TASK-4

AI FOR CYBER SECURITY

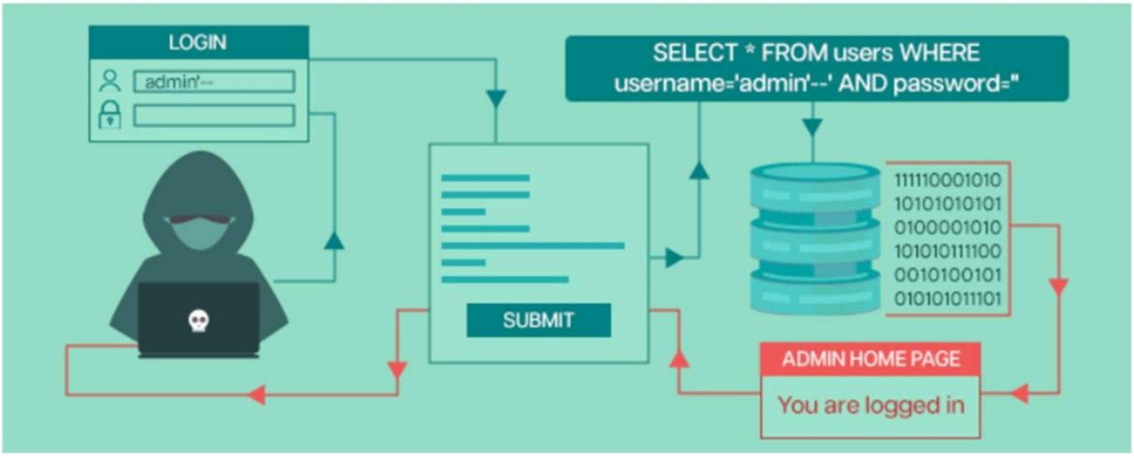
P. MONISH

21BCE9517

28-08-2023

Understanding top 10 web Application attacks

1.SQL Injection

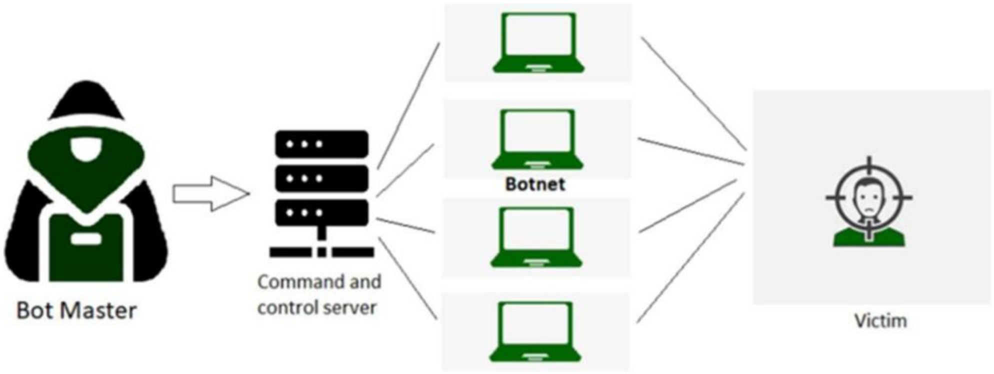


- In SQL injection attacks, malicious SQL queries are injected into input fields or URLs, taking advantage of poorly sanitized user inputs.
- If successful, attackers can manipulate the database, steal data, or even gain unauthorized access to the server.
- .Developed in the 1970s, Structured Query Language (SQL) is a language for accessing and manipulating data from the database. An application can communicate with the database using SQL statements.
- With the use of SQL statements, the application can perform some standard SQL commands such as “SELECT,” “UPDATE,” “INSERT,” “DELETE,” “CREATE,” and “DROP.”

Attackers use the input fields in web applications to run arbitrary queries (injection) on the server. Hence, the attack process is called SQL Injection or SQLi attack.

They gain access to information that is not intended to be displayed. These injection attacks are categorized as ‘high impact severity’ by OWASP Top 10.

