Secure your dapp

We recommend implementing security controls, such asHTTPS andContent Security Policy (CSP) , to improve the security of your dapp and protect your users.

caution The following security advice isn't exhaustive.

Use HTTPS

HTTPS can protect your dapp against attackers who might try to eavesdrop or tamper the communication channel between your dapp and your users. HTTPS encrypts data transmitted between the web server and the user's browser, making it difficult for attackers to intercept or modify the data.

To secure your dapp using HTTPS, obtain an SSL/TLS certificate from a trusted certificate authority (CA). For exampleLet's Encrypt offers free SSL/TLS certificates.

Install the certificate it on your web server. If you're using a static website hosting service, it might have a default way to enable HTTPS on your dapp.

Use Content Security Policy

Content Security Policy (CSP) is a security feature that can protect your dapp against various types of attacks, such ascross-site scripting (XSS) andclickjacking .

CSP defines a set of policies that the browser must follow when displaying the dapp. See the full list of CSP directives that you can enable for your dapp in theMDN CSP reference documentation .

Use CSP with a server setup

If you dapp uses a server setup, enable CSP by setting theContent-Security-Policy header in all responses from your server. For example, in Express.js, add the following middleware at the top of your server:

app . use ((req , res , next)

=>

{ res . setHeader ("Content-Security-Policy" , "default-src 'self'; frame-ancestors 'none'") ; next ( ) ; }) ; In a header, this looks like the following:

Content-Security-Policy: default-src 'self'; frame-ancestors 'none' Seemore examples of CSP in popular web frameworks and languages.

Use CSP with a static site

If your dapp uses a third party hosting provider, and you can't set a customContent-Security-Policy header in the server responses, you can enable CSP by using the HTML tag .

Add this tag to thehead section of an HTML file to instruct the browser on which CSP directives should be followed. For example:

< head

> < meta http-equiv = " Content-Security-Policy " content = " default-src ' self ' ; frame-ancestors ' none ' " /> </ head

Configure your CSP

CSP configuration is specific to each dapp. We recommend starting with the following secure and restrictive CSP:

default-src 'self'; frame-ancestors 'none' * default-src 'self' * - By default, your dapp's code can't load or connect to content from outside * your domain. * frame-ancestors 'none' * - Your dapp can't be embedded within the webpage of another domain (to * preventclickjacking attacks *).

From here, you can make adjustments as needed by your dapp to support the content you want to load. For example, if your dapp loads images hosted onOpenSea , you can enable this by addingimg-src 'opensea.io' to your CSP:

default-src: 'self'; frame-ancestors 'none'; img-src: 'opensea.io' For more information and common use cases for CSP, see theMDN CSP documentation .

Edit this page