RT0706 – Web Security
The Attacker’s View of the Web

Who Am I ?

Anthony Maia

@piosky1

https://cs.piosky.fr

Teams / Discord

Table of Contents

1

Class Presentation

What is this class about?

Class Presentation

Lectures 15 hours

1. The attacker's point of view
2. Reconnaissance and mapping
3. Vulnerability detection and exploitation

Practical work 15 hours (graded)

- Two-person team
- You will need
  - Kali Linux
  - A proxy
  - https://www.root-me.org account

Exam

- **Multiple-choice questionnaire (40 questions - 1 hour)**

2

General Overview

Why a web security class?

General Overview

Why a web security class?

- Daily use of web applications exposed to anyone with an Internet connection
- Originally the web was static
- Many of today's websites have grown significantly more complex
- There is now a much larger and accessible attack surface
- But many new frameworks and technologies improve the web security

General Overview

Why do we will study web security from an offensive point of view?

- Way more fun and efficient
- Be aware of the attacks and the related risks and impacts
- Take security into consideration while developing and integrating web solution

General Overview

Why offensive security is needed to secure a web application?

- Real attack simulation with an hacker mindset
- Risk, impacts and vulnerabilities are assessed in real conditions
- External point of view (a person who did not develop the application)
- Security recommendations
- Some flaws or bugs are hard to see in a static condition
- Business logic flaws cannot be found by scanners
- Creation of proof of concepts

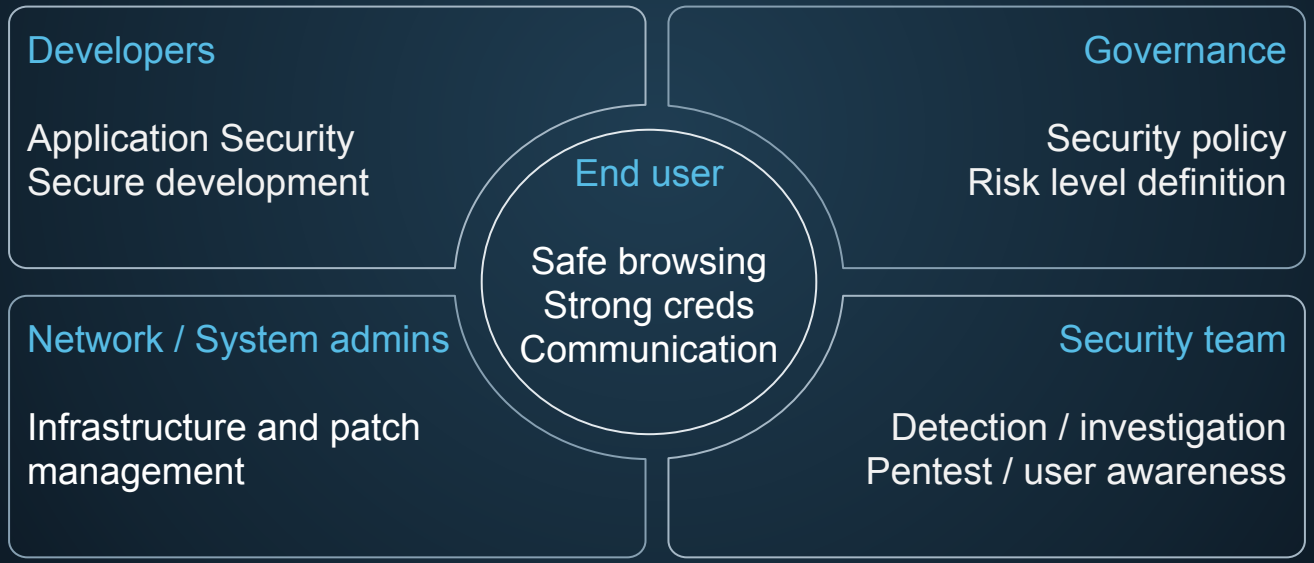
General Overview

Difference between attackers and defenders

- It's hard to attack but even more to defend
- Attackers need one vulnerability to get in, defenders need to patch them all
- Attackers always have a head start on defenders
- You need to understand the attacker mindset and techniques to defend yourself
- You need to know defender countermeasures to bypass them or stay stealthy

Security Actors

Security is hard, new techniques and vulnerabilities are released every day, and everyone should feel concerned



Attacker’s Motivations

Motivations	Methods
Financial	Data steal / ransomware
Curiosity / personal interest	Full domain control
To gain reputation / fame	Crash / availability / defacement
Revenge attack	Full domain control / source code backdoor
Political	Long term espionage

