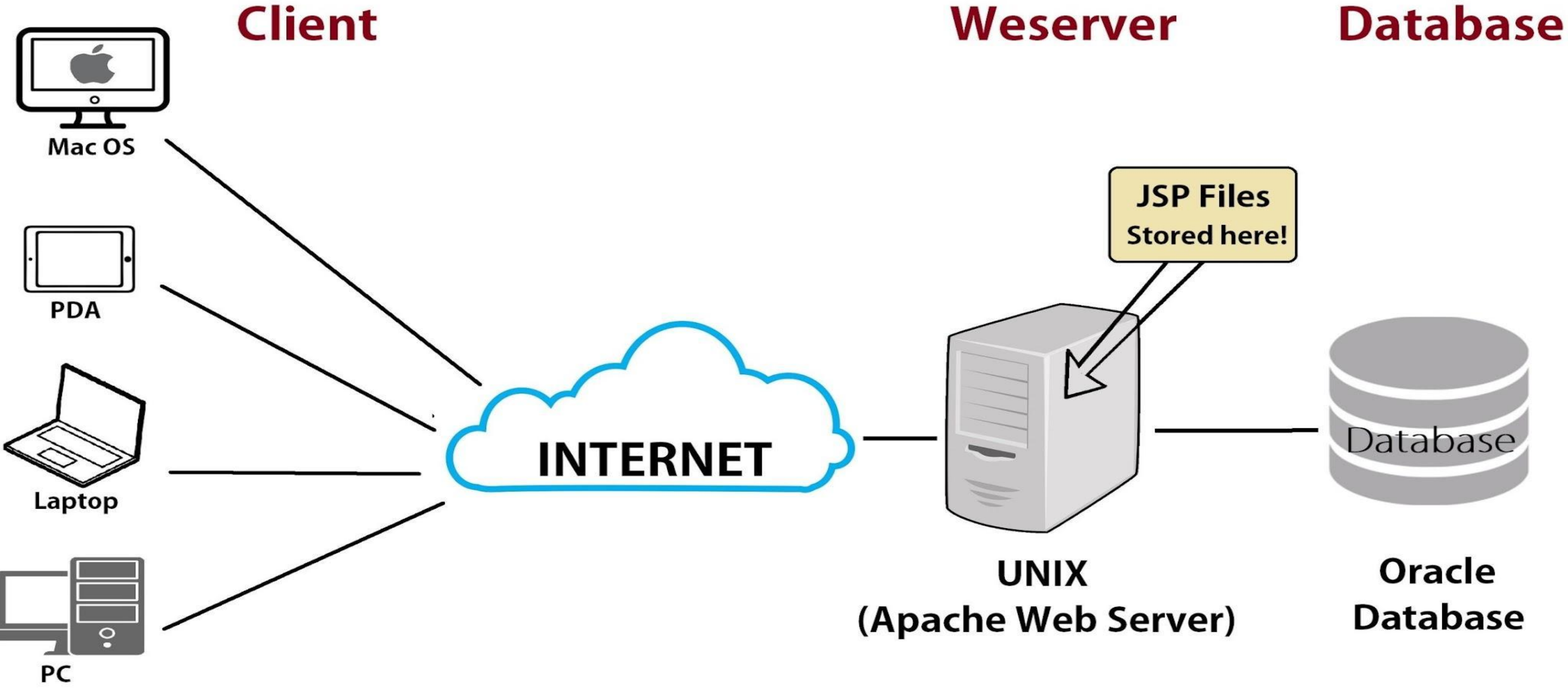
Web Application Penetration Testing

How to start hacking on systems ?

