WEB APPLICATION SECURITY

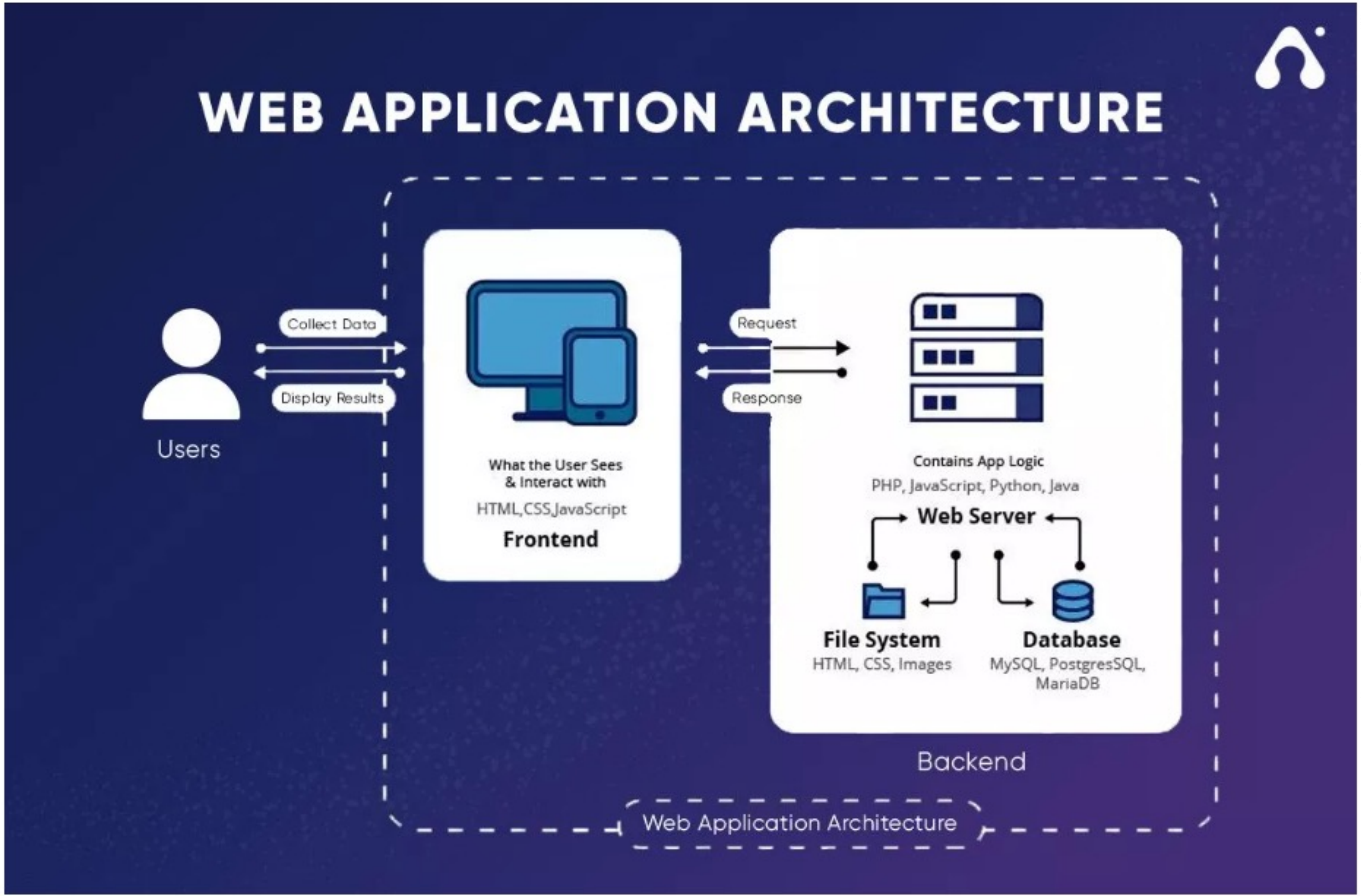
WHAT IS A WEB APP?

A web application is a type of app that can be accessed through a web browser.

