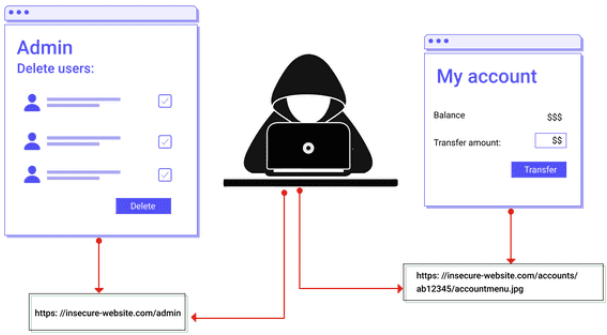
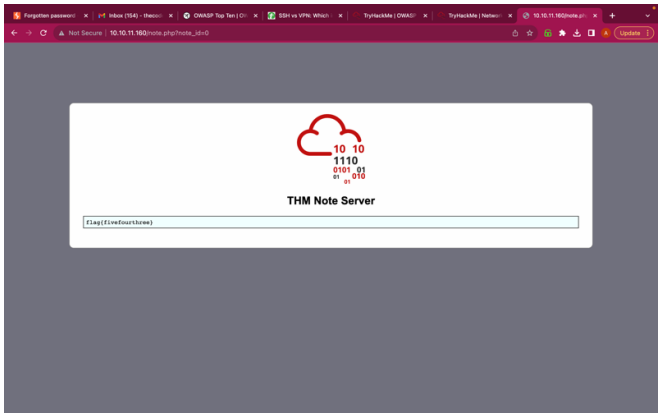
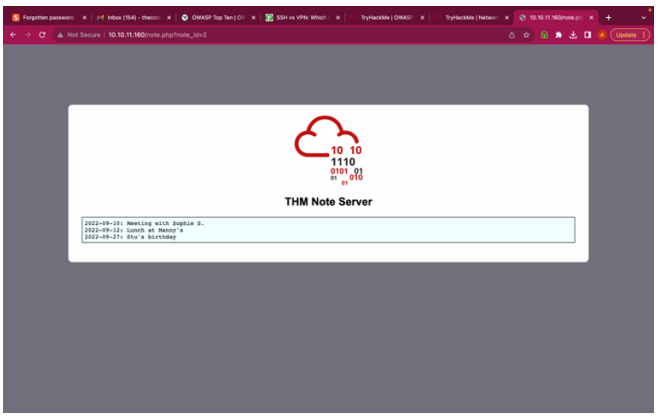
**ADITI RAI**
**ASSIGNMENT : 1**

# Top 5 OWASP  Vulnerabilities Demo

**A01:2021-Broken Access Control** : A user can access resources or perform actions that they are not supposed to be able to .



