Section 9: 2FA/MFA

Before Reading...

2FA & MFA are advanced level concepts. If you haven't learned how to use 1Password effectively or are still struggling to be confident using it, **do not read this section yet**. We would advise spending more time with 1Password until you've gained enough experience to attempt more advanced level functionality and concepts. These concepts can be somewhat confusing, so once you're sure you've got the basics of 1Password down, return here and continue reading to expand your knowledge of passwords and security.

Introduction to 2FA & MFA

This guide is meant to give you an overall understanding of Two-Factor Authentication and Multi-Factor Authentication. In addition, there's an example for how to add Time-based One Time Passwords as a second factor on an account to 1Password. The example is for Microsoft 365 but is general enough for most other kinds of accounts.

Understanding Multi-Factor Authentication

Multi-Factor Authentication (MFA) is a method for identifying and granting access to users on computers, applications, websites, and other portals. Two or more factors are used in the process of identifying and allowing access for a user. A user must have/submit all factors in order to be granted access to whatever platform they are attempting to use.

> ## A Note about Passkeys □□
>
> Passkeys are a newer system of linking people and accounts with biometric authentication and no actual password. The thought being that a password cannot be compromised if it doesn't exist. As more companies and services begin adopting this authentication method, you will start seeing this appear as an option more and more. While this has merit for home use, this particular feature is not as useful to us at a company level, as it doesn't provide a balanced approach to both security and flexibility. **We highly recommend that passkeys never be used for company purposes.**

Background

The typical user credentials most of us are used to these days include a user name and password. In the past, a memorable string of characters (e.g. "abcd1234" or "mydogmax") was secure enough. The more characters— and types of characters (e.g. numbers `0 - 9`, letters `a - z` and `A - Z`, specials `. , _ - ! @ # $ ...` etc)— the more secure it was because it was harder to guess.

The Need for More Security □□

However the ever increasing power of computing allows attackers/hackers to guess short passwords rather easily. This can be helped by increasing the number of characters used and using less predictable patterns.

For example, `55` is much easier to guess than `25797142593691473697` and `apple` is much easier to guess than `ixqvt`.

But can you imagine trying to type— much less remember— a crazy password like `{oGxQS"ipZ)*2]'#pnP9/\sK`? No! Which is why we try to create passwords that are easy to remember and type, like `pumpkin.Engine82`. But, a short password like that isn't enough. We could go longer (`pumpkin.Engine82-harpIcon_Door!Toadstool`) but the longer it gets, the harder it is to remember. Plus, even a password like this could be guessed with enough time. We need a better way. This is where MFA comes in to save the day.

What is Multi-Factor Authentication, really?

Put simply, we use multiple "factors" such as passwords, physical items, access to lines of communication, and biometric information to authenticate a user instead of a single factor like just a password. You might already be familiar with a type of MFA called **Two Factor Authentication** (2FA). The most common 2FA people know of today uses SMS text messages or email sent to your phone or computer with a one time use code.

Two Factor Authentication □□ □□

Two Factor Authentication (2FA) is a type of MFA. Basically, 2FA is MFA but just using two factors. That's why you will sometimes hear 2FA and MFA used interchangeably. Having a second factor besides a password adds a layer of work for attackers/hackers to perform beyond just guessing your password. They would have to guess your password *and* have access to your phone or email. And while it's trivial for us to just have our phone with us or log into our email accounts, it's another step for nefarious actors that isn't so trivial.

> “□□
>
> ## A Note about Two Step Verification (2SV)
>
> You might run across a similar term to 2FA called **2SV** or "Two Step Verification. Generally you'll see this terminology from big tech companies like Google or Microsoft. For the most part you can think of 2FA and 2SV as being similar but there are some specific differences. Two Step Verification involves multiple steps where you provide one factor at a time. Two Factor Authentication can involve steps or not, but like 2SV uses two factors. So you could think of 2SV as a type of 2FA, but there's more nuance around the terms that are outside the scope of this guide. Just be aware that the term exists and can sometimes be confused with 2FA.

Why we don't like using SMS for 2FA □□

SMS text messages may be convenient, but unfortunately they're not as secure as you might think. Hackers have found ways to spoof sim cards, reroute text messages and calls, or even just convince your carrier to hand over your number to them on another phone using social engineering. And because a lot of platforms allow you to reset your password just by having the second factor (one time code via SMS), attackers sometimes don't even bother with the password anymore — opting to just attack the path of least resistance. For this reason, we shy away from using SMS for 2FA and instead use something else.

Additionally, 1Password does not support SMS. But 1Password enables us to use a form of 2FA even with shared credentials, for certain types of accounts.

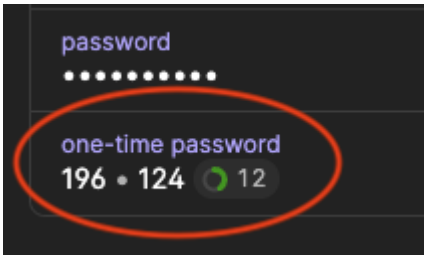
What We Use Instead for 2FA

Many platforms, accounts, and portals have the option of using another kind of second factor called **Time-based One Time Password** (TOTP). 1Password can store the TOTP along with your usual password so you never have to remember either. It works somewhat like the codes you might receive on SMS, but instead they are never transmitted to you once you have TOTP set up. They are instead generated by your device locally.

What is a Time-based One Time Password?

Imagine if your password changed every minute. That would make things complicated! How would you know which password is next? Well, someone figured that out. When you set up a TOTP, you are given a "secret code" usually in the form of a QR code. The "secret code" is just a really long string of characters. You don't need to know much about the "secret code" but if you place it in a 1Password item, the app can use that "secret code" for something really clever.

Once the "secret code" is entered into a 1Password item alongside your normal password, you'll see a short number — a string of six digits — that changes every 30 seconds.



This short number is a one-time password that will only work once for that moment in time; usually 30 seconds. It's a pseudo-random number that is generated locally on your device using a special algorithm that uses the "secret code" and the time of day as inputs. The special algorithm spits out a six digit number unique for you at that specific moment. Essentially, it's like having a password that is only valid for a short period of time; a one-time use password. And because it's generated using the current time, it is a time-based "one time password" — i.e. TOTP.

Because the service/website you're trying to log into also has a copy of the "secret code" and the current time, when you try to log in, they're able to generate the exact same "random" number on their end. And as long as both your clock and their clock are in sync, you will both generate the same number. When you send that one-time password they know it's you because the numbers match. But once it's been used, it can't be used again. Attackers can't intercept it and use it before you can. This makes it vastly more secure than SMS 2FA. The "secret code" is nearly impossible to guess, even if attackers managed to obtain a few of those six-digit codes, they couldn't reverse engineer the secret.

Why Not Just Use TOTP Without a Password?

That's a good question. Having multiple factors makes things more secure. The more different factors we have, the more complicated it is for an attacker to obtain the credentials they need to gain access where they're not supposed to.

1Password: Everything in One Place

Some of you may be saying: What?!? Aren't we invalidating the extra security by saving both the user/pass and the second factor in a single system? Yes and no; having 2FA enabled still protects us against casual sharing and phishing, since for the 2FA to work you need a code that's only valid for 30 seconds.

Our 1Password accounts are protected by a very long Access Key plus user/pass (and team URL,) and so even without 2FA enabled on the 1Password account itself, there are still security measures present - the "one password of 1Password" is just for you, not hackers!

1Password



As stated before, 1Password cannot support text message (SMS) based 2FA. Therefore we are relying on app-based TOTP as a second factor for all accounts where available.

For accounts that you only use (i.e. not shared), please enable 2FA with a TOTP when available. **Do not use proprietary authenticator apps for 2FA** such as the *Microsoft Authenticator* or the *Google Authenticator*; **only use 1Password**. We only want to use a TOTP when available for 3rd party apps (i.e. 1Password).