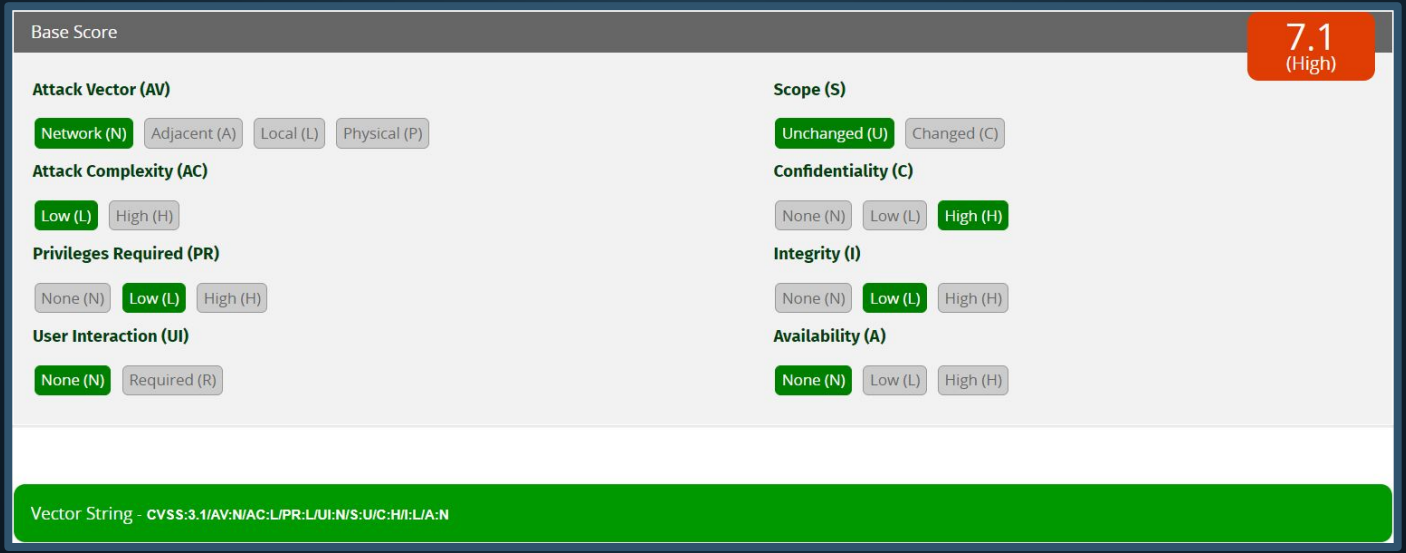
Vocabulary

- Threat: Agent or actor that can cause harm
- Vulnerability: A flaw someone can exploit to cause harm
- Risk: Where the threat and vulnerability overlap
- Exploit: Code or technique that a threat uses to take advantage of a vulnerability

Security Standards

Common Vulnerability Scoring System (CVSS) v3.1

- Used to rate a vulnerability
- https://www.first.org/cvss/calculator/3.1



Security Standards

Common Weakness Enumeration (CWE)

- Used to classify a vulnerability
- https://cwe.mitre.org

CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Weakness ID: 79
Abstraction: Base
Structure: Simple

Status: Stable

Presentation Filter: Complete

Description

The software does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

Extended Description

Cross-site scripting (XSS) vulnerabilities occur when:

1. Untrusted data enters a web application, typically from a web request.
2. The web application dynamically generates a web page that contains this untrusted data.
3. During page generation, the application does not prevent the data from containing content that is executable by a web browser, such as JavaScript, HTML tags, HTML attributes, mouse events, Flash, ActiveX, etc.
4. A victim visits the generated web page through a web browser, which contains malicious script that was injected using the untrusted data.
5. Since the script comes from a web page that was sent by the web server, the victim's web browser executes the malicious script in the context of the web server's domain.
6. This effectively violates the intention of the web browser's same-origin policy, which states that scripts in one domain should not be able to access resources or run code in a different domain.

Security Standards

Common Vulnerabilities and Exposures (CVE)

- Used to report a vulnerability
- https://cve.mitre.org

CVE-ID	
CVE-2018-10207	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

An issue was discovered in Vaultize Enterprise File Sharing 17.05.31. An attacker can exploit Missing Authorization on the FlexPaperViewer SWF reader, and export files that should have been restricted, via vectors involving page-by-page access to a document in SWF format.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:https://www.excellium-services.com/cert-xlm-advisory/cve-2018-10207/

Security Standards

OWASP Testing guide (OTG)

OWASP Application Security Verification Standard (ASVS)

- Used to guide the assessment and ensure a level of security checks
- https://owasp.org/www-project-web-security-testing-guide
- https://cheatsheetseries.owasp.org

3

Penetration Test

What is a penetration test?

Penetration Test

What is a penetration test?

"It is an authorized simulated attack on a computer system, performed to evaluate the security of the system." - Wikipedia

Penetration Test

What is the job of a penetration tester?

- To model the actions of real-world threats
- To find vulnerabilities and exploit them to determine business risks
- To recommend appropriate defenses

Legislation

Permission to test

- Getting permission from the client or internal authority (CISO) is primordial
- The permission must be written and signed

Article 323-1

Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 60 000 euros d'amende.
Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système, la peine est de trois ans d'emprisonnement et de 100 000 euros d'amende.

