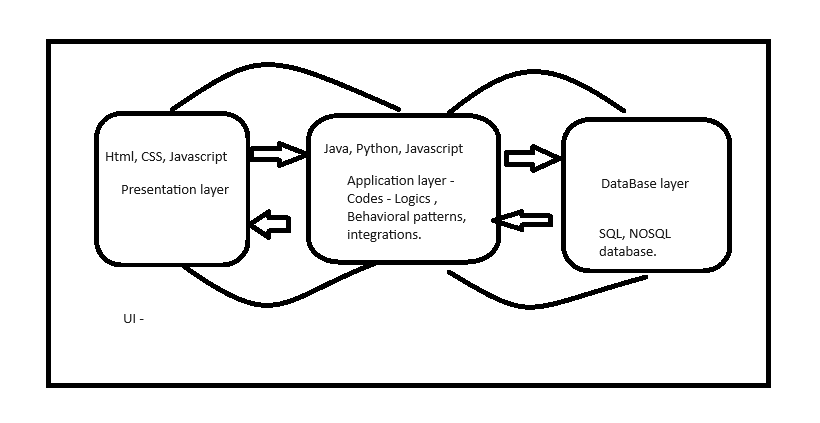
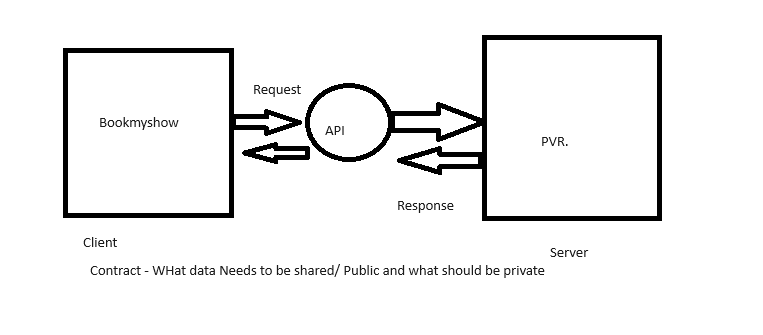
1. **Client Server Architecture**
2. **3 tier architecture.**

****

1. **What is an API**

****

**Network :**

**Http, https protocol.**

**DNS protocol.**

**API - Application programming interface - interface between 2 applications.**

1. **Bookmyshow and PVR.**
2. **Zomato and Google maps.**
3. **Ola/uber with Google Maps.**
4. **Swiggy with Banks.**
5. **Dominos with Maps**
6. **Dominoes with Zomato.**
7. **PhonePe with Banks.**

**Security Testing**

Security testing is an integral part of software testing, which is used to **discover the weaknesses, risks, or threats in the software application** and also help us to stop the nasty attack from the outsiders and make sure the security of our software applications.

The primary objective of security testing is to find all the potential ambiguities and vulnerabilities of the application so that the software does not stop working. If we perform security testing, then it helps us to identify all the possible security threats and also help the programmer to fix those errors.

It is a testing procedure, which is used to define that the data will be safe and also continue the working process of the software.

Examples of Security Breach :

1. Citizens data loss from Government Official Vaccine Website.
2. WazirX Data loss.
3. Hackers stole 2 million Customers data from Dominos.
4. JW Marriott - Customers data lost - how ? - They used login Credentials of employees.

Conclusion - Login, Logout feature is very important from Security point of view.

### 

### **The goal of Security Testing:**

The goal of security testing is to:

* To identify the threats in the system.
* To measure the potential vulnerabilities of the system.
* To help in detecting every possible security risk in the system.
* To help developers fix security problems through coding.

Functional Testing -

Non functional testing - Security tester, Penetration testing

In Depth explanation :

* **Evaluate the system’s ability to withstand an attack**: Security testing evaluates the system’s ability to withstand different types of attacks, such as network attacks, social engineering attacks, and application-level attacks.
* Social Engineering attack - Involves Psychological Manipulation that tricks victims into sharing sensitive data.
* **Phishing** - Mails, Login website - Impersonating login and will compromise your credentials.

They will make it look like it’s okay, but it will be redirecting u to a malicious website and it can attack ur system, it can copy ur credentials, it can make ur banking info to be compromised.

1. UI - will be an imposter
2. Mail imposter.
3. Link imposter.

* **Ensure compliance**: Security testing helps ensure that the system meets relevant security standards and regulations, such as HIPAA, PCI DSS, and SOC2.
* **Provide a comprehensive security assessment:** Security testing provides a comprehensive assessment of the system’s security posture, including the identification of vulnerabilities, the evaluation of the system’s ability to withstand an attack, and compliance with relevant security standards.
* **Help organisations prepare for potential security incidents**: Security testing helps organisations understand the potential risks and vulnerabilities that they face, enabling them to prepare for and respond to potential security incidents.
* **Identify and fix potential security issues before deployment to production:** Security testing helps identify and fix security issues before the system is deployed to production. This helps reduce the risk of a security incident occurring in a production environment

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **Explain the term vulnerability.**

### Vulnerability is a weakness or flaw in an IT system, network, or other resources that can be exploited to gain unauthorised access. Any intruder or bug can quickly attack the system due to its vulnerability. Using the vulnerability, an attacker can run malicious code, set up malware, or steal confidential information.

