OASIS

ChatGPT & other web apps may have full read access to your **entire OneDrive**

Elad Luz
Research Lead

Published on
May 28, 2025

A Quick Summary

Websites that feature OneDrive file uploads such as ChatGPT, Slack, Trello, ClickUp, Zoom and more, receive access to your entire OneDrive content—rather than just the specific files selected for upload—and may maintain this access for extended periods. This issue arises from the lack of fine-grained OAuth scope for OneDrive, which causes the official OneDrive File Picker implementation to request read access to the entire drive - even when uploading just a single file. Furthermore, sensitive secrets used for this access are often stored insecurely by default.

Oasis reached out to Microsoft which took note of the report and may consider improvements in the future.

Oasis estimates hundreds of apps are affected, and reached out to popular app vendors that make use of the OneDrive File Picker, briefing about the issue before public disclosure.

Detail Explanation

OAuth. Background

OAuth is an industry-standard protocol for sharing a user's data from one website with a web application hosted on a different site.

For example, in the case of OneDrive, when a user wants to share a file from their OneDrive with a web application (e.g. for file upload), OAuth allows the user to grant such permission to the application. After the user consents, an access token is issued to the web application, granting it limited access to the user's data. OAuth is widely adopted and supported by numerous services across various platforms.

For detailed information about how OAuth works read our OAuth 2.0 with Microsoft: Start here article.

Primary issue: Excessive Permissions in the OneDrive File Picker

OAuth - Scope

A scope defines one or more permissions that an application is requesting from the user. Each service provider, such as Microsoft or Google, offers a predefined set of scopes for their APIs—like Microsoft Graph—which cannot be modified. Ideally, these scopes should be fine-grained and tailored for specific use cases to ensure that an app only receives the permissions it truly needs.

When a user consents to OAuth access, they are presented with a list of requested scopes, which represent the permissions the application is asking for. The user cannot grant or deny individual permissions—they must either consent to all requested scopes or deny access entirely.

The OneDrive File Picker - User Consent

The OneDrive File Picker is a Microsoft-maintained component that allows a web application to interact with a user's OneDrive by displaying a dialog for browsing and selecting files. To use the picker, a developer registers an Azure application with the appropriate permissions. The first time a user interacts with the picker—whether to upload or download a file—they are prompted to give consent to the scopes requested by that Azure app.

The Issue with File Picker Scopes

In general, all versions of the OneDrive File Picker request permissions that allow them to:

- **Read your entire OneDrive drive** in "file upload" scenarios.
- **Write to your entire OneDrive drive** in "file download" scenarios.

Moreover, **version 7.0** of the file picker in specific requests **both read and write permissions for the upload process.**

These permissions are valid for at least an hour (more on this later).

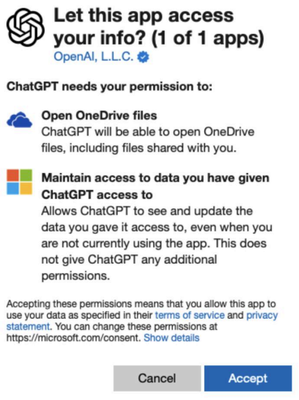
In simple terms, any web application that uses the OneDrive File Picker has access not just to the file you select to upload/download, but to your entire OneDrive. Even worse, this access might persist after the file upload is complete.

This clearly represents a case of **excessive permissions.**

Users supplied their consent, aren't they aware of this?

You could argue that since the user consents, there is no security issue.

Imagine you have just clicked on an "upload" button and are presented with the following consent-



Let this app access your info? (1 of 1 apps)
OpenAI, L.L.C.

ChatGPT needs your permission to:

Open OneDrive files
ChatGPT will be able to open OneDrive files, including files shared with you.

