Replay Attack - what it is, how it works, and how to prevent token reuse

Gridinsoft Help Center

What it is

A replay attack is when an attacker captures a valid network message (like a login or payment request) and sends it again to trick a system into granting access or repeating an action. The attacker doesn't need to read or change the contents-just re-transmit them at the right time. Replay usually rides on a compromised or untrusted network path (malware on the device, rogue Wi-Fi, or local MitM) and succeeds when the protocol lacks freshness checks such as nonces, timestamps, or one-time tokens.

Why it matters

Replay can bypass passwords and some MFA flows by reusing session cookies, bearer tokens, or previously valid requests. That can mean unauthorized logins, duplicated transactions, or account changes without the victim realizing.

How it works - quick tour

- Intercept: capture an authenticated request or token on a weak or compromised path.
- Store: hold the message until it's useful or the token is still valid.
- Re-send: transmit the same bytes to the server to repeat the original effect.
- Pivot: use the resulting session to pull data, change settings, or move laterally.

Red flags

- Identical requests hitting an API with the same payload and headers within short intervals.
- Successful logins that skip fresh MFA challenges, followed by token reuse from new IPs.
- Duplicate transaction IDs or nonce failures in logs.
- Sudden activity from untrusted networks shortly after a user's legitimate action.

Prevent it

- Enforce TLS with strict certificate validation; block captive portals and rogue APs on sensitive devices.
- Add freshness: nonces, one-time tokens, signed requests, and short token lifetimes.
- Bind tokens to context (client TLS key, device ID, or IP range) and invalidate on change.
- Use SameSite+HttpOnly+Secure cookies; rotate sessions after auth and on privilege changes.
- Require server-side replay detection (nonce store, timestamp windows, idempotency keys).
-