Article 323-2

Le fait d'entraver ou de fausser le fonctionnement d'un système de traitement automatisé de données est puni de cinq ans d'emprisonnement et de 75 000 euros d'amende.

Article 323-3

Le fait d'introduire frauduleusement des données dans un système de traitement automatisé ou de supprimer ou de modifier frauduleusement les données qu'il contient est puni de cinq ans d'emprisonnement et de 150 000 euros d'amende.

Penetration Test Point of Views

Black box

- No knowledge of the application and the infrastructure (attacker)
- Non-authenticated point of view (no credentials)

White box

- Advanced knowledge of the application and the infrastructure (developer)
- Access to source and to architecture schemas
- Access to privileged accounts (admin / moderator…)

Grey box

- Partial knowledge of the application and the infrastructure (end user)
- Access to the application using a standard user account

Vulnerability Scan VS Penetration Test

Vulnerability scan	Penetration test
Automated	Manual and automated (tools)
Time saving (minutes)	Time consuming (days)
Scheduled	Annually (or after significant changes)
Passive	Active
Report false positive	Rules out false positive
Programmed	Intuitive
N/A	Risk aware and evaluation
N/A	Exploitation and Post-Exploitation

Offensive Security Consultant

An attacker mindset

- Knowledge of how the application can be attacked
- Creation of PoC in order to prove that the application is vulnerable
- Exploit the vulnerability
- Adapt to the environment and combine vulnerability to exploitation paths

Offensive Security Consultant

A consultant mindset

- Recommendations based on experience and technologies
- Advices to prevent potential future flaws
- Code review approach may be used
- Reporting
- Communication with developers, security engineers management and board

Penetration Test Approaches

Web vulnerability assessment	Web penetration test
Limited to the web application	The web server is the starting point
Follows ASVS methodology	More instinctive
Tends to be exhaustive	Focus on most critical vulnerabilities
Makes a PoC for exploitation	Exploits and makes post-exploitation
Consultant with developer skills	Consultant with post-exploitation skills
Grey / White box approach	Black / Grey box approach

Penetration Test Methodology

- OSINT
- Commonly skipped
- Can expose architecture diagrams, credentials, e-mails
- Surprising results that can improve the security practices

Recon

Penetration Test Methodology

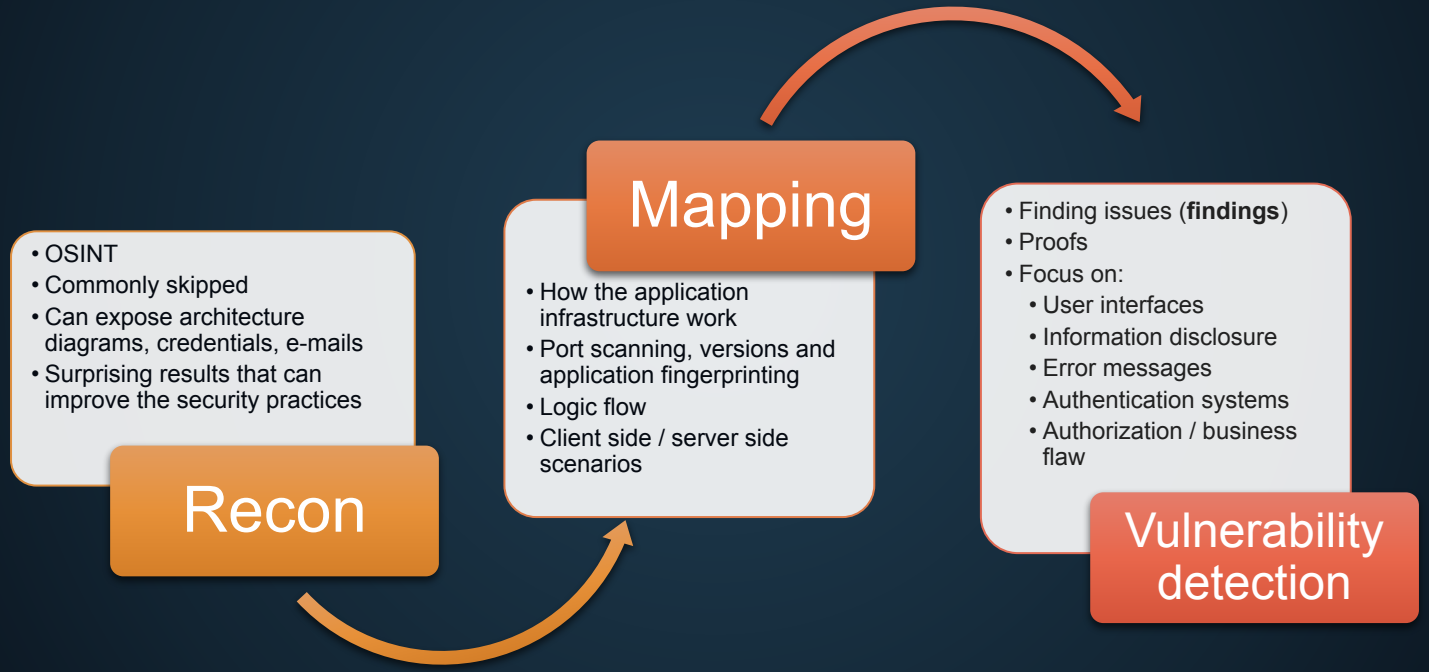
Recon

- OSINT
- Commonly skipped
- Can expose architecture diagrams, credentials, e-mails
- Surprising results that can improve the security practices

