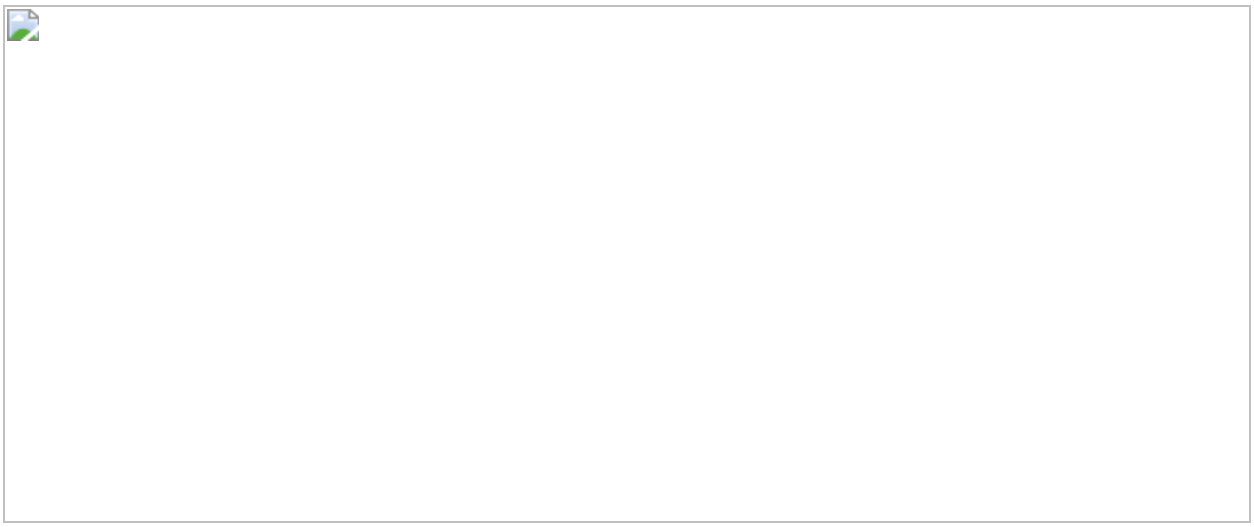


Week 1 - Assignment

Top 5 Vulnerability Exploitation

`1. Broken Access Control`

- `Description`

  Broken Access Control is a vulnerability that allows unauthorized users to access protected resources or perform actions that they should not be able to perform. This can occur when there is a failure to properly authenticate and authorize users or when access control rules are not properly enforced.

  For example, if a web application does not properly check user permissions when accessing certain pages or performing certain actions, an attacker could exploit this vulnerability to gain access to sensitive data or perform unauthorized actions. This could have serious business impact, such as loss of confidential information, reputational damage, financial loss, or legal liability.

  To perform this vulnerability, an attacker would attempt to bypass access control mechanisms by exploiting weaknesses in authentication and authorization mechanisms, such as brute-forcing passwords, exploiting session management flaws, or manipulating access control tokens. It is important to properly design and implement access control mechanisms to prevent this vulnerability and protect sensitive resources.

