# Use Cases

## Glossary

There are currently two roles in the COPS Platform system. The role grants access and their security clearance determines their viewing and editing capabilities.

* Administrator: The administrator assigns username and passwords to employees. They can assign or edit security context for each user and data. The administrator interacts with the database directly.
* Employee: We consider everyone using the system to be a normal user. Access to data is purely based on security context assigned by the administrator. Employees interact with the container and the service, but not with the database directly.
* Service: The service runs inside of a container, and allows employees to make API calls. The service makes requests to the database, and allows the employee to send/receive data to/from the database if permitted.

## System Preconditions

The data storage has a file with user names and security context for each user. Users who are not on this list cannot access the system. Only the admin can access this list from the database.

\*\* This will change if we implement IdAM (stretch goal)

## INDEX

UC1: Database Interaction (Admin)

UC2: Database Interaction (Service)

UC3: Container Runtime

UC4: Log Transactions (SG)

UC5: Access Logs (SG)

UC6: Authenticate Users (SG)

## UC1: Data Interaction (Admin)

**1.1 Preconditions**

An admin account has been pre-authorized in the system and has logged in.

**1.2 Main Flow**

An admin can configure [S1], populate [S2], and modify the database [S3].

**1.3 Sub-flows**

* [S1] The admin runs SQL commands to: create tables, modify table security context, and modify column security context.
* [S2] The admin has access and run permissions for a program which populates the database.
  + In our case, we will just populate the database with randomly generated data.
  + In the general case, the client will likely populate the database with existing data.

**1.4 Alternate Flows**

* [E1] If the user is trying to submit information to a security level not allowed to them, the system returns an error message saying “Access denied as you can only submit data in your own security context”.
* [E2] If the user is trying to access information from a higher security level, the system returns an error message saying “Access denied”.

**1.5 Logging**

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction code** | **Description** | **Logged in username** | **Transaction type** |
| 100 | Database initialization | Admin | Create |
| 101 | Database configuration | Admin | Edit |
| 102 | Populate database | Admin | Create |
| 103 | Access Denied | Not Admin | Error |

**1.6 Data Format**

|  |  |
| --- | --- |
| **Field** | **Format** |
| Data | Follows format for the organization/ file they edit |

**1.7 User Story**

## 

## UC2: Database Interaction (Service)

**2.1 Preconditions**

A RESTful service has an account set up with the database. The database has been populated and configured by the admin with the proper labels

**2.2 Main Flow**

A service can read data [S1], write data [S2] and view reports [S3].

**2.3 Sub-flows**

* [S1] Service sends a read request to the database via an API call:
  + e.g. GET /api/table\_name/id

The database returns the specified data to the service [E1][E2].

* [S2] Service sends a write request to the database via an API call:
  + e.g. PUT /api/table\_name/id
  + Data to write is contained in the request body

The data is written to the database, and a success response is returned to the service [E3].

* [S3] Service sends a request to view reports to the database via an API call:
  + e.g GET /api/reports/table\_name/filter

The database returns the specified report to the service [E4][E5].

**2.4 Alternate Flows**

* [E1] If the service is trying to read data with a higher security level, an error response is returned with the message “Access denied: Cannot read data higher than your security level”.
* [E2] If the service is trying to read data outside of its permitted categories, an error response is returned with the message “Access denied: Cannot read data outside of permitted categories”
* [E3] If the service is trying to write data outside of its security context, an error response is returned with the message “Access denied: Cannot write data outside of your security context”.
* [E4] If the service is trying to view reports with a higher security level, an error response is returned with the message “Access denied: Cannot view reports higher than your security level”.
* [E5] If the service is trying to view reports outside of its permitted categories, an error response is returned with the message “Access denied: Cannot view reports outside of permitted categories”.

**2.5 Logging**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transaction code** | **Description** | **Logged in username** | **Secondary user** | **Transaction type** |
| 200 | Successful data submission | User | N/A | Submit |
| 201 | Retrieve data | User | N/A | View |
| 202 | View report | User | N/A | View |
| 203 | Failed access | User | Security level attempted to access | View |
| 204 | Failed data submission | User | Security level attempted to access | Submit |

**2.6 Data Format**

|  |  |
| --- | --- |
| **Field** | **Format** |
| Data | Follows format for the organization/ file they edit |

**2.7 User Story**

## UC3: Container Runtime

**3.1 Preconditions**

The user has authenticated themselves in the COPS Platform system.

**3.2 Main Flow**

A user can make a service request to the system [S1].

The system can create an isolated container to run the requested service for the user [S2].

The system can monitor all containers with running services [S3].

The system can destroy a container once the user disconnects from the service [S4].

**3.3 Sub-flows**

* [S1] User requests a service from the system with the following information [E1]:
  + SELinux security contexts of the user
  + Name of the requested service
* [S2] The system creates a container with the enforced security context of the user [E2]. Starts the requested service on the container and establishes a connection with the user [E3].
* [S3] The system monitors all the running container instances for any changes of state [E3].
* [S4] The service in a container is stopped when the user disconnects.The system destroys the running container with the stopped service [E4].

**3.4 Alternate Flows**

* [E1] The service request contains invalid SELinux security labels and/or an invalid service name. Returns an error response to the user.
* [E2] An error occurs with the creation of the container. The system sends an error response to the user.
* [E3] A running container enters another lifecycle state (‘Stopped’, ‘Paused’, ‘Dead’). The system attempts to set the container back to a ‘Running’ state.
* [E4] An error occurs when the system attempts to destroy the container. The system attempts to destroy the container again.