1. Identify yourself
2. Learn how the system is builded
3. Learn how system works
4. Learn hacking steps
5. Learn about common vulnerabilities and attacks on system you want to pentest
6. Practice on vulnerabilities and attacks
7. Learn CVES

Web Development

1. Front End
    a. HTML
    b. CSS
    c. Javascript
2. Backend
    a. Server Side (PHP , Python, JS, .NET , Ruby , CGI)
    b. DBMS (MYSQL , Mariadb , Oracle , PostgreSQL , Redis , Mongo DB , SQLite)
    c. SQL



Client

Weserver

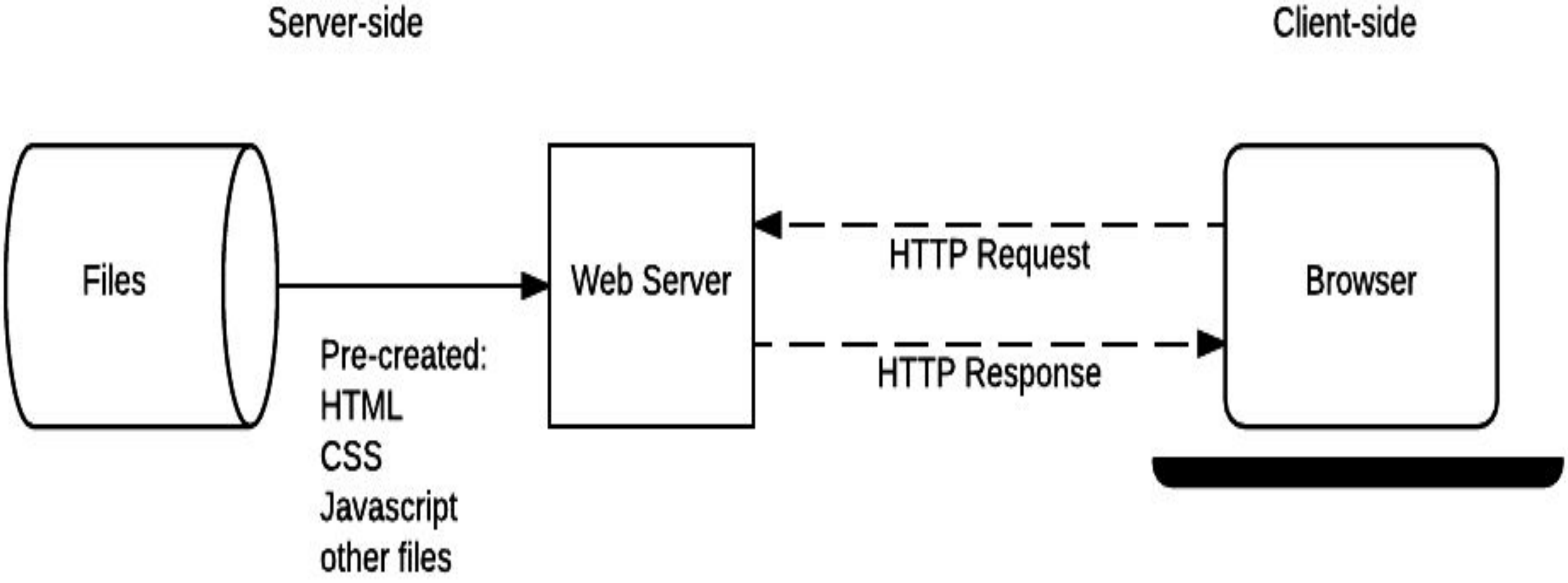
Database

UNIX
(Apache Web Server)