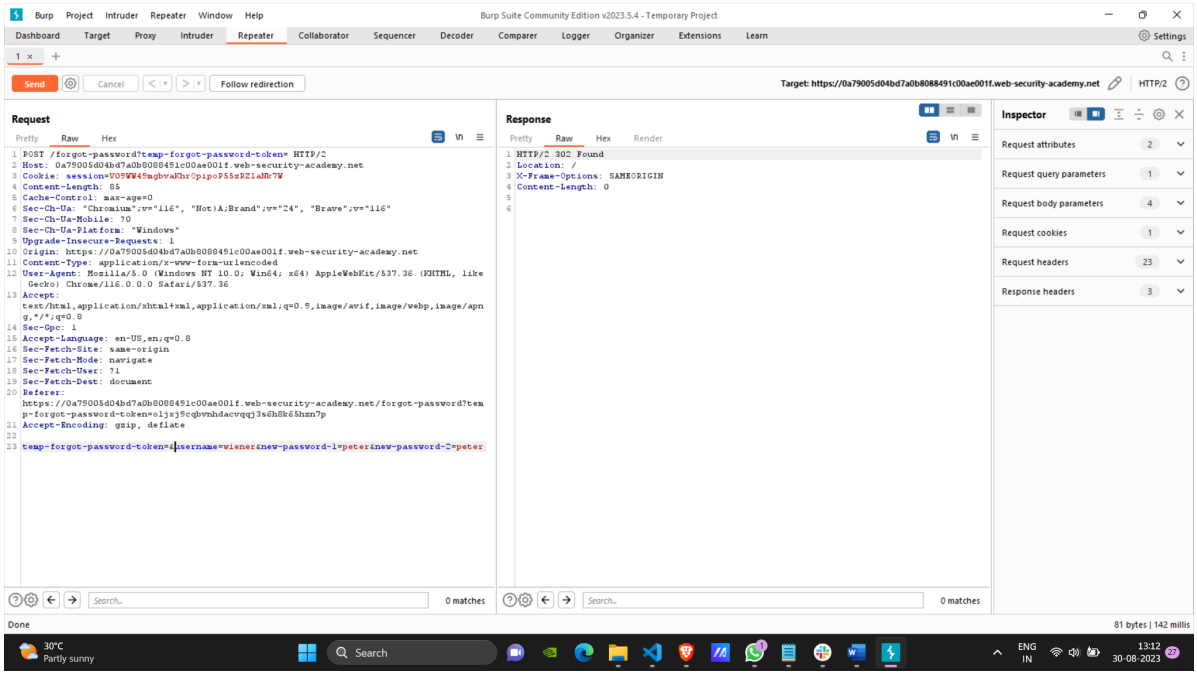
- `Business Impact`

  Broken Access Control can have serious business impact, such as loss of confidential information, reputational damage, financial loss, or legal liability. If an attacker gains unauthorized access to sensitive data or performs unauthorized actions, it could compromise the security of the entire system and lead to significant consequences for the organization. It is important to properly design and implement access control mechanisms to prevent this vulnerability and protect sensitive resources.

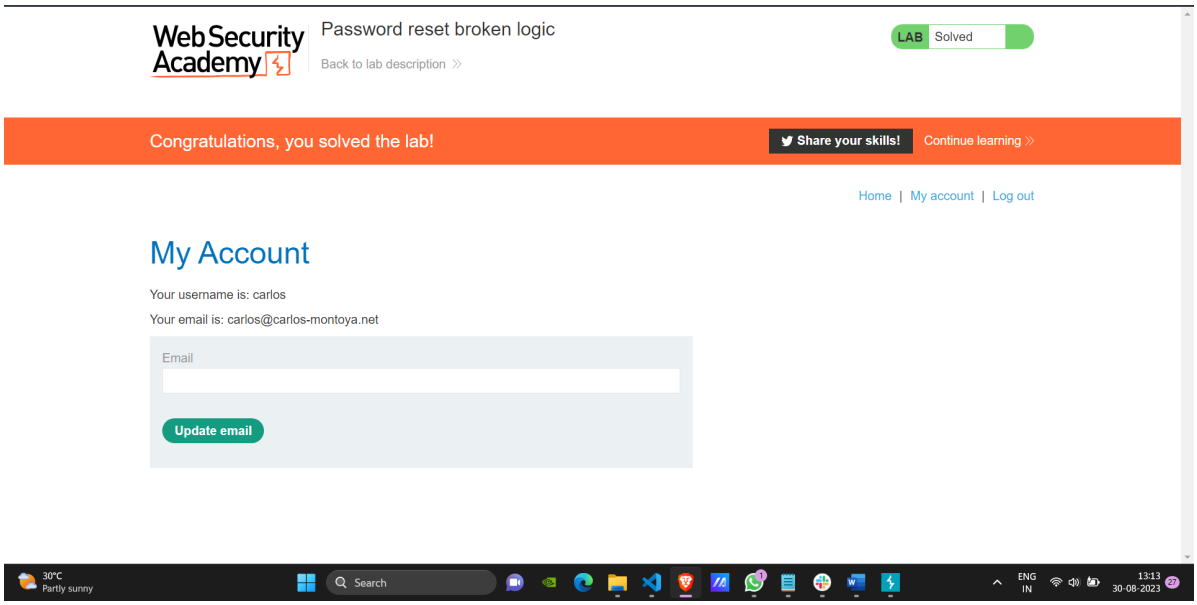
- `Performing the vulnerability`

  Using PortSwigger Lab we will try to access and change password for a user names carlos.

