Mitigation Write-up: MAC Forgery Assignment

1. How Does HMAC Solve the Problem?

HMAC (Hash-based Message Authentication Code) is a secure construction that mitigates the vulnerabilities of naive hash-based MACs, such as `MAC = hash(secret || message)`. HMAC uses two layers of hashing combined with the secret key to produce a MAC that is resistant to length extension attacks. Specifically, HMAC computes the MAC as follows: `HMAC(key, message) = hash((key XOR opad) || hash((key XOR ipad) || message))`, where `opad` and `ipad` are constant padding values. This process ensures that the key is mixed with the message in a way that prevents an attacker from manipulating the internal state of the hash function. Unlike the insecure construction, HMAC does not allow an attacker to append data to the message and compute a valid MAC without knowing the secret key. By using a robust keying mechanism and double hashing, HMAC provides strong guarantees of integrity and authentication.

2. Why Did the Attack Fail After Using HMAC?

The length extension attack succeeded against the original `MAC = hash(secret || message)` construction because MD5 is vulnerable to length extension, allowing an attacker to append data and compute a new valid MAC. However, when the system was modified to use HMAC, the attack failed. This is because HMAC's design prevents the attacker from reusing the hash's internal state to extend the message. In the demonstration, the forged message and MAC generated during the length extension attack (e.g., appending `&admin=true` to `amount=100&to=alice`) were verified using the original MD5-based MAC function, and the server accepted them. After switching to HMAC, the same forged message and MAC were tested, but the verification failed. This happened because HMAC incorporates the secret key in a way that requires knowledge of the key to produce a valid MAC for any message, including the extended one. The double-hashing and key-padding mechanism of HMAC ensures that the attacker cannot forge a valid MAC without the secret key, thus protecting the system from such attacks.