**Phishing** In most cases, hackers send out fake emails that look as if they're coming from legitimate sources, such as financial institutions, eBay, PayPal -- and even friends and colleagues.

**What are the factors that may cause vulnerabilities?**

The following factors can cause vulnerabilities:

**Design flaws:** System vulnerabilities can allow hackers to attack systems quickly due to design flaws.

Example - Atm MACHINE : AFTER MONEY WITHDRAWAL, U GO HOME,

bUT what if the software has not logged out from ur account?

**Passwords:** Hackers can quickly obtain the information if they know the password.

India

Joker123

money\*\*\*

()!@$@!902@#@01S@$#@$@mike

**Poor and complex coding:** Complex and poorly implemented code can cause vulnerabilities in software systems.

Hardcoding.- anyone sees ur code will see the data. And some data can be confidential.

Ur data should always be separated from the code- Data driven code.

String password;

driver.fe(by.id(“”)).sendkeys(password);

Example - Using Public access specifiers instead of private.

**Management:** Inadequate data management can lead to many vulnerabilities.

**Human Error:** Human errors are the most common vulnerabilities.

Leaving confidential info on ur desktop without security, desk,

opening personal emails in our client machines.

## 

## 

## 

Performance issue

Example - Government sites- Competitive exams Results Day.

## **Distributed denial-of-service attacks**

In a distributed denial-of-service (DDoS) attack, multiple compromised machines attack a target, such as a server, website or other network resource, making the target totally inoperable.

The flood of connection requests, incoming messages or malformed packets forces the target system to slow down or to crash and shut down, denying service to legitimate users or systems.

### **How to prevent DDoS attacks(Or any attacks )**

To help prevent DDoS attacks, companies should take the following steps:

* Implement technology and tools to monitor networks visually and know how much bandwidth a site uses on average. DDoS attacks offer visual clues so administrators who understand the normal behaviours of their networks will be better able to catch these attacks.
* Ensure servers have the capacity to handle heavy traffic spikes and the necessary mitigation tools necessary to address security problems.
* Update and patch firewalls and network security programs.
* Set up protocols outlining the steps to take in the event of a DDoS attack occurring.

**Principle of Security Testing:**

Below are the six basic principles of security testing:

* **Confidentiality** - It is a security process that prevents the leak of the data from the outsider's because it is the only way where we can ensure the security of our data.
* Encryption -

Example - Bank details when making a transaction/ or at any other time, especially PIN, passwords.

* **Integrity** - In this, we will secure those data which have been changed by the unofficial person. The primary objective of integrity is to permit the receiver to control the data that is given by the system.

KYC confirmation from Organization (bank)

Approving loans, approving House loans, car loans,

Employee- timesheet approval, it should be done by the manager.

* **Authentication** - The authentication process comprises confirming the **individuality of a person**, tracing the source of a product that is necessary to allow access to the private information or the system.

Example - Login - username and password.

What is 2FA?

Username and password

Otp - mail/ phone.

Research on Security measures taken by different organizations :

1. Github - Shaik, Istiyak
2. Paypal - Santosh, Vikas
3. Google pay - ANkita, Prashant.
4. Amazon - Shivani, Pratik.
5. Snapchat - Jaydev, Sagar.

Presentation - Pointers on how an organisation has implemented security in their org in terms of authentication, authorization, encryption, etc.

* **Authorization** - It is the process of defining that a client is permitted to perform an action and also receive the services. The example of authorization is Access control.
* I am having authorization to edit the document, and trainees are not authorised to edit this document.
* Trainees only have authorization to view only access.
* **Availability** - In this, the data must be retained by an official person, and they also guarantee that the data and statement services will be ready to use whenever we need it.

How easily i can get information about ICICI bank customer details?

* **Non-repudiation** - It is used as a reference to digital security, and it a way of assurance that the sender of a message cannot disagree with having sent the message and that the recipient cannot repudiate having received the message.
* Examples - phonepe, Facebook messenger, slack, gmail.

Transaction - of money, messages, chat windows,

* It combines authentication and integrity: non-repudiation authenticates the identity of a user who performs a transaction, and ensures the integrity of that transaction.

### 

### 

### **Types of Security Testing:**

1. **Vulnerability Scanning:** Vulnerability scanning is performed with the help of automated software to scan a system to detect known vulnerability patterns.

Tool name - OWASP ZAP - Scanning, Burp Suite

1. **Security Scanning:** Security scanning is the identification of network and system weaknesses. Later on, it provides solutions for reducing these defects or risks. Security scanning can be carried out in both manual and automated ways.
2. **Penetration Testing:** Penetration testing is the **simulation** of the attack from a malicious hacker. It includes an analysis of a particular system to examine for potential vulnerabilities from a malicious hacker who attempts to hack the system.

Password - iot1234!

