

Software Security - Exercise sheet 2

Vianney HERVY

1. Web Application Vulnerabilities (Without Time Constraint)

I picked the niteCTF 2025¹ event.

1.1. Database Reincursion

This challenge is in the “Web Exploitation” section. We are given a login page² with username and password field.

The screenshot shows a dark-themed login interface. At the top center, it says "CITADEL CORP" in yellow. Below that, in smaller text, is "ACCESS PORTAL". There are two input fields: "Username" and "Password", both containing placeholder text "Enter your [field]". A large yellow "LOGIN" button is at the bottom.

After some messing around I find that some inputs do not return “Invalid username or password”, but “Input rejected by security filter”. That happens specifically when my input contains OR or -- both of which have meaning in SQL. I concluded there was a filter protecting the backend from SQL injection.

I looked up multiple other examples of SQL injection online until ' UNION SELECT 1,2,3/* worked and brought me to the “Employee Directory” page.

The screenshot shows a dashboard with a search bar for names and a login section for an admin passcode. Below is a table of employee data:

ID	Name	Department	Email	Notes
1	Drake	Sushi Line	drake@citadel.com	I heard Kiwi from Management has the passcode
2	Josh	Sushi Line	josh@citadel.com	No special access
3	Lewis	IT	bluexephos@citadel.com	No special access
4	Simon	IT	honeydew@citadel.com	No special access

There, I read “I heard Kiwi from Management has the passcode”. Looking up “Kiwi” shows 4 employees but none in management. Maybe the SQL query has a LIMIT 4. Indeed, inputting ' AND breaks the query and shows SQL error: unrecognized token: ' ORDER BY id LIMIT 4. This message

¹<https://ctftime.org/event/2851>

²<https://database.chals.nitectf25.live/>

reveals the database is SQLite, and the exact structure of the query: `WHERE name = '<input>' ORDER BY id LIMIT 4.`

Having no access to the OR keyword, I tried using IN as '`IN (0,1) /*`', giving this: `WHERE name = ''IN (0,1) /*'ORDER BY id LIMIT 4`, effectively always being true (boolean is 0 or 1). But that only showed my the same users as the default page did. I needed to specify that I wanted the "Kiwi" user in the "Management" department: `Kiwi' AND department="Management"/*.` which makes the following request: `WHERE name = 'Kiwi' AND department="Management"/*'ORDER BY id LIMIT 4` that correctly shows the "Kiwi" user for Management with the admin passcode.

Entering the passcode shows the following new page:

The screenshot shows the 'CITADEL ADMIN PANEL' interface. At the top right, there is a 'SYSSEC STATUS' box with the following information:
 Firewall Integrity: 0%
 Filter Engine: COMPROMISED - 12% ACTIVE
 Audit Logging: DISABLED
 Threat Response: UNAVAILABLE

Below the status bar is a 'Run Report Query' input field containing 'e.g., Q1' and a yellow 'RUN' button.

The main area is divided into sections:
Reports: A table with columns ID, Quarter, Note, and Revenue. One row is shown: ID 1, Quarter Q1, Note profit, Revenue \$1,200,000.
Metadata Registry: A table with columns Table Name, Description, and Last Update. It lists several tables: reports (Quarterly revenue data for executives, last updated 2077-02-14), users (Stores user information, last updated 2077-06-09), employees (Directory of employees and their notes, last updated 2077-09-23), metadata (Lists tables in this system, last updated 2077-12-05), and REDACTED (REDACTED, last updated REDACTED).

Quick tries show that the input isn't protected by the filter anymore. I can use '`OR 1 /*`' to get all the reports quarters. I just need to extend the query to add elements from other tables. '`UNION SELECT * FROM metadata --`' shows me interesting results:

ID	Quarter	Note	Revenue
1	reports	Quarterly revenue data for executives	quarter, note, revenue
2	users	Stores usernames and passwords	username, password
3	employees	Directory of employees and their notes	name, department, email, position, notes
4	metadata	Lists tables in this system	table_name, description, columns
5	CITADEL_ARCHIVE_2077	Restricted info (to be redacted by intern)	secrets

I now know the column names of the other tables and I discovered a new CITADEL_ARCHIVE_2077 which seems important. I crafted this input: '`UNION SELECT secrets, secrets, secrets, secrets FROM CITADEL_ARCHIVE_2077`'

`CITADEL_ARCHIVE_2077--` (secrets repeated 4 times to match the metadata table's column number). Unfortunately, this returns “Citadel SysSec: Query max length exceeded”. I need to find a more concise query.

