General security options

Last Modified on 08/07/2024 3:04 pm EDT

Use these options to add further security protections to your knowledge base. Your Chief Security Officer will be happy.

Basic authentication

Sometimes, to facilitate integration with a third-party tool, it's useful to have an account that uses basic authentication (basic auth). Basic auth uses an email address and a password, similar to readers set up directly in KnowledgeOwl.

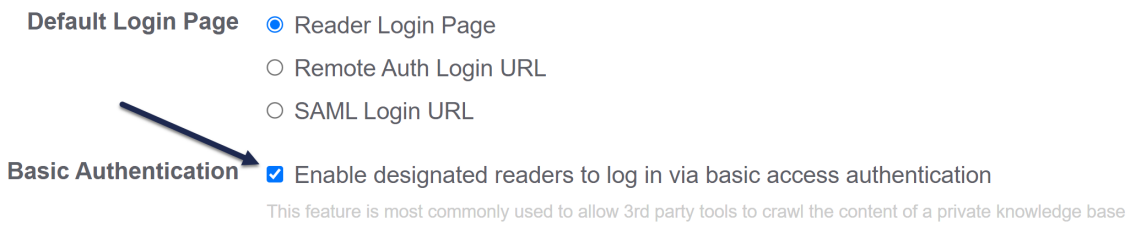
Basic auth can be useful if you have your knowledge base access restricted in one format, but you'd like to give a third-party tool its own reader account to authenticate with. We see this most often used for tools that crawl your knowledge base for various purposes (such as Amazon's Kendra or other chatbot/search integrators).

To enable Basic Authentication in KnowledgeOwl, you need to enable the overall setting in **Settings > Security** and then configure an individual reader account to use basic auth. See below for more detailed instructions.

Setup

First, enable basic authentication for your knowledge base overall:

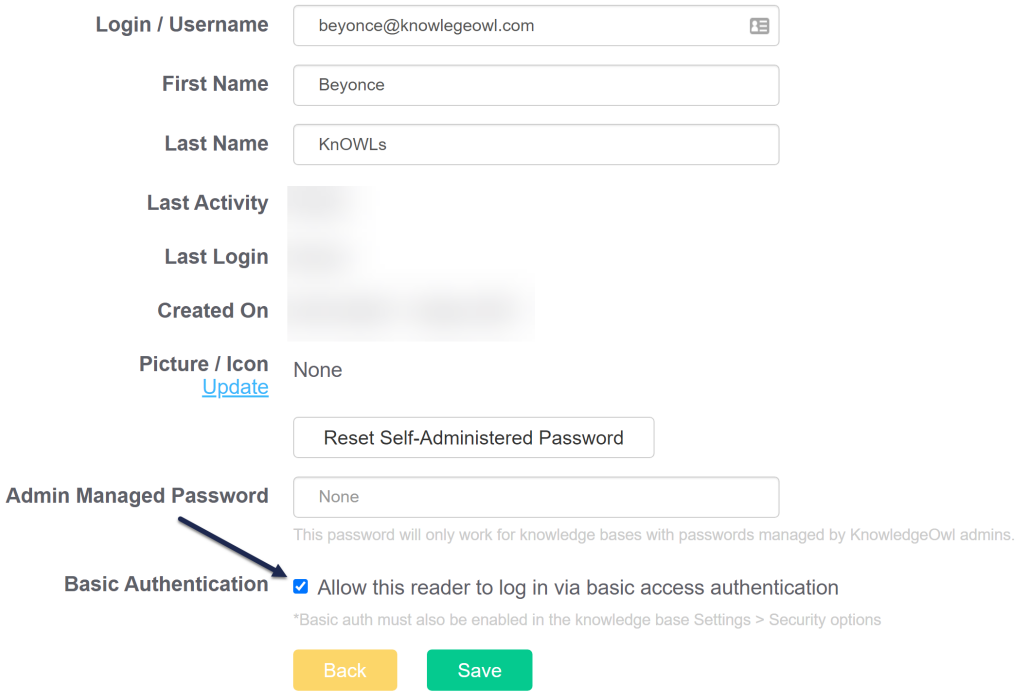
1. Go to **Settings > Security**.
2. In the **Security Settings** section, look for the **Basic Authentication** subsection.
3. Check the box next to "Enable designated readers to log in via basic access authentication."



