Members: Brandon Peh (bcp39), Louise Lee (zl245), Ruixin Ng (rn279), Wang Zilong (zw243)

Threat Model

Threats include:

- Hackers who wish to gain access to files that they are not supposed to see or know the existence of and use these files for malicious purposes.
- Curious people who know there may be other files stored on the server and wish to access them or their file names (so they can guess the content)
- Unhappy viewers who wish to deny availability or change the files that they are able to see.
- Mischievous people who want to delete all the files and cause trouble for fun.

We will assume that no one stores anything of too much financial or personal importance. The sensitivity of the files that we will expect to be stored is at the level of sharing cute cat videos or sharing code between a team doing a CS project.
In terms of the threat level, we are only concerned with threats up to disgruntled employees under Schneider's Taxonomy of Cybersecurity Threats. We will not deal with criminals or anything with that level of resources and motivation or more because hacking our system would not result in much financial gain.
We will assume that the server that the files are hosted on is physically secure, and that the network that the files are distributed through is not tempered with (no Dolev-Yao attacker).
We will assume that the admin is also trusted and will not perform any insider malfeasance

System Purpose

To provide a secure file sharing system for users.
Users are allowed to register and upload files to a private repository that should not be viewed or modified by others without authorization. They can then share files with other users and control whether other users are allowed to download or overwrite those files or edit the permissions on the files. All actions on files and folders are logged.
Users with viewing permissions can see the log associated with a file or folder. Changing the permissions on a folder sets all subfiles and subfolders to have the same permissions unless there are other non-conflicting permissions (which should remain).
System administrators are allowed to a log of all user account actions (creation of accounts, change of passwords and account information).
We will require all users to be logged in before they can perform any actions. We will use a 2FA system (the second factor being email) to log in.

Functional Requirements

User Types: We separate them into:

- Admin - the system administrator (there is only one admin for accountability)
- Owner - the user who uploads the file
- Editor - users with edit rights to a file
- Viewer - users with view but not edit rights to a file

Note that Owners are automatically Editors and Editors are automatically Viewers. Users refer to all individuals that have an account registered in the system. An admin is not a user and the admin uses a separate account for administrative purposes described below. Admins have some other functions that are listed below: being able to see user account logs.

The event log of a file or folder refers to a log that keeps track of all operations that have been performed on a file or folder: creation, editing (through overwriting), renaming and deleting. The event log of a user refers to a log that keeps track of user account activity: creation, file operations, and deleting.

Members: Brandon Peh (bcp39), Louise Lee (zl245), Ruixin Ng (rn279), Wang Zilong (zw243)

User Type	Assets	Importance	User Story
Owner	Files	M	As an owner, I can upload new files to the system.
Owner	Folders	S	As an owner, I can create new folders in the system.
Owner	Folders	W	As an owner, I can upload new folders in the system.
Editor	Folders/files	S	As an editor, I can upload new files to folders that are shared with me.
Editor	Files	S	As an editor, I can overwrite files that are shared with me.
Editor	Folders/File	S	As an editor, I can share folders and files that I have uploaded with other users, giving them either read/write access.
Editor	Folders/files	C	As an editor, I can rename folders and files that are shared with me.
Editor	Previous versions of files	W	As an editor, I can rollback my files to a previous version up to the last 5 days.
Viewer	Files	M	As a viewer, I can download files that I have access to view (either uploaded by me or shared with me by other users).
Viewer	Event logs	S	As a viewer, I can view the event log of a folder or file that is shared with me.
User	User accounts	M	As a user, I can sign up for a user account.
User/Admin	User accounts	C	As a user/admin, I can change my password.
User/Admin	User accounts	W	As a user/admin, I can require 2-factor authentication via email in order to log into my account.
Admin	Event logs	S	As an admin, I can monitor the the logs of user account activity as well as file activity.
Admin	Server space	C	As an admin, I can restrict the total size of files of all users.
Admin	User accounts	W	As an admin, I can add or remove users from my system.