2. DENIAL-OF-SERVICE (DOS) / DISTRIBUTED DENIAL-OFSERVICE



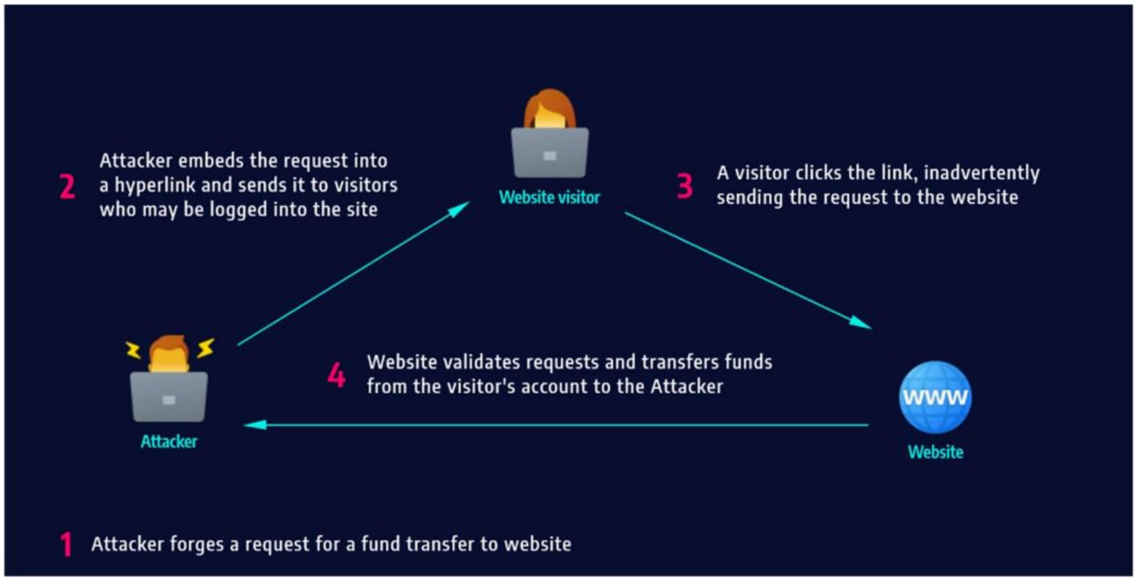
- Distributed Denial of Service (DDoS) is a type of DOS attack where multiple systems, which are trojan infected, target a particular system which causes a DoS attack.
- A DDoS attack uses multiple servers and Internet connections to flood the targeted resource.
- A DDoS attack is one of the most powerful weapons on the cyber platform.
- When you come to know about a website being brought down, it generally means it has become a victim of a DDoS attack. This means that the hackers have attacked your website or PC by imposing heavy traffic.
- Thus, crashing the website or computer due to overloading.

.

3.Cross Site Request Forgery (CSRF)

- Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they’re currently authenticated.
- With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker’s choosing.

- If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth.
- If the victim is an administrative account, CSRF can compromise the entire web application

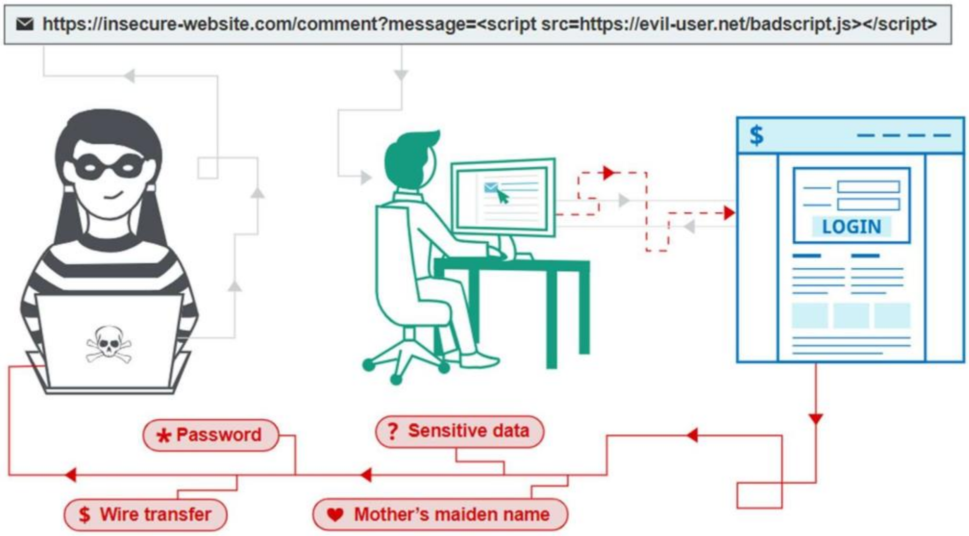

2 Attacker embeds the request into a hyperlink and sends it to visitors who may be logged into the site
Website visitor
3 A visitor clicks the link, inadvertently sending the request to the website
4 Website validates requests and transfers funds from the visitor's account to the Attacker
Attacker
WWW
Website
1 Attacker forges a request for a fund transfer to website

- This blog post introduces our newest addition to our pentestion arsenal, the ssh-putty-brute.ps1.
- This tool can turn the well-known PuTTY SSH client (putty.exe or plink.exe) into a reliable SSH login brute force tool which in addition also evades any Antivirus or endpoint protection solution.