Maintain access to data you have given ChatGPT access to
Allows ChatGPT to see and update the data you gave it access to, even when you are not currently using the app. This does not give ChatGPT any additional permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://microsoft.com/consent. Show details

The phrasing "Will be able to open OneDrive files" may appear to suggest that only selected files will be accessible (remember - users are in a scenario where they are about to upload files), but in reality, the scope grants access to all of the user's OneDrive content. Similarly, unless you are a seasoned developer familiar with the concept of "offline access", the permission to "maintain access" could be vague as well. It's worth considering whether this wording clearly communicates the level of access being granted.

The lack of fine grained scopes for the file picker use case makes it impossible for users to distinguish between malicious apps that target all your files and legit apps that ask for excessive permissions just because there is no other secure option.

Application developers were unnecessarily aware of this

In File Picker versions prior to **8.0**, the entire authentication process was encapsulated within the File Picker itself, which meant developers themselves were less likely to be aware of the excessive permissions their applications were receiving.

Vendors developing Web apps are also at risk, as security incidents could result in severe consequences, leaking a lot of files from a lot of their users. The granted access itself may even violate compliance regulations.

Should all OAuth scopes be fine-grained ?

While not every application can offer perfectly granular permissions, this case presents a unique opportunity. Since Microsoft manages both the scope definitions and the File Picker implementation, there's potential for a more precise alignment between what the picker does and the access it requires. The File Picker is a key feature in a file hosting service, and with Microsoft's level of control, we're hopeful that future iterations will offer more tailored scopes.

Later in this article, we look at how other popular file hosting services handle similar scenarios with much narrowly scoped permissions.

Secondary issue: Insecure Storage of Sensitive Secrets

OAuth involves the exchange of sensitive secrets that should be securely stored. This is particularly important for preventing unauthorized access to user data.

After the user provides consent or logs in, **the web application receives an Access Token**. This token allows the app to make API requests on behalf of the user to access their resources.

The Access Token is typically **valid for one hour** and cannot be revoked.

Older File Picker versions (6.0-7.2)

Those versions encapsulated the authentication part using an "Implicit Flow", exposing **sensitive tokens in URL fragments** in an insecure manner. Furthermore, File Picker version 7.0 also **writes sensitive tokens to the browser's localStorage in plain text**, adding to the exposure.

Newest File Picker version (8.0)

This version requires developers to take care of the authentication themselves, which would usually be done using the **MSAL** (Microsoft Authentication Library) as recommended in the official documentation, and most likely using the "Authorization Flow".

Developers should be aware that MSAL **stores sensitive Tokens in the browser's session storage** in plain text.

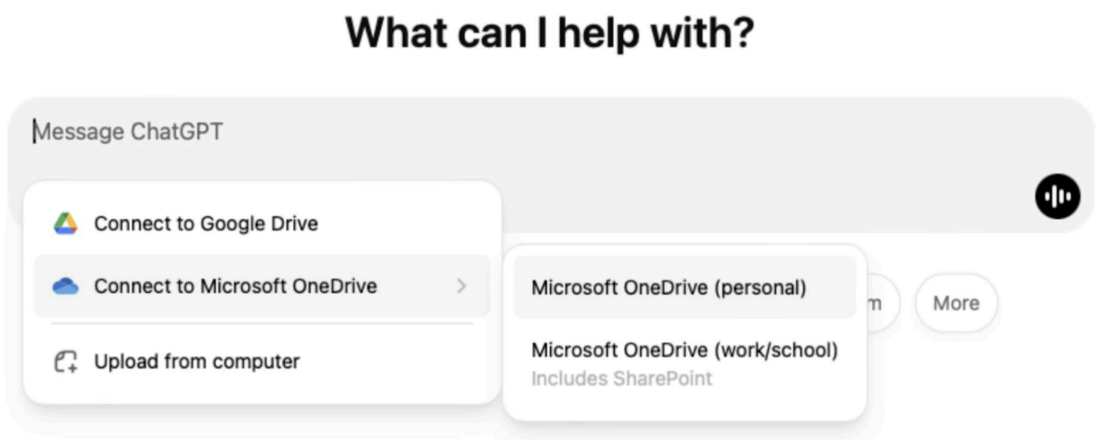
On Authorization flows a Refresh Token may also be issued which essentially lengthens the access period (e.g. this is the case for ChatGPT). Refresh Tokens should only be requested when necessary, as they provide ongoing access to the user's data. Refresh Tokens can be revoked.