Members: Brandon Peh (bcp39), Louise Lee (zl245), Ruixin Ng (rn279),
Wang Zilong (zw243)

Security Goals

	Asset & Value to stakeholders	Stakeholder(s)	Harm	Security Goal
1.	Files and folders (Stakeholders require files for personal usage, such as work or entertainment)	Users	Unauthorized modifications of the content of files in the system could damage the integrity of the files to the users.	The system shall prevent modification of files by unauthorized principals. [I]
2.	Files and folders	Users	Unauthorized deletion of files in the system could deny legitimate users access to the content that they need.	The system shall prevent deletion of files by unauthorized principals. [A]
3.	Files and folders	Users/Admin	Unauthorized access to files (viewing/knowing about their existence) could infringe on the privacy of users.	The system shall prevent unauthorized principals from viewing the content of the files stored on the server. [C]
4.	User Permissions (Stakeholders need user permissions to access the files that they need for personal usage)	Users	Unauthorized modification of user permissions could deny users the ability to modify the files/folders they need.	The system shall prevent unauthorized principals from changing the user permissions of a file/folder. [A]
5.	User permissions	Users	Escalation of privileges could result in unauthorized actions.	The system shall prevent privilege escalation of unauthorized principals. [I]
6.	User permissions	Users	Giving too many user permissions to a user could allow a user to invade another user's privacy by accessing the personal files of another user.	The system shall prevent too many user permissions from being handed out. [C]
7.	Server space (Stakeholders need server space to host this system)	Admin	Allowing users to upload files to the server without restriction on total file size could cause	The system shall prevent excessive usage of the

			the server to run out of storage capacity very quickly.	server's storage capacity. [A]
8.	Server space	Admin	Allowing users to upload files of arbitrarily large size could slow down the system too much for other users.	The system shall restrict the size of files stored on the server based on their file type. [A]
9.	Previous versions of files (Stakeholders need previous versions of files to roll back to them if needed)	Users	Unauthorized access to the previous versions of files could compromise the security of the files and result in an invasion of privacy.	The system shall prevent unauthorized access of the previous versions of files. [C]
10.	User accounts (Stakeholders require an account to store and access files on the system)	Users	Unauthorized access to user accounts could infringe upon the privacy of users and might lead to unauthorized modifications of files, thereby undermining its integrity.	The system shall prevent unauthorized access to user accounts. [C]
11.	User accounts	Users	Authorized access to password files could lead to unauthorized access to user accounts, which would infringe upon their privacy.	The system shall prevent unauthorized access to password files. [C]
12.	Event logs of files (Provides detailed description of R/W history of files of a user, which allows stakeholders to track down what happened at different time intervals)	Users/Admin	Damage to/denial of access to event logs of files could damage the system's ability to mitigating and recovering from harm to files and folders. (Specifically, this will make it difficult for the system to rollback changes.)	The system shall prevent an unauthorized access to the event logs of changes made to files. [I]
13.	Event logs of users (Stakeholders are able to keep track of user activities and note unusual activities, if any)	Admin	Damage to/denial of access to event logs of users could damage the system's ability to hold principals accountable and prevent them from causing further damage.	The system shall prevent an unauthorized access to the event logs of users. [I]

Members: Brandon Peh (bcp39), Louise Lee (zl245), Ruixin Ng (rn279), Wang Zilong (zw243)

Feasibility Analysis (for each security goal listed above, using the details listed above)

1. This security goal is feasible by implementing audit mechanisms. Furthermore, we can also include a rollback scheme to undo any unauthorized modifications to the system.
2. This security goal is feasible by implementing authentication and audit mechanisms. The authentication mechanisms will prevent unauthorised principals from deleting the file/folder. The audit mechanisms will allow us to determine which user to hold accountable for the deletion.
3. This security goal is feasible by implementing authentication mechanisms and encrypting the contents of files and folders. The authentication mechanism, paired with privilege separation, ensures that users do not gain permission to accounts and files/folders that they should not have access to. By encrypting the contents of files and folders, unauthorised principals (including admins) cannot access the contents or metadata of files and folders.
4. This security goal is achieved by user authentication and privilege separation. Only owners and editors of a file are granted the privilege to disseminate permissions.
5. Similar to the argument posited in (4), this security goal is feasible by limiting principals who can disseminate information.
6. This security goal is extremely feasible because the admin can simply assign a restricted storage usage for users, so that the storage capacity of the server is properly regulated.
7. This goal is feasible since we can enforce limits on the total size of files of any user.
8. A limit is enforced on the maximum file size that can be uploaded.
9. Only Owners and Editors are granted permission to rollback the files, and hence are the only principals who are able to download the previous version of files. Also, previous versions of files are encrypted, thereby further ensuring that unauthorized principals cannot access to the contents of the files.
10. This security goal is feasible by implementing authentication schemes such as user IDs and passwords to authenticate a user. It is the first step of defense against hackers who wish to infiltrate our system.
11. Password files can only be written to by the appropriate principals. Passwords are hashed and salted. Passwords must include a number and a special character.
12. Event logs of files are can only be viewed by principals with at least Viewer permissions. The files are encrypted and decrypted on client side, and cannot be downloaded nor modified by any user and admin.
13. Event logs of users are secured are only accessible to Admins.