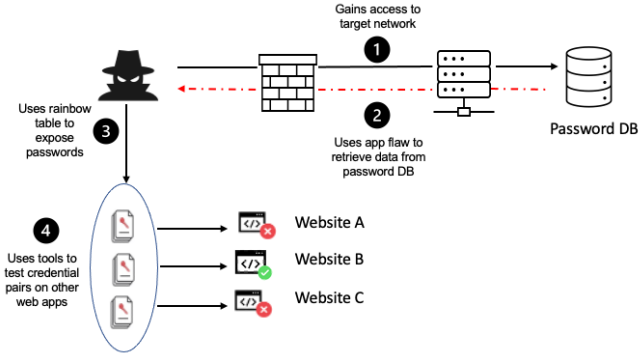



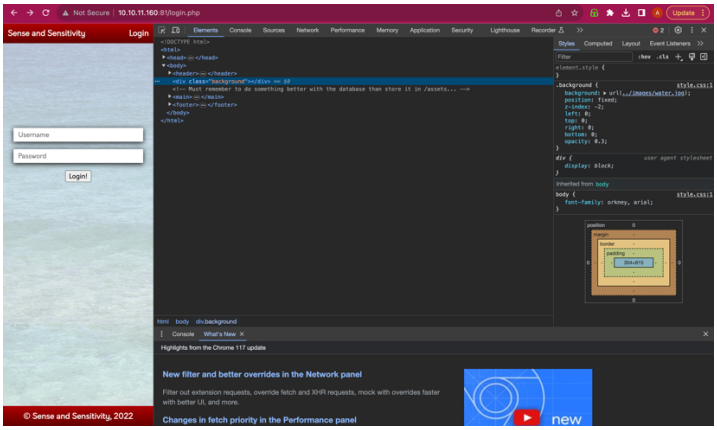



**A02:2021-Cryptographic Failures** : Cryptographic failures are where attackers often target sensitive data, such as passwords, credit card numbers, and personal information, when you do not properly protect them. This is the root cause of sensitive data exposure.

**What are some Common Examples?**

- Sensitive data is transmitted (via HTTP, FTP, SMTP, etc) or stored in clear-text (database, files, etc).
- Use of old or weak cryptographic algorithms.
- Use of weak or default encryption keys or re-use of compromised keys.

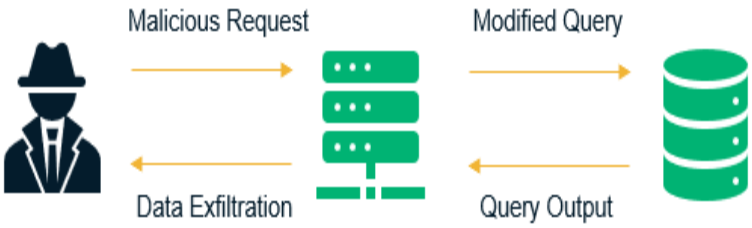


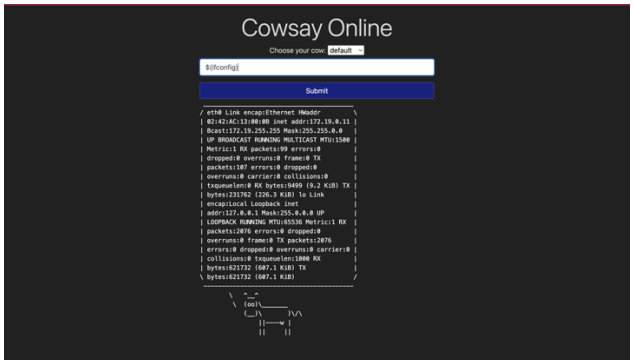




```
Aditis-MacBook-Air-2:Downloads aadi$ file webapp.db
webapp.db: SQLite 3.x database, last written using SQLite version 3022000, file counter 255, database pages 7, 1st free page 5, free pages 1, cookie 0x6, schema 4, UTF-8, version-valid-for 255
Aditis-MacBook-Air-2:Downloads aadi$ sqlite3 webapp.db
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.
sqlite> .tables
sessions  users
sqlite> PRAGMA table_info(users);
0|userID|TEXT|1||1
1|username|TEXT|1||0
2|password|TEXT|1||0
3|admin|INT|1||0
sqlite> SELECT * FROM users;
4413096d9c933359b898b6202288a650|admin|6eea9b7ef19179a06954edd0f6c05ceb|1
23023b67a32488588db1e28579ced7ec|Bob|ad0234829205b9033196ba818f7a872b|1
4e8423b514eef575394ff78caed3254d|Alice|268b38ca7b84f44fa0a6cdc86e6301e0|0
sqlite>
```