When a TOTP for 2FA is not available, please use stronger passwords instead. If the account forces you to use 2FA (i.e. you cannot leave 2FA disabled) then use SMS based 2FA. You may also use SMS based 2FA when it makes sense to do so, but preference is for not using 2FA and just using stronger passwords.

For accounts that you and others use (i.e. shared), **do not enable 2FA yourself.** The IT staff will set up 2FA for you where appropriate.

Note: At this time we are not using 2FA for the 1Password account itself. Again, there is already more than just a user/pass needed for access and initial setup, so TOTP along side your master password is unnecessary.

To reiterate:

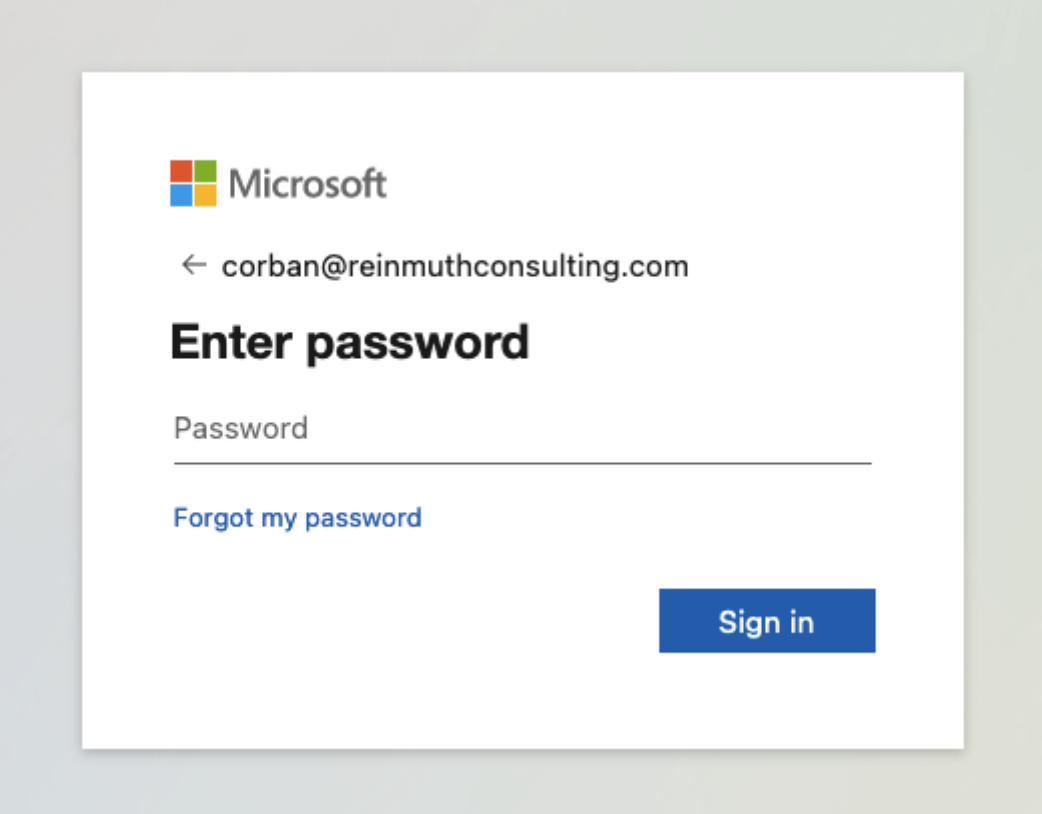
- **If** the account is added to a **1Password** item that is in ***your vault*** (accounts that you only use, i.e. not shared) it is strongly preferred that the account have 2FA enabled with an app-based TOTP.
  - **If** 2FA with a TOTP on a 3rd party app (i.e. 1Password) is not available, do not use 2FA. We instead use stronger passwords (or SMS 2FA when it makes sense).
  - **If** 2FA is forced for the account, use SMS 2FA
- **Do not enable 2FA yourself for accounts that are shared.** The IT staff will handle setting up 2FA for shared accounts.

How to add 2FA (TOTP) in 1Password

Note: For demonstration, we will be using Microsoft 365 as the account enabled with 2FA and adding the TOTP into 1Password. The process will be different for each website/service. So for each unique situation, you will have to use your best judgement. Be aware also that not all website/services support TOTP. In those cases where TOTP is not supported for 3rd party apps, we do not enabled 2FA. However if 2FA is required/forced, we will then use SMS as a last resort.

Step 1: Sign into the service/website

In this case — since we are using Microsoft 365 as an example — we will be going to `office.com` and signing in.

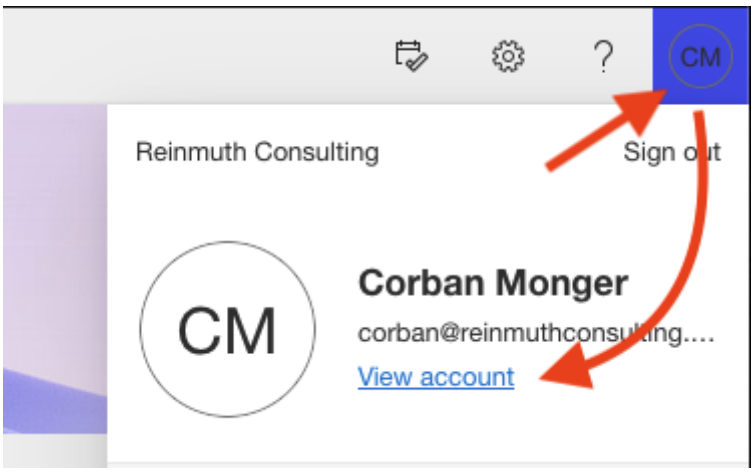


Enter the account credentials (username and password), and click Sign in.