4. **Save** your changes.

Once you have basic auth enabled for the knowledge base as a whole, you need to designate a reader account to use that authentication:

1. Click on your profile icon/name in the upper right.
2. Select Readers from the dropdown to access the Readers area of your account.
3. You can choose to create a new reader account to use with basic auth, or edit an existing reader account to use as your basic auth account. Be sure the reader account is set up with a self-administered password.
4. Once you're in the **Edit Reader** screen viewing the details of the reader account you'd like to use, look for the **Basic Authentication** subsection.
5. Check the box next to "Allow this reader to log in via basic access authentication".



6. Be sure the reader has access to the appropriate knowledge base(s) and/or reader group(s).
7. **Save** your changes.

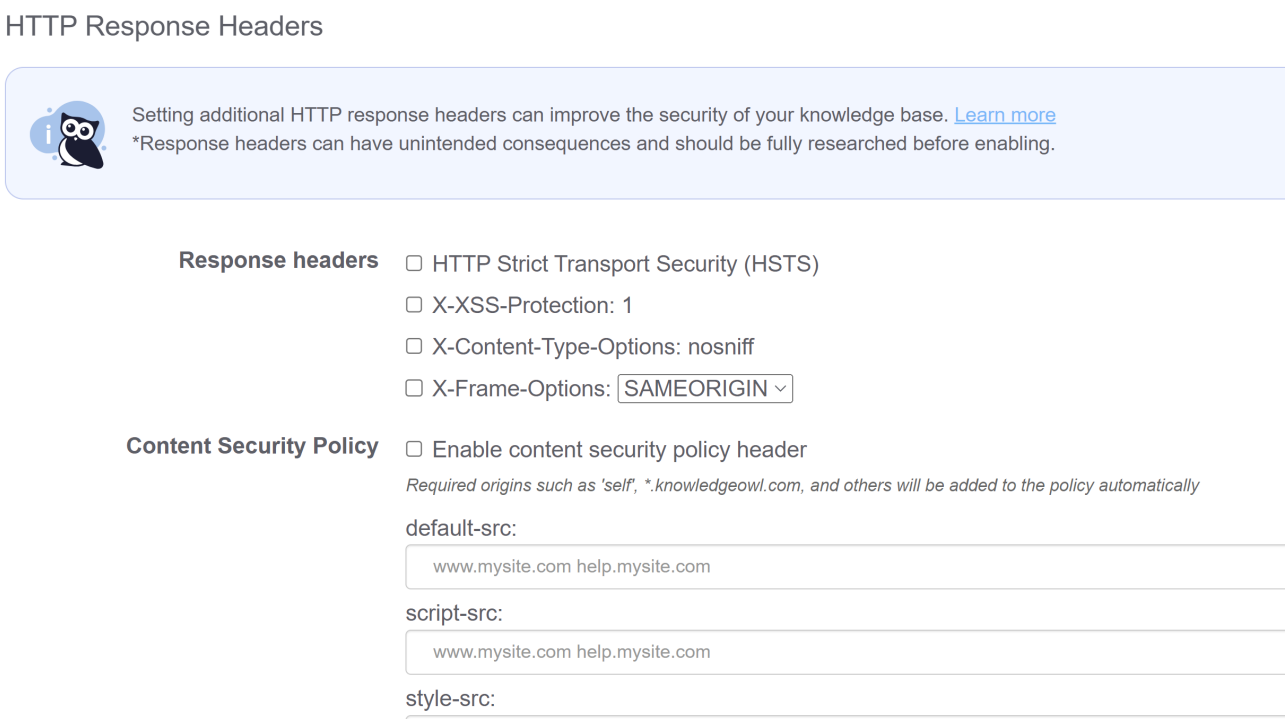
You can now use the email address and password you set this account up with as the username/password for your third-party tool.

