Section 8 – Bruteforce attacks

1.Cluster bomb brute force

* In this type of attack both username and password field are bruteforced to proceed.
* We use burpsuite interceptor and apply CLUSTER BOMB attack.
* Cluster bomb attack: This type of attack is carried out when there are multiple input fields.
* Flagging results can be used for obtaining a filtered result.

2.Hydra Bwapp Form Bruteforce

* Hydra is an advanced tool to perform bruteforce attacks.
* The syntax for his tool can be tricky.
* Example: hydra 192.168.1.11 http-form-post "<path>"username:^USER^&password:^PASS^ -L users.txt -P pass.txt

3.Hydra SSH Attack

* SSH: Secure shell (Network Protocol)
* Check the website for any open SSH port
* Perform hydra command to access target SSH.

Section 9 - Sensitive Data Exposure

1.Sensitive Data Exposure Example

* This is not a vulnerability. This scenario happens when developer forgets to remove any important information during production that can be used to perform an attack.

Section 10 - Broken access control

1.Broken access control theory

* Broken access control happens when users act outside of their intended permissions.
* The vulnerability is called Insecure direct object reference.

2.Accessing password with BAC

* In this topic passwords are retrieved by using broken access control. Entering extra piece of information where it is not required leads to displaying of extra information.

3.Ticket price IDOR

* This topic covers a vulnerability in which burpsuite interceptor is used to intercept the data and manipulate the data to a favorable outcome for the attacker.

Section 11 - Security Misconfiguration

1.Security Misconfiguration - Default App Credentials

* This scenario happens when admins of the server hasn't changed the credential for a very long time or from the defaults.
* Example: Mainly routers gateway has the default username and password, which is available in the web.

Section 12 - Cross site scripting - XSS

1.XSS Theory

* This is one of the biggest vulnerabilities on a webpage.
* Similar to HTML injection but here JavaScript codes are injected here.
* There are three types of XSS
  + Reflected XSS
    - In this XSS type a malicious script is reflected off of a web application to the victim's browser.
    - This is a non-persistant attack
  + Stored XSS
    - In this XSS type a malicious script is injected directly into a vulnerable web application.
    - This is a persistant attack.
  + DOM XSS
    - In this XSS type a attack payload is executed as a result of modifying the DOM environment (Document object model) in the victim's browser used by the original client side script.

2.Changing Page content with XSS

* Basic test for testing XSS vulnerable sites is to popup an alert using "<script>alert()</script>".
* Script to popup IP address as alert "<script>alert(window.location.hostname)</script>".
* Script to popup document cookies "<script>alert(document.cookies)</script>".
* Every user will be affected by stored XSS attack who uses the vulnerable website.
* Fields like comment section which is XSS vulnerable can be exploited easily.
* Script to change contents of webpage "<script>document.querySelector('#id').textContent='I am hacker'</script>".

3.Bypassing simple filter

* Use inspect element to inspect the page for any information regarding the filter.

4.Downloading a file with XSS vulnerability

* Using the scripts download files from web.

Section 13 - SQL Injection

1.SQL Injection Theory

* SQL injection is a technique used to exploit user data through web page inputs by injecting SQL commands as statements. Basically, these statements can be used to manipulate the application’s web server by malicious users.

2.Guide To Exploiting SQL Injection

* One of the biggest and most serious vulnerabilities

3.Getting entire database, Extracting Password from database, Bypassing filter in SQL query, Blind SQLi

* As injection happens once the database query is taking and unfiltered arguments as user input. Usually those inputs are something like usernames or passwords.
* It could also be something like an input on a page where you search for a specific product.
* We inject SQL syntax inside the query and tick the application to execute our code just because it didn't filter user input.

Query: Select [Element] From [Table] Where [Condition]

* SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.
* We can retrieve hidden data using this method.