Mapping

- How the application infrastructure work
- Port scanning, versions and application fingerprinting
- Logic flow
- Client side / server side scenarios

Penetration Test Methodology



Penetration Test Methodology

Recon

- OSINT
- Commonly skipped
- Can expose architecture diagrams, credentials, e-mails
- Surprising results that can improve the security practices

Mapping

- How the application infrastructure work
- Port scanning, versions and application fingerprinting
- Logic flow
- Client side / server side scenarios

Vulnerability detection

- Finding issues (**findings**)
- Proofs
- Focus on:
  - User interfaces
  - Information disclosure
  - Error messages
  - Authentication systems
  - Authorization / business flaw

Exploitation

- Get new accesses (admin)
- Get a **Reverse Shell (RSH)**
- Get sensitive information
- Ensure persistence
- Then post-exploitation

Findings and Recommendations

Findings must be

- Proven: track record
- Repeatable: by developers or security engineers / consultants
- Explainable: independently of the audience

Recommendations

- From generic to detailed for the situation
- Should be specific to the technologies (language / framework / OS / web server…)
- Should be adapted and realistic according to the client constraints

4

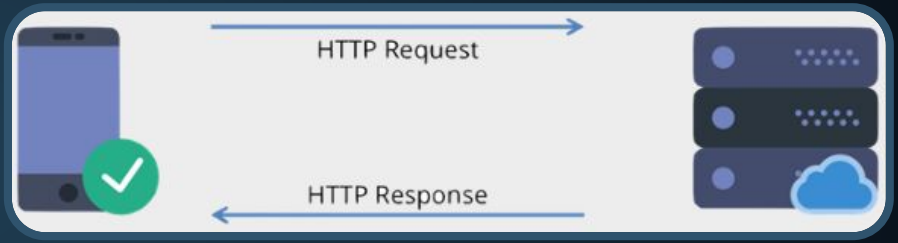
Web

How the web works?

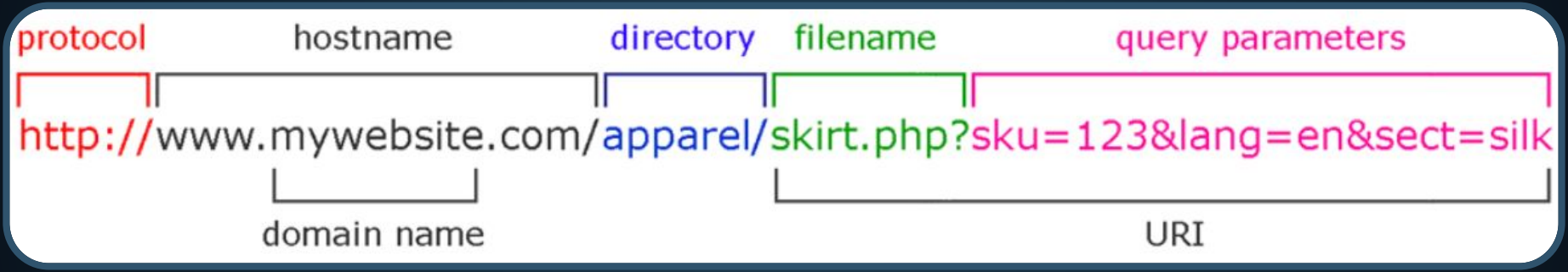
Hypertext Transfer Protocol

HTTP

- Client / server architecture
- Request / response protocol
- HTTP/1.1 is defined in RFC 2616
- HTTP/2.0 is defined in RFC 7540
  - Binary protocol



URL


protocol
hostname
directory
filename
query parameters
http://www.mywebsite.com/apparel/skirt.php?sku=123&lang=en&sect=silk
domain name
URI

HTTP Request

Method File HTTP version Headers

```
GET /index.html HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
Connection: Keep-Alive
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Host: www.example.com
Referer: http://www.google.com?q=dingbats
```

Blank line

Data – none for GET

HTTP Response

HTTP version

Status code

Reason phrase

Headers

```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Set-Cookie: …
Content-Length: 2543

<HTML> Some data... whatever ...</HTML>
```

