GCM NONCE REUSE BUGS AS AN EXAMPLE OF EASY TO MISUSE CRYPTO CONSTRUCTIONS

Hanno Böck, Aaron Zauner

https://github.com/nonce-disrespect/nonce-disrespect

"There's also an annoying niggle with AES-GCM in TLS because the spec says that records have an eight byte, explicit nonce. Being an AEAD, the nonce is required to be unique for a given key. Since an eight-byte value is too small to pick at random with a sufficiently low collision probability, the only safe implementation is a counter. [...] Thankfully, all the major implementations use a counter and I did a scan of the Alexa, top 200K sites to check that none are using random values - and none are." (Blog post by Adam Langley)

NONCE

Two encryptions with same Key+Nonce: broken

XORing cancels out key, Forbidden Attack by Joux

HOW TO SELECT NONCE?

Counter: good (repeats after 2^64 encryptions, unrealistic)

Random: risky (likelyhood of repeating nonce becomes realistic around 2^29 and high around 2^32)

Repeating: broken

THE SPEC (RFC 5288 / TLS 1.2)

Each value of the nonce_explicit MUST be distinct for each distinct invocation of the GCM encrypt function for any fixed key. Failure to meet this uniqueness requirement can significantly degrade security. The nonce_explicit MAY be the 64-bit sequence number.

BAD SPEC

Tells the implementor to make sure nonces must be distinct, but gives no advice how to do so properly.

INTERNET-WIDE SCAN

184 hosts with repeating nonces

72445 hosts with random looking nonces

AFFECTED DEVICES

Duplicate nonce (high severity vuln): Radware, (unnamed - disclosure pending)

Random nonce (low severity vuln): IBM Lotus Domino, A10, Sangfor

Probably more.

WHAT TO DO?

Spec for TLS 1.3 and Chacha20/Poly1305 does it better: Nonce is defined by spec, faulty implementations will thus be unable to connect to correct implementations.

SIV

Synthetic IV: Avoid nonce reuse issue on algorithm level.

Adds overhead / complexity, but avoids nonce reuse issues.

CONCLUSION

Specs should try to avoid pitfalls for implementors if possible.

If that's not possible specs should be specific on how to avoid pitfalls.

AES-GCM in TLS 1.2 fails in both regards.