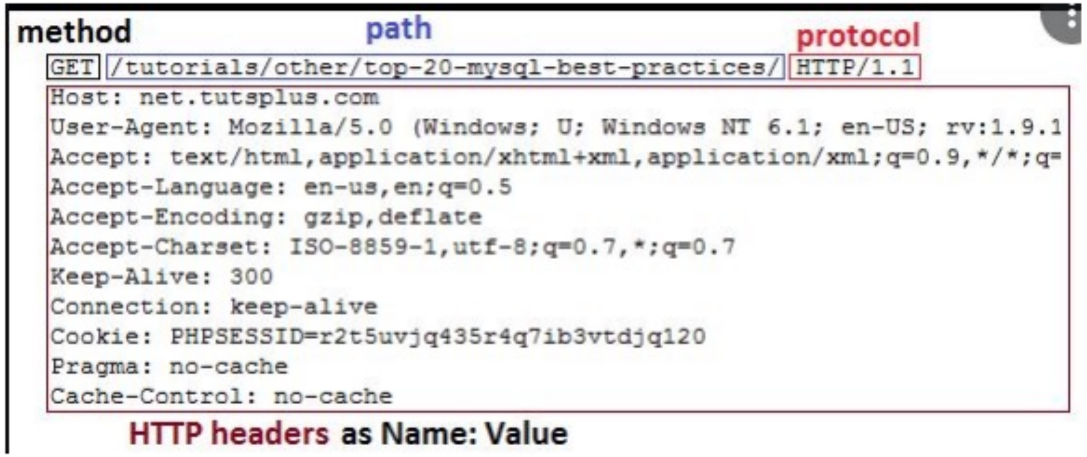
WHAT IS A WEB BROWSER?

A web browser is an application for accessing websites. E.g Google chrome, edge, Firefox, safari, etc.

ARCHITECTURE

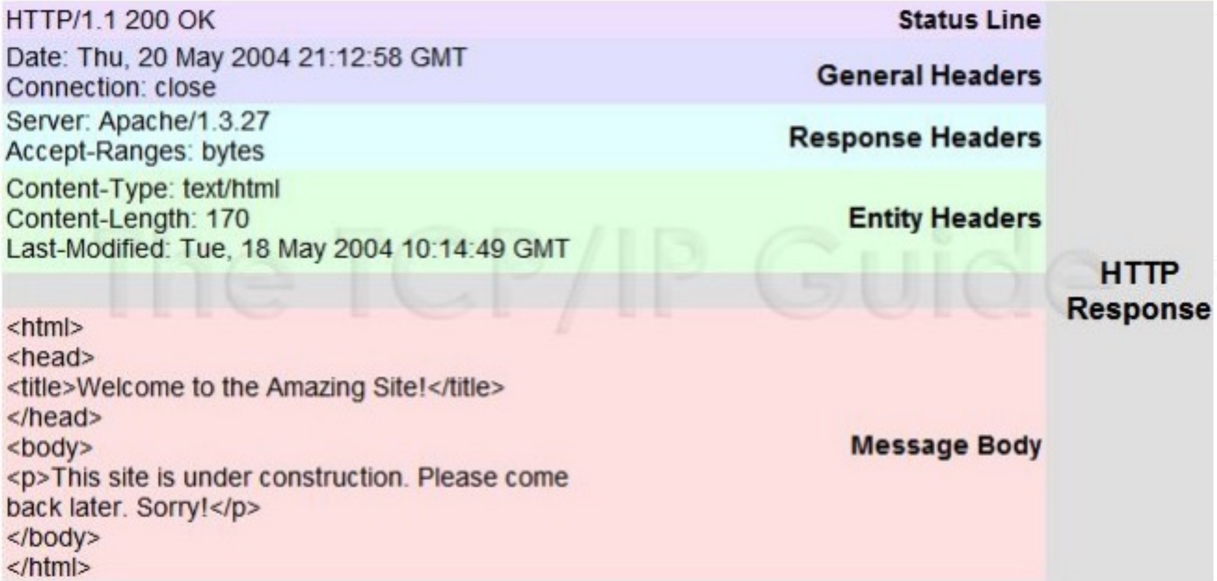

WEB APPLICATION ARCHITECTURE
Users
Collect Data
Display Results
What the User Sees & Interact with
HTML,CSS,JavaScript
Frontend
Request
Response
Contains App Logic
PHP, JavaScript, Python, Java
Web Server
File System
HTML, CSS, Images
Database
MySQL, PostgresSQL, MariaDB
Backend
Web Application Architecture