  Here, we *change the name in username parameter* while querying and posting, hence we can access a account names carlos




```
temp-forgot-password-token=&username=carlos&new-password-1=peter&new-password-2=peter
```



2. Cryptographic Failures

- `Description`

  Cryptographic failures are vulnerabilities that occur when the cryptographic mechanisms used to protect data are implemented incorrectly or are insufficiently strong. This can result in sensitive data being exposed or compromised.

  Cryptographic failures can occur in many ways, such as using weak encryption algorithms, improperly storing cryptographic keys, or failing to properly authenticate digital signatures. These vulnerabilities can be exploited by attackers to bypass security controls and gain unauthorized access to sensitive information.

  For example, if a web application uses weak encryption algorithms to protect sensitive data, an attacker may be able to easily decrypt the data and gain access to confidential information. Similarly, if cryptographic keys are not properly protected, attackers may be able to steal the keys and use them to access encrypted data.

  To prevent cryptographic failures, it is important to properly implement strong cryptographic mechanisms, such as using up-to-date encryption algorithms, properly storing cryptographic keys, and using digital signatures to ensure the integrity of data.

- `Business Impact`

Cryptographic failures can have serious business impact, such as loss of confidential information, reputational damage, financial loss, or legal liability. If an attacker is able to exploit a cryptographic vulnerability, they may be able to gain access to sensitive data and compromise the security of the entire system. It is important to properly implement strong cryptographic mechanisms to protect sensitive information.

- `Performing the vulnerability`

  This vulnerability takes place when important things like password are not stored after encryption but stored in direct normal text. For our example,

3. Injection

- `Description`

  Injection vulnerabilities occur when untrusted data is sent to an interpreter as part of a command or query. This can allow an attacker to execute malicious code or access sensitive information. Injection vulnerabilities can occur in many different contexts, such as SQL, LDAP, or XML.

  For example, if a web application does not properly validate user input when constructing SQL queries, an attacker may be able to inject malicious SQL code into the query and gain access to sensitive data. Similarly, if an application uses untrusted data to construct LDAP queries, an attacker may be able to inject LDAP commands to gain unauthorized access to the system.

  To prevent injection vulnerabilities, it is important to properly validate and sanitize all input data, and to use parameterized queries whenever possible. It is also important to limit the privileges of interpreters to reduce the impact of successful attacks.

