

Forum:[Hack Forums](#)

Thread: [Symmetric Encryption & Decryption in Unity WebGL games.](#)

T3n - April 21, 2025, 1:07 pm

Hello big brains, i'm planning on rewriting an already existing game in Unity (the old game is flash based & not available anymore. Its a multiplayer 2D MMORPG. I'm planning to use a websocket as server & want to use Symmetric messages encryption between client & server. The problem is that Unity WebGL games can be decompiled & people can see the structures of functions & strings outside the functions. The question is - how do i hide the decryption key inside the client ? Make it up at run time or just hide it somewhere?

Any answer is welcome.

Thanks,
T3n.

PercevalFox - April 21, 2025, 2:51 pm

(Apr 21, 2025 01:07 PM)T3n Wrote: [CITE](#) Hello big brains, i'm planning on rewriting an already existing game in Unity (the old game is flash based & not available anymore. Its a multiplayer 2D MMORPG. I'm planning to use a websocket as server & want to use Symmetric messages encryption between client & server. The problem is that Unity WebGL games can be decompiled & people can see the structures of functions & strings outside the functions. The question is - how do i hide the decryption key inside the client ? Make it up at run time or just hide it somewhere?

Any answer is welcome.

Thanks,
T3n.

Hey, I'm not 100% sure, i've never done that, but i can give you some quick security tips:

1. No hard-coded AES key

Attackers will pull it out of your WebGL build.

2. Use ephemeral ECDH key exchange

Client & server each generate a key pair, swap public keys over WSS, then derive a fresh AES-GCM session key.

3. Always WSS/TLS depending how you do things

Even if you do your own encryption, run over `wss://` to block MITM (man in the middle, IDK if you know this kind oof attack).

4. Compile to WASM + obfuscate

– Enable Unity IL2CPP for WebGL

– Run an obfuscator or code-stripping tool for avoid reotr-engineering

5. Integrity check & key rotation

– On startup, verify a hash of your JS/WASM against a server value

– Renegotiate ECDH every X minutes or messages

That way You don't hide a key, you create it at runtime via a secure protocol, and you never leave it hard-written.. ?

I think, this is the best secure way to do, but it's kinda difficult if you don't know "exactly" what to do and how to do it ^^

BTW Good luck bro =)

T3n - April 21, 2025, 8:43 pm

(Apr 21, 2025 02:51 PM)PercevalFox Wrote: [\[CITE\]](#) Hey, I'm not 100% sure, i've never done that, but i can give you some quick security tips:

1. No hard-coded AES key

Attackers will pull it out of your WebGL build.

2. Use ephemeral ECDH key exchange

Client & server each generate a key pair, swap public keys over WSS, then derive a fresh AES-GCM session key.

3. Always WSS/TLS depending how you do things

Even if you do your own encryption, run over `wss://` to block MITM (man in the middle, IDK if you know this kind oof attack).

4. Compile to WASM + obfuscate

– Enable Unity IL2CPP for WebGL

– Run an obfuscator or code-stripping tool for avoid reotr-engineering

5. Integrity check & key rotation

– On startup, verify a hash of your JS/WASM against a server value

– Renegotiate ECDH every X minutes or messages

That way You don't hide a key, you create it at runtime via a secure protocol, and you never leave it hard-written.. ?

I think, this is the best secure way to do, but it's kinda difficult if you don't know "exactly" what to do and how to do it ^^

BTW Good luck bro =)

MITM was the first thing i learned ages ago x), big thanks. Your suggestion sounds perfect, i'll do it that way.

PercevalFox - April 22, 2025, 12:17 am

(Apr 21, 2025 08:43 PM)T3n Wrote: [\[CITE\]](#) MITM was the first thing i learned ages ago x), big thanks. Your suggestion sounds perfect, i'll do it that way.

You're very welcome my friend =)

Okey, if you are aware of MITM, that's a very good thing for your project =)