HTTP REQUEST AND RESPONSE



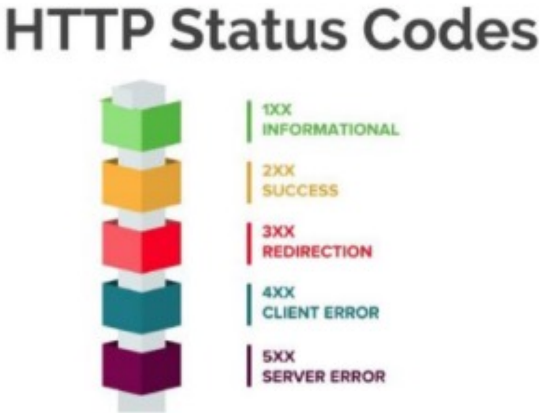
```
GET /tutorials/other/top-20-mysql-best-practices/ HTTP/1.1
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Pragma: no-cache
Cache-Control: no-cache
```

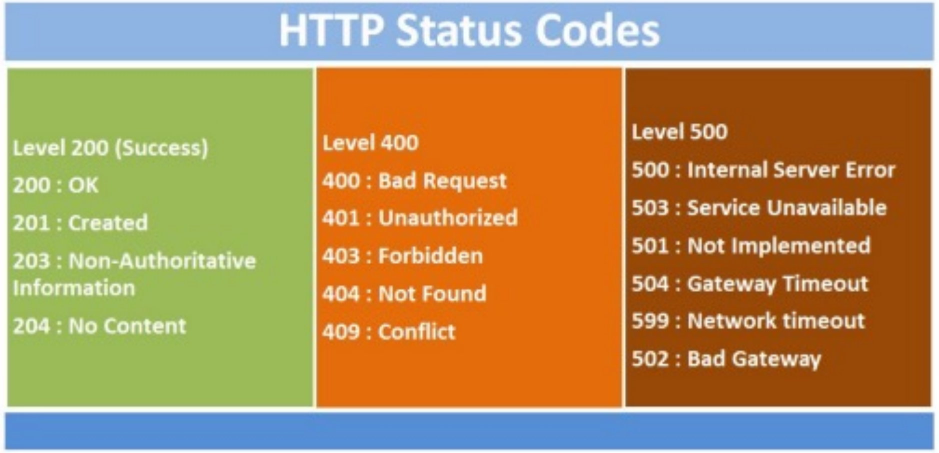
HTTP headers as Name: Value



```
HTTP/1.1 200 OK
Date: Thu, 20 May 2004 21:12:58 GMT
Connection: close
Server: Apache/1.3.27
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 170
Last-Modified: Tue, 18 May 2004 10:14:49 GMT

<html>
<head>
<title>Welcome to the Amazing Site!</title>
</head>
<body>
<p>This site is under construction. Please come
back later. Sorry!</p>
</body>
</html>
```





HTTP Status Codes

Level 200 (Success)	Level 400	Level 500
200 : OK	400 : Bad Request	500 : Internal Server Error
201 : Created	401 : Unauthorized	503 : Service Unavailable
203 : Non-Authoritative Information	403 : Forbidden	501 : Not Implemented
204 : No Content	404 : Not Found	504 : Gateway Timeout
	409 : Conflict	599 : Network timeout
		502 : Bad Gateway

HTTP METHODS

GET -used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

HEAD- Same as GET, but it transfers the status line and the header section only.

POST- used to send data to the server.

PUT- Replaces all the current representations of the target resource with the uploaded content.

DELETE- Removes all the current representations of the target resource given by URI.

CONNECT- Establishes a tunnel to the server identified by a given URI.

OPTIONS- Describe the communication options for the target resource.

WEB APP SCANNERS

→ Nessus https://www.tenable.com/downloads/nessus?loginAttempted=true

→ ZAP

→ Burp Suit

→ Nikto

→ Dirb

→ Dirbuster

NIKTO

Nikto is an open-source web server and **web application scanner**. Nikto can perform comprehensive tests against web servers for multiple **security threats**, including over 6700 potentially dangerous files/programs. Nikto can also perform checks for outdated web servers’ software, and version-specific problems.

```
# nikto –help
# nikto  –host scanme.nmap.org
# nikto   –h  10.10.10.10
```

DIRB

DIRB is a **Web Content Scanner**. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary-based attack against a web server and analyzing the responses.

```
# man dirb
# dirb scanme.nmap.org
```

HOW TO DOWNLOAD NESSUS

1- Download Nesuss on kali (https://www.tenable.com/downloads/nessus?loginAttempted=true)

select the platform to be (**Linux-Debian-amd64**)