Key - 2

Cryptographed password - kqv345#

1. **Risk Assessment:** In risk assessment testing security risks observed in the organisation are analysed. Risks are classified into three categories i.e., low, medium, and high. This testing endorses controls and measures to minimise the risk.
2. **Security Auditing:** Security auditing is an internal inspection of applications and operating systems for security defects. An audit can also be carried out via line-by-line checking of code.
3. **Ethical Hacking:** Ethical hacking is different from malicious hacking. The purpose of ethical hacking is to expose security flaws in the organisation’s system.
4. **Posture Assessment:** It combines security scanning, ethical hacking, and risk assessments to provide an overall security posture of an application.
5. **Network security testing**: Network security testing is a type of testing that focuses on identifying vulnerabilities in the network infrastructure. It includes testing firewalls, routers, and other network devices to identify potential vulnerabilities.
6. **API security Testing** - Using postman we can do Security testing, by testing the authentication, authorization of the API’s.

APi will also have tokens(passwords).

WHen u test ur API’s with tokens , then ur testing the authentication.

1. Make sure that all the 3 layers are secured **individually** and in **Integration**.(Component Testing, Integration Testing wrt Security)

Investigate the Security of the application and explain how security is approached in a particular app.

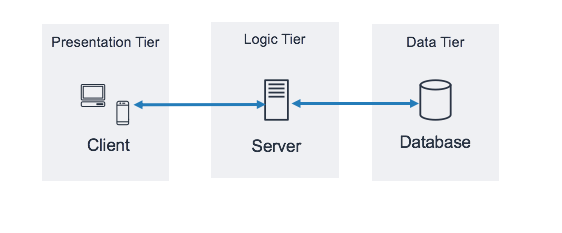
1.Github - Anjali , Khalid, shibham gaikwad, Jani , Kanishk

2.Facebook - Alekya, ABhishek baral, Prince, Rajit , Sakshi

3.Instagram - Ritick, Tanfeez, Vaishnavi, Goutam,Kavita.

4. PhonePe - Shivam, Rajesh, Sharmila, Karan, Rohan

5. Gmail - Hirak, Annie, Shubham srivastav, Ragini,



**Information gathering.(Tools and websites - Wayback machine)**

**About WayBack Machine :**

Here we can check the history of any website, we can get to know how the website works at previous versions and we can track.

The previous versions are not supposed to work, sometimes developers forget to close

The endpoints , so hackers can exploit the endpoints.

**SQL Injection**

SQL injection is a type of cyber attack that targets the SQL (Structured Query Language) databases used in many web applications. It occurs when an attacker is able to manipulate an application's input in such a way that it is interpreted as SQL commands. This can happen when input from a user is not properly validated or sanitised before being used in an SQL query.

This can allow the attacker to execute arbitrary SQL queries, read sensitive data from the database, modify database data, and even delete data

Example :

Suppose you have a simple web application that allows users to log in using a username and password. The application uses the following SQL query to authenticate users:

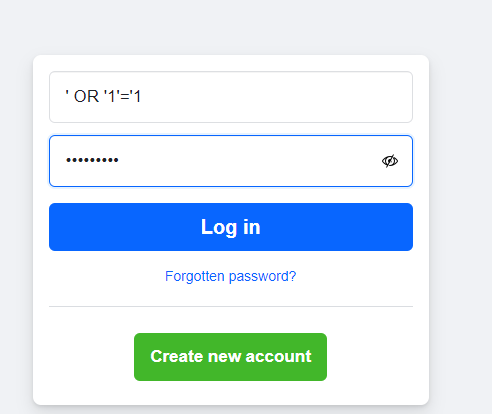
SELECT \* FROM users WHERE username = 'akshay’' AND password = ‘password’

The input\_username and input\_password variables are populated with the values entered by the user in the login form. However, the application does not properly validate or sanitize these inputs, making it vulnerable to SQL injection.

An attacker can exploit this vulnerability by entering a specially crafted input in the username field, such as:

' OR '1'='1

When this input is inserted into the SQL query, it modifies the query to become:



For Normal login , the following query will be executed :

SELECT \* FROM users WHERE username = 'akshay’' AND password = ‘password

For Cyber attack login , the query manipulation will be done in the following way :

SELECT \* FROM users WHERE username = '' OR '1'='1' AND password = 'input\_password'

In this modified query, the OR '1'='1' condition always evaluates to true, effectively bypassing the password check. As a result, the query returns the first user from the users table, granting unauthorised access to the application.

**SQL Injection** - Injecting a SQL syntax inside the text fields to manipulate the query and gain unauthorised access is called SQL injection.