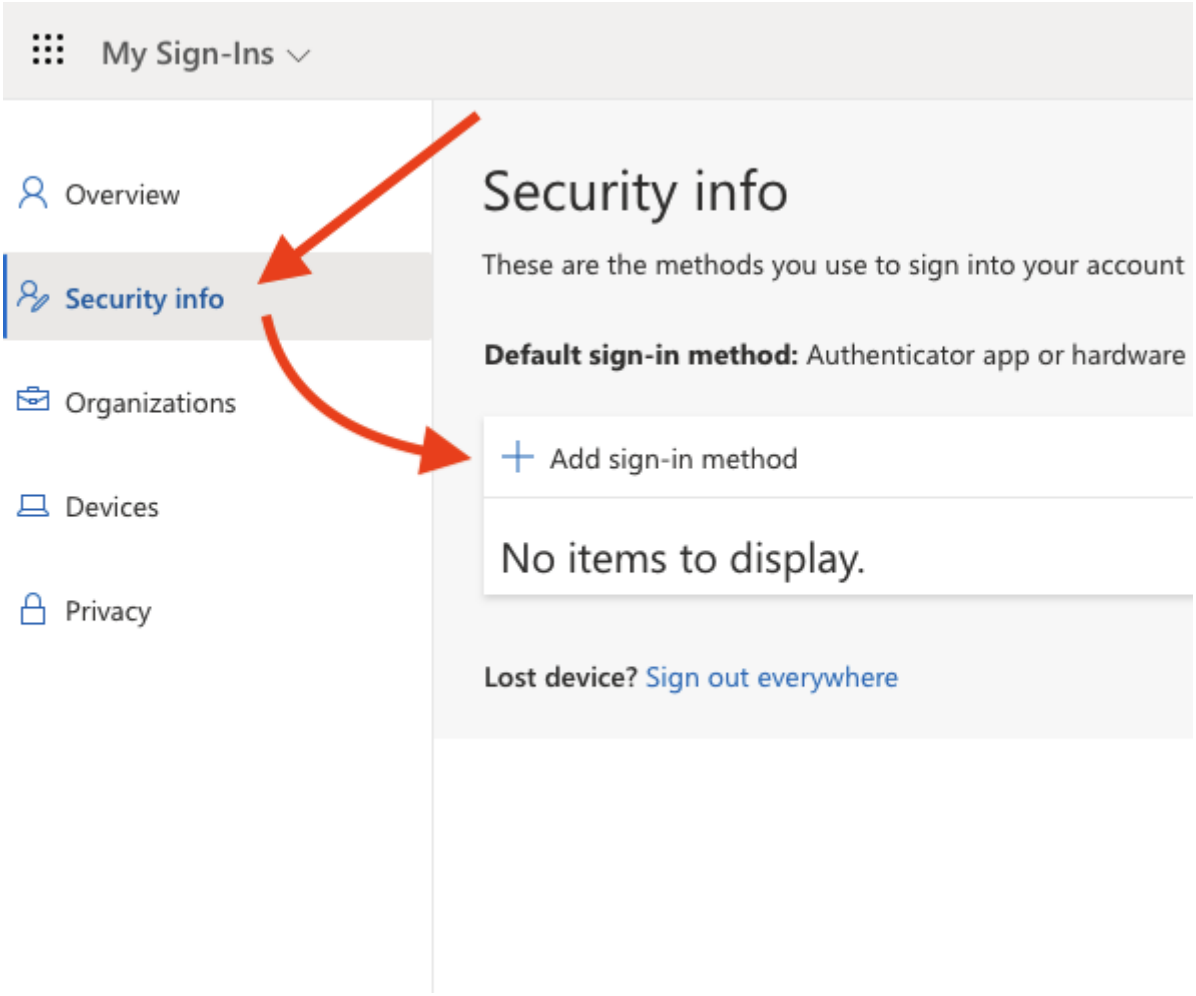
Step 2: Locate 2FA Settings

It may not be obvious at first where to find this. Usually it will be in something like "Account" or "Settings" and under a heading like "Security."

In the case of Microsoft 365, head up to the top right of the site, and click the circle with the account initials, then on View account.



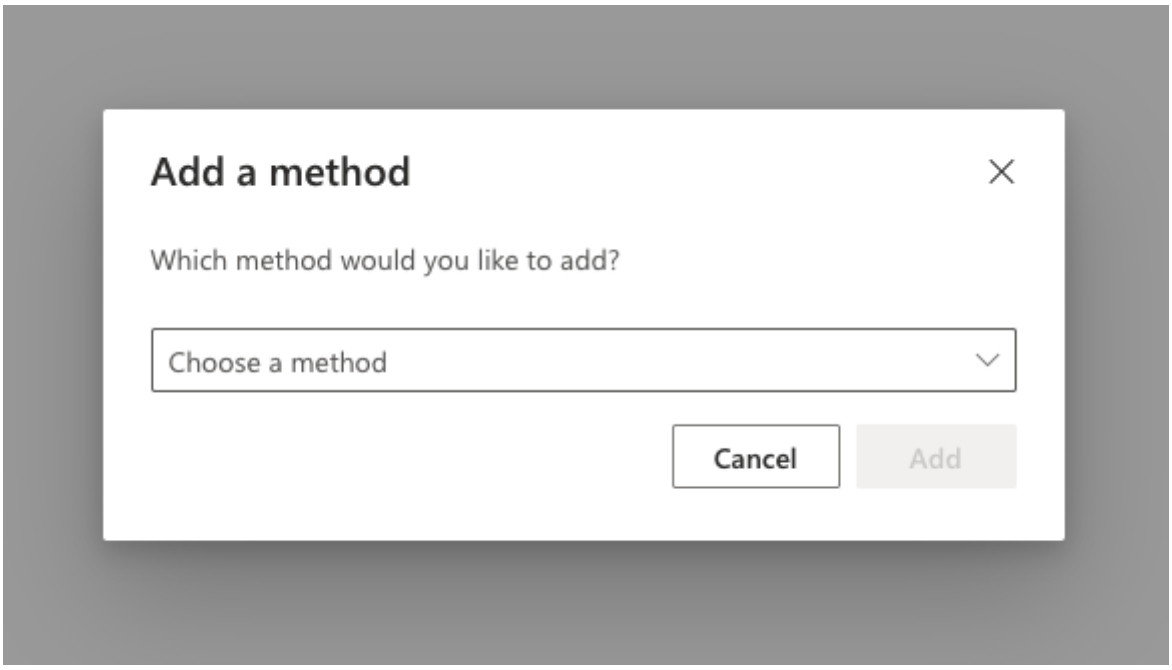
From there click on the Security info heading and then click on Add sign-in method.



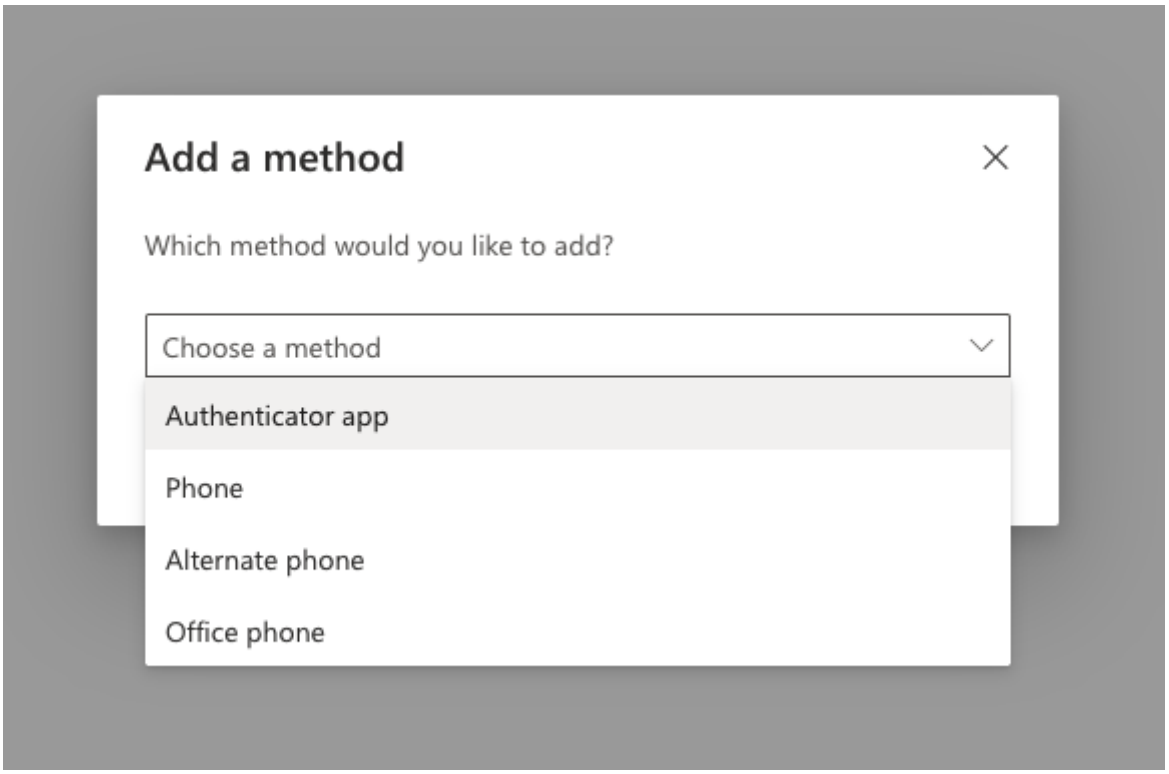
Step 3: Add or Enable TOTP 2FA

At this point you will then need to "add an authenticator" to your account. Our "authenticator" app is 1Password itself. Some services have their own apps. We do not use them. We only use 1Password. If you do not seen an option to use TOTP on a 3rd party authenticator stop and do not set up 2FA. If the service is forcing 2FA to be enabled and you have no choice, in that case you will need to use SMS as a last resort.