2- Open the terminal and navigate to the Nessus directory **cd /Downloads/Nessus***

3- run **sudo dpkg -i Nessus*.deb**

4- start the nessus service **sudo /bin/systemctl start nessusd.service**

5- Open firefox in kali and type **https://kali:8834** then follow the prompt

6- Click **Continue** then select **register for Nessus essential**

7- Enter your name and email to register

8- Create username and password when prompted

9- run from kali terminal **systemctl enable nessusd**

10- run from kali terminal **systemctl start nessusd**

INJECTION ATTACK

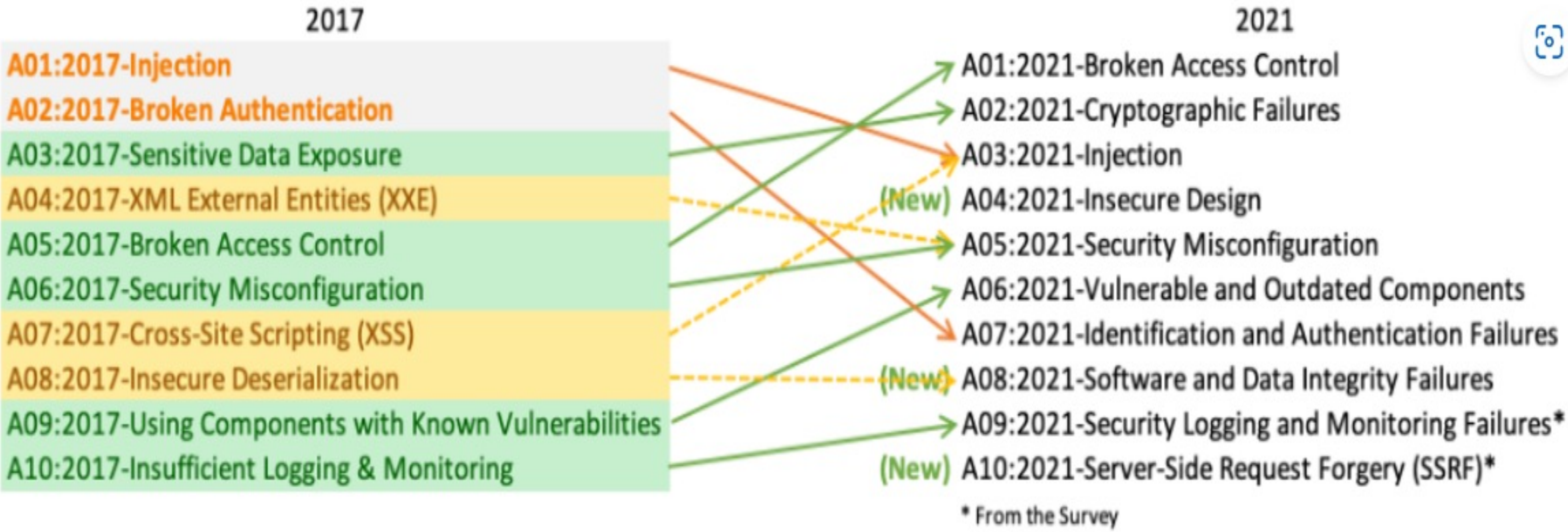
Injection attacks refer to a broad class of attack vectors. In an injection attack, an attacker supplies untrusted input to a program.

OWASP

Open Web Application Security Project

The list of top 10 most critical web application risk

OWASP TOP 10



2017

A01:2017-Injection

A02:2017-Broken Authentication

A03:2017-Sensitive Data Exposure

A04:2017-XML External Entities (XXE)

A05:2017-Broken Access Control

A06:2017-Security Misconfiguration

A07:2017-Cross-Site Scripting (XSS)