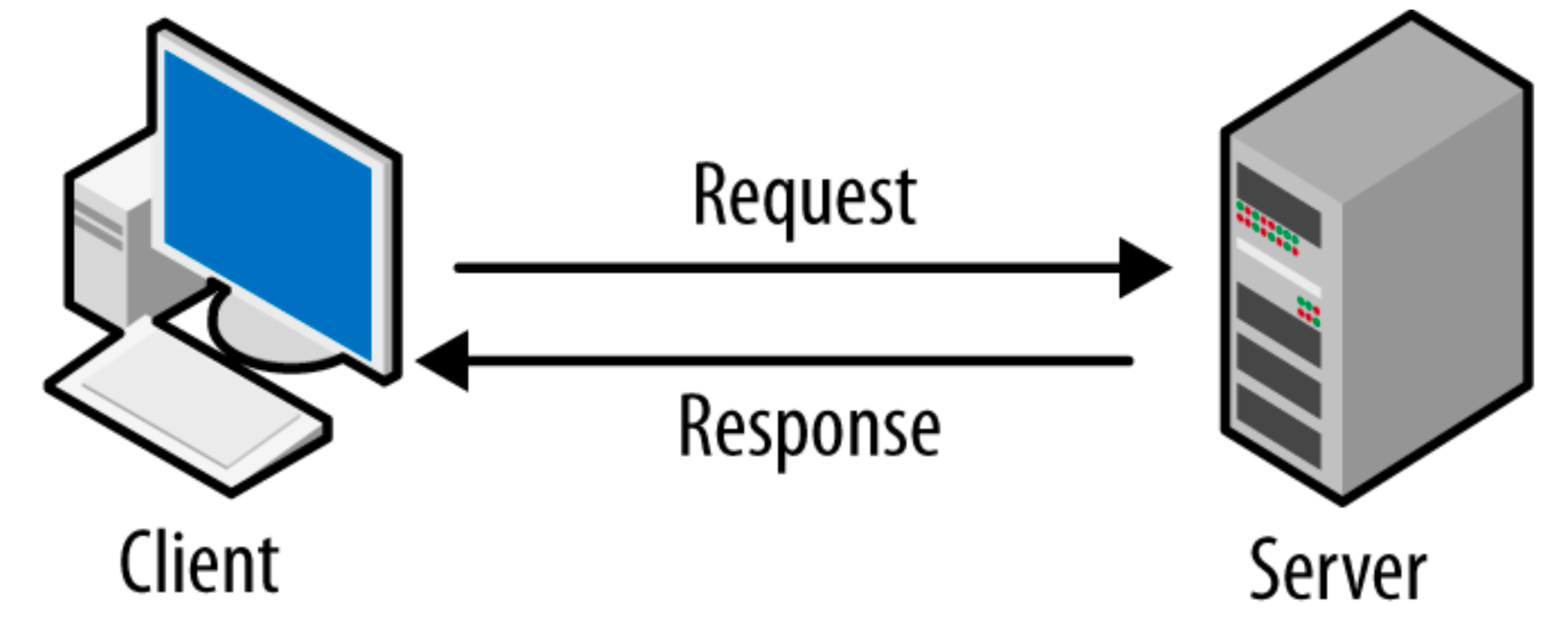
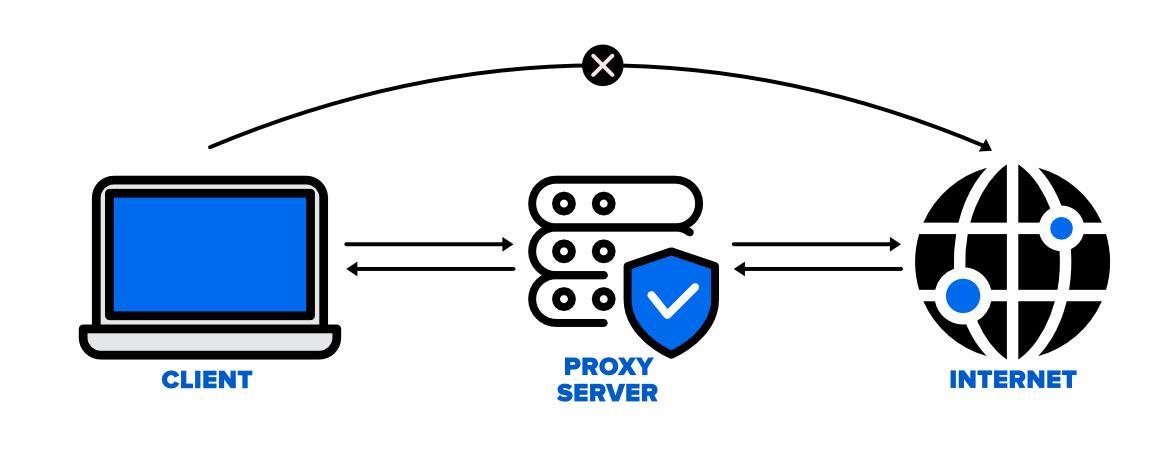
**Cross-Site Scripting(XSS)**

XSS, or Cross-Site Scripting, is a type of security vulnerability commonly found in web applications. It occurs when a web application allows users to input data that is then **displayed on the website without proper validation or escaping**. Attackers can exploit this vulnerability by **injecting malicious scripts** (such as JavaScript) into the web application, which are then executed in the context of the victim's browser.

XSS vulnerabilities can be used by attackers to steal sensitive information, such as cookies or session tokens, impersonate users, deface websites, redirect users to malicious websites, or perform other malicious actions.

1. Information Gathering. - Tools and websites - Wayback machine.

Terms Related to Security testing :

1. Client and Server
2. **A proxy server** acts as an intermediary between your device and the internet. When you use a proxy server, your internet traffic is routed through the proxy server before reaching its destination. This means that the website or service you are accessing sees the IP address of the proxy server instead of your own IP address.
3. 