HTTP response headers

In order to improve the security of your knowledge base, you can enable some additional HTTP response headers and/or Content Security Policies that will be returned by our servers that give additional instructions to the reader's browser. The effects of these headers vary and should only be enabled by someone with a clear understanding of what they do.

You can find these settings under **Settings > Security** in the **HTTP Response Headers** section. Below we'll provide a

brief high level description of what each of these headers is used for.



HTTP Response Header Options

HTTP Strict Transport Security (HSTS)

The HTTP Strict Transport Security header informs the browser that it should never load a site using HTTP and should automatically convert all attempts to access the site using HTTP to HTTPS requests instead. See MDN Web Docs for more info.

X-XSS-Protection: 1

Enables XSS filtering (usually the default in browsers). If a cross-site scripting attack is detected, the browser will sanitize the page (remove the unsafe parts). See MDN Web Docs for more info.

X-Content-Type-Options: nosniff

The X-Content-Type-Options response HTTP header is a marker used by the server to indicate that the MIME types

advertised in the Content-Type headers should not be changed and be followed. This is a way to opt out of MIME type sniffing, or, in other words, to say that the MIME types are deliberately configured. See MDN Web Docs for more info.

X-Frame-Options: SAMEORIGIN/DENY

The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in an `<iframe>`, `<embed>`, or `<object>`. Sites can use this to avoid click-jacking attacks, by ensuring that their content is not embedded into other sites. See MDN Web Docs for more info.

Content Security Policy

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement to distribution of malware. If you're pulling external resources, the CSP determines which domains are allowed to do certain things, such as run scripts, load resources, and so on. See MDN Web Docs for more info.

You can include domains with and without the scheme (such as https:, wss:, etc.), 'unsafe-eval', and domains with wildcards (*) for any of the CSP directives here.

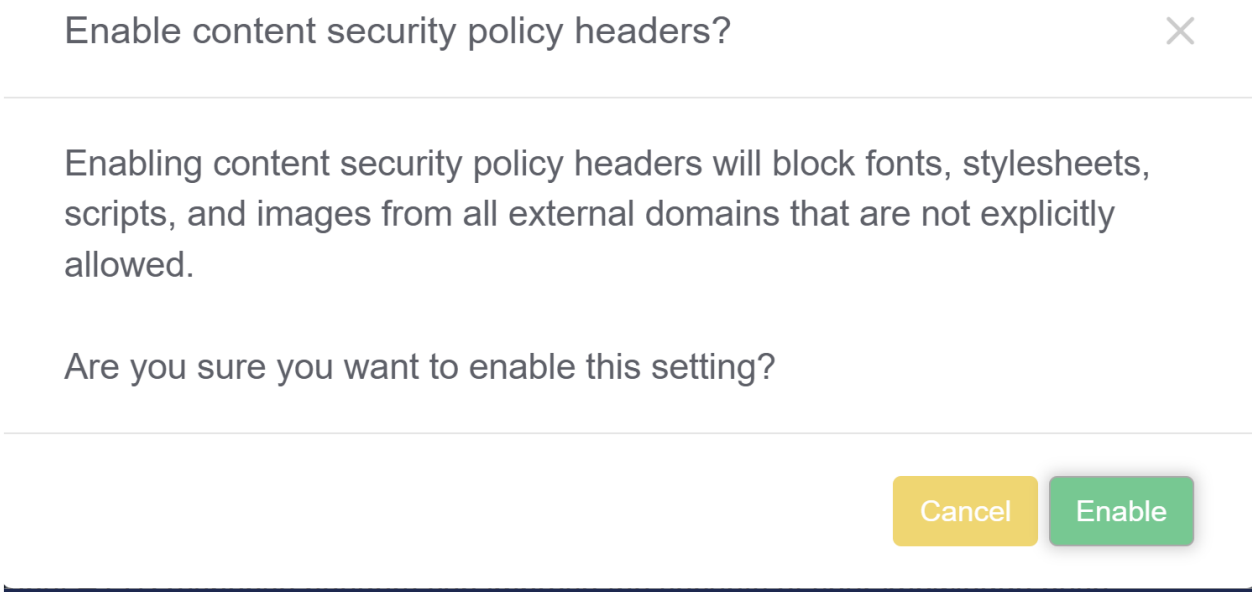
Proceed with caution
These options have the most potential to cause unintended consequences for your knowledge base. If you are referencing any remote JavaScript, fonts, images/files, or CSS, make sure you add the remote domains into this policy or they'll be blocked.