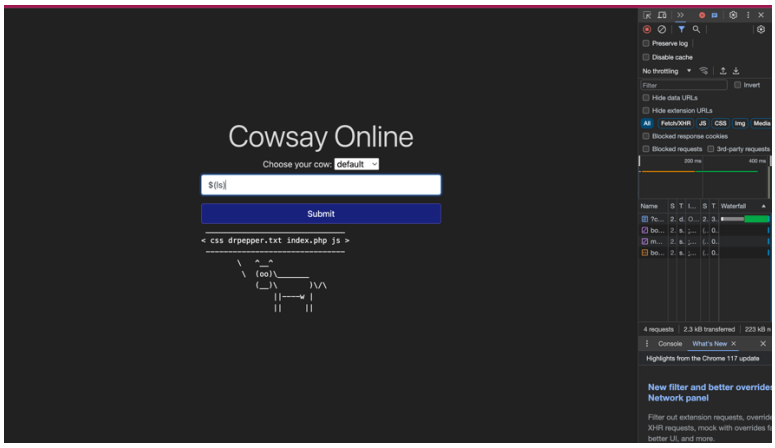

**A03:2021-Injection** : Injection is an attacker's attempt to send data to an application in a way that will change the meaning of commands being sent to an interpreter. For example, the most common example is SQL injection, where an attacker sends "101 OR 1=1" instead of just "101".

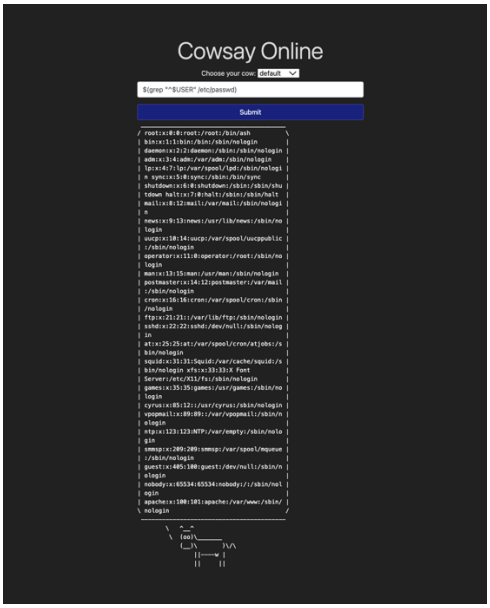
During an injection attack, untrusted inputs or unauthorized code are "injected" into a program and interpreted as part of a query or command. The result is an alteration of the program, redirecting it for a nefarious purpose.

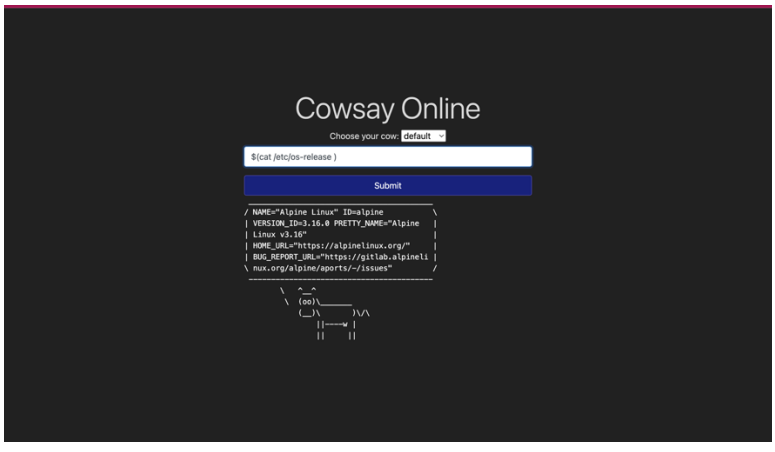


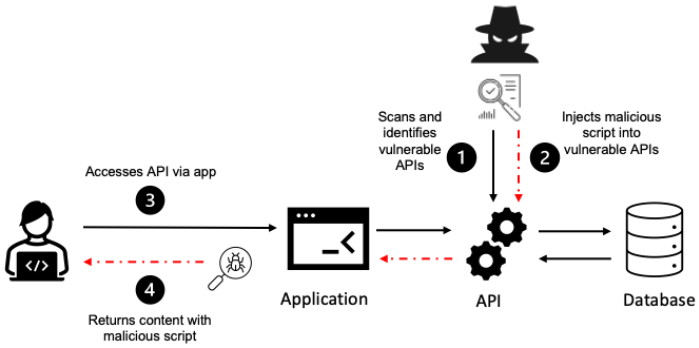
```
passthru — Execute an external program and display raw output

passthru("perl /usr/bin/cowsay -f $cow $mooing");
```