A08:2017-Insecure Deserialization

A09:2017-Using Components with Known Vulnerabilities

A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

A03:2021-Injection

(New) A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

(New) A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures*

(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

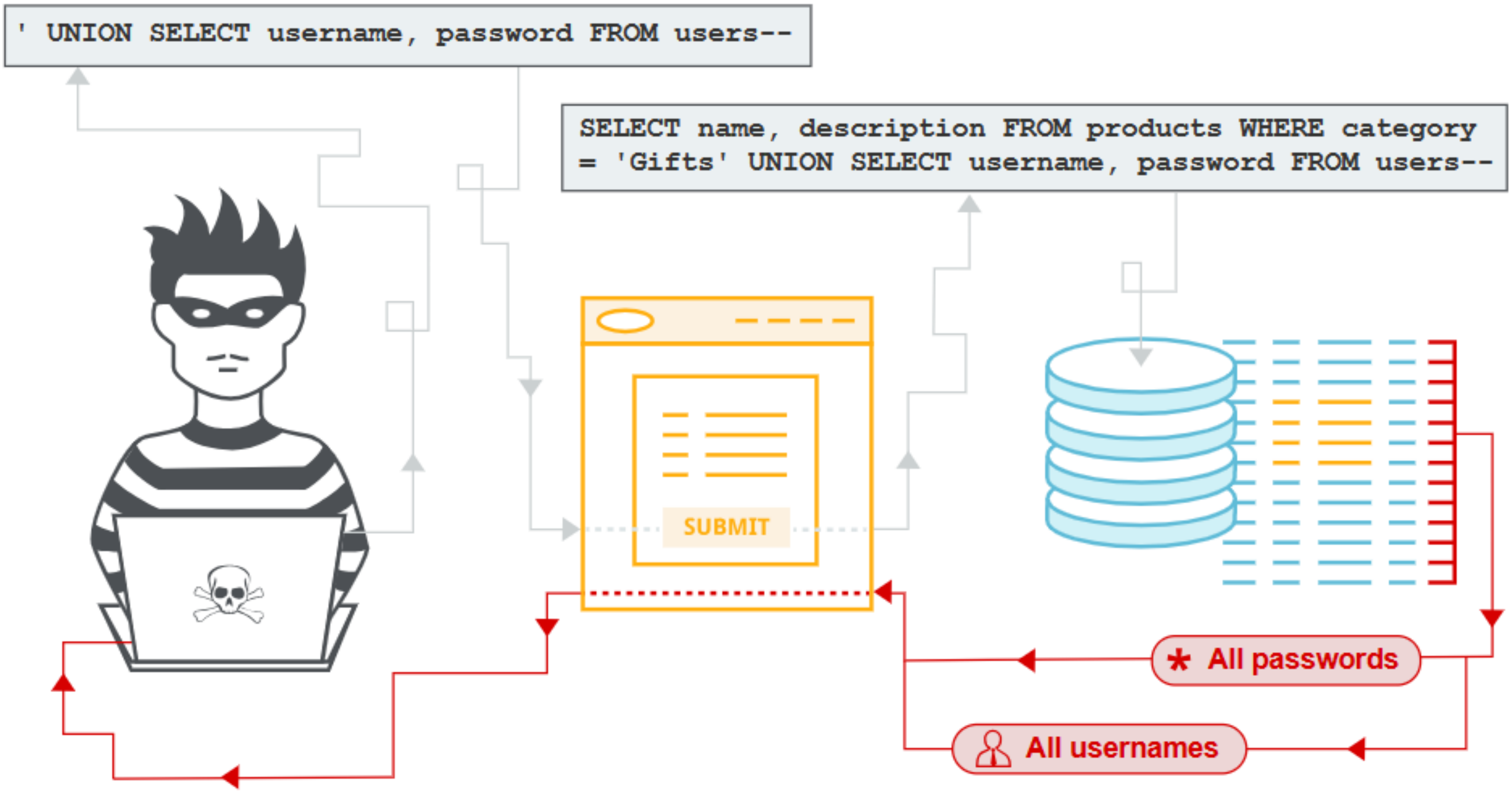
COMMAND EXECUTION ATTACK

Command execution attacks refer to a type of security vulnerability where an attacker exploits a weakness in a system or application to execute arbitrary commands on the targeted machine.

- `admin' --`
- `admin' #`
- `admin'/*`
- `' or 1=1--`
- `' or 1=1#`
- `' or 1=1/*`
- `') or '1'='1--`
- `') or ('1'='1--`

SQL INJECTION (SQLI)

SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.



```
' UNION SELECT username, password FROM users--
```

```
SELECT name, description FROM products WHERE category
= 'Gifts' UNION SELECT username, password FROM users--
```

