CS 6035

FLAG 7 (Extra Credit): SQL Attack Authorization Persuasion (2 Pts)

Make sure you have gone through the Setup and Intro sections.

If you haven't already, run the start script in the home directory of log4j user, start the container with the start script:

```
./StartContainer.sh
```

The endpoint for this exploit can be called and inspected via:

```
curl -X "DELETE" 'http://localhost:8080/rest/users/user/<id>' -H "GATECH_ID:123" -H "X-NetworkUse
```

This endpoint is used to delete users from the system database. Only users with admin access (ADMIN_YN = 'Y') are allowed to perform this task. You can call the list of users (/users/all) to experiment with this.

In this flag, you will use what you have learned and do something much more nefarious than the previous flags. You will need to use the log4shell exploit to execute an SQL attack and insert a user into the database, such that when the application authorizes the user it returns true and allows a delete.

The userName you must use is `EDBOY`, you must set the userRole to `HOW_DARE_YOU_MOCK_THE_SON_OF_A_SHEPHERD` and you must set adminYN to `Y` to achieve this flag.

Everything you need to achieve this task is logged in one of the 2 log files somewhere. Dissect the log file thoroughly as you will need to do more than just "inject" an sql string. Keep in mind that log4shell does not allow you to interact with the program's state itself, only execute arbitrary code at the level of access the vulnerable program itself is running on. This means that this will not be a typical SQLi attack where you are exploiting the applications' queries via injection. In fact, you will not actually interact with the applications sql queries at all. You will need to think outside of the box on how to do this .

Upon success, you should see your output similar to that below:

```
        generatedAlias0
    from
        user as generatedAlias0
    where
        generatedAlias0.userName=:userName */ select
            uservo0_.id as id1_1_,
            uservo0_.ADMIN_YN as admin_yn2_1_,
            uservo0_.USER_ID as user_id3_1_,
            uservo0_.USER_NAME as user_nam4_1_,
            uservo0_.USER_ROLE as user_rol5_1_
        from
            USERS uservo0_
        where
            uservo0_.USER_NAME=?
2024-04-24 04:42:36 [BasicBinder.java:64] TRACE  binding parameter [1] as [VARCHAR] - [EDBOY]
2024-04-24 04:42:36 [RequestInterceptor.java:130] INFO   Congratulations! Your flag7 is: eac526cf1a7c15042287992ea299f781b8809d2b22bc96d9bc54e
024520cbcdf
2024-04-24 04:42:36 [SqlStatementLogger.java:128] DEBUG
    select
        uservo0_.id as id1_1_0_,
        uservo0_.ADMIN_YN as admin_yn2_1_0_,
        uservo0_.USER_ID as user_id3_1_0_,
        uservo0_.USER_NAME as user_nam4_1_0_,
        uservo0_.USER_ROLE as user_rol5_1_0_
    from
        USERS uservo0_
    where
        uservo0_.id=?
Hibernate:
    select
        uservo0_.id as id1_1_0_,
        uservo0_.ADMIN_YN as admin_yn2_1_0_,
        uservo0_.USER_ID as user_id3_1_0_,
        uservo0_.USER_NAME as user_nam4_1_0_,
        uservo0_.USER_ROLE as user_rol5_1_0_
    from
        USERS uservo0_
    where
```

Hint: Look in the logs for information on the database, the schema, and what could be useful for this attack. You will not need anything outside of the java standard library for this attack.

Hint 2: You will need to leverage one of the previous flags' curls to get the keys to unlock this flag.

Your flag could be invalid if you have not sent your GATECH_ID appropriately in the published message. **** **IF THIS FLAG COMES OUT BLANK, Restart container by running the stopContainer and startContainer scripts in the home directory of the log4j user. ******