Defining the script-src or default-src policy may also prevent the KnowledgeOwl Modern Widget from rendering (Widget 2.0 will not be affected).

To enable CSP:

We recommend that you:

1. First, review the individual directives below and add appropriate domains/settings to individual CSP directives (such as default-src, script-src, etc.). See the sections below for more information on each directive.
2. Once you've reviewed all directives and added appropriate domains and resources, check the box to **Enable content security policy header**.
3. You'll be presented with a pop-up confirming you're ready to enable. You must select **Enable** to proceed.



Enable content security policy headers confirmation pop-up

4. CSP won't be fully enabled until you **Save**.
5. We recommend navigating around your knowledge base after you've enabled the CSP to ensure that you didn't overlook any necessary domains or resources. Console errors can help you identify things you missed!

default-src

This CSP directive is used as a fallback when any of the others don't have values specified. See MDN Web Docs for more information. We include default-src ***.knowledgeowl.com**, **'self'**, **'unsafe-inline'**, and a few other required resources here even if you don't specify them.

script-src

This CSP directive specifies valid sources for JavaScript. This includes not only URLs loaded directly into `<script>` elements, but also things like inline script event handlers like `onclick`. See MDN Web Docs for more information.

Fancy Box
If you're using the Fancy Box plugin for images, include **cdnjs.cloudflare.com** in this list. You may also want to add **'unsafe-eval'** to this list. See MDN Web Docs for more information.

Before you enable, consider checking these places for domains you may be running scripts from and be sure you've added them to this directive:

- **Settings > Style > Custom HTML:** Body, Article, etc.
- **Settings > Style > Custom Head**
- **Library > Snippets:** Check individual snippets that may be running scripts themselves

style-src

This CSP directive specifies valid sources for stylesheets. See MDN Web Docs for more information.

Fancy Box
If you're using the Fancy Box plugin for images, include **cdnjs.cloudflare.com** in this list.

Before you enable, consider checking these places for references to external stylesheets and be sure you've added them to this directive:

- **Settings > Style > Custom HTML:** Body, Article, etc.
- **Settings > Style > Custom Head**
- **Library > Snippets:** Check individual snippets that may be running scripts themselves

font-src

This CSP directive specifies valid sources for fonts loaded using `@font-face`. See MDN Web Docs for more information.

Before you enable, consider checking these places for references to font-face rules containing external resources:

- **Settings > Style > Fonts:** If a Custom Font is selected, be sure to add that domain.
- **Settings > Style > Custom CSS:** Look for any font-face rules.
- **Settings > Style > Custom Head:** Look for any font references outside of KnowledgeOwl.
- **Library > Snippets:** Check individual snippets to be sure they don't have font-face rules.

img-src

This CSP directive specifies valid sources for images and favicons. See MDN Web Docs for more information.

Before you enable, consider checking these places for references to images outside of KnowledgeOwl:

- **Settings > Style > Custom Head**
- **Settings > Style > Custom CSS**
- **Settings > Style > Custom HTML:** Body, Homepage, Article, etc.
- Individual snippets or articles containing external images in the body or in article thumbnails/banners

Requiring login to view files/images

The Security Settings for your knowledge base (**Settings > Security** and **Settings > SSO**) determine the general security requirements for readers to access your knowledge base.