SQL INJECTION MITIGATION

Input validation or **input sanitization**

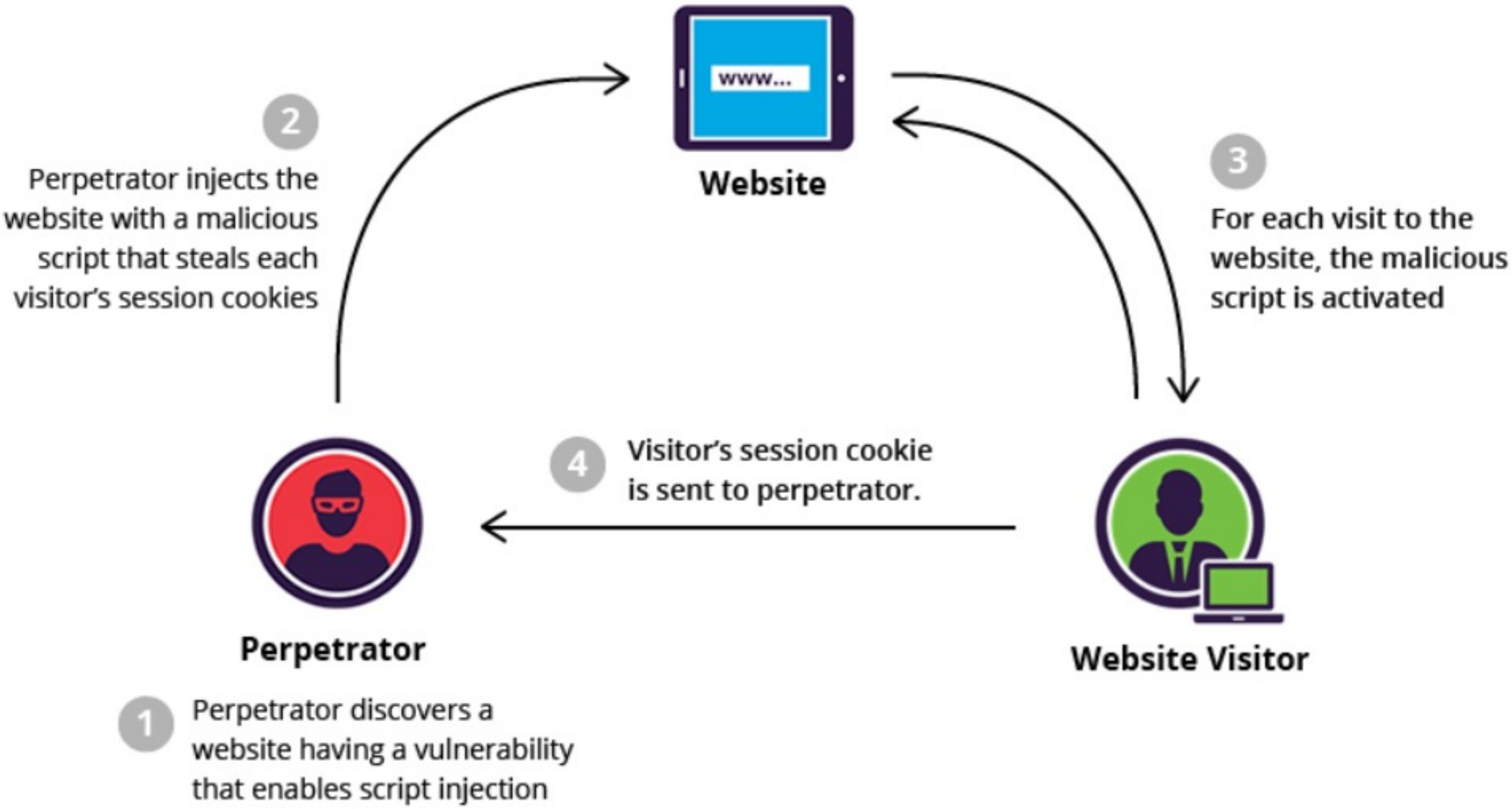
VIDEO EXPLAINING SQL INJECTION

What is SQL injection? - Web Security Academy – YouTube

CROSS-SITE SCRIPTING (XSS)

Cross-site scripting is an exploit where the attacker attaches code onto a legitimate website that will execute when the victim loads the website.

XSS



1 Perpetrator discovers a website having a vulnerability that enables script injection

2 Perpetrator injects the website with a malicious script that steals each visitor's session cookies

3 For each visit to the website, the malicious script is activated

4 Visitor's session cookie is sent to perpetrator.

PRACTICE SQL QUERIES

https://www.w3schools.com/sql/sql_syntax.asp

https://highon.coffee/blog/nikto-cheat-sheet/