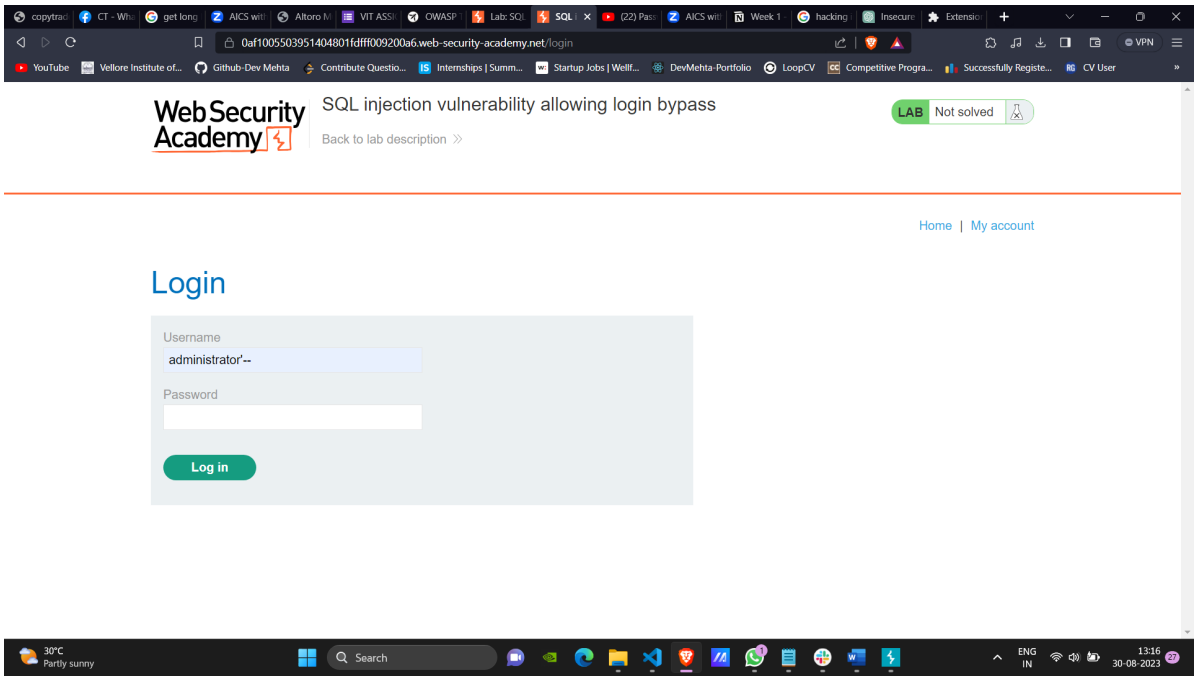
- `Business Impact`

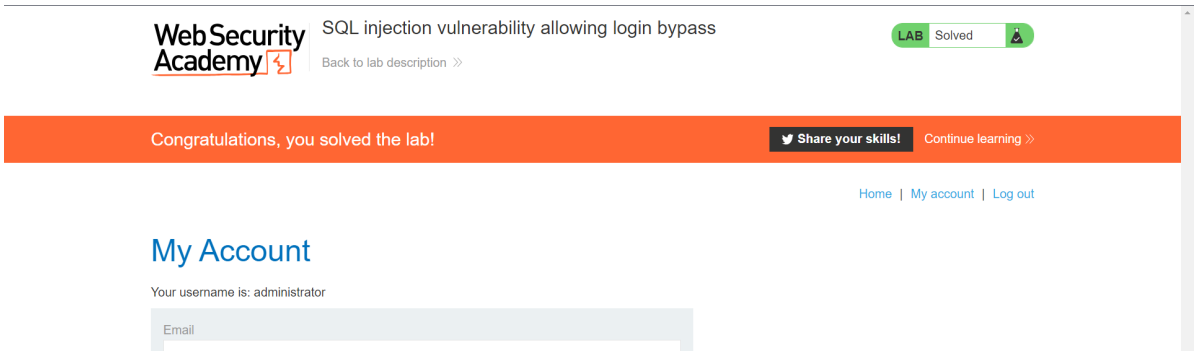
  Injection vulnerabilities can have serious business impact, such as loss of confidential information, reputational damage, financial loss, or legal liability. If an attacker is able to exploit an injection vulnerability, they may be able to gain access to sensitive data or execute unauthorized commands, compromising the security of the entire system.

- `Performing the vulnerability`

Again using PortSwigger Labs we are able to show login bypass using SQL Injection!

We use ‘— which comments the password in a way and hence allows us to login





4. Insecure Design

- `Description`

An insecure design vulnerability, often referred to as a security design flaw, is a type of software or system vulnerability that arises from poor or inadequate design choices in the architecture of a software application, system, or network. These design flaws can create pathways for attackers to exploit, compromise, or

manipulate the system, potentially leading to unauthorized access, data breaches, service disruptions, or other security incidents.

Unlike more traditional vulnerabilities that stem from coding errors, insecure design vulnerabilities are rooted in fundamental architectural decisions. Examples of insecure design vulnerabilities include improper access controls, insufficient authentication mechanisms, lack of encryption for sensitive data, and improper handling of user inputs.

- `Business Impact`

  Insecure design vulnerabilities within software systems can have profound business implications, including the heightened risk of data breaches, eroded customer trust, financial losses from regulatory fines and legal actions, tarnished reputation leading to reduced customer acquisition and retention, operational disruptions causing downtime and inefficiencies, and a competitive disadvantage as security concerns drive customers towards more secure alternatives. Addressing these vulnerabilities is crucial to safeguarding sensitive data, maintaining business continuity, and preserving the organization's reputation and bottom line.

- `Performing the vulnerability`

`5. Security Misconfiguration`

- `Description`

  Security misconfiguration refers to the improper implementation or settings of security controls within software applications, systems, or networks. It occurs when default configurations, unnecessary features, or weak settings are left unchanged, creating vulnerabilities that attackers can exploit. These misconfigurations can expose sensitive data, grant unauthorized access, and weaken the overall security posture of an organization.