Data

Cookies

Gmail Authentication Request

Request

Raw | Params | Headers | Hex

```
POST /_/signin/sl/lookup?hl=fr&_reqid=40282&rt=j HTTP/1.1
Host: accounts.google.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://accounts.google.com/signin/v2/identifier?continue=https%3A%2F%2Fmail.google.com%2Fmail%2F&servi
ce=mail&sacu=1&rip=1&flowName=GlifWebSignIn&flowEntry=ServiceLogin
X-Same-Domain: 1
Google-Accounts-XSRF: 1
Content-Type: application/x-www-form-urlencoded;charset=utf-8
Content-Length: 1854
Cookie: 1P_JAR=2018-09-22-09;
NID=139=vlsmoAgQo5KJXq57oW6uPboTDC69NRfX0tTDB9J7SXwN2oM9G4z3pK_MB-Yun6oZsN7Mzsj-Db7NI4BVrI4fG-X9mBjrRWg
VIqMmEohN1QWhwoRAyX0gXCqPicrrQAClxTqCK6GRLePN; CONSENT=WP.271a76; OGP=-5061451:;
GAPS=1:XiNZM8rvxl7_VPsoKL7g4u2m_92Z5Q:6F2xiNjay3pdZItf
DNT: 1
Connection: close

continue=https%3A%2F%2Fmail.google.com%2Fmail%2F&service=mail&rip=1&f.req=%5B%22fake%40gmail.com%22%2C%
22AEThL1xH10I7ezFik0f0SVkdXTLSkflARBSPqzSk7G8EdGiFrow6P10qZIxNfm-d0nRzVcqCL22vcLqS8ird2aRDGsHhRTFg-wFt2
9UIw2jm_RjzCwGD8-fwZMFICqn0tFsgDlYrBCPBAdbCjdwYNhaIzn0vu-A-VxluWWjLeUQuTWFt4cywubs%22%2C%5B%5D%2Cnull%2
C%22FR%22%2Cnull%2Cnull%2C2%2Cfalse%2Ctrue%2C%5Bnull%2Cnull%2C%5B2%2C1%2Cnull%2C1%2C%22https%3A%2F%2Fac
counts.google.com%2FServiceLogin%3Fcontinue%3Dhttps%253A%252F%252Fmail.google.com%252Fmail%252F%26servi
ce%3Dmail%26sacu%3D1%26rip%3D1%22%2Cnull%2C%5B%5D%2C4%2C%5B%5D%2C%22GlifWebSignIn%22%5D%2C1%2C%5Bnull%2
Cnull%2C%5B%5D%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnu
ll%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2C%5B%5D%5D%2Cnull%2Cnull%2Cnull%2Ctr
ue%5D%2C%22fake%40gmail.com%22%5D&bgRequest=%5B%22identifier%22%2C%22!jY6ljqSCTjR47nehQTpEM8PkL-TYy6MCA
AAAqVIAAAA0mQFCl_LtyDwwsVf0BC4hbUTHrTv2irG08syTmlQZveJLPD4zd67RW1H1MA2GCM1WND3-a0FpEhPkL-1SwoJTp48G8SJ8
BZPpKwrl-4e3Sowv_KTdMeaysqnFY_hhIch8xGdF6qjnlT2VLAryWETQnpQdGMymdmPVoKli7rX-2di8rl6N0aeWp2r9bQEcfbd7QZu
LpqnkrtB2jrqFlctD7wwtojJDqM3G44kTUQ2L82GVsgEox4x0_rP8ykl JQJC9HPsY1LBcdqDfq0-htD8hF50udVcpH3QDP03x3PMZZ0
F3JqPKyXExljfe8o_zqqWVTTIOvM0j6Doclwzlotmlf45VhtVMX2VRyY9AU9DeytucjCf5p13HjD7UJhYzzTasXjCJvL01SWHwBDCT6
mrx_njhror_uh001VoMeL1GAWsJ5ZH_4A%22%5D&azt=AFoagUWYNbasNsGV_Mtfr4uJL1IG2ikECw%3A1537607465053&cookiesD
isabled=false&deviceinfo=%5Bnull%2Cnull%2Cnull%2C%5B%5D%2Cnull%2C%22FR%22%2Cnull%2Cnull%2C%5B%5D%2C%22G
lifWebSignIn%22%2Cnull%2C%5Bnull%2Cnull%2C%5B%5D%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnul
l%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%2C%
5B%5D%5D%5D&gmscoreversion=undefined&checkConnection=youtube%3A494%3A1&checkedDomains=youtube&pstMsg=1&
```