4 CROSS SITE SCRIPTING (XSS):

- This type of attack is more likely to target websites with scripting flaws. The injection of malicious code into web applications is known as Cross-Site Scripting.

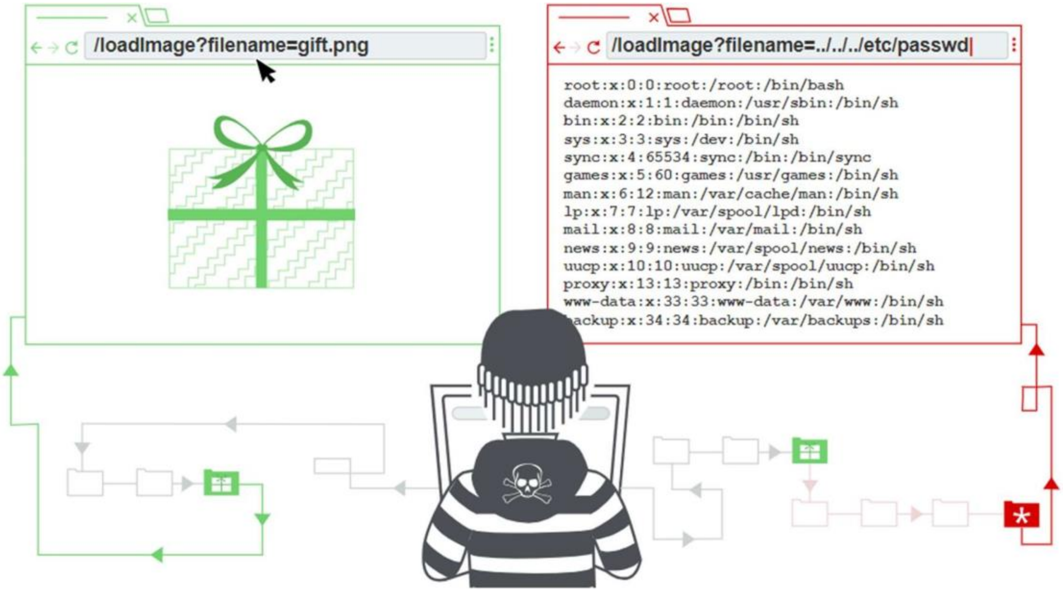
- The script will give the hacker access to web app data such as sessions, cookies, and so on.


https://insecure-website.com/comment?message=<script src=https://evil-user.net/badscript.js></script>
$
LOGIN
Password
Sensitive data
Wire transfer
Mother's maiden name

- Cross-site scripting works by manipulating a vulnerable web site so that it returns malicious JavaScript to users.
- When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.

5 DIRECTORY TRAVERSAL:

- Directory Traversal Attack is usually effective on older servers with vulnerabilities and misconfiguration.
- The root directory is where web pages are stored, however, in this attack, the hacker is after directories outside of the root directory.


/loadImage?filename=gift.png
/loadImage?filename=../../../etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

Path traversal is also known as directory traversal. These vulnerabilities enable an attacker to read arbitrary files on the server that is running an application. This might include:

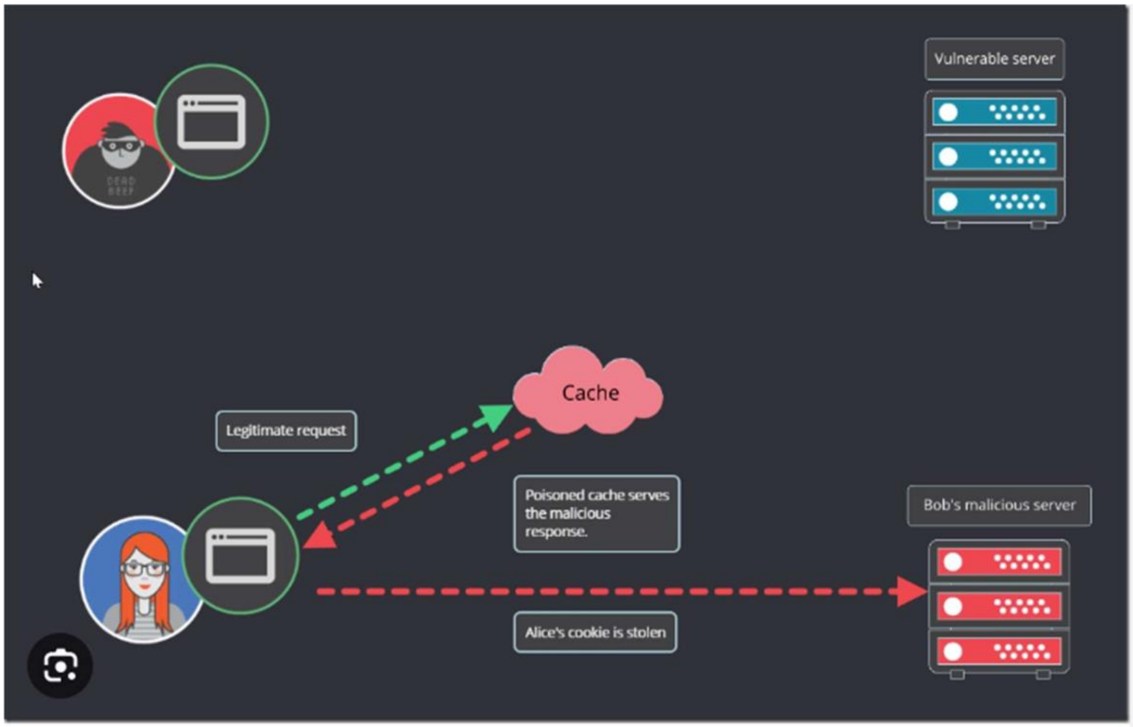
- Application code and data.
- Credentials for back-end systems.
- Sensitive operating system files.

In some cases, an attacker might be able to write to arbitrary files on the server, allowing them to modify application data or behavior, and ultimately take full control of the server.

6 HTTP RESPONSE SPLITTING ATTACK:

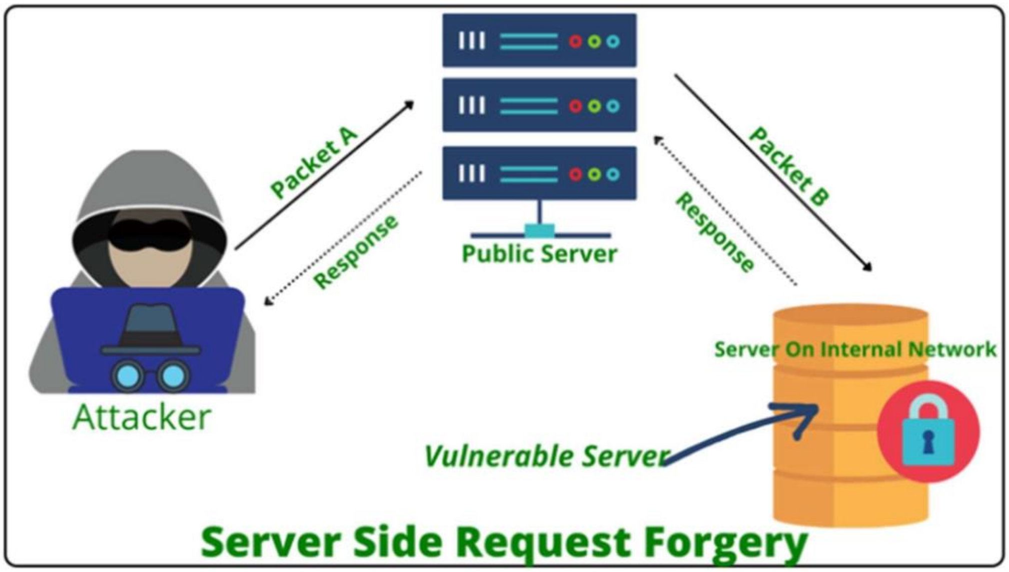
- HTTP Response Splitting is a protocol manipulation attack, similar to Parameter Tampering.