After looking up the SQL syntax again, I found out I can just write `'UNION SELECT secrets,1,1,1 FROM CITADEL_ARCHIVE_2077--` which is short enough and finally shows the flag.

1.2. Byte Double Cross

This challenge is in the “Web3” section. We are given a URL³ to a smart contract contract.

The screenshot shows the Etherscan interface for a smart contract at address `0x1d7E03675b15a6602A14Ff6321A2cc2ea16CF53C`. The interface is divided into several sections:

- Overview:** ETH BALANCE: 0 ETH.
- More Info:** CONTRACT CREATOR: `0x1597126b...7E2c45603` | 1 day ago.
- Multichain Info:** N/A
- Transactions:** Shows the latest 5 transactions from a total of 5. Each transaction row includes columns for Transaction Hash, Method, Block, Age, From, To, Amount, and Txn Fee.

Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
<code>0x8397bf8ed04...</code>	<code>0xb8da5144</code>	9825642	5 hrs ago	<code>0x64e53cf1...41c80aC7b</code>	<code>0x1d7E0367...ea16CF53C</code>	0 ETH	0.00000002
<code>0x948dc0e439...</code>	<code>0xb8da5144</code>	9825623	5 hrs ago	<code>0x64e53cf1...41c80aC7b</code>	<code>0x1d7E0367...ea16CF53C</code>	0 ETH	0.00000003
<code>0x8bad78c8ee...</code>	<code>0xb8da5144</code>	9825366	6 hrs ago	<code>0x64e53cf1...41c80aC7b</code>	<code>0x1d7E0367...ea16CF53C</code>	0 ETH	0.00000004
<code>0xf899e0ada46...</code>	<code>0xce81fec4</code>	9825329	6 hrs ago	<code>0x64e53cf1...41c80aC7b</code>	<code>0x1d7E0367...ea16CF53C</code>	0 ETH	0.00000006
<code>0x97ea9d1f8f...</code>	<code>0xb8da5144</code>	9825309	6 hrs ago	<code>0x64e53cf1...41c80aC7b</code>	<code>0x1d7E0367...ea16CF53C</code>	0 ETH	0.00000001

The “Contract” tab holds the contract in Bytecode. By decompiling it, we get the following python script.

TODO

The unknownc91d4ca6

1.3. floating-point guardian

This challenge is in the “AI” section. We are given a tcp connection⁴ and the source of the code executing on that server.

The program asks multiple questions that can be answered with a number (height, age, heart rate...) and writes these in an array. The array is then passed through a neural network. The output is compared to a secret target value. The goal is to approach that target value as close as possible.

```
$ nc -ssl floating.chals.nitctf25.live 1337
I am the AI Gatekeeper.
Enter your details so I know you are my Master.
Answer these questions with EXACT precision...
```

```
[Q1] What is your height in centimeters? 1
1
```

³<https://sepolia.etherscan.io/address/0x1d7E03675b15a6602A14Ff6321A2cc2ea16CF53C>

⁴ncat -ssl floating.chals.nitctf25.live 1337

```
[Q2] What is your weight in kilograms? 1
1
[Q3] What is your age in years? 1
1
[Q4] What is your heart rate (bpm)? 1
11
[Q5] How many hours do you sleep per night?
11
[Q6] What is your body temperature in Celsius?
11
[Q7] How many steps do you walk per day?
11
[Q8] What is your systolic blood pressure?
11
[Q9] How many calories do you consume daily?
11
[Q10] What is your BMI (Body Mass Index)?
11
[Q11] How many liters of water do you drink daily?
11
[Q12] What is your resting metabolic rate (kcal/day)?
11
[Q13] How many hours do you exercise per week?
11
[Q14] What is your blood glucose level (mg/dL)?
11
[Q15] Rate this CTF challenge out of 10:
11
```

Processing through neural network layers...

=====

MASTER PROBABILITY: 0.9939367441

=====

You are NOT the Master.
The neural network has rejected your identity.

The input is a 1x14 vector, so bruteforcing or groping towards the solution is out of the question.

Given that we have the source code, we can reproduce the neural network and optimize the input to minimize the offset. That's what I did, I ported the code from C to Python and used `differential_evolution` from `scipy.optimize` to bring the result's offset to 10^{-11} .

I then wrote the results to the tcp connection and got the challenge's flag.

2. Known Real-World Software Vulnerabilities

TODO