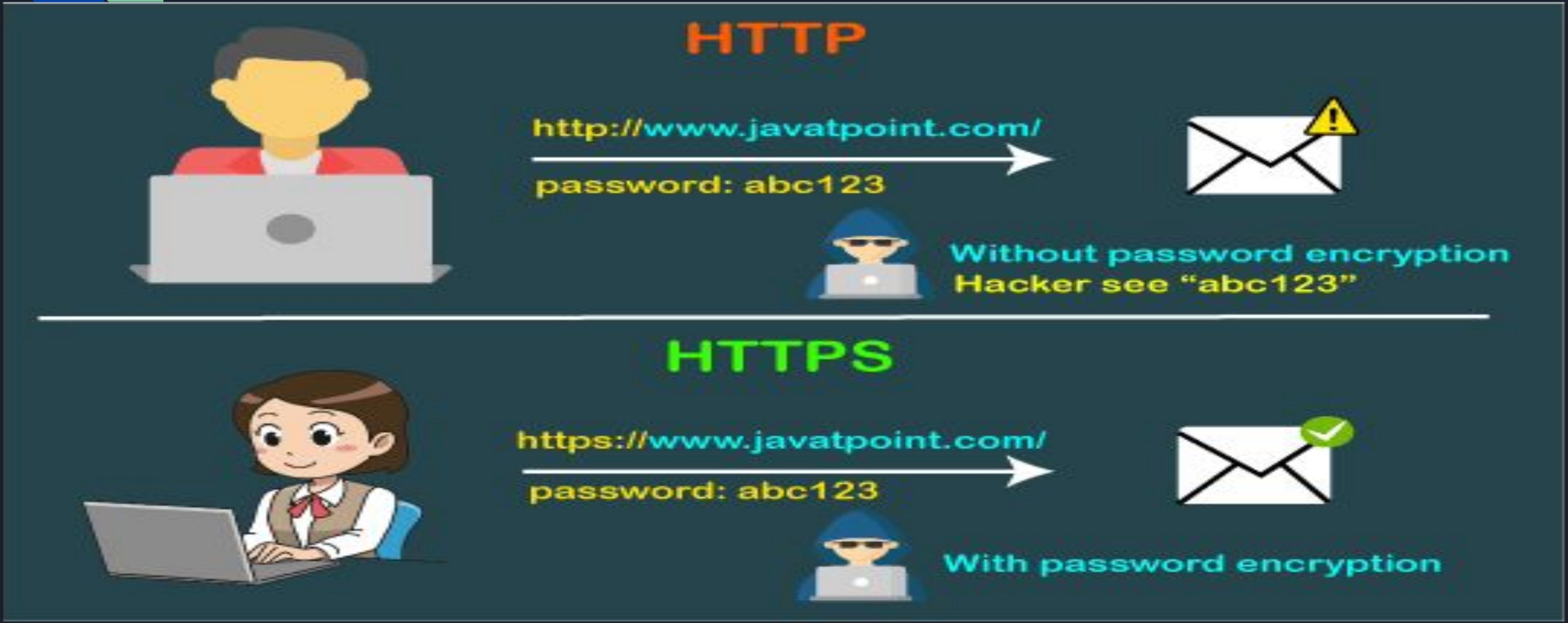
Oracle
Database

Web Client Browsers (Client Side)