- Only programs that use HTTP to exchange data are vulnerable to this attack. Because the entry point is in the user viewable data,
- it works just as well with HTTPS. The attack can be carried out in a variety of ways.



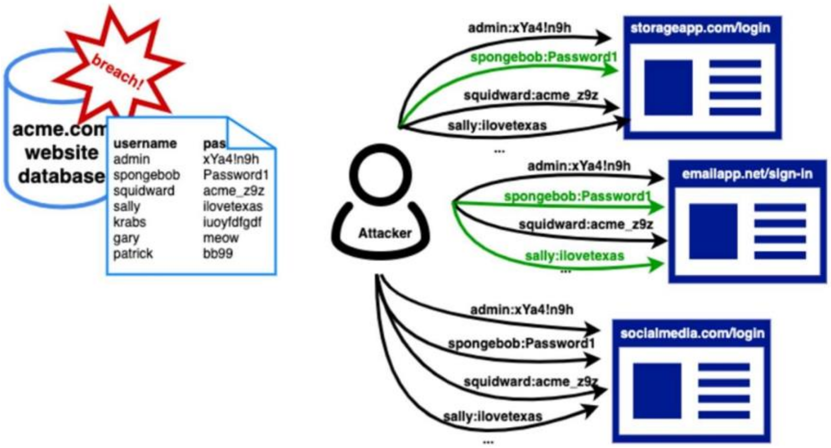
7. Server-Side Request Forgery (SSRF):

- In a Server-Side Request Forgery (SSRF) attack, the attacker can abuse functionality on the server to read or update internal resources.
- The attacker can supply or modify a URL which the code running on the server will read or submit data to, and by carefully selecting the URLs,
- the attacker may be able to read server configuration such as AWS metadata, connect to internal services like http enabled databases or perform post requests towards internal services which are not intended to be exposed.



8.**Brute Force and Credential Stuffing**:

- Credential stuffing is the automated injection of stolen username and password pairs (“credentials”) in to website login forms, in order to fraudulently gain access to user accounts.
- Since many users will re-use the same password and username/email, when those credentials are exposed (by a database breach or phishing attack, for example)
- submitting those sets of stolen credentials into dozens or hundreds of other sites can allow an attacker to compromise those accounts too.
- Credential Stuffing is a subset of the brute force attack category. Brute forcing will attempt to try multiple passwords against one or
- multiple accounts; guessing a password, in other words. Credential Stuffing typically refers to specifically using known (breached) username / password pairs against other websites.

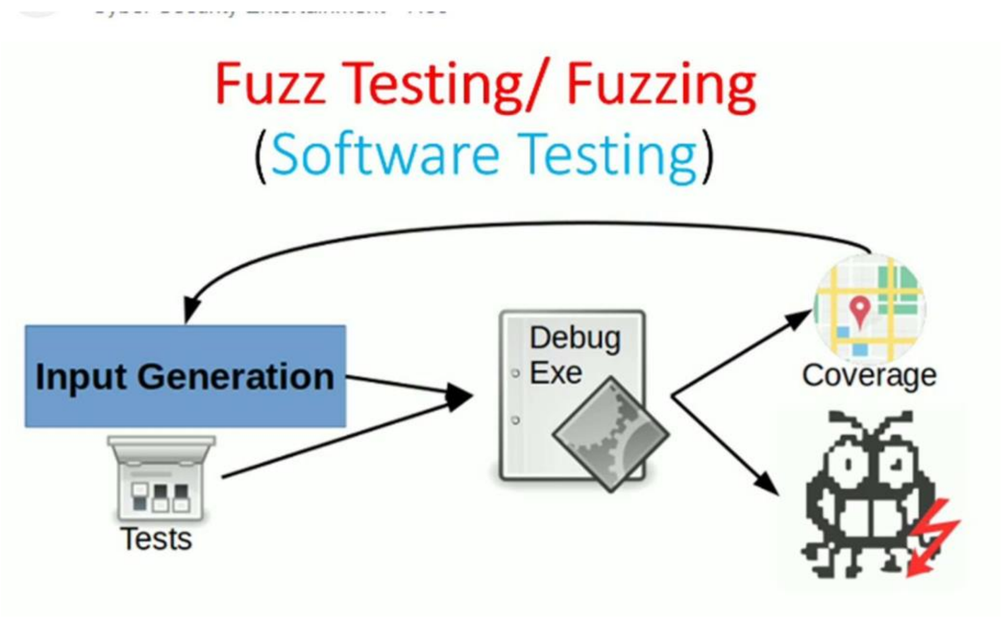


Diagram

username	pas
admin	xYa4!n9h
spongebob	Password1
squidward	acme_z9z
sally	ilovetexas
krabs	iuoyfdfgdf
gary	meow
patrick	bb99