Scenario Based Questions :

* Cross-Site Scripting (XSS):
  + Scenario: Imagine you are testing a web application form that accepts user input and displays it on the webpage. How would you test for XSS vulnerabilities in this form?
  + Expected Answer: Look for places where user input is displayed without proper validation or encoding. Try entering HTML/JavaScript code in the input fields to see if it gets executed on the webpage.
* SQL Injection:
  + Scenario: You are testing a login form that uses SQL queries to authenticate users. How would you test for SQL injection vulnerabilities in this form?
  + Expected Answer: Try entering SQL commands in the username or password fields to see if the application responds unexpectedly or gives access without the correct credentials.
* Sensitive Data Exposure:
  + Scenario: A web application stores user passwords in plaintext in its database. How would you test for sensitive data exposure in this application?
  + Expected Answer: Look for places where passwords or other sensitive information is stored or transmitted without encryption. Check if the application uses secure protocols like HTTPS.
* Broken Authentication:
  + Scenario: You are testing a web application that allows users to log in using a username and password. How would you test for broken authentication vulnerabilities?
  + Expected Answer: Try various login scenarios, such as using a blank password, entering a username that does not exist, or entering a password that is too short. Check if the application handles these cases securely.
  + We can also recommend to add 2AF, End - End Encryption.
* Security Misconfiguration:
  + Scenario: You are testing a web application that is configured to display detailed error messages in production. How would you test for security misconfiguration in this application?
  + Expected Answer: Look for places where sensitive information, such as stack traces or database connection strings, is exposed to users. Check if the application uses default passwords or settings.
* Clickjacking :
  + Scenario: A web application allows users to perform sensitive actions, such as transferring money, by clicking on a button. How would you test for clickjacking vulnerabilities in this application?
  + Expected Answer: Try embedding the application in an iframe on a malicious website and overlaying transparent elements on top of the buttons. Check if the buttons can still be clicked

Hands on Activity :

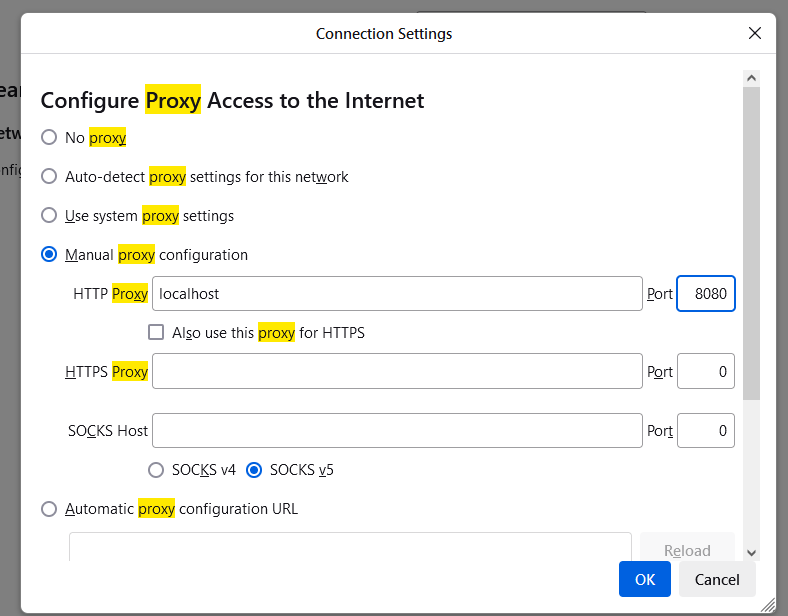
BurpSuite :

1. Proxy Interceptor

Steps :

1. Go to Firefox browser and set the proxy to local host 8080.

Settings -> Network Settings - > Proxy -> Add manual proxy.



1. Download Certificate from : <http://burpsuite>
2. Go to: https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-excessive-trust-in-client-side-controls
3. Register account and login to application.
4. Again go to the Portswagger mentioned link and click on Access the Lab.
5. We will get a e commerce app, click on my account.
6. There we have to login using credentials , username - wiener , password is - peter.
7. Update ur email id.
8. go to home and click on View details ,
9. Go to BurpSuite , go to proxy and click on Intercepter On.
10. Now go back to application and then click on Add to cart.
11. The Request will be sent to the Intercepter first and then to the server.

Proxy Interceptor using burp suite browser :

Open Burp Suite :

1. Open the browser from burpsuite, Open the portswagger link, register user and login to application.
2. Before add to cart click on interceptor on and then add to cart.
3. You will get a post request.
4. Make changes in the request and then click on forward.
5. Then interceptor off.
6. The changes will be reflected in the UI.

A brute force attack is a type of cyber attack where an attacker tries to gain unauthorised access to a system or account by systematically trying all possible combinations of usernames, passwords, or encryption keys until the correct one is found.