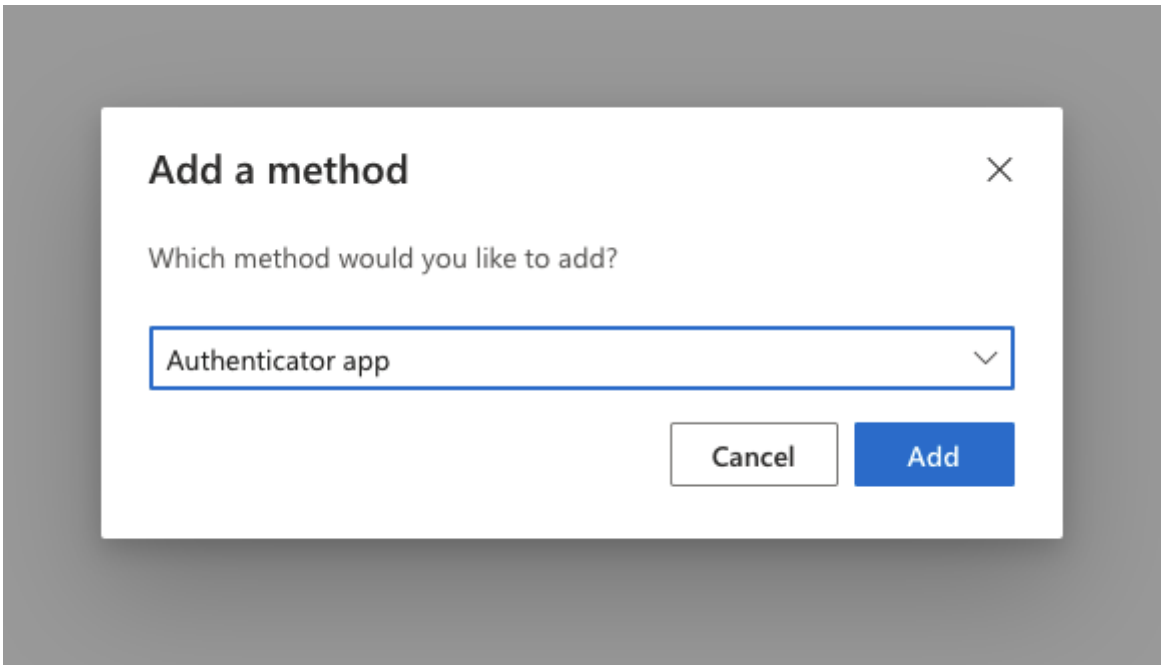
On Microsoft 365, after you clicked "Add sign-in method," a prompt should have appeared with a dropdown menu containing multiple options.



Click "Choose a method" to expand the dropdown menu and select "Authenticator app" from the list.

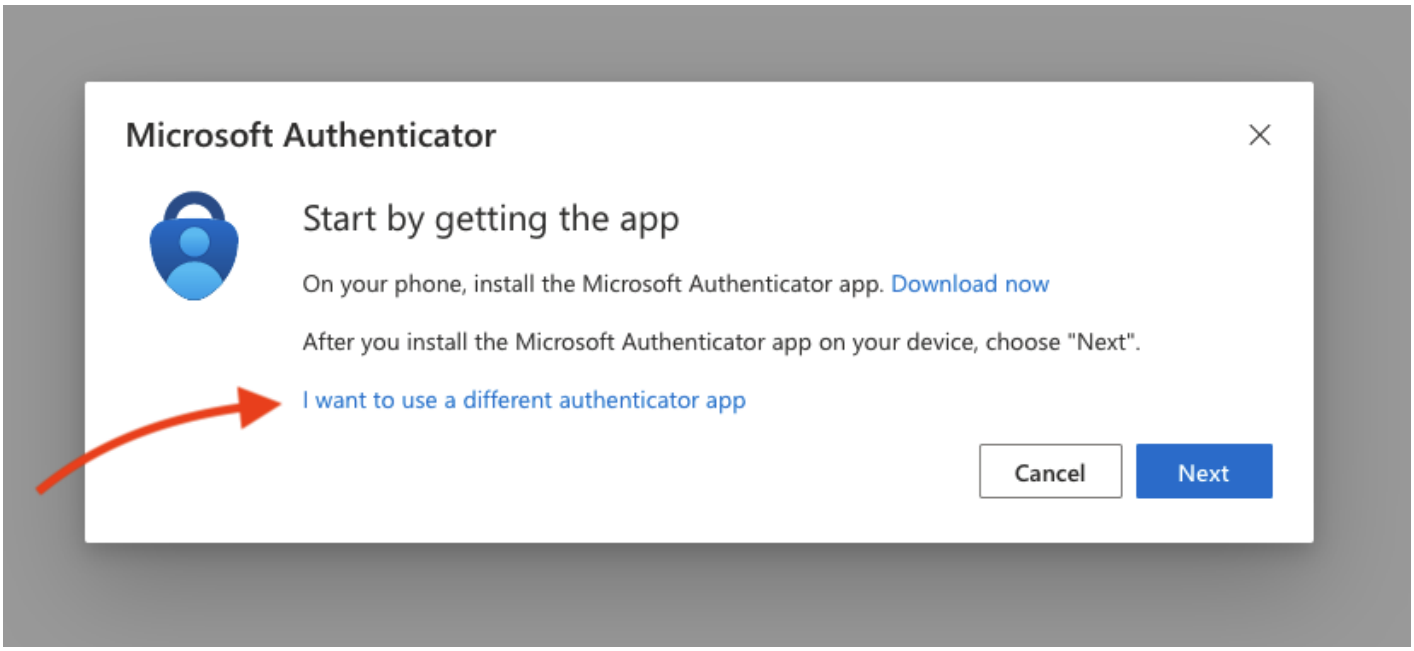


Click `Add` to continue.

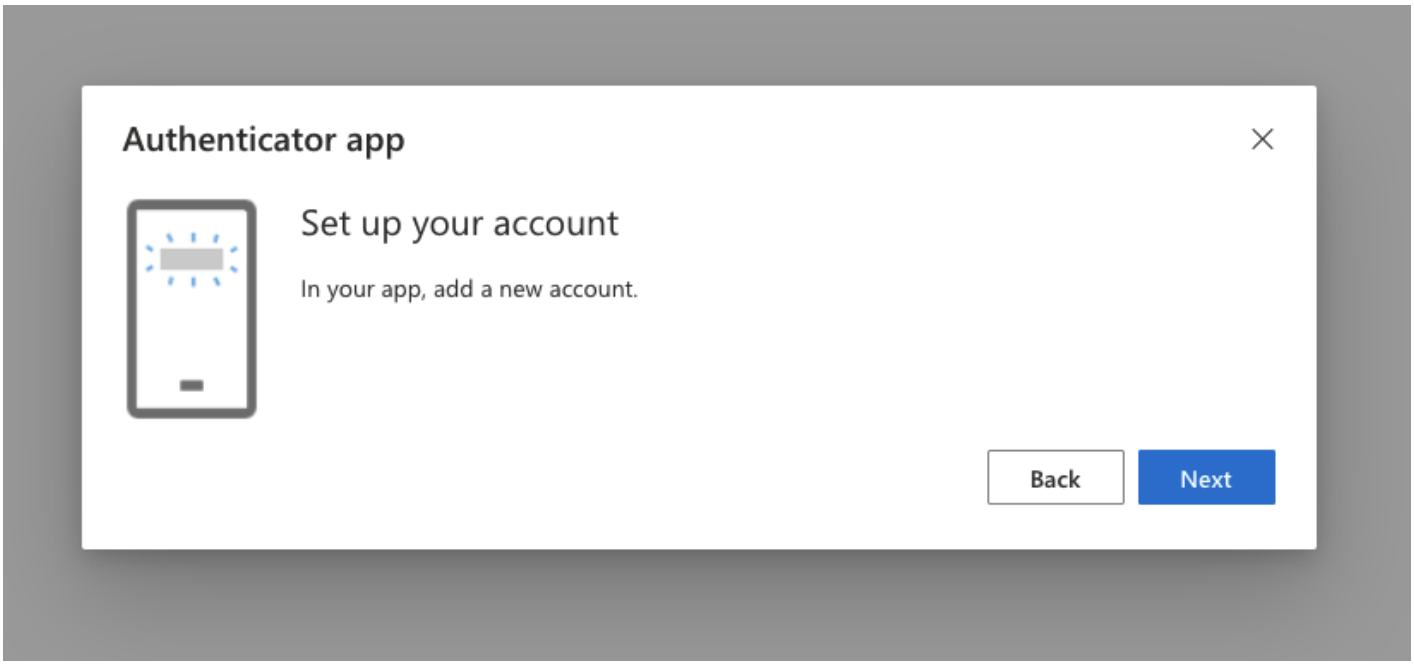


Microsoft will prompt you to use their authenticator app. We do not use their app.