You can read more about the different types of flows and tokens in our OAuth: Start here article.

Affected Vendors. Several Examples

OpenAI

On the ChatGPT website, for logged-in users, uploading using OneDrive is available by clicking on the clip icon left to the prompt box.



Given the astounding number of users (reported 400+ millions per month) it is likely that millions have already granted access to their OneDrive.

ChatGPT website makes use of Version 8.0 of the Microsoft OneDrive File Picker.

Collaboration Tools

Several popular collaboration tools make use of the OneDrive File Picker to enable file sharing and attachment features. Examples include:

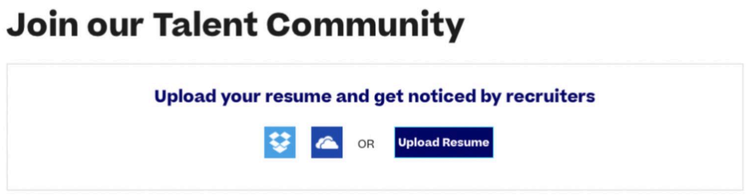
- Slack allows users to share OneDrive files directly within channels and messages. Slack OneDrive Integration
- ClickUp enables users to attach OneDrive files to tasks and documents. ClickUp OneDrive Integration
- Trello supports attaching OneDrive files to cards via a dedicated Power-Up. Trello OneDrive Integration

Phenome

Phenome's talent acquisition platform is integrated with various major companies' websites, helping with job advertising, candidate applications, and the hiring process.

Consider a scenario where an employee from Organization A discreetly applies for a position at Organization B, uploading their résumé directly from their corporate OneDrive. In doing so, Organization B may inadvertently gain access to the employee's entire OneDrive account, including potentially sensitive or confidential files belonging to Organization A - possibly a direct competitor

The following is a cropped screenshot of what the Phenome CV upload buttons look like-


Join our Talent Community
Upload your resume and get noticed by recruiters
OR
Upload Resume

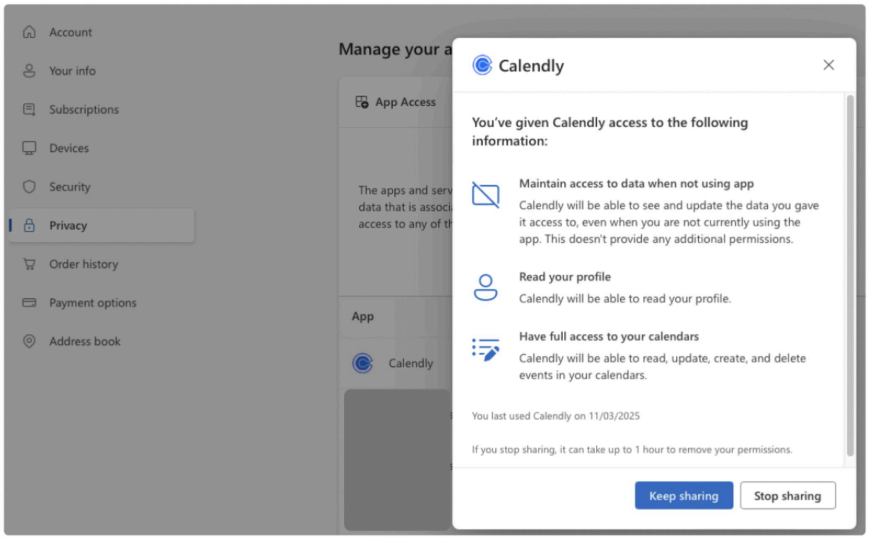
Mitigation

Checking Whether You've Previously Granted Access to a Vendor

If you're concerned about the potential risks raised by these issues, it might be a good time to review the third-party access you've granted to your account.