Web Server And Client



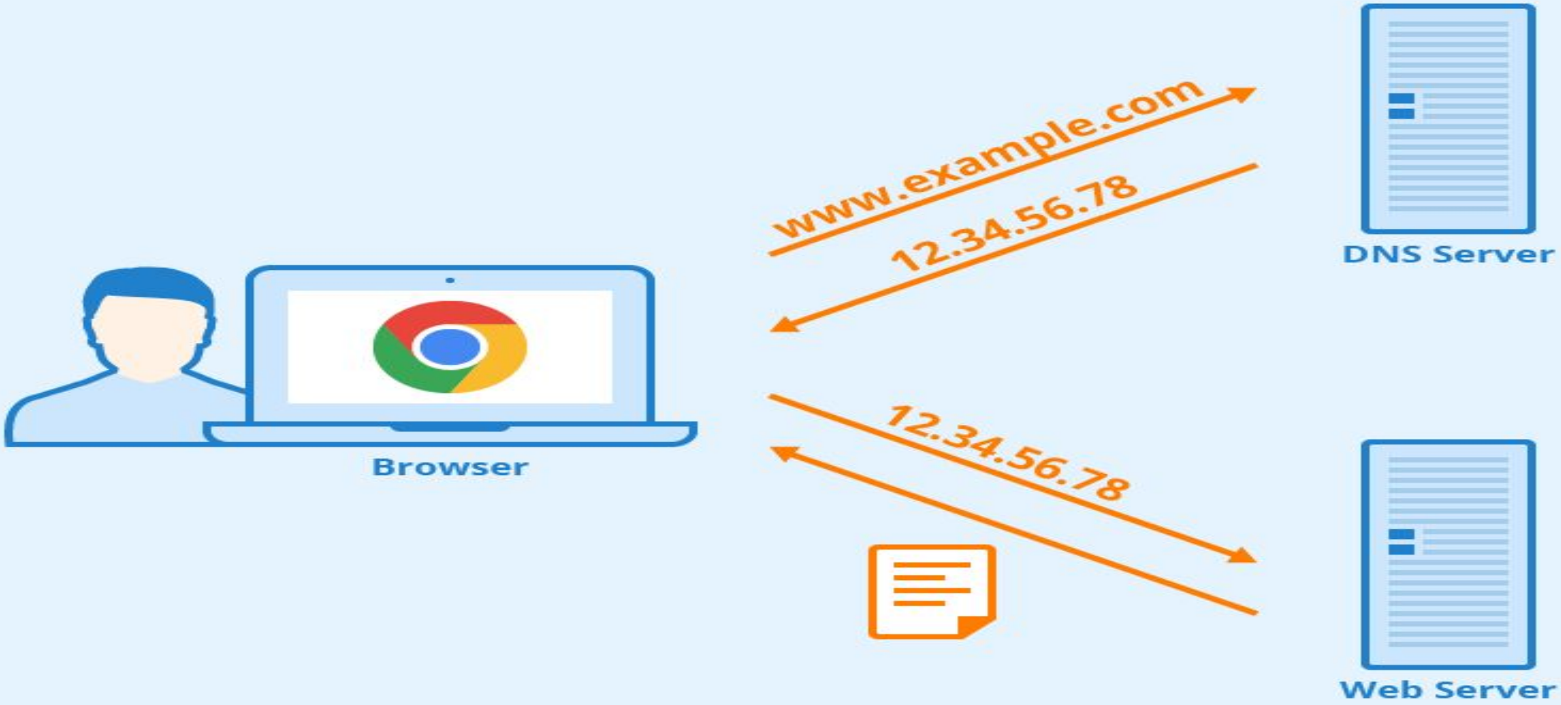
Web Protocols



Without password encryption
Hacker see “abc123”