Click the link that says "I want to use a different authenticator app" near the bottom.



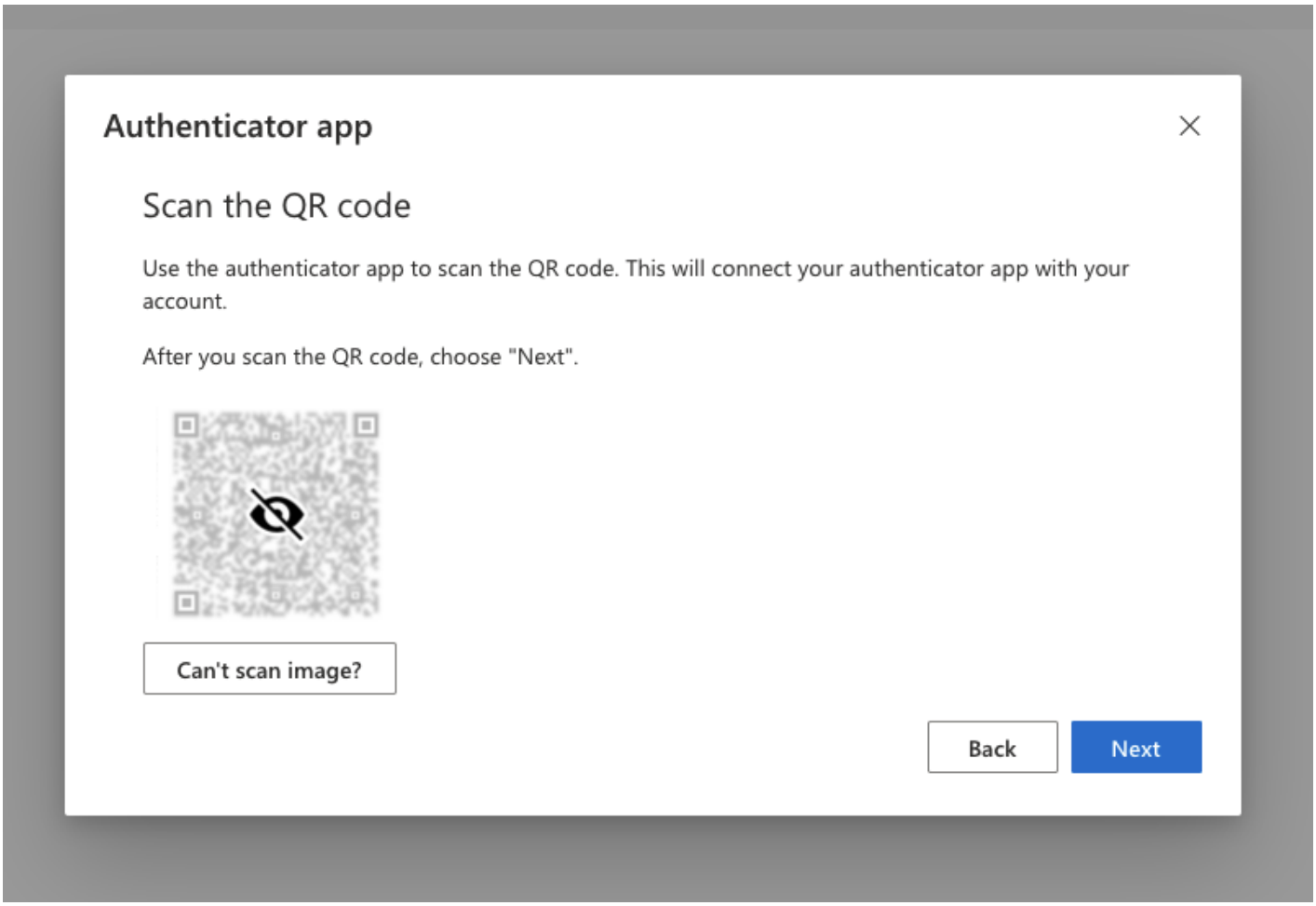
You will then be prompted to "add a new account" but this language can be ignored. What they mean is to prepare the item in 1Password, however you should already have an item in 1Password at this point. Click `Next` to continue.



Step 4: Scan the QR Code Using 1Password

When enabling 2FA with TOTP in most services, you should be shown a QR code. However in other cases you may be asked to follow a different set of instructions not covered here. Follow whatever steps are necessary to add the TOTP "secret code" to 1Password or contact IT for assistance.

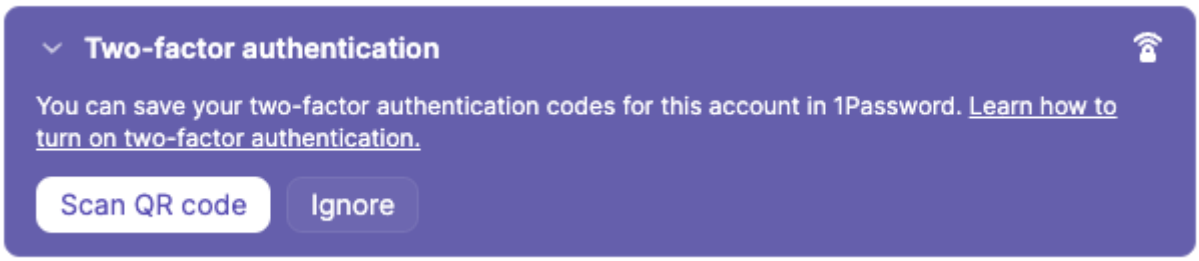
You should then see the following prompt including a QR code. In the example image below, the QR code has been altered for security purposes and will look different for you.



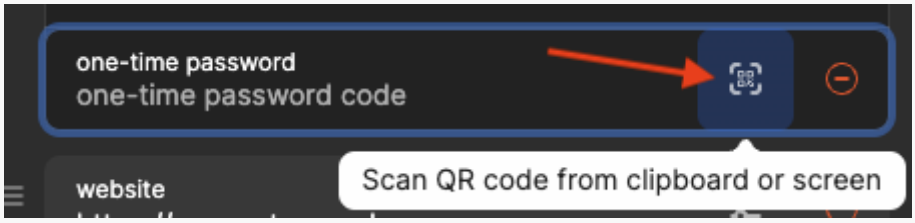
Over in 1Password — within the item for the Microsoft 365 account — you should see a banner:



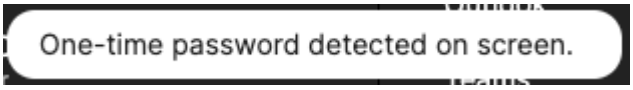
Clicking on it reveals the following:



Click the `Scan QR code` button.

> ### I can't find the Two-factor authentication banner in 1Password! □□♂
>
> If you don't see the banner and there isn't a one-time password already set up for the account, you can add the TOTP by editing the account item, clicking on `+ add more` and then choosing `One-Time Password`. You can then click the Scan QR code button indicated by the image below.
>
> 
>

1Password will attempt to look for the QR code on your screen. Make sure the QR code from the website is visible when telling 1Password to perform a scan. If successful, you should see the following notification at the bottom of the 1Password app.