The files you upload to your knowledge base--PDFs, Excel sheets, screenshots, Word documents, etc.--do not automatically use this same security.

By default, even if your knowledge base requires login, the files you've uploaded do *not* require login. This is by design so that you can give readers the link to specific documents and they can easily download the file by clicking on that link or URL without having to log in to your knowledge base.

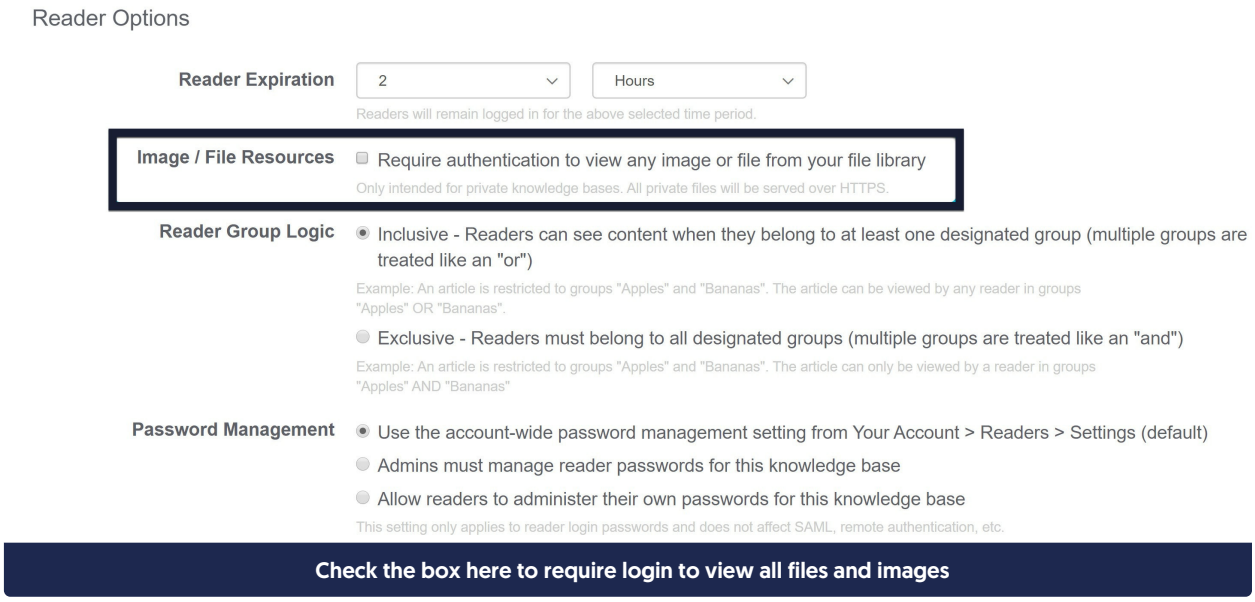
However, you can adjust your security settings so that readers have to be logged in to access files and images stored within your knowledge base. If you make this change, the URLs you'll see will change slightly to a different Cloudfront URL.

With authentication required, if you share a hyperlink directly to a file stored in KnowledgeOwl, anyone accessing that link will be prompted to log in to the knowledge base using the default authentication method before they'll be able to view the file.

Dashboard display
When this secure file library setting is enabled, you'll no longer see your knowledge base's logo when you're viewing the KnowledgeOwl app dashboard (/app/switch-project). They'll only be hidden from the app switch-project view; they're still shown everywhere else!