With password encryption

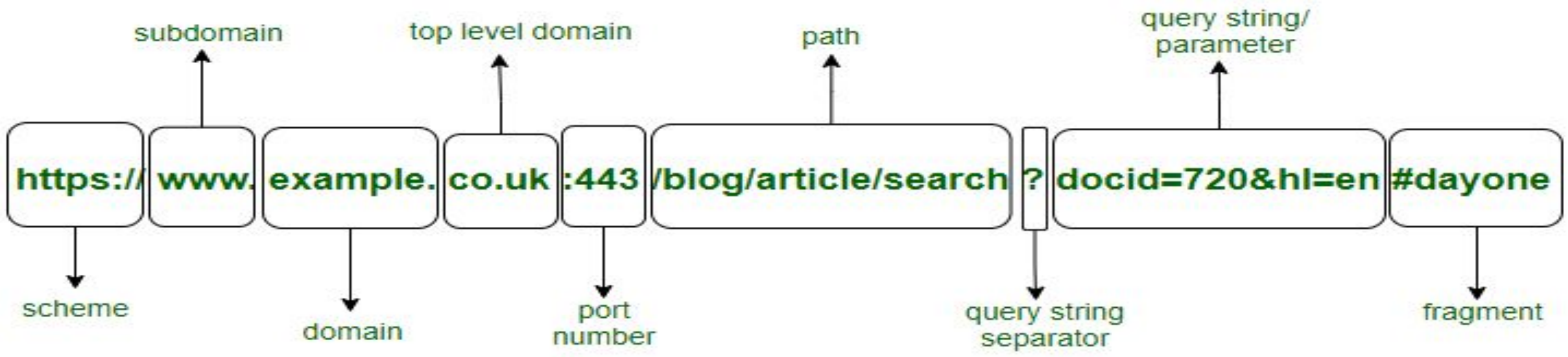
Domain Name System



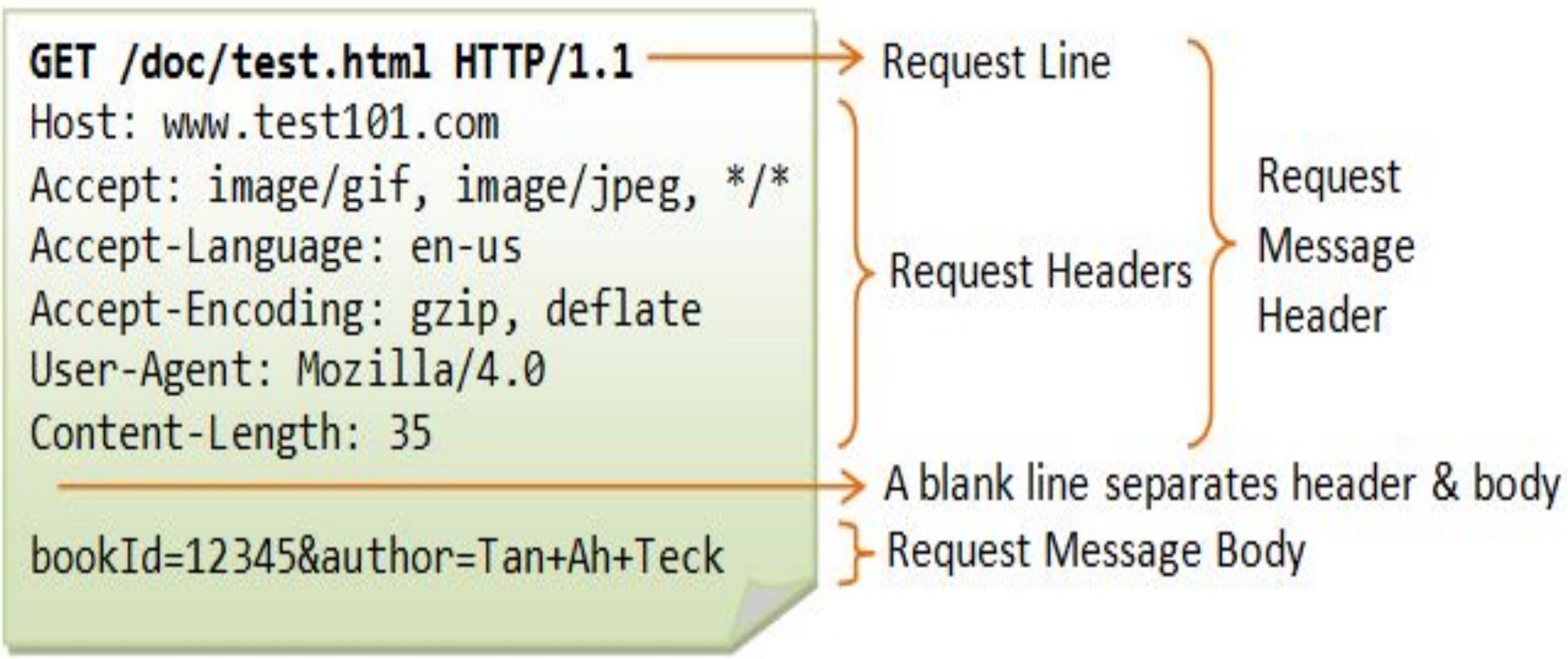
URL Structure

Parts of a URL

URL : https://www.example.co.uk:443/blog/article/search?docid=720&hl=en#dayone



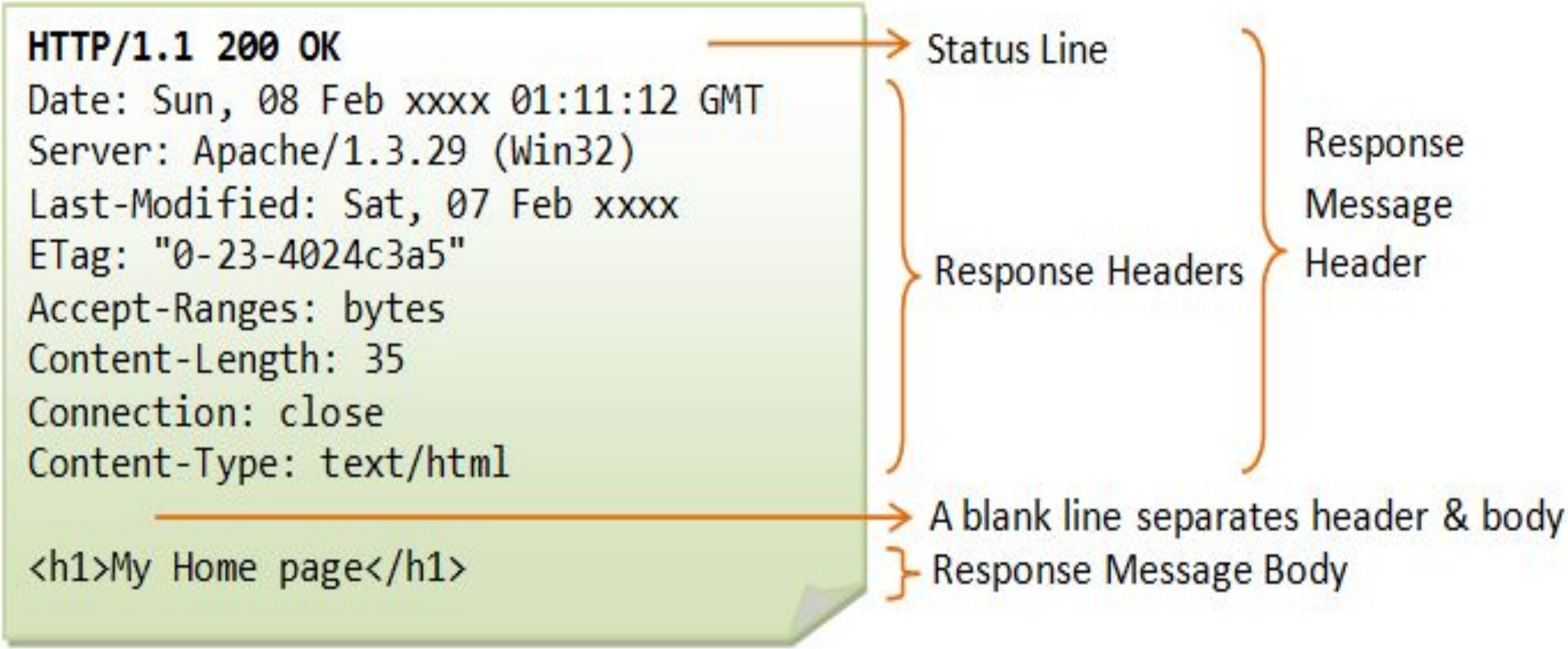
HTTP Request Headers



```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

HTTP Response Headers



```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

HTTP Request Methods

GET	POST	PUT	DELETE	PATCH	HEAD
retrieve data from server	add data to an existining file or resource	update(replace) an existing file or resource in server	delete data from server	update a resource partially (modify)	retrieve the resource's headers

- **CONNECT** is used to open a two-way socket connection to the remote server;
- **OPTIONS** is used to describe the communication options for specified resource;
- **TRACE** is designed for diagnostic purposes during the development.
- **HEAD** retrieves the resource's headers, without the resource itself.

HTTP Status Codes

Level 200

200: OK
201: Created
202: Accepted
203: Non-Authoritative Information
204: No content