Potential Issues at this Stage...

QR code detection failed.

If the One-time password is not detected on screen, you will see the following notification at the bottom of the 1Password app instead of the previously mentioned one.

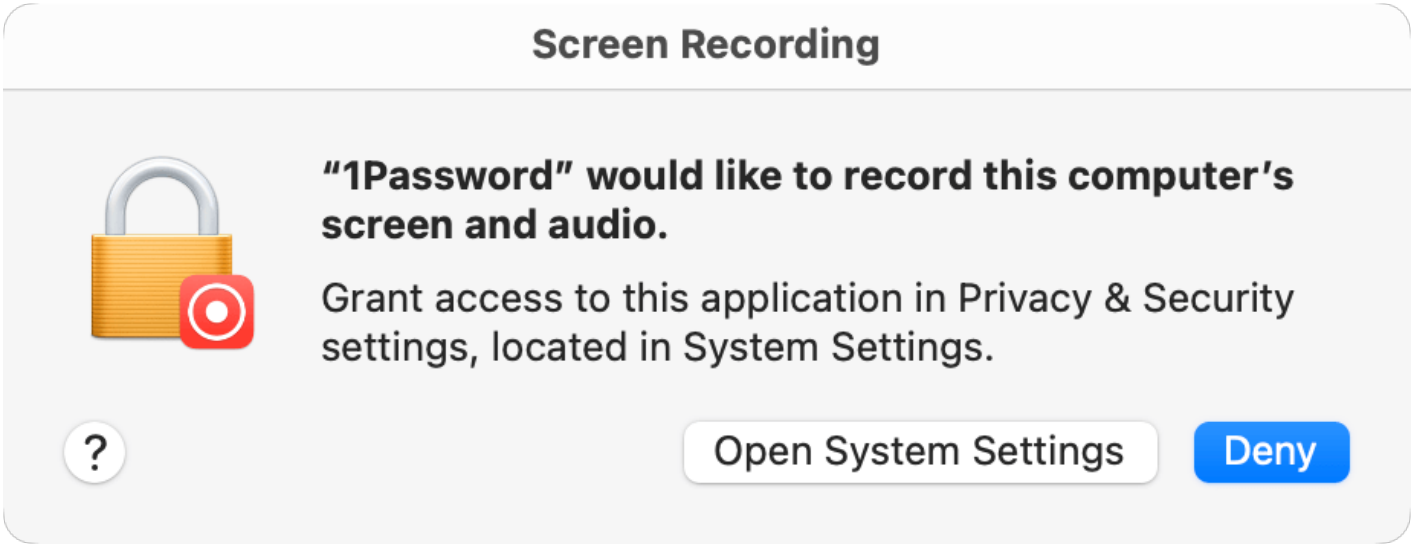


There are a few possible reasons for this:

- □□ The Screen Recording Permission has not been enabled.
- □□ The QR code is not visible on screen
- □□ The QR code is too small

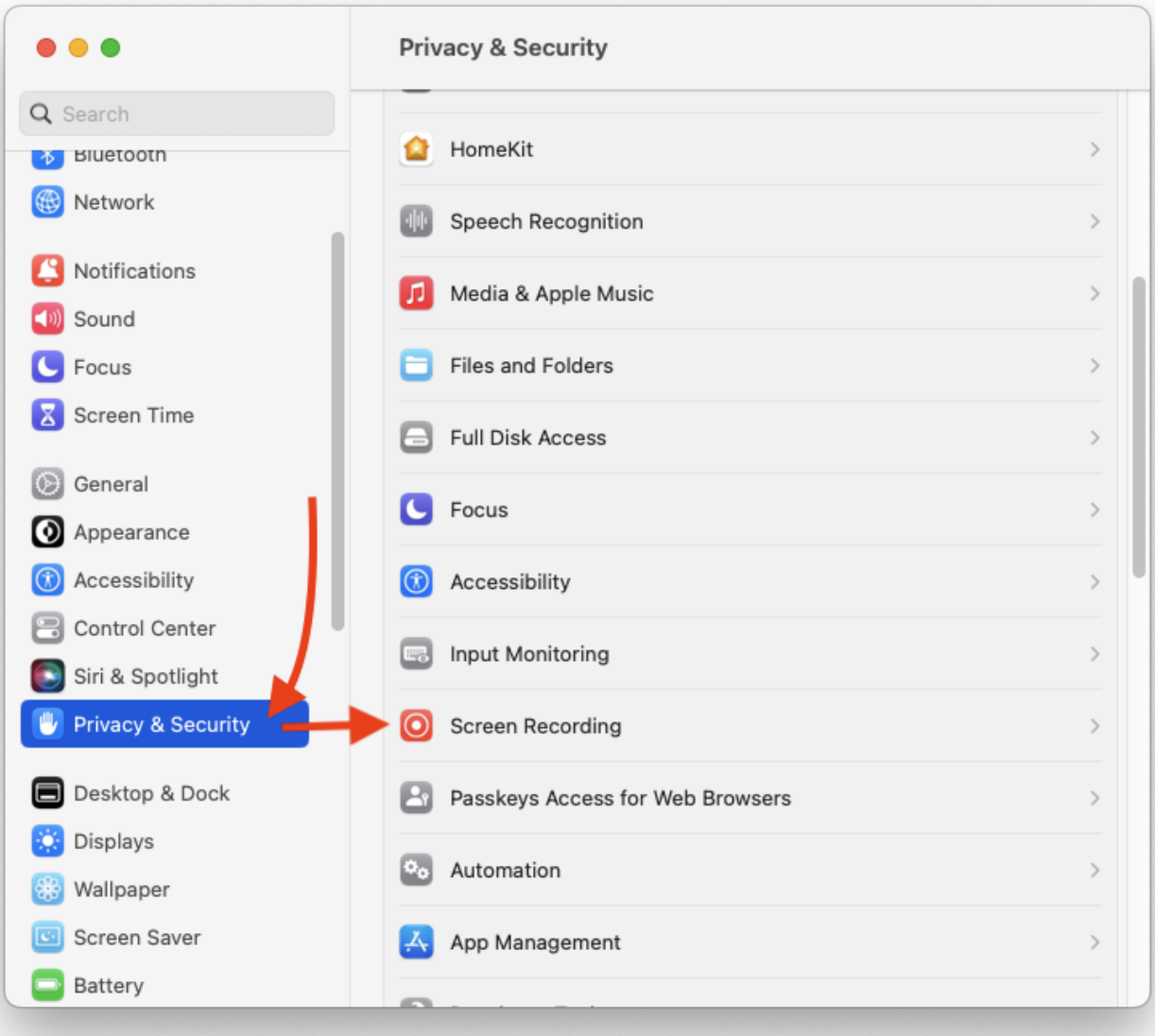
See the below answers for these possible solutions. If you verify all three and are still having trouble, contact IT for help.

