What is a signature replay attack?

A signature replay attack reuses a valid cryptographic signature to authorize an action more than once, or in a context it was not meant for. Smart contracts often accept off-chain signatures (for gasless approvals, meta-transactions, permits), and if the signed message lacks a nonce, a deadline, a chain ID, or the contract address, an attacker can replay it, repeating a withdrawal or replaying it on another chain or contract. The fix is binding each signature to a unique, single-use, scoped context (the EIP-712 pattern).

HOW IT WORKS

01 How it works and example

Replay arises when the signed data omits binding fields:

- No nonce: a signature authorizing "transfer 100" can be submitted repeatedly, draining the account.
- No deadline: an old signature stays valid forever and can be used much later.
- No chain ID: a signature valid on one chain is replayed on another (cross-chain replay).
- No contract address / domain: a signature for one contract is accepted by another using the same scheme.

The attacker simply re-submits the captured signature. Documented for defensive context.

> SCOPE EVERY SIGNATURE
>
> *A safe signed message includes a nonce (single-use), a deadline, the chain ID, and the contract address (domain). The EIP-712 typed-data standard bundles these so a signature works once, here, before it expires.*

HOW TO DEFEND

- Include a unique nonce per signature and mark it used, so each signature works only once.
- Add a deadline/expiry so old signatures cannot be replayed later.
- Bind to the chain ID and contract address (the EIP-712 domain separator) to stop cross-chain and cross-contract replay.
- Use EIP-712 typed structured data rather than raw message signing.
- Audit every signature-verification path for missing nonce, deadline, or domain binding.

SOURCES

[1] SWC Registry: Smart Contract Weakness Classification

[2] Ethereum.org: Smart contract security

[3] MITRE CWE-294: Authentication Bypass by Capture-replay