Gmail Authentication Response

Response

Raw | Headers | Hex


```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Frame-Options: DENY
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Date: Sat, 22 Sep 2018 09:12:03 GMT
Content-Disposition: attachment; filename="response.bin"; filename*=UTF-8''response.bin
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
Server: GSE
Set-Cookie: GAPS=1:GuKR2jmC_G36TbLEuYD0iCg-WRjLdw:0Hml0msoCWN6pfI0;Path=/;Expires=Mon, 21-Sep-2020
09:12:03 GMT;Secure;HttpOnly;Priority=HIGH
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39,35"
Connection: close
Content-Length: 1281

)]}'

[[["gf.alr",1,"AEThLlw5Wka4dgU5rPYE6Ydtl-wshJHTmNPBqZCXCM0hUIX1CYoWKCcgdCaJ3PprbkuzowF549kA03U0hzBbwHX
jhX_PR_2MbQnyDWSmcICs_0DhLeP7kjfztYbTNzrisKfbNd-EeDoXo7-YY6uudejmmepDMBPD1qnGAMHdTxunL5dgvsXhZrnMg44ja
FyZaugAjXC10uwQG-AayriK9P1hvW6KC1GGo_uTa7D4_V3-wBzsipAmH8QYibUsB7DM29R0IZEJMbDM-kuHPJNcCW3Zs0t6kjSkCsm
xc1CU3bNufXUAkPHAVa_H0S0Uj4LWN5cDyx_GhHrr0EhLqcykXKEomZTy4uV9k7X6t0b9r2X_3T8hIBPmuxhH92t51__8jZLRxdFXk
Hof",[["fake@gmail.com",null,null,null,null,"fake@gmail.com","gmail.com",null,null,2]
]
,null,null,null,["gf.sisr",1,null,null,[[[null,null,"type:
FIRST_AUTH_FACTOR\n",1,null,"INITIALIZED",null,null,1,7,null,null,null,null,null,null,"fake@gmail.com"
,"https://lh3.googleusercontent.com/-XdUIqdMkCWA/AAAAAAAAAAI/AAAAAAAAAAA/AAN31DW9H02KiG1iLAZ2o37QtDUrA
-Huww/mo/photo.jpg",null,null,1,1,{"1001":[1]
,"5001":[]
```

User-Agent

- Web browsers are transposed as User-Agent
  - The string defines the browser in use
  - Sent in HTTP request
- User-Agent can be spoofed easily
- Tools use specific user-agent so change them
  - Nikto / Nmap / SQLMap...

Request Methods

Methods	Meaning
GET	Used to get a web resource from the server
HEAD	Used to get the header from GET method
POST	Posts data on the server
PUT	Asks the server to store the dat (inverse of GET)
DELETE	Asks the server to DELETE data
TRACE	Asks the server to return an action trace for diagnosis
OPTIONS	Asks the server to return the list of supported request methods

Status Code

Code	Meaning
1xx	Hold on
2xx	Here you go
3xx	Go away
4xx	Client messed up
5xx	Server messed up

Status Code

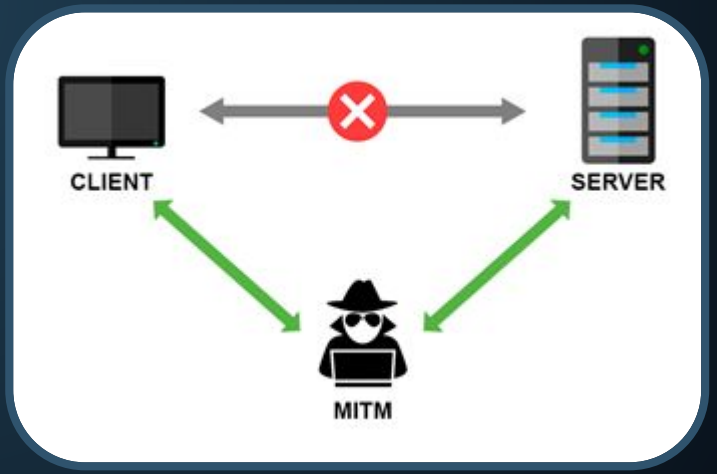
Level 200	Level 400	Level 500
200: OK (Success)	400: Bad request	500: Internal server error
	401: Unauthorized	501: Not implemented
	403: Forbidden	502: Bad gateway
	404: Not found	503: Service unavailable
		504: Gateway timeout