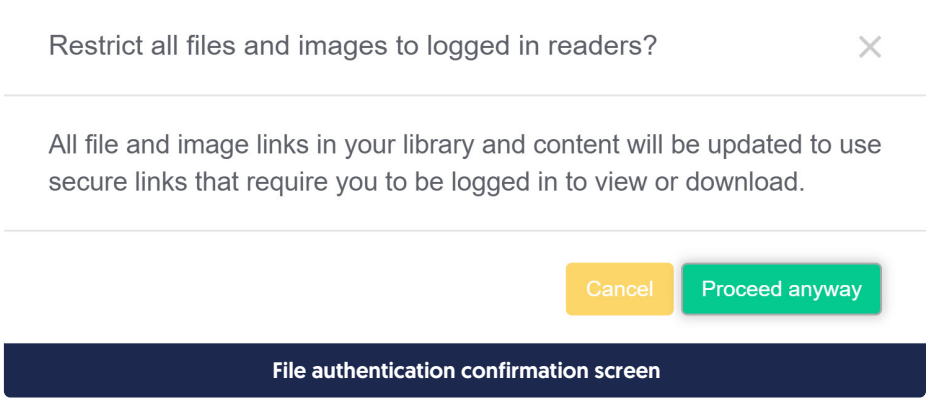
If you would like to require that someone must log into your knowledge base before accessing files:

1. Go to **Settings > Security**.
2. In the **Reader Options** section, check the **Image / File Resources** box next to "Require authentication to view any image or file from your file library."



Check the box here to require login to view all files and images

3. You'll receive a warning asking you if you're sure you want to do this. Click **Cancel** to keep files unauthenticated; click **Proceed anyway** to continue with requiring login to view files.



File authentication confirmation screen

4. Be sure to **Save** your changes.

Once these changes are saved, we will programmatically update the URL for most of your files referenced in your knowledge base's theme, within articles, and within article thumbnails/banners and category icons. If you are using URL redirect categories or articles pointing to files stored in KnowledgeOwl, you may need to manually update those URLs.

Some customers who require file authentication have noticed some issues with their upper left logos not loading properly on initial page load. If you make this change and notice issues with your logo, please contact us and let us know you're having issues. We can move your logo file to unrestricted storage so it will load faster.

Spam protection

If any part of your knowledge base is publicly available, you're probably interested in preventing spam from reaching you!

There are three main areas you can get spam from:

- **The Contact Form: either the full contact form in the live knowledge base, or the Contact tab of Contextual Help Widget (2.0).**
- **If Comments are enabled and you are not checking the box to "Only allow logged in readers and authors to leave comments". See Comment restrictions and permissions.**
- **Bogus subscription sign-ups: only generally possible if public subscriptions have been enabled**

KnowledgeOwl provides two ways for you to prevent spam:

ReCAPTCHA

See Add reCAPTCHA for more information on setting up reCAPTCHA.

Pros

- **Free service provided by Google**

- Once set up, you don't have to think about it
- Most readers are familiar with reCAPTCHA processes, since they're used so many places
- Generally very effective at blocking spambot traffic

Cons

- Requires you to set up one or more site keys and secrets with Google and get them configured
- People can get caught in a reCAPTCHA loop, depending on the type of reCAPTCHA you've selected--this is why we recommend using the checkbox version rather than the "select all pictures of xx" version
- reCAPTCHA is a Google-supported tool, and particularly if you have GDPR requirements or concerns, reCAPTCHA might not be a viable option

Honeypot

Honeypots are an alternative way to handle spam protection. Honeypots are used in a variety of ways, but the basic gist is that they create something that is enticing and somewhat irresistible to bad actors.

For things like contact forms, this means that instead of making all readers complete an action or test before they can submit a form, a honeypot might include some hidden form fields that no human will see. Spambots do see them and generally fill them out. Submissions with these fields completed are ignored.

Honeypots might also include time or repeat submission restrictions, where they'll flag repeated submissions from the same reader within xx seconds of each other, or flag submissions that took fewer than xx seconds to fill out.

Our built-in honeypot function works similarly to these options (though for security reasons, we can't tell you the full details!).

Pros

- Simple setup: check a box, Save, and you're done; no registration or site keys to configure
- Better end-user experience for your average human reader (no tests/tasks to complete)

Cons

- If someone seriously wants to attack and spam you, they can figure a honeypot out and bypass it, so they aren't as effective as reCAPTCHA when it comes to dedicated malicious attackers