200 usernames ,200 password

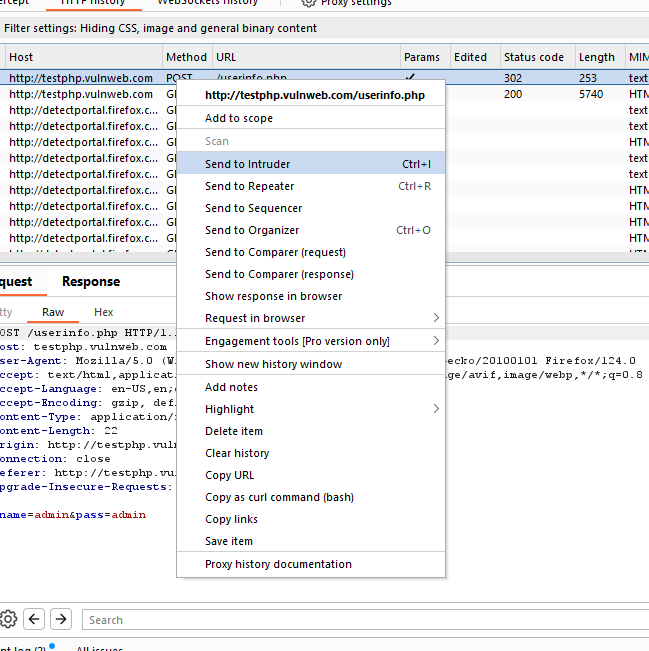
Brute force attack using BurpSuite : :

1. Set up Proxy in 127.0.0.1:8080 and go to a login page :

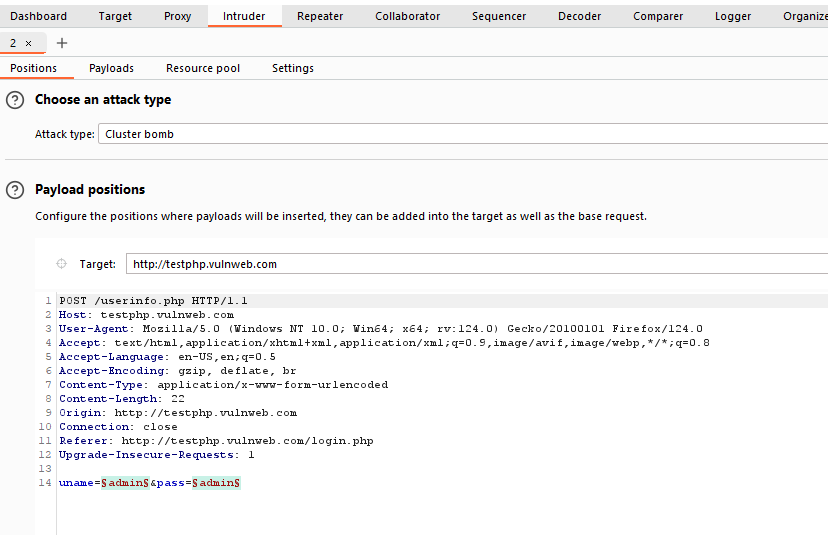
First without any proxy :

[.vulner http://testphpb.com/login.php](http://testphp.vulnweb.com/login.php)

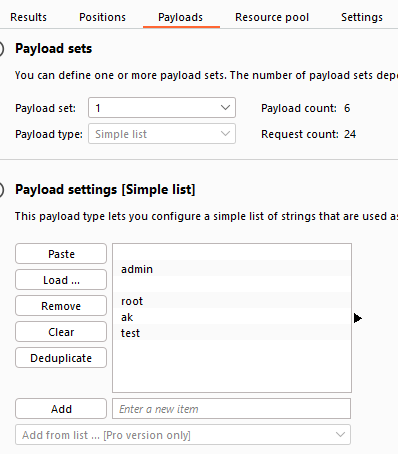
1. Then Open BurpSuite and make sure that , there also proxy is configured to the same localhost.
2. Make sure the website is visible in the http history of burpsuite.
3. Add the website to intruder by right click on URl as shown below :

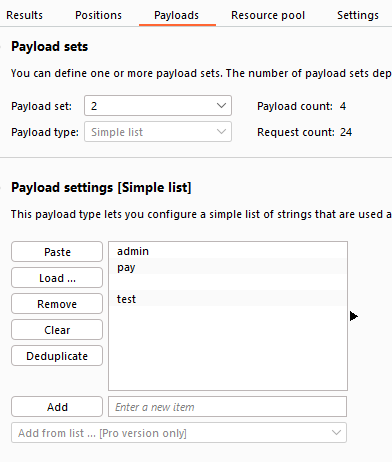


1. Enter details of username and password and click on Login.
2. All the login details are visible as a post request in Burp Suite.
3. Go to the Intruder section, change the attack type to cluster bomb.



9) Add payloads of username and password as shown :





10 ) Then click On Start attack.