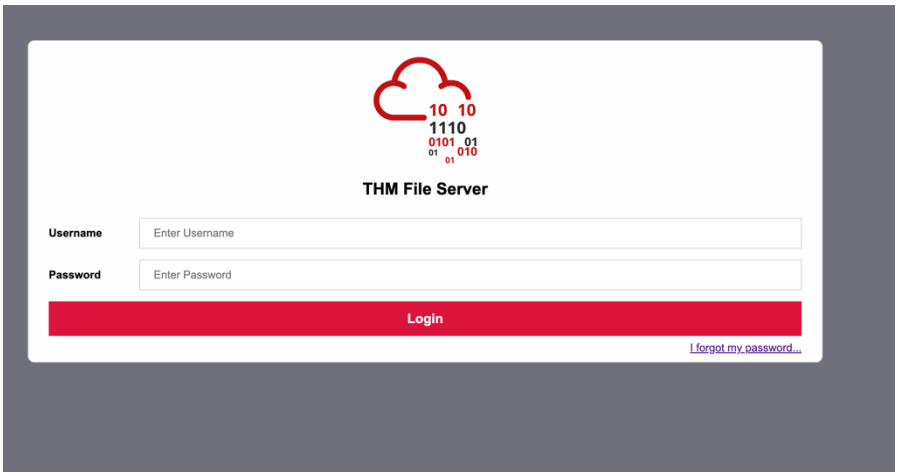


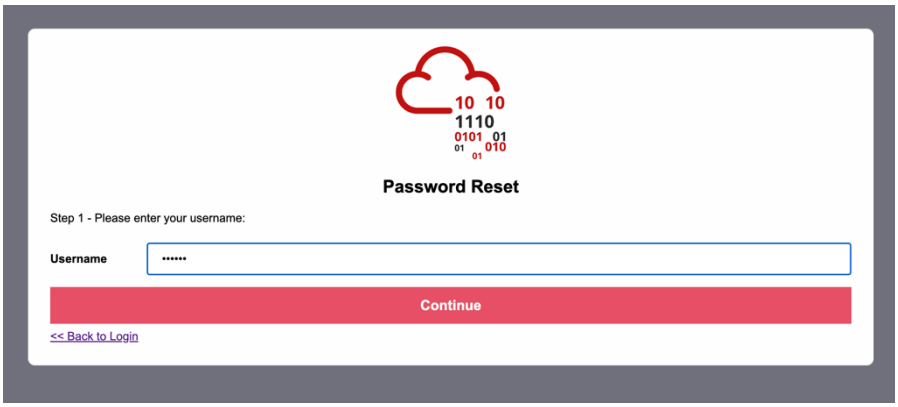



**A04:2021-Insecure Design** : Insecure design is a broad category representing different weaknesses, expressed as “missing or ineffective control design.” There is a difference between insecure design and insecure implementation. We differentiate between design flaws and implementation defects for a reason, they have different root causes and remediation. A secure design can still have implementation defects leading to vulnerabilities that may be exploited. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks. One of the factors that contribute to insecure design is the lack of business risk profiling inherent in the software or system being developed, and thus the failure to determine what level of security design is required.

