# 604.https://stackoverflow.com/questions/69395126/verify-user-is-owner-of-an-nft-via-metamask-connection-make-sure-connected-user

**T:**Verify user is owner of an NFT via MetaMask connection? Make sure connected users public eth address is the same as of the NFT?

**Q:**I need to verify on my own domain/server the user which has connected his MetaMask wallet is the owner of a specific NFT in order to allow him special functions? Basically, I want to give the user access to an area that only the owner of this NFS would have.  
  
My original NFT is sold in opensea but I can't use the opensea hidden-area option to just give the user a hidden password since the next owner (after reselling) and the old owner would have the same password and old owners could still access like this. But I need that only the current owner has access.  
  
My user/visiter can already connect with MetaMask at my own domain and I get the public ETH address of the active account but since this is only javascript and my backend is PHP I can't just post the MetaMask info to my PHP backend since this would be easy to trick/hack.  
  
How can I make sure the current connected MetaMask Account is the same as the NFT owner (which I know) and allow to access a URL only for this user?  
  
My current state is that the user connects his MetaMask and I use opensea API to check who is currently the owner of the NFT. I can compare both eth addresses but the flaw in this is obviously that I use ajax to send the MetaMask public address to my backend which is only for testing since this is of course zero save!  
  
Thank you in advance for any idea, help, tip I can get.  
  
PS: My backend is PHP

**C1:**Thanks for sharing your answer! Question for you. I am considering something similar to you. Why did you want a password for your user to access somewhere? Was that just part of the user experience that you desired? Could you have just "verified with MetaMask that the user is the legit owner of the NFT" and been done with is and given them functionality that way?

1 **Answer**

**A1:**After hours and days of researching, I found a solution that works for me.  
  
Here are the needed steps.  
  
 ● Use the MetaMask API to let the user connect with your site. This is pretty easy and good explained in the MetaMask API.  
  
 ● Once you want to verify the MetaMask owner is the legit owner of an NFT you need first query OpenSea (or another place) the current owner of the NFT. In my case, I use the OpenSea API for my specific NFTs. Once you got the owner you are ready to verify.  
  
 ● On your site you need to ask the user to sign a custom message with MetaMask. There are different options to do that. More about this here: MetaMask Signing. I send for example a short text message with a unique code that I first created in my PHP Backend. Doing that I also save the code and custom message into my MySQL.  
  
 ● Once the user has signed you get the signing code which you can send back to your PHP backend via ajax etc. without a problem. Only the owner of the Account which you requested in the signing code is able to sign with the correct account.  
  
 ● Once you got the signing code in your PHP Backend you can use Fast Elliptic Curve Cryptography in PHP and php-ecrecover to check the code against the unique code you created before and the message the user signed. As a response, you get the Signer Account ETH Address and you are ready to compare. If the Signers ETH Address is the same as the NFT owner you are ready to go and you can consider the signer the owner of the NFT.  
  
I believe this is safe to use but I am not an expert on that. In my case, this only authorize an NFT owner for a certain closed area in my community page and there are not really high risks involved but maybe somebody raises some security thoughts on that. However, I found that other NFT pages and even Opensea work similarly.  
  
I hope this points someone in the right direction, I lost quite some time figuring this out because most solutions are Node.js etc. but not with PHP backends.

**C1:**this is really cool; I'm new to this as well; question, does this have to be metamask? What if the user stores it a trusted wallet?