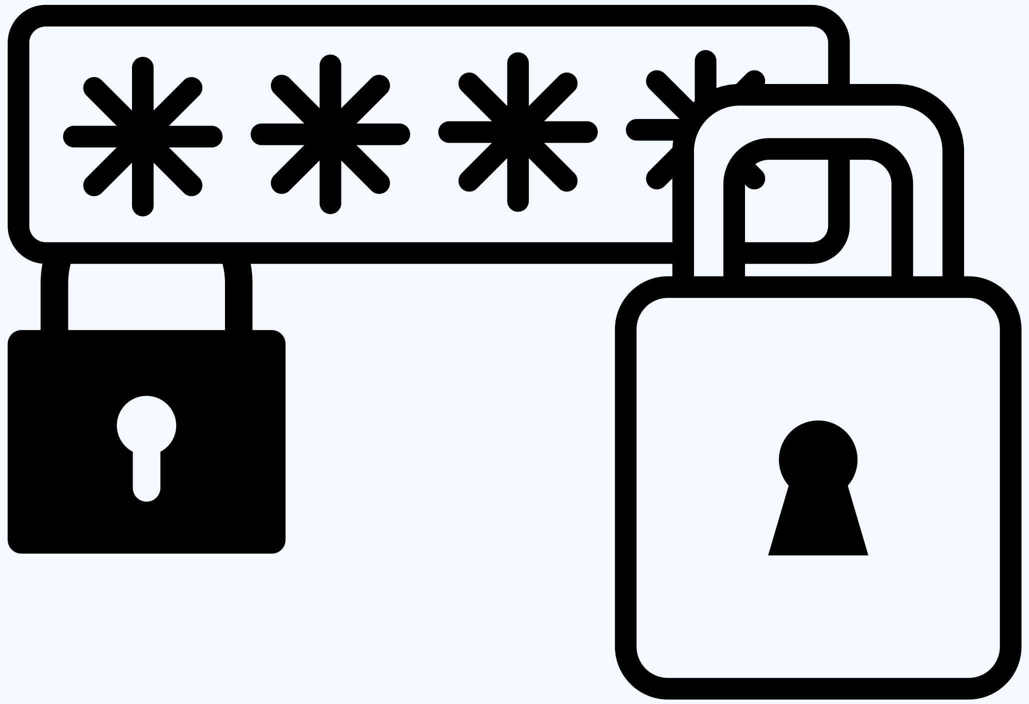
1. Intercept original message & MAC
2. Use hashpumpy to:
    - Generate hash padding
    - Append &admin=true
    - Create forged MAC
3. Server accepts forged message if using naive hash



WHY IT WORKS?

Hash Internals:

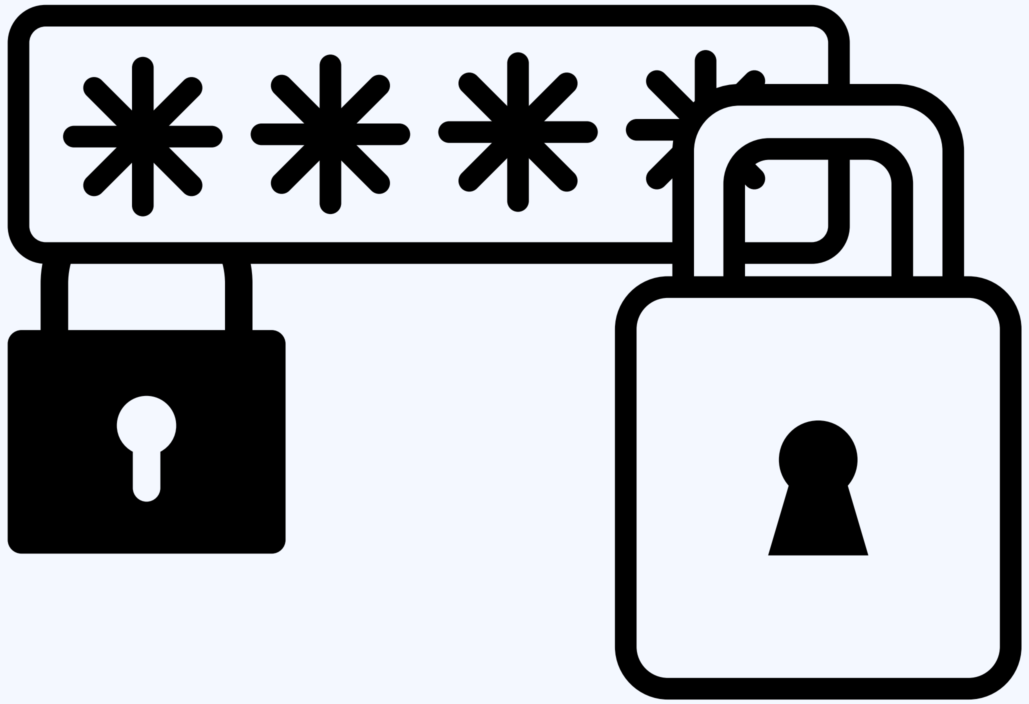
- MD5/SHA-1 process data in blocks (512 bits)
- Attacker uses original MAC as internal state
- Hash continues with appended data
- No need for the key