□□ Screen Recording Permission

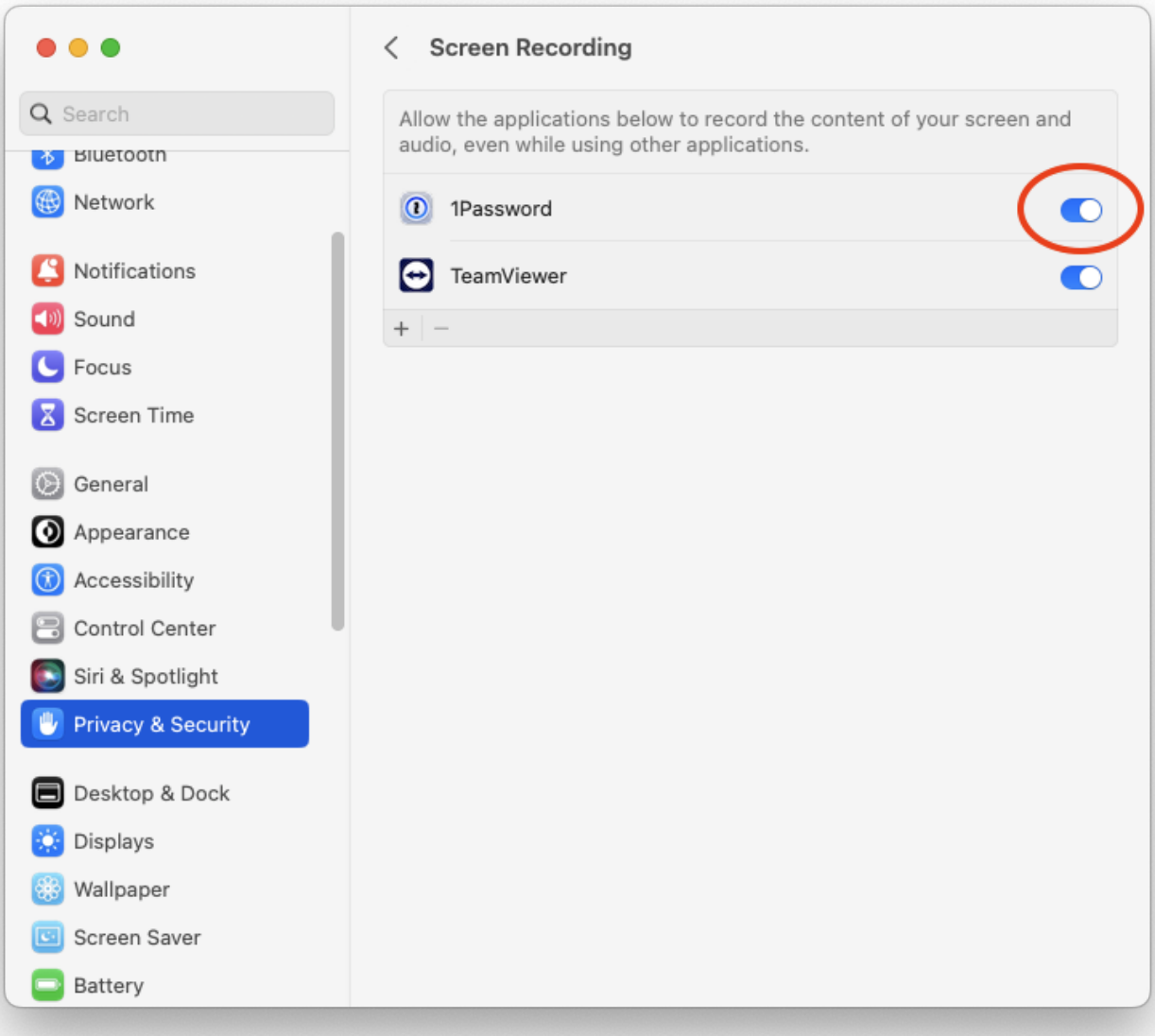


If this is the first time you've tried to scan QR codes with 1Password, you may be prompted to grant screen recording permission to the 1Password app.

Open System Settings and go to Privacy & Security and then click Screen Recording.



Next, in the list of apps displayed, enable Screen Recording for 1Password.



You may need to use Touch ID or Use Password to change the setting. Then you may be prompted to Quit & Reopen the 1Password app. Do so.

The QR code is not visible on screen

Make sure the window with the QR code is visible on your screen. Because the app uses a screen capture and looks at your entire screen, if you can't see the QR code, neither can the app! Position the windows such that you can see the full QR code at the same time you click the `Scan QR code` button in 1Password.

The QR code is too small

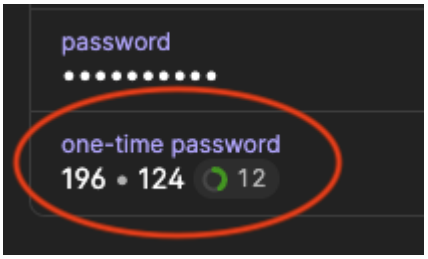
If the QR code is plainly visible, the app might be having difficulty recognizing the QR code because of the size and resolution of the image. In your web browser, try increasing the view size by going to the View menu and selecting Zoom or Zoom In. You can also press Command + Plus (`⌘` `+`) on your keyboard. Keep zooming in and trying to Scan QR Code in the 1Password app until successful.

If you've zoomed in beyond 200% and you still receive a "detection failed" message, contact IT for help.

Don't forget to return your browser's zoom back to 100%! (`⌘` `-` or `⌘` `0`)

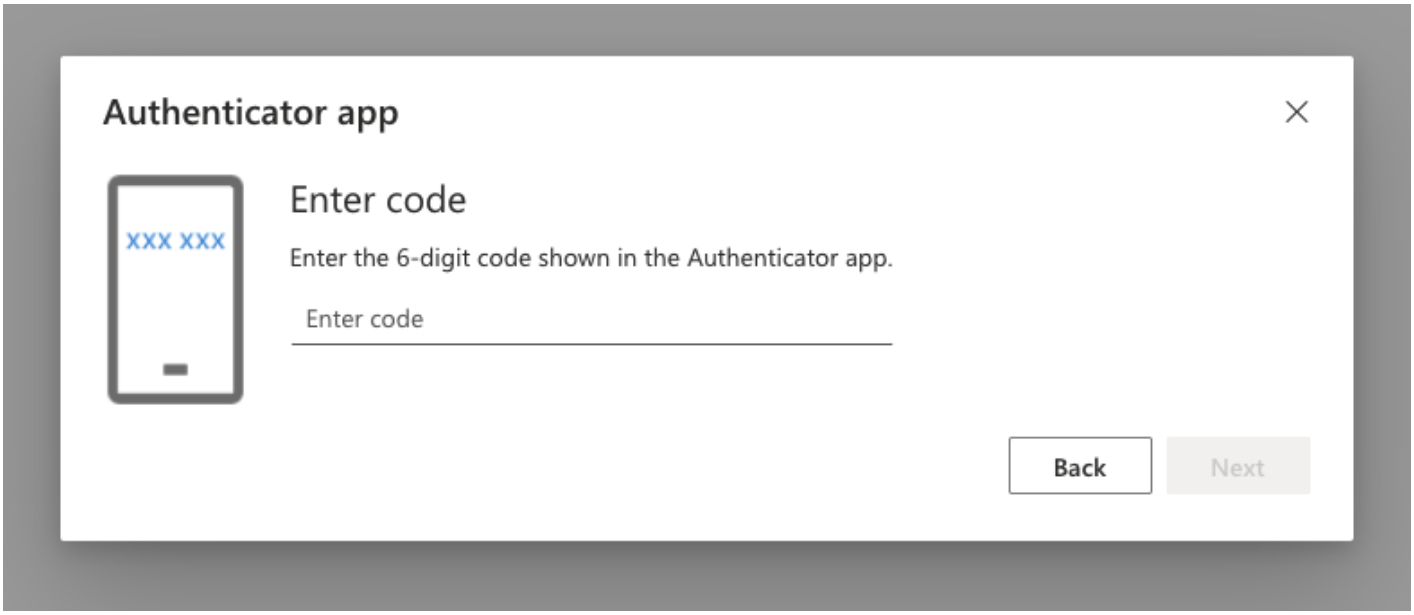
Step 5: Finishing up

If the Scan was successful, you should now see a six digit code that changes every 30 seconds, indicated by a countdown timer next to the code.