Details of Implementation of Security Mechanisms

1. Event logs of files (audit): This is done through logs that record what a user does in the system. For example, when a user uploads a file, the action is recorded in a log.
2. User authentication: By implementing a registration and login mechanism to our system, it ensures that users do not gain access to files that they do not have the permission to. A 2FA scheme might also be implemented to further guarantee that only that user has access to their account and files.
3. Event logs of users (audit): A log of user activity is maintained, which keeps track of updates (eg change of password, creation of new account) to user accounts.
4. Privilege separation: Users can be granted different levels of permissions to a file/folder (i.e. read/write permissions) and this is determined by an admin. All users (excluding owners) by default are not granted permission to read nor write to a file/folder.
5. Encryption: All files and folders of users will be encrypted and decrypted on client side. Every user will have an individual private key.

Members: Brandon Peh (bcp39), Louise Lee (zl245), Ruixin Ng (rn279), Wang Zilong (zw243)

Essential Security Elements

Authentication:
Authentication allows us to identify who is allowed to access our system. It is the first barrier in preventing the threats listed above from accessing our system and compromising the confidentiality and integrity of our file sharing system. Only legitimate users of the system are allowed to access the system and we would like to keep any threats from accessing the system in the first place. Authentication is therefore essential in providing a first line of defense against any intruders.

Furthermore, authentication also allows us to bind principals of our system to the actions that they perform. Each user of the system is associated with a user ID, and whenever an action is performed by a user, this can be recorded in an event log. Any malicious actions by insiders or a hacked account can be traced with the event log. Thus, authentication can hold users accountable for their actions.

Authorization:
Authorization is another critical aspect of our system because there are various types of users in our system. We want to differentiate between the different kinds of actions that different users can perform, so a mechanism is needed to determine what actions can be performed. A user can either intentionally or unintentionally cause harm to the system if given too many permissions. In our system. Authorization is crucial because we do not want users to have access to assets that they should not view or modify as we must uphold the principle of least privilege. Appropriate authorization must also be given to legitimate users to ensure the availability of files to these users.

Audit:
As mentioned in authentication, we require audit mechanisms such as event logs that can record and review actions. This allows us to hold users accountable for the actions that they perform in the system. Malicious users who abuse the permissions given to them will have their account suspended. This audit mechanism is important in deterring potential attackers as well as ensuring that users of the system act responsibly. Furthermore, apart from deterrence, monitoring the event logs can allow the admin to spot any suspicious actions that have been performed. This allows the admin to take the appropriate measures such as performing a system rollback in order to revert the system back to a safe state. These reasons demonstrate the importance of the audit mechanism in our system.

Confidentiality:
Confidentiality is key in our system as well. Since our system implements file sharing, users who upload personal files would naturally want to keep their files secret and secure. This makes confidentiality important because we would want to protect these files (assets) from any unauthorized disclosure. A key feature of our system is that a user is able to share files to a certain group of users. However, at the same time, the user who shares the files might not want other unauthorized viewers to see the file. For example, a team that has their code stored using this system will not want other non-team members to see their code. Given that some users want to share sensitive information with a restricted group of people,

confidentiality is therefore an important security element that we have to implement for our system.

Integrity:
Integrity is also integral to our system. Users who do not have the right to edit files should not be able to make changes to files - owners should be assured that only editors have the right to make changes to files. Aside from privilege restrictions, our system also detects all changes and records it in an event log. This will provide some deterrence through accountability.