For Private Accounts

1. Log in to your **Microsoft Account.**
2. In the left or top pane, click on **"Privacy".**
3. Under **"App Access"**, select the list of apps that have access to your account.
4. Review the list of apps, and for each app, click on **"Details"** to view the specific **scopes** and **permissions** granted.
5. You can "Stop Sharing" at any time, consider an Access Token takes about an hour to expire regardless of when you clicked stopped sharing. This would however revoke a Refresh Token if present.



Calendly

You've given Calendly access to the following information:

Maintain access to data when not using app
Calendly will be able to see and update the data you gave it access to, even when you are not currently using the app. This doesn't provide any additional permissions.

Read your profile
Calendly will be able to read your profile.

Have full access to your calendars
Calendly will be able to read, update, create, and delete events in your calendars.

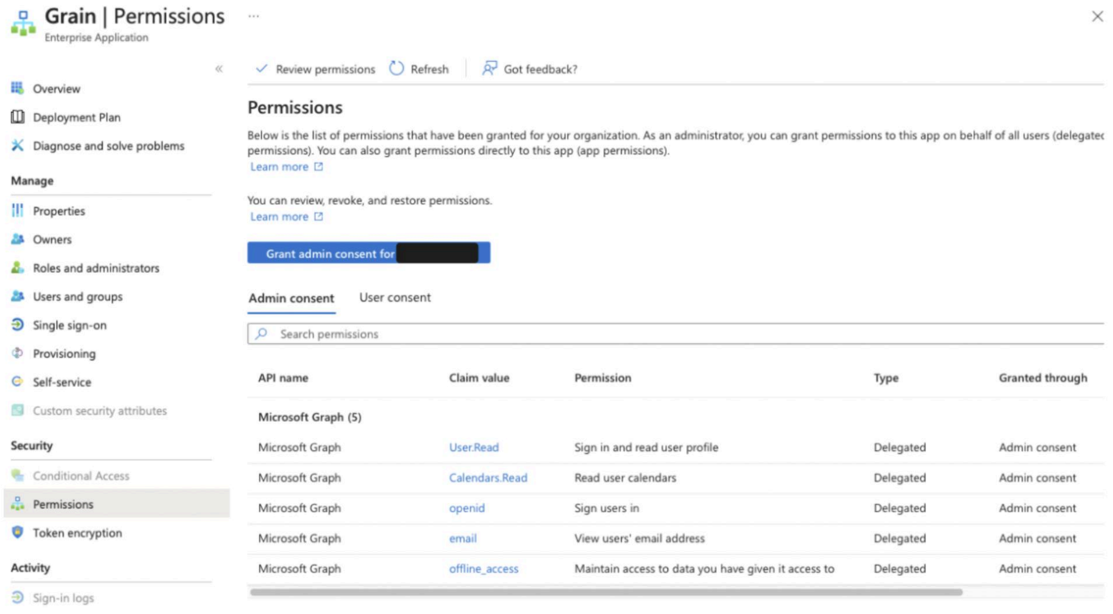
You last used Calendly on 11/03/2025

If you stop sharing, it can take up to 1 hour to remove your permissions.

App Access Details Screen

For Organitzations

1. In the **Entra Admin Cente**r, navigate to the list of enterprise applications.
2. The list will display an **Application ID** column (Client ID) and an **Object ID** column (also known as the **Service Principal Object ID**).
3. Currently, there is no way to filter for **Delegated Applications** directly from this screen.
4. To check the permissions granted to each app, click on an application, then click on the **"Permissions"** button in the left pane. This will list all granted **scopes** and you can verify whether they are **delegated**.
5. You can also view the **user** who granted the permissions.



