secure report

This report outlines the vulnerabilities in using naive constructions of Message Authentication Codes (MACs) with concatenated secrets and messages , and recommends HMAC as a secure alternative.

Identified Security Issue

Vulnerable Construction:

MAC= hash(secret || message)

Why This Is Insecure:

- Allows **length extension attacks**.
- If a hacker knows hash(secret || message), they can:
  - **Guess the length** of the secret.
  - **Append additional data** to the original message.
  - **Forge a valid MAC** that bypasses authentication checks.

Underlying Problem:

- Most hash functions (e.g., MD5, SHA-1, SHA-256) process input in blocks.
- The internal state can be extended with attacker-controlled input, producing a valid hash without knowing the secret.

Recommended Mitigation

Use HMAC (Hash-based Message Authentication Code)

Advantages of HMAC:

- Designed to withstand length extension attacks.
- Properly mixes the key and message using:
  - Inner hash: hash((key $\oplus$ ipad) || message)
    -Here, the secret key is combined with the

inner padding , then concatenated with the message, and hashed.

- Outer hash: hash((key ⊕ opad) || inner_hash)
  -Then, the result of the inner hash is concatenated with the secret key combined with
  outer padding and hashed again.

- Remains secure even if the underlying hash function is weak.

Why use inner and outer padding?

- The two different paddings (ipad and opad) ensure the key is mixed with the message in **two distinct ways**, preventing attackers from manipulating or extending the hash.
- This double hashing **hides the internal hash state** so attackers cannot perform length extension attacks.
- It also **strengthens security** by making it difficult to reverse-engineer the key or forge a valid MAC.

Conclusion:

Simply hashing a secret with a message is unsafe because attackers can change the message and still create a valid hash. HMAC fixes this by mixing the key and message in two steps, making it secure against these attacks. Always use HMAC for strong message protection.