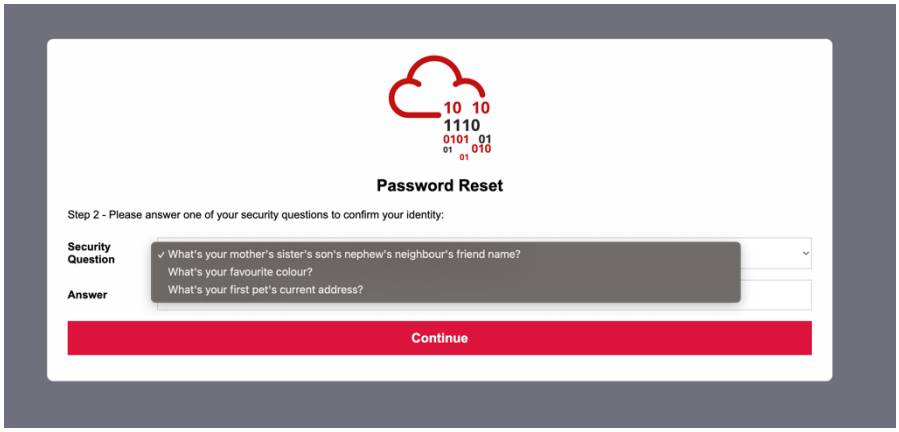



Choose forgot password



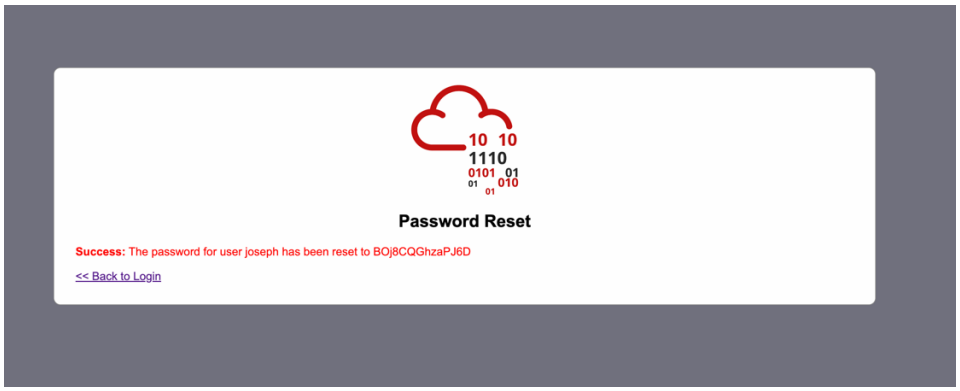

Enter the username given in task.



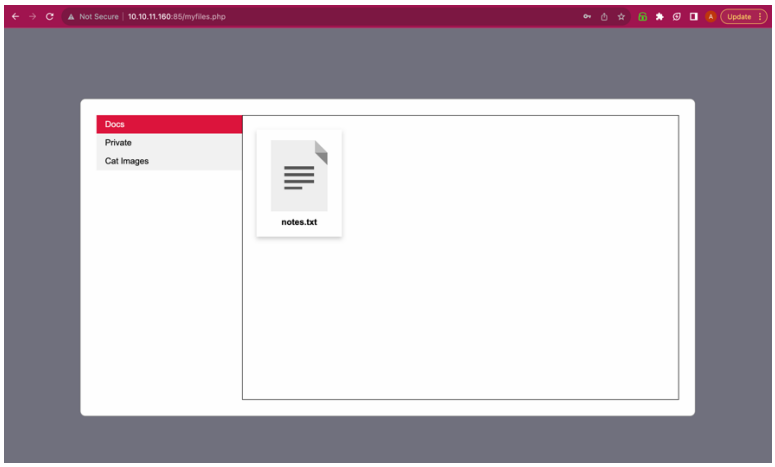

Check for the easy to guess security question.
And attempt guesses.

Example: What's you favourite colour?

Tried : black, red, blue, yellow, white, green
Success : green




Now you can login using the USERNAME : joseph and PASSWORD : BOj8CQGhzaPJ6D




Here is Successful Login into the Account due Insecure design of the web app .

# A05:2021-Security Misconfiguration