**Chapter 11 - Blind SQL Injection – CSRF – Clickjacking**

1. Why is it called a **blind** SQL injection?

It does not show any error message.

1. Why is it **difficult to exploit**?

They only return information when the application is given SQL payloads that return a true or false response. By observing the response, the attacker can extract sensitive information.

1. What are the two types of blind SQL injection?

* Boolean-based
* Time-based

1. What is a SQLI vulnerability when it comes to blind injection?

It is when there is no actual transfer of data.

1. How is an attacker able to reconstruct information (blind sql injection)?

By sending requests and observing the resulting behavior of the DB server.

1. Explain Blind SQL injection using an example.

* Instead of receiving user details, the attacker can figure out if the user is present or not.
* We can use the Boolean-based payload: 1' and 1=1; - -(True🡪present)

1. How do we get information about the version of the SQL database?

1’ AND substring(@@version,2,1) =1; --

1’ AND substring(@@version,2,1) =0; --

I’m basically extracting the second character to check whether the version is 1 or 0.

1. How to find the length?

1’ AND length (database ()) =1; -- (1 character if it’s true)

1’ AND length (database ()) =2; -- (2 characters if it’s true)

1’ AND length (database ()) =3; -- (3 characters if it’s true)

1’ AND length (database ()) = 4; -- (4 characters if it’s true)

1. How do you find the name of the database?

1’ AND substring (database (),1,1) =’a’; --

1’ AND substring (database (),1,1) =’b’; --

1. What is the time-based Blind SQL injection ()?

* The attacker sends SQL queries to the database which forces the database to wait for a specific amount of time before giving a response.
* The response time will indicate whether the query is true or false
* Example: 1' and if((select+@@version) like "10%",sleep(2),null);- -

1. How can you find SQLI vulnerabilities?

|  |  |  |
| --- | --- | --- |
| Black Box Texting | White Box Testing | Automated Application Tools |
| 1. Map the application. 2. Fuzz the application:   Submit SQL-specific characters and look for anomalies.  Look for time delays when submitting different payload queries.  Submit queries that have true or false conditions | Enable web server logging.  Enable database logging.  Map the application.   1. Visible functionality 2. Regex search   Code Review  Test any potential SQLI vulnerabilities. | * Sqlmap * Web Application Vulnerability Scanners like portswigger |

1. How is the web application firewall used?

When there’ s a non-web attack, it’s blocked by the network firewall.

When there’s a web attack, it’s blocked by the web application firewall.

1. What are the two ways in which we can mitigate SQL injection?

It is when use blacklisting or whitelisting for characters that are accepted or denied in the user input fields.

1. What is whitelisting?

Creating a list of approved characters in an efficient method to defend against SQL injection attacks.

Once the list is ready, the application should disallow all requests containing characters outside it.

1. What is blacklisting?

It’s not a recommended way to protect against SQLI attacks because it is prone to failure. It works if the developer makes sure the input field accepts no special characters, other than the characters required.

1. What is the web application firewall?

Specialized form of application firewall

Enables secure application delivery for HTPP-based applications.

Network traffic is inspected at the application layer.

It helps prevent attacks like SQL injection, cross-site scripting, cross-site request forgery, sensitive data exposure, and improper system configuration.

1. What are SQL injection attacks?

Sending SQL commands that trigger the backend SQL to give authorized access, sensitive company data, modify database, and execute administration operation on the database.

1. What is Cross-site scripting?

When a web server fails to validate data coming from another site, compromising the server, and enabling the attacker to obtain sensitive information.

1. What is sensitive data exposure?

When web applications aren’t secured enough. This permits the attackers to exfiltrate personally identifiable information and protected health information.

1. What is cross-site request forgery?

It’s when a fraudulent user uses HTTP request, including the user’s cookie session and sends it to a vulnerable website to steal the victim’s data and hijack their account.

It’s an attack where the attacker causes the victim user to carry out an action unintentionally while the user is authenticated.

1. What are the conditions for Cross Site Request Forgery?

* Relevant action
* Cookie-based session handling
* No unpredictable request parameters

1. What is the Black Box testing perspective?

* Map the application.
* Identify all the application functions that have the 3 conditions
* Create a PoC Script to exploit CSRF: using <img>--🡪get and for POST-🡪form with hidden fields.

1. What is the White Box Testing perspective?

* Identify the framework that’s being used.
* How does the framework protect itself from attacks.
* Review code.
* Review sensitive functionality.

1. What is a Clickjacking attack?

It is when the attacker fools the user into thinking they’re clicking on one thing and they end up clicking on another thing.

1. How can we protect against CSRF attacks?

Use CSRF tokens tied with sessionID to verify the authentication of a user.

1. What are some ways used to protect against attacks?

* Use CAPTCHA
* Use two factor authentication.
* Use Firewall
* Use CSRF Token
* Black Testing/Listing
* White Testing/Listing
* Hashing Passwords
* Disable HTTTP redirections.
* Don’t send responses data back to clients