Grain | Permissions
Enterprise Application

Permissions

Below is the list of permissions that have been granted for your organization. As an administrator, you can grant permissions to this app on behalf of all users (delegatec permissions). You can also grant permissions directly to this app (app permissions).
Learn more

You can review, revoke, and restore permissions.
Learn more

Grant admin consent for [redacted]

API name	Claim value	Permission	Type	Granted through
Microsoft Graph (5)				
Microsoft Graph	User.Read	Sign in and read user profile	Delegated	Admin consent
Microsoft Graph	Calendars.Read	Read user calendars	Delegated	Admin consent
Microsoft Graph	openid	Sign users in	Delegated	Admin consent
Microsoft Graph	email	View users' email address	Delegated	Admin consent
Microsoft Graph	offline_access	Maintain access to data you have given it access to	Delegated	Admin consent

Screenshot of the permissions of a dummy app taken from a dummy tenant

Using Azure CLI

Examining each application individually in the Entra Admin Center can be tedious. Alternatively, you may prefer to use the Azure CLI to review access by listing the Service Principals in the tenant (see here for full reference)-

```
az ad sp list
```

One can then list the delegated permissions given service principal (see here for full reference)-

```
mgc service-principals oauth2-permission-grants list --service-principal-id {servicePrincipal-id}
```

Checking Whether a Site Is Using the OneDrive File Picker

To check if a site is using the OneDrive File Picker:

1. **Attempt to upload or download a file to OneDriv**e through the site's interface.
2. When the consent prompt appears, check whether OneDrive permissions are being requested.

To determine the **exact version** of the OneDrive File Picker in use, you can inspect the page source or look for the relevant URLs being fetched using the browser's developer tools.

Expect to find these urls per version-

- **Version 8.0:**
  - https://onedrive.live.com/picker/_layouts/15/FilePicker.aspx
  - https://*.sharepoint.com/_layouts/15/FilePicker.aspx
  - https://*.sharepoint.com/*/_layouts/15/FilePicker.aspx
- **Version 7.2:**
  - https://js.live.net/v7.2/OneDrive.js
- **Version 7.0:**
  - https://js.live.net/v7.0/OneDrive.js
- **Version 6.0:**
  - https://js.live.net/v6.0/OneDrive.js

Notice some websites have copied those sources locally and possibly altered them, in such cases one would need to find other means to locate the File Picker use.

Mitigation for web apps

If possible, temporarily remove the option to upload files using OneDrive through OAuth until Microsoft provides a secure alternative. In the meantime, consider exploring safer, simpler workarounds, such as supporting shared "view-only" file links from OneDrive, understanding that this flow may be less convenient for users.

If removing the File Picker is not a feasible solution for you, we recommend the following:

- **Avoid using Refresh Tokens.**
  Do not request the "offline access" scope.
  Remove any code or storage logic related to keeping and using refresh tokens.
  If any refresh tokens are currently stored, dispose of them in a secure manner.
- **Store your Access Tokens in a secure manner** and dispose of them when no longer needed.
  Review the code that handles access tokens to ensure they are not exposed to third parties (e.g., by being stored in session or local storage).

File Picker on Other Leading File Hosting Services

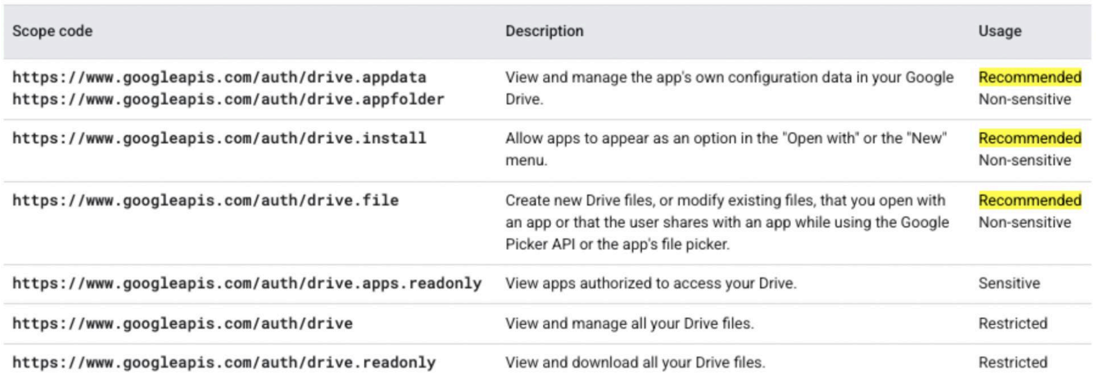
Google Drive

Google Drive offers a fine-grained scope for web applications, allowing them access only to files that the app has created or those that have been explicitly shared with it.