How to do security testing for a static application

* 1.Code Review: Perform a thorough review of the application's source code to identify potential security vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure direct object references.
* 2. Security Headers: Ensure that the application's HTTP headers include security headers such as Content Security Policy (CSP), Strict-Transport-Security (HSTS), and X-Frame-Options to protect against common attacks.
* 3. File and Directory Permissions: Ensure that the application's files and directories have the correct permissions set to prevent unauthorised access.
* 4. Secure Communication: Ensure that the application uses secure communication protocols such as HTTPS to protect data transmitted between the client and server.
* 5.Authentication and Authorization: Ensure that the application implements strong authentication mechanisms and properly enforces authorization rules to prevent unauthorized access to sensitive resources.
* 6. Input Validation: Ensure that the application validates and sanitizes user input to prevent common vulnerabilities such as SQL injection and XSS.
* 7.Error Handling: Ensure that the application handles errors gracefully and does not reveal sensitive information to users.
* 8.Third-party Libraries: Ensure that third-party libraries used by the application are up to date and do not have known security vulnerabilities.
* Example - Log4j
* Security Testing Tools: Use security testing tools such as Burp Suite, OWASP ZAP, and Nessus to perform security scans and identify vulnerabilities in the application

How to do Security Testing for Dynamic Websites

Here are some steps to perform security testing for a dynamic application:

* Identify Security Requirements: Understand the security requirements specific to your application, including regulatory compliance (e.g., GDPR, HIPAA), authentication requirements, and data encryption standards.
* Threat Modeling: Create a threat model to identify potential security threats and vulnerabilities in your application. This helps prioritise security testing efforts.
* Perform Security Testing: Conduct various types of security testing, including:
  + Vulnerability Assessment: Use automated tools like OWASP ZAP, Nessus, or Qualys to scan for common vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure server configurations.
  + Penetration Testing: Perform manual testing to simulate real-world attacks on your application. This helps identify vulnerabilities that automated tools might miss.
  + Authentication and Authorization Testing: Verify that the application properly authenticates users and enforces access controls to protect sensitive data.
  + Data Security Testing: Ensure that sensitive data is encrypted both at rest and in transit. Test for data leakage vulnerabilities.
  + Session Management Testing: Check for vulnerabilities related to session management, such as session fixation and session hijacking.
* API Security Testing: If your application uses APIs, perform security testing to ensure that APIs are secure and protected against common vulnerabilities.
* Security Configuration Testing: Review the security configuration of your web servers, databases, and other components to ensure they are properly configured to mitigate common security risks.
* Regular Security Audits: Conduct regular security audits to identify and address new security vulnerabilities that may arise as the application evolves.
* **Security Training:** Provide security training to developers, testers, and other stakeholders to increase awareness of security best practices and common vulnerabilities.
* **Incident Response Planning**: Develop an incident response plan to quickly address security breaches if they occur and minimize their impact

1. Install Kali Linux OS.

Step 1 :

Download Kali Linux :

https://www.kali.org/get-kali/#kali-installer-images

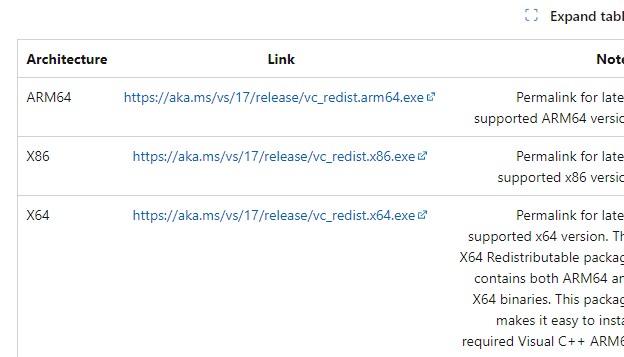
Download Virtualbox :

<https://www.virtualbox.org/wiki/Downloads>

To install Virtual box we need C++ 2019:

Download C++ from link below :

<https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170>



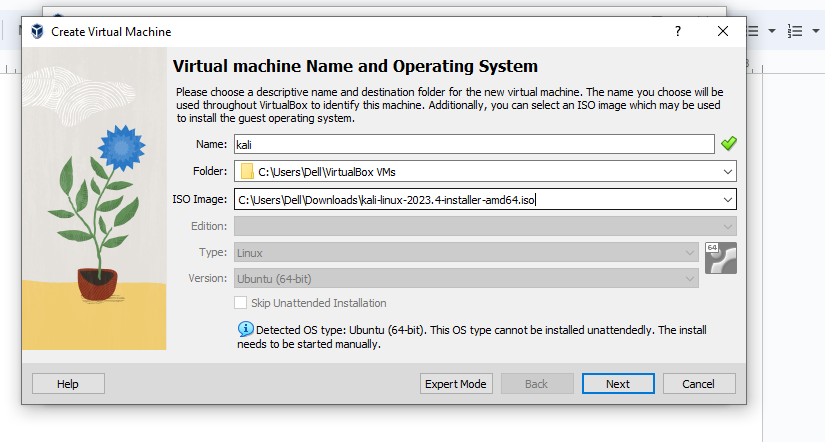
Next :

We need to set up Kali Linux in Virtual Box.

Step 1 : Open Virtual Box.

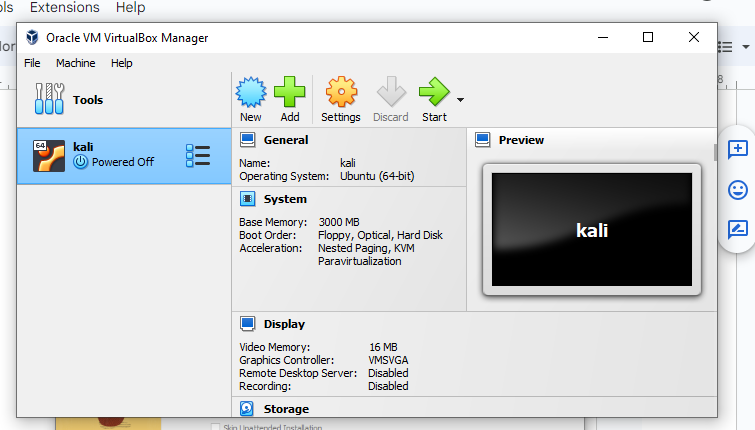
Step 2 : Click on New .

