Mitigation of MAC Forgery Attacks using HMAC

Introduction

The naive MAC construction using hash(secret || message) is vulnerable to length extension attacks due to the properties of hash functions like MD5 and SHA1. To prevent such attacks, we use a more secure construction called HMAC (Hash-based Message Authentication Code).

Why HMAC is Secure

HMAC avoids the vulnerabilities of simple concatenation by applying a two-step hashing process:
1. It first applies an inner hash using the key XORed with an inner pad.
2. Then it applies an outer hash using the key XORed with an outer pad.

This structure ensures:
- The internal state of the hash function is never exposed.
- The attacker cannot predict or forge a valid MAC even if they know the hash output and the message.

Demonstration Results

When the naive implementation was tested using hashpumpy, the length extension attack successfully forged a valid MAC. However, once the implementation was switched to HMAC, the same attack failed entirely. The server rejected all forged messages, confirming that the vulnerability was fully mitigated.

Conclusion

Using HMAC provides strong protection against length extension and other forgery attacks. It should always be used in real-world applications instead of insecure constructions like hash(secret || message).