- **Scope**: https://www.googleapis.com/auth/drive.file
  - **Permissions**:
    - The app can create new files in the user's drive.
    - The app can **list, read, and modify only files** that it has previously created or interacted with.
    - "Interaction" with a file created by something other than the app can only occur if the user has explicitly selected it (e.g. using a file picker).

This approach ensures that web apps have **limited access** to the user's files, providing a safer, more controlled experience.

Listing other available scopes, one will notice the existence of https://www.googleapis.com/auth/drive.readonly Which appears to be similar to the one required by the OneDrive file picker-



Scope code	Description	Usage
`https://www.googleapis.com/auth/drive.appdata` `https://www.googleapis.com/auth/drive.appfolder`	View and manage the app's own configuration data in your Google Drive.	Recommended Non-sensitive
`https://www.googleapis.com/auth/drive.install`	Allow apps to appear as an option in the "Open with" or the "New" menu.	Recommended Non-sensitive
`https://www.googleapis.com/auth/drive.file`	Create new Drive files, or modify existing files, that you open with an app or that the user shares with an app while using the Google Picker API or the app's file picker.	Recommended Non-sensitive
`https://www.googleapis.com/auth/drive.apps.readonly`	View apps authorized to access your Drive.	Sensitive
`https://www.googleapis.com/auth/drive`	View and manage all your Drive files.	Restricted
`https://www.googleapis.com/auth/drive.readonly`	View and download all your Drive files.	Restricted

Notice the usage column of “drive.readonly” states “Restricted”, which means apps requiring such access have to go through verification and a security assessment by Google, more about this here.

Dropbox

Dropbox offers a file picker solution through its **Chooser SDK,** which does not rely on a typical OAuth flow. Instead, it uses a proprietary endpoint where it passes the app key to retrieve the selected files.

Key Points:

- This is not an OAuth flow, meaning the app does not request traditional OAuth scopes from the user.
- There is **no user consent** for granting specific permissions like in traditional OAuth; the app simply gets a file (or a list of files) that the user selects via the picker.
- The app key is used for identification, and files are returned based on the user’s selection.

This model minimizes the need for broad permissions and avoids unnecessary exposure of the user's data, offering a more streamlined and secure file access approach.

These examples highlight that other leading file hosting services have made deliberate choices to **limit** the scope of access granted to web apps, providing users with more control over their data.

End Note

A scope that is not fine-grained enough, combined with a vague user prompt, creates a dangerous combination for both personal users and organizations.

Oasis Security has provided its customers with detailed insights regarding service principals owned by third parties with OneDrive access based on user consent.

About Oasis Security

Oasis Security is the management and security solution for non-human identities. Purpose-built to address the unique challenges of visibility, security, & governance of NHIs across hybrid cloud environments.

With advanced AI-driven analytics, Oasis automatically discovers NHIs, assesses their risks, and identifies ownership across the entire environment. Its integrated, policy-based governance capabilities ensure seamless orchestration of the NHI lifecycle, including remediation and compliance management—all within a single, unified solution.

Leading organizations across a wide range of industries use Oasis to foster innovation and collaboration among security, identity, and engineering teams, enabling secure digital transformation and cloud adoption.

The Oasis Research team

Our dedicated research team is committed to enhancing security in the field of identity. We take pride in our responsible and professional collaboration with vendors to address vulnerabilities and strengthen overall security.

The authors

Elad Luz, Research Lead at Oasis Security, LinkedIn

Elad Luz has over 20 years of research experience in fields such as software vulnerabilities, reverse engineering, network protocol analysis, threat detection, and ML. Prior to Oasis, he served as a CDR Research Lead at Wiz and as the Head of Research at CyberMDX. He has publicly disclosed over 20 vulnerabilities, demonstrating a strong commitment to enhancing security across multiple platforms.

Overview