Step 3 : We need to provide the location of Kali Disk image in **ISO Image**



We need to allocate RAM and ROM to the virtual container.

After finishing ,we should get this screen. :



Next Step :

We need to click on Start, then Kali Linux will be opened.

Then we need to install Kali Linux with Graphical install option,

Then we have to setup hostname, domain should be blank, username and password.

Password is important, we need to remember this .

**After installing Kali Linix , we have to download and install OWASP ZAP in Kali Linux.**

**Explain about various tools available in Kali Linux in Top left corner (Dragon symbol)**

**Tools related to Information gathering, Vulnerability analysis.**

**Install ZAP :**

**OWASP ZAP download link :**

<https://www.zaproxy.org/download/>

AFter downloading ZAP :

We need to update the OS

Open shell command prompt :

Sudo apt-get update

Step 1 :

AFter that , open terminal from the ZAP downloaded location

In command prompt :

Chmod 777 <filename>

Now we have changed the file to executable file .

Now we need to change the user access to super user and then execute the command :

Sudo ./<filename>

Now we need to open ZAP, found in dragon icon applications.

Just click on ZAP icon.

ZAP Server will be started.

SO ZAP acts as an intermediate between your KaiI and Browser.

So to make browser trust the ZAP we add certificate so that the browser can recognize this tool.

How to add?

Tools - > Options -> Dynamic SSL certificate -> New screen - > Save the certificate.

In your Browser go to three lines search for SSL certificate and add import the certificate.

**Now we have to connect the browser with the ZAP tool.**

We need to add Foxy Proxy extension in our browser,

Go to Foxy proxy(Firefox Extension) official website, and just click on Add to firefox.

Now in the Extensions tab , we can see Foxy proxy extension.

In this Foxy proxy extension, we need to add ZAP

We need to click on Options section in Foxy proxy ,

In a new tab the Foxy proxy will be loaded.

In that we need to select Proxies in the second tab and click on Add .

We need to add ZAP to this Foxy proxy,

After clicking on add we will gte various options :

Just add title as ZAP, hostname as localhost, port as 8080 and click on save.

After this we can browse on firefox, everything will be recorded in ZAP,

All the Requests, responses , alerts, vulnerabilities will be displayed in ZAP tool.

**Scanners in ZAP :**

First we have to enter the website in the Firefox.

To Start Active Scan -

Go to Tools → Active Scan → Sites —> Mention the Site that u want to scan. – > Select a request → Then start Active Scan

This will start the Active scan process and all the details are displayed in the dashboard.

Here in Alerts we can see the Vulnerability alerts which can be reported to Security Analysts.

**AJAX SPIDER :**

**First load the website in Firefox(Prerequisite)**

**To start the AJAX SPIDER :**

Go to Tools → AJAX SPIDER → Select-> Select the website and the request →

Check box advanced options → In advanced options we will get the max duration , crawl depth, crawl states , no of browser to be opened etc. and then click on

Start SCAN.

Now the spider scanning will be displayed in the Dashboard.

Here we can see all the URLS, links that are present in the website, backend, APIS

And we can see all the information related to links which can be Vulnerable.

**NMAP :**

**Nmap is a command prompt tool. It doesn't have a UI.**

**So all the commands in the NMAP help us to attack, make a system vulnerable.**

**How to start NMAP?**

**Go to Terminal and give command**

**Nmap <IPaddress/URL>**

**Example -**

**1)Command to connect to a particular IP**

**Nmap 192.190.23.12**

**Sudo Nmap 192.190.23.12 (recommended command )**

**2)This is used to Scan the IP and shows whether we can connect to a particular IP.**

**Sudo Nmap -Pn -sP <IPAddress>**

**3)Command to Ping a particular IP**

**Ping <Ipaddress>**

**Press control c to stop the ping**

4) Command to show port vulnerabilities for a particular website :

**Sudo nmap www.moolyaed.com**

This will display us all the ports, service and status.

Through this information the attacker can determine which port is not secured.

5) Command to stop the service for a particular website/IP for a few milliseconds :

**nmap <IPaddress/Website> -max-parallelism 800 -Pn --script http-slowloris --script-args http-slowloris.runforever=true**

**Note : Press control z to stop the nmap attack.**

**6) Command to list the top ports in a website :**

Sudo nmap --top-ports 10 [www.moolyaed.com](http://www.moolyaed.com)

This will list down all the top 10 ports for the website, here the attacker gets the details of all the secured and unsecured ports.

**7) Command to list the Vulnerability scan of a website -**

nmap -Pn --script vuln [www.moolyaed.com](http://www.moolyaed.com/)