- `Business Impact`

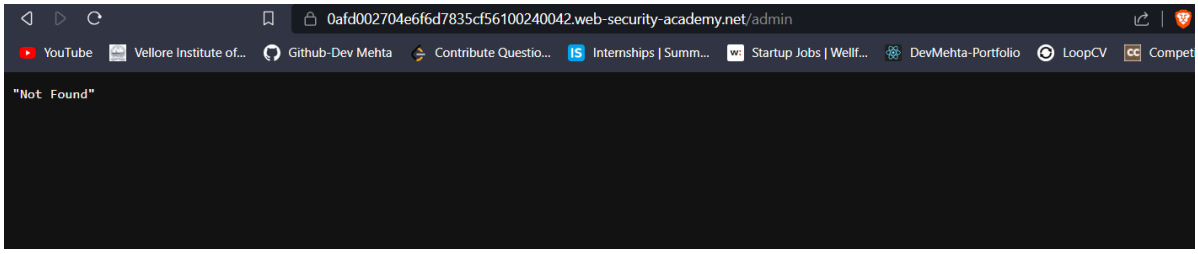
  Security misconfigurations can lead to significant business impact, including unauthorized access to sensitive information, data breaches resulting in legal and financial liabilities, operational disruptions causing downtime and loss of productivity, reputation damage leading to customer loss and diminished trust, regulatory non-compliance leading to fines, and increased resource allocation for incident response and mitigation efforts. Proper configuration management and

regular security assessments are essential to prevent these risks and maintain a strong security stance.

- `Performing the vulnerability`

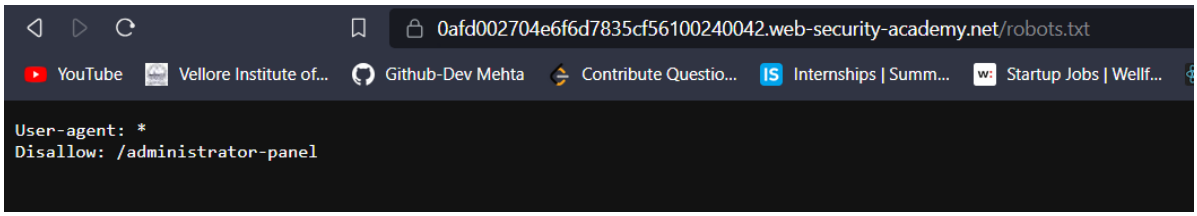
  We try to get to admin panel which is a vulnerability, if the security is correct it wont allow us. We will again use PortSwigger Lab

  Try 1:

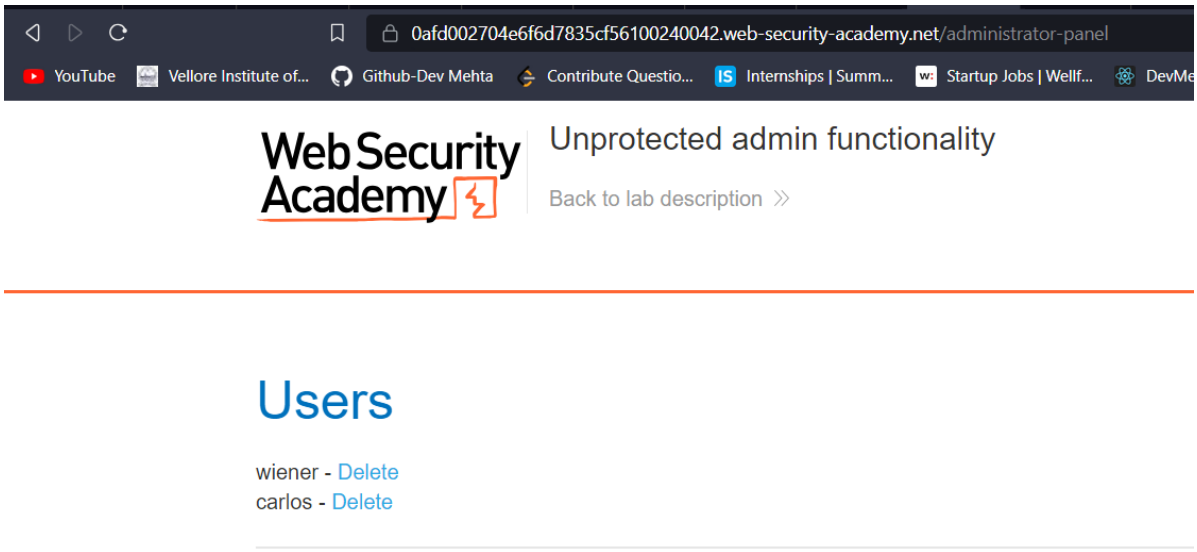


  When I directly add */admin* in link it does not work

  Try 2:



  When I put */robots.txt* it gives the admin panel link



On adding */administrator-panel,* I get the access of admin shows security misconfiguration