9. Fuzz Testing (Fuzzing)

- Fuzz testing, also known as fuzzing, is a technique used to discover vulnerabilities in a web application by sending it random or invalid input data.
- The goal of fuzz testing is to identify how the web application responds to different inputs and to find errors and crashes.
- Fuzz testing can be performed manually or with the help of automated tools.
- Fuzz testing can uncover vulnerabilities that may not be detected by other security testing methods such as penetration testing.
- To perform effective fuzz testing, a tester needs to understand the web application’s input and output mechanisms and the types of data that the application processes.



10. Command Injection

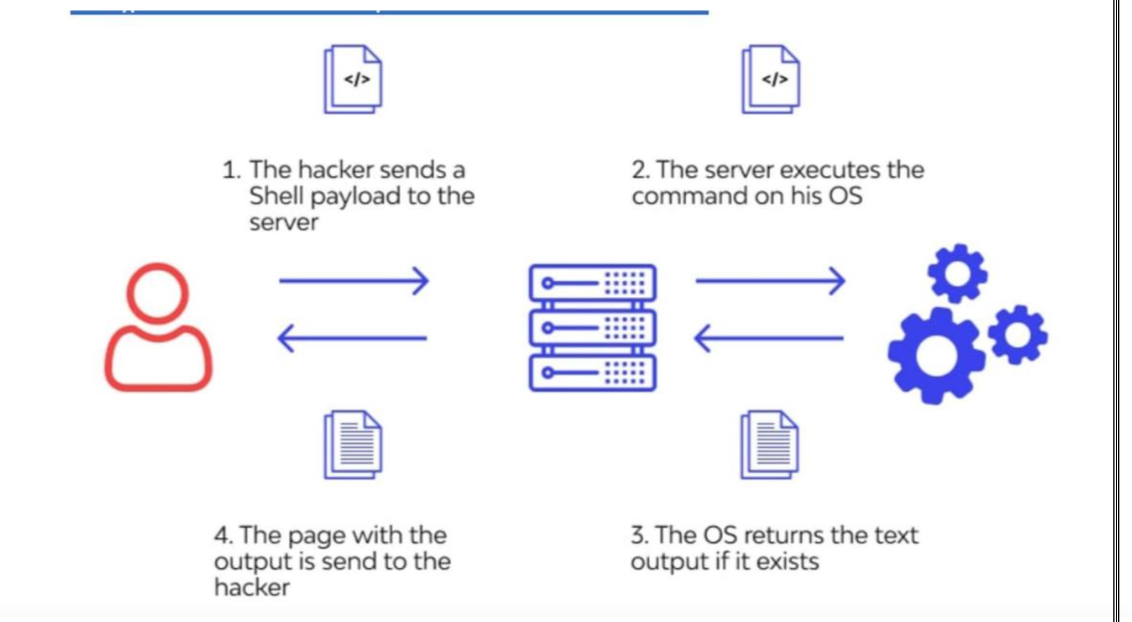
- Command injection attacks occur when attackers inject malicious commands into input fields or parameters that are executed by the server's operating system.
- This can lead to unauthorized access and data manipulation. Code Injection Vs. Command Injection
- As both aim to disintegrate the host server and implicate injecting manipulated elements, it is apparent to consider them alike. However, that’s not 100% true.

- Code injection interests exploited code introduction using an app and banks upon the ill-handling of non-trustful data inputs by the end-user.

- All the attacks, using this mode of action, happen due to the absence of (one of multiple essential) end-user data validation at any stage. The code introduction can be done locally or over the internet.

- Speaking of the harms caused, injecting corrupted code can only hamper the targeted system/application.

- For instance, if a threat actor introduces a corrupted PHP code, then the code will be highly driven by the host machine's PHP functionalities and permissions. Its execution is simple and looks very much similar to Trojan horses.



1. The hacker sends a Shell payload to the server
2. The server executes the command on his OS
3. The OS returns the text output if it exists
4. The page with the output is send to the hacker