ATTACK WORKS

client.py on vulnerable server.py

- Attacker forges: amount=100&admin=true
- Server accepts forged MAC
- Admin privilege gained without key



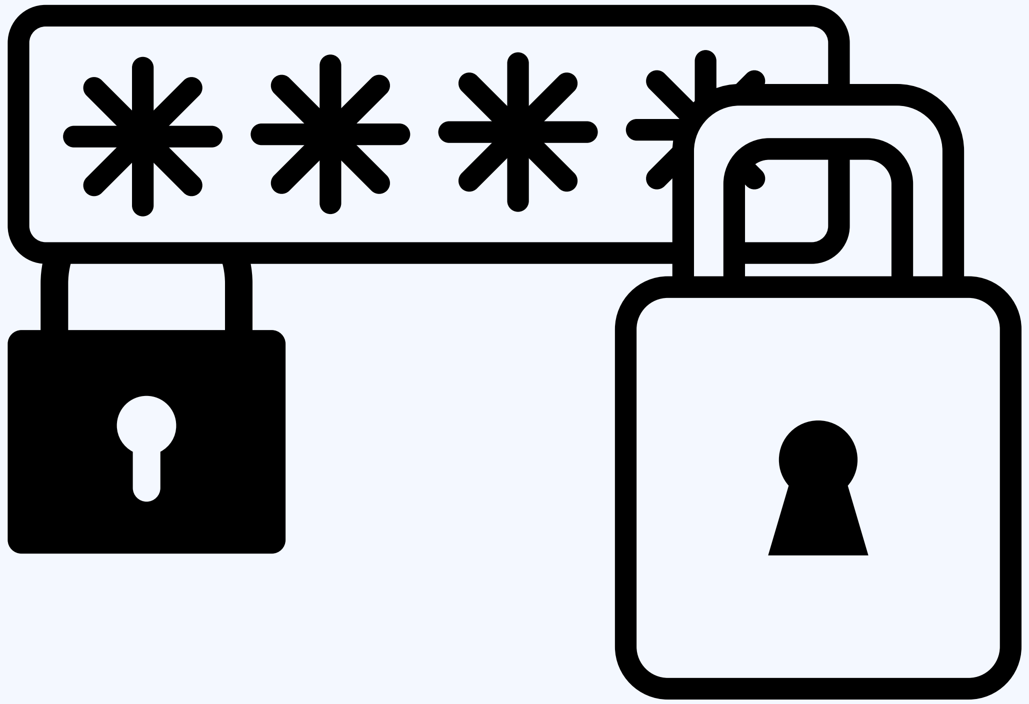
CRYPTOGRAPHIC DEFENSE

Why Naive MACs Fail

- hash(secret || message) is vulnerable to length extension

How HMAC Fixes It

- Uses: HMAC(key, message)
- Internally: H((K ⊕ opad) || H((K ⊕ ipad) || message))
- Prevents length extension
- Key never exposed