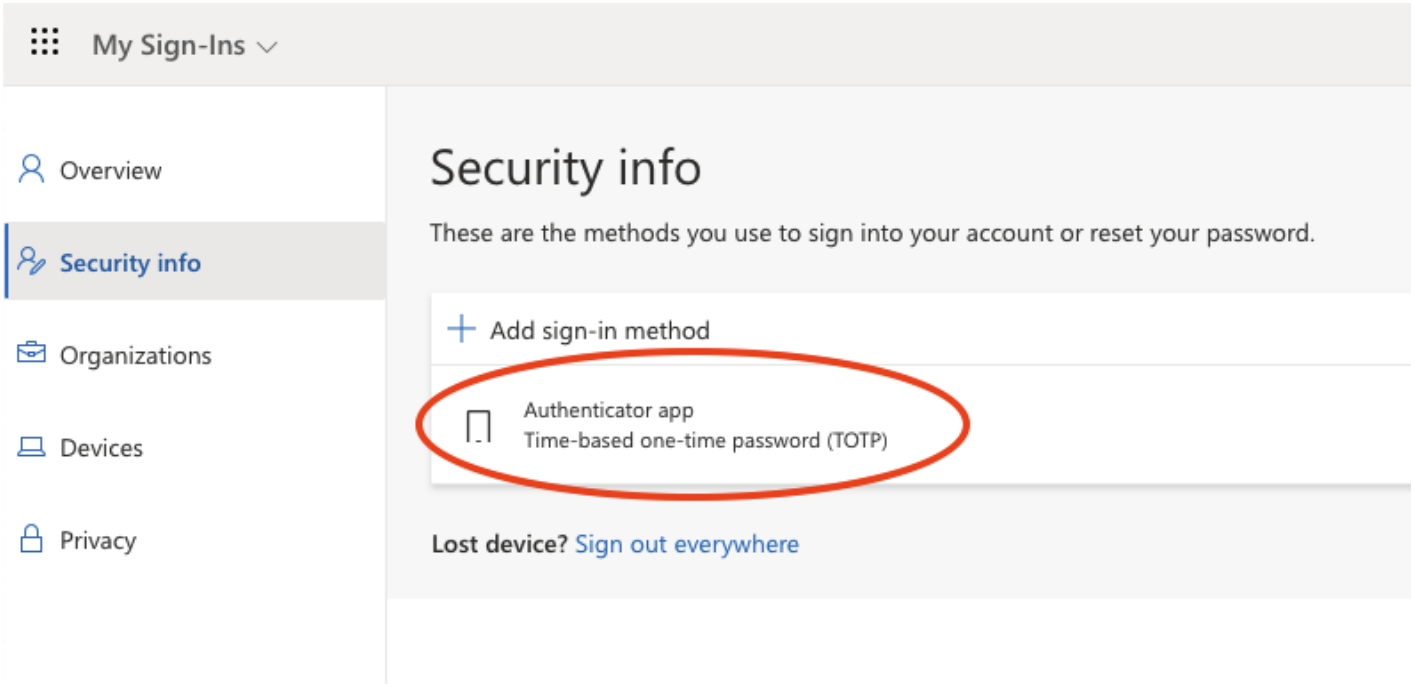


Back over on `office.com` click Next to continue. You will then be prompted to enter one of the six digit codes that is now being displayed in the 1Password app under the heading "one-time password." This is to verify that the TOTP was added correctly and is generating the codes as expected.

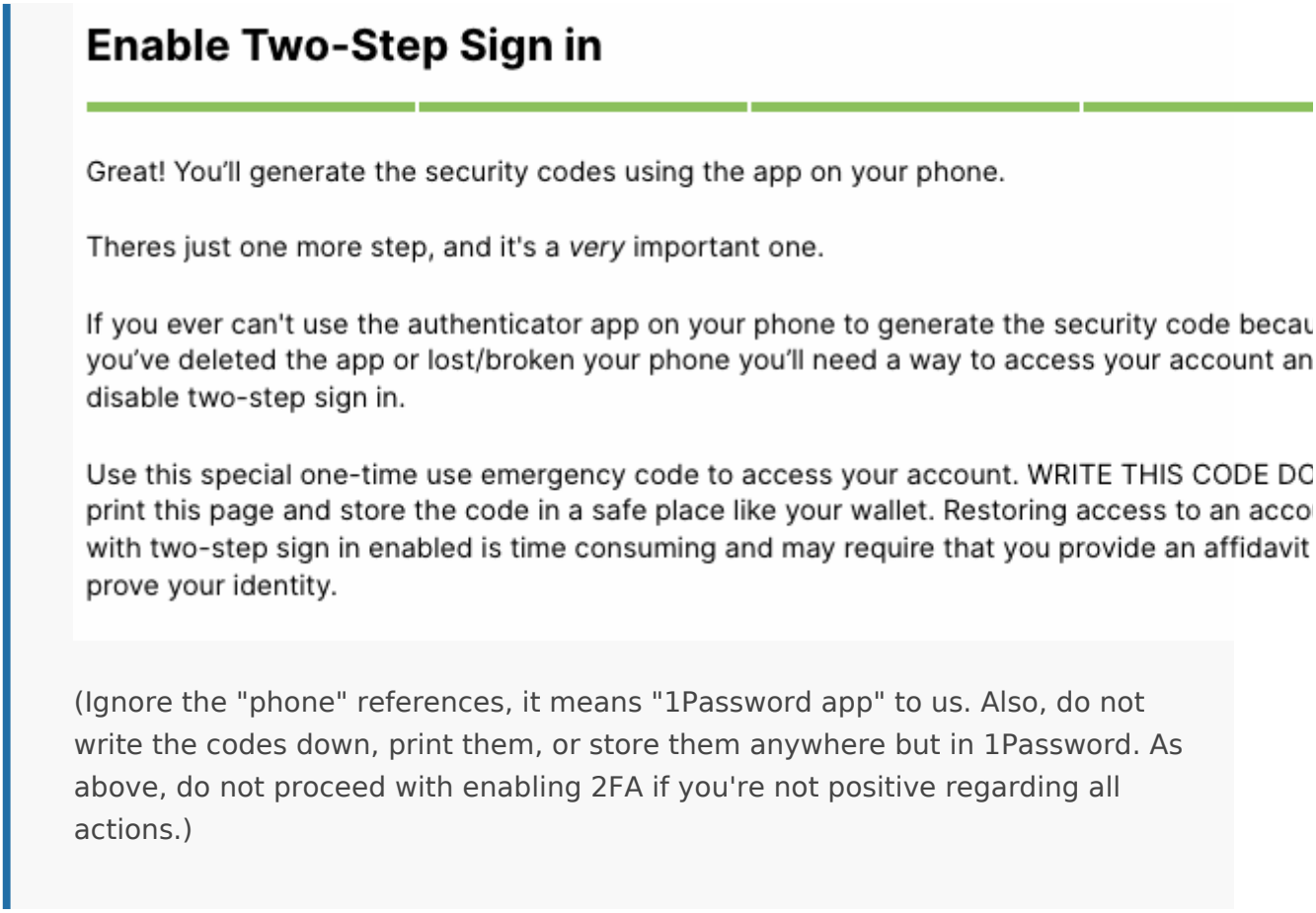
Enter the code displayed, and hit Next. If it fails, the code entered may have just expired, so try, try again. If it fails multiple times, contact IT for help.



If successful, you should now have a TOTP added to the account.



> ## Important note!
>
> **Often you will be offered either a recovery code, or a bunch of one-time codes. This is a rescue system and it is critical that all codes provided are saved to the 1Password item!**
>
> If it's a single code save it in a new field in the same item as a password (not the main password!) Should look like this:
>
> 2FA enabled, emergency recovery code
> •••••••••• Copy
>
> Again, do not replace any password with this one. Do not replace the main account password, do not replace the "one-time password" which is the primary 2FA system. This is a third, completely separate entity.
>
> If you are not confident you're doing this correctly, **do not proceed!** This warning is from Hover but succinctly covers why:



(Ignore the "phone" references, it means "1Password app" to us. Also, do not write the codes down, print them, or store them anywhere but in 1Password. As above, do not proceed with enabling 2FA if you're not positive regarding all actions.)

Done!

You've successfully added a TOTP to an item in 1Password!

Revision #24
Created 2022-07-11 22:08:40 UTC by Mark Reinmuth
Updated 2025-02-14 22:00:09 UTC by Corban Monger