HTTPS

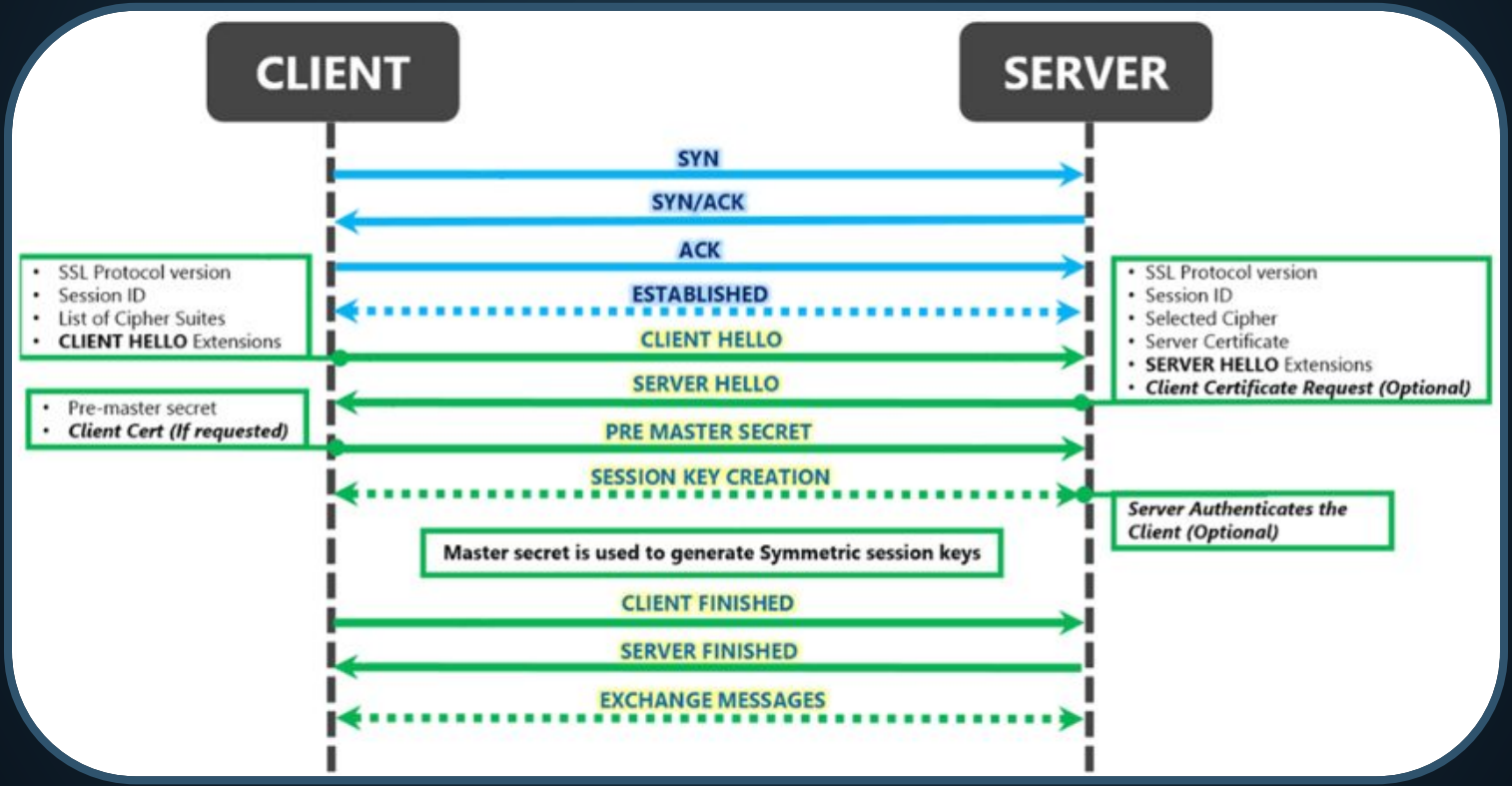
SSL/TLS are two common options for HTTP encryption

- TLS is the successor of SSL
- TLS adds two different methods to lower the chance of hash collision

It prevents Man-in-The-Middle attack



HTTPS

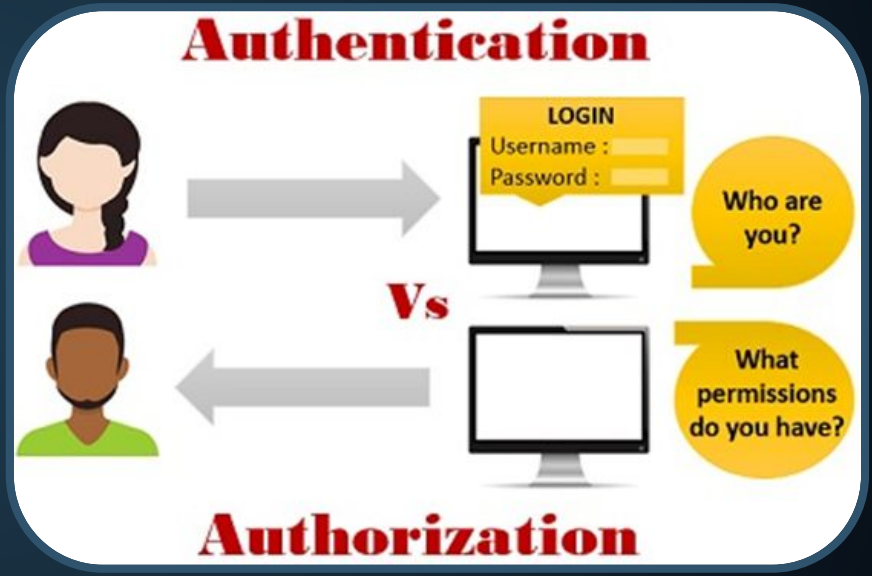

CLIENT
SERVER
SYN
SYN/ACK
ACK
ESTABLISHED
CLIENT HELLO
SERVER HELLO
PRE MASTER SECRET
SESSION KEY CREATION
CLIENT FINISHED
SERVER FINISHED
EXCHANGE MESSAGES
• SSL Protocol version
• Session ID
• List of Cipher Suites
• CLIENT HELLO Extensions
• Pre-master secret
• Client Cert (If requested)
• SSL Protocol version
• Session ID
• Selected Cipher
• Server Certificate
• SERVER HELLO Extensions
• Client Certificate Request (Optional)
Server Authenticates the Client (Optional)
Master secret is used to generate Symmetric session keys