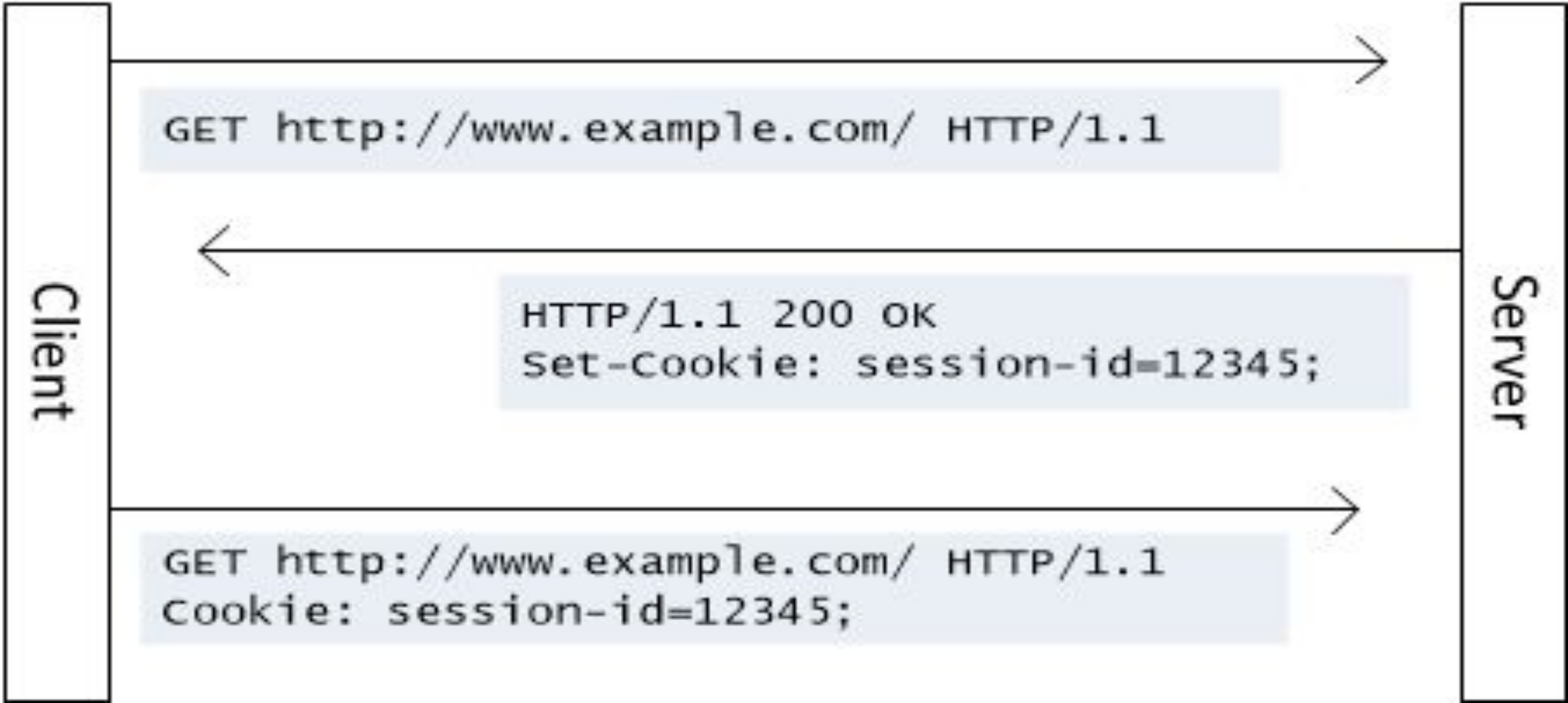
Level 400

400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
409: Conflict

Level 500

500: Internal Server Error
501: Not Implemented
502: Bad Gateway
503: Service Unavailable
504: Gateway Timeout
599: Network Timeout

HTTP Cookies

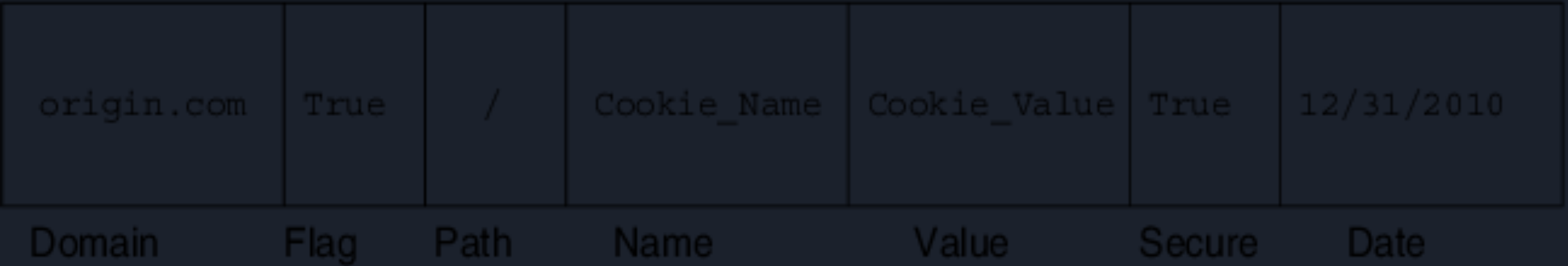


```
GET http://www.example.com/ HTTP/1.1
```

```
HTTP/1.1 200 OK
Set-Cookie: session-id=12345;
```