**3.5 Logging**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transaction code** | **Description** | **Logged in username** | **Secondary user** | **Transaction type** |
| 300 | Service Request | User |  | Other |
| 301 | Container  Created | User | Service | Other |
| 302 | Service Started | User | Service | Other |
| 303 | Container State Change | User | Service | Other |
| 304 | Service Stopped | User | Service | Other |
| 305 | Container Destroyed | User | Service | Other |

**3.6 Data Format**

|  |  |
| --- | --- |
| **Field** | **Format** |
| SELinux labels | Must adhere to the following format:  *sensitivity [: category, ... ]*  Examples:  s0:c0.c25 ( all categories in range c0-c25)  s5:c5,c45,c255 ( only categories 5, 45, and 255)  Sensitivity number value must be in the range 0-15.  Category number value must be in the range 0-255. |
| Service Name | TBD (Must be a valid Service) |

**3.7 User Story**

## UC4: Log Transactions (SG)

\*\* Use Log aggregator

Maintaining confidentiality and integrity of data is paramount. Complete log files are critical for performing forensics on inappropriate access (create, read, update, delete) of data and on the inappropriate granting of system privileges to users.

**4.1 Preconditions**

None

**4.2 Main Flow**

Any event which creates, views, edits, or deletes information is logged [S1]. Login failures, valid authentication, and log outs are also logged [S2].

**4.3 Sub-flows**

* [S1] For creating, viewing, modifying, or deleting information, the following information is recorded: the username of the logged in user, any appropriate secondary username of the user whose information is being accessed, a transaction type corresponding to the given action, and the current timestamp. Individual audit codes related to specific use cases are presented within each Use Case description. The subflow and transaction values are based on Use Case. For example, any in the range of 100-199 are for UC1, any in the range of 300-399 are in UC3, etc. The exception is for authentication [S2].
* [S2] The values from range 0-99 are logging events which do not exist in any use case but are concerned with the system as a whole. Logging associated with authentication [UC2] is also in this range. Miscellaneous transaction codes 1-99 are presented in Section 3.5 below.

**4.4 Alternate Flows**

None

**4.5 Logging**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transaction code** | **Description** | **Logged in username** | **Secondary user** | **Transaction type** |
| 0 | Failed login | IP address | N/A | Other |
| 10 | Successful login | user | N/A | Other |
| 5 | User locked | user | N/A | Other |

### 

## UC5: Access Logs (SG)

**5.1 Preconditions**

A user is a registered user of the COPS system (UC1). The user has authenticated himself or herself into the system (UC2).

**5.2 Main Flow**

The user logs in and sees the top entries in his or her access log [S1]. The user can send a request to view his or her access log [S2] separately. The user can then choose the beginning and end date for the period of time they would like to view their access log for [S3] and send the request. The resulting list should include the following for each access:

* Name of accessor
* Role of accessor
* Date and time of access
* Transaction type

**5.3 Sub-flows**

* [S1] When the user logs in, they can see ten most recent events.
* [S2] By default, upon requesting a detailed list, the user is presented with a list of all entries sorted by dates, most recent access first.
* [S3] The user may enter a date range to view all entries within the range [E1].

**5.4 Alternate Flows**

* [E1] The user sends an invalid date, or sends an end date that is before the start date. No events are returned and the user has the opportunity to send different dates.

**5.5 Logging**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transaction code** | **Description** | **Logged in username** | **Secondary user** | **Transaction type** |
| 500 | Access logs viewed | User | N/A | View |

## UC6: Authenticate Users (SG)

\*\* IdAM will be used

**6.1 Preconditions**

None

**6.2 Main Flow**

A user enters their username and password to gain role/ security level based entry into the COPS system [E1]. A session that has been inactive for more than ten minutes is terminated [S1]. Upon successful authentication, the user will be able to send requests to interact with data. An authenticate session ends when the user sends a request to log out or close the system.

**6.3 Sub-flows**

* [S1] Electronic sessions must terminate after ten minutes of inactivity. Ensure that authentication is reset after a period of inactivity that exceeds ten minutes.

**6.4 Alternate Flows**

* [E1] The user may try three times. After three failed attempts with a user id, lock out that username for 60 minutes. If the last 6 login attempts from a given IP address fail, across any number of users or non-existent usernames, that IP address will be locked out for 60 minutes. After the 60 minute lockout-period, a user gets 3 more attempts, and an IP address gets 6 more attempts.
* [E2] If a user or IP address is locked out 3 times in a 24-hour period, the offending user or IP address will be banned from the system until re-authorized by a system administrator.

**6.5 Logging**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transaction code** | **Description** | **Logged in username** | **Secondary user** | **Transaction type** |
| 0 | Failed login | IP address | N/A | Other |
| 1 | Successful login | user | N/A | Other |
| 2 | Logged out | user | N/A | Other |
| 3 | User locked out | user | N/A | Other |
| 4 | IP locked out | IP address | N/A | Other |
| 5 | User banned | user | N/A | Other |
| 6 | IP banned | IP address | N/A | Other |

**6.6 Data Format**

|  |  |
| --- | --- |
| **Field** | **Format** |
| time | YYYY-MM-DD HH:MM:SS |
| ip | A valid IPv4 or IPv6 address. A String in the form "X.X.X.X" where each X is a integer (base 10) from 0 to 255, or "X:X:X:X:X:X:X:X" where each X is a hexadecimal number from 0000 to FFFF (no 0x- prefix), and extraneous zeros may be omitted. |
| user | A valid user within the system |

**6.7 User Story**