Authentication and Authorization

User identification on the application

- Credentials
- Client-side certificate
- Multi-Factor Authentication

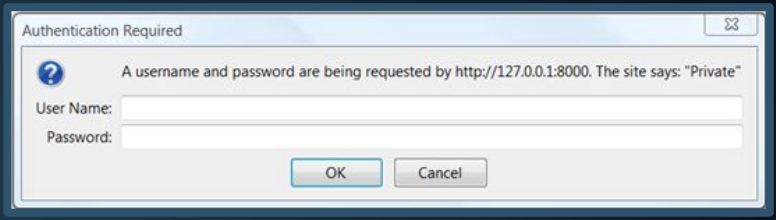
Authentication is not authorization

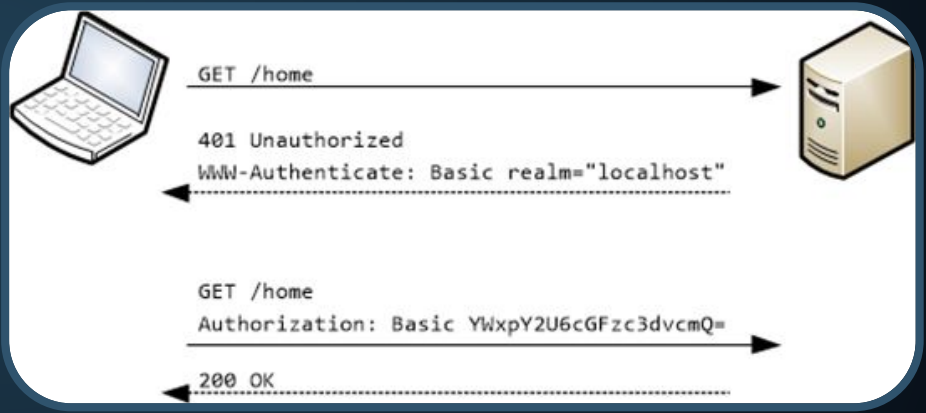


Basic Authentication

Defined in RFC 2617

- Username and password are base64 encoded
- Server identifies itself through “realm”
- Credentials sent for each request





Attacker point of view

- No account lockout (Brute force)
- No maximum login attempt
- Sniffed from HTTP (MiTM) or retrieved from a browser

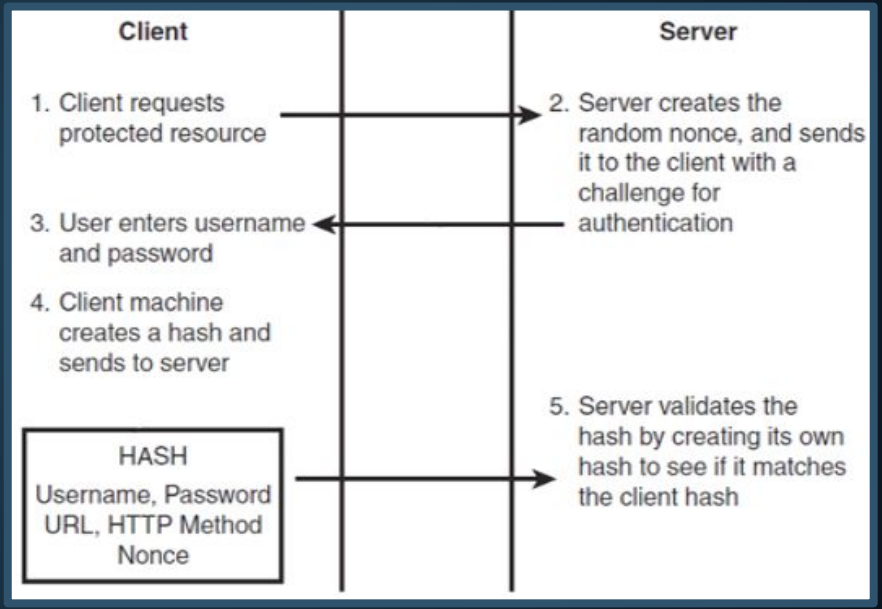
Digest Authentication

Defined in RFC 2617

- Designed to improve Basic Authentication
- Similar but uses MD5 and has a nonce (salt)

Attacker point of view

- No account lockout
- No maximum login attempt
- Nonce predictability
- MiTM attack



Integrated Windows Authentication (IWA)