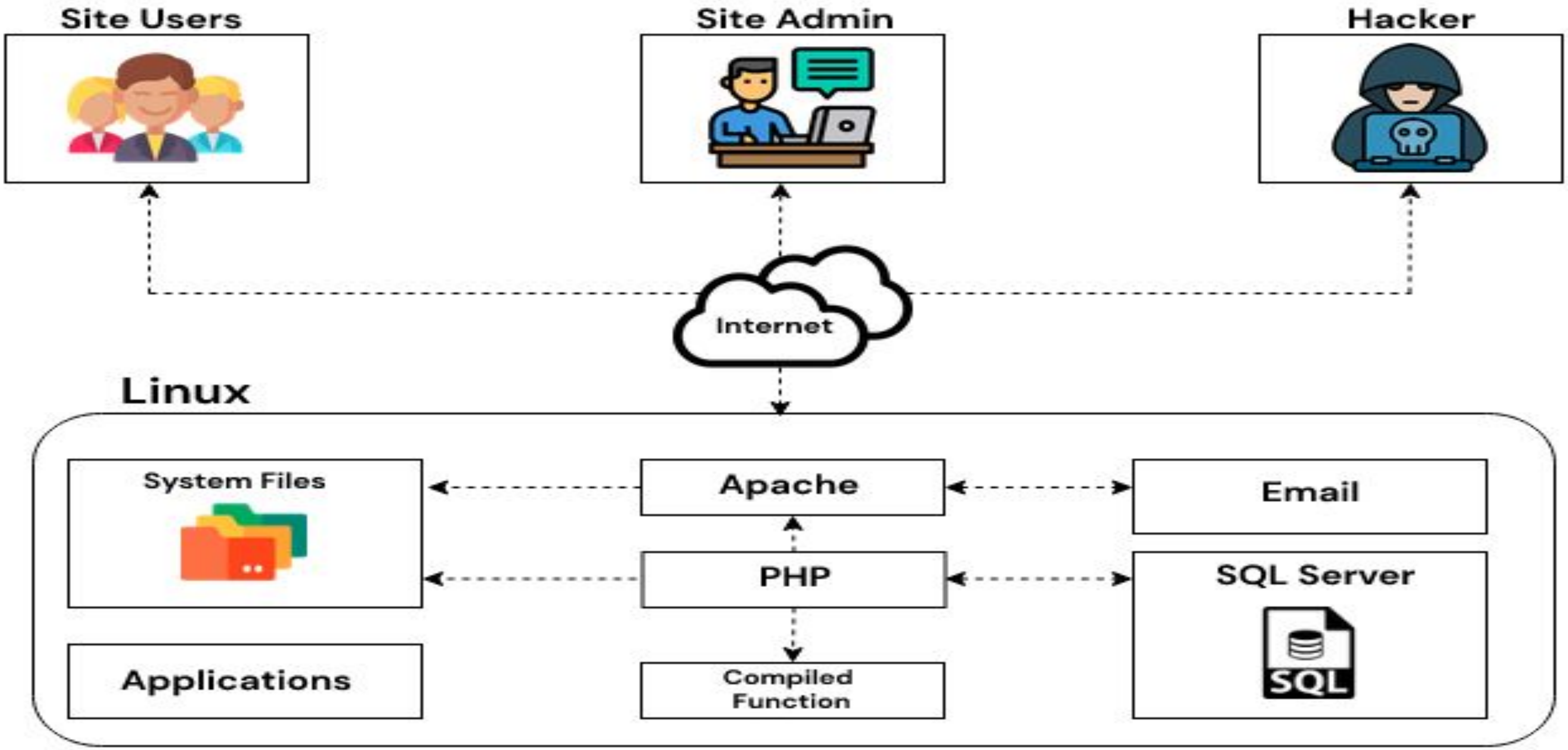
```
GET http://www.example.com/ HTTP/1.1
Cookie: session-id=12345;
```

Cookie Structure



origin.com	True	/	Cookie_Name	Cookie_Value	True	12/31/2010
Domain	Flag	Path	Name	Value	Secure	Date

Hacking a Web Server


Site Users
Site Admin
Hacker
Internet
Linux
System Files
Apache
Email
PHP
SQL Server
SQL
Applications
Compiled Function