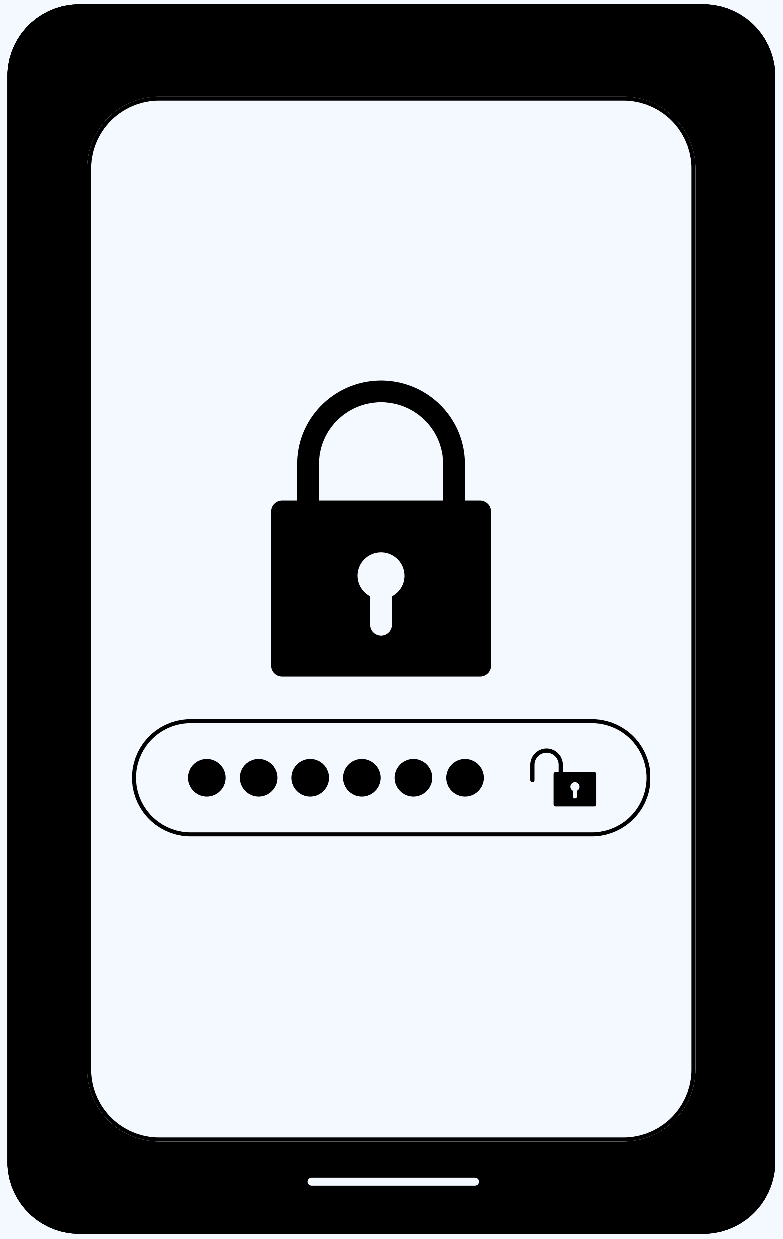


Cyber Protect Data Threat Security Attack Firewall Malware

SECURE SERVER

client.py on secure_server.py

- HMAC-SHA256 used
- Forged message rejected
- MAC cannot be extended



CONCLUSION

- MACs protect data integrity & authenticity
- Naive use of hash functions is risky
- Length Extension attack is a real threat
- HMAC is a secure alternative
- Our demo showed how attacks can succeed/fail depending on the design

Cyber Protect Data Threat Security Attack Firewall Malware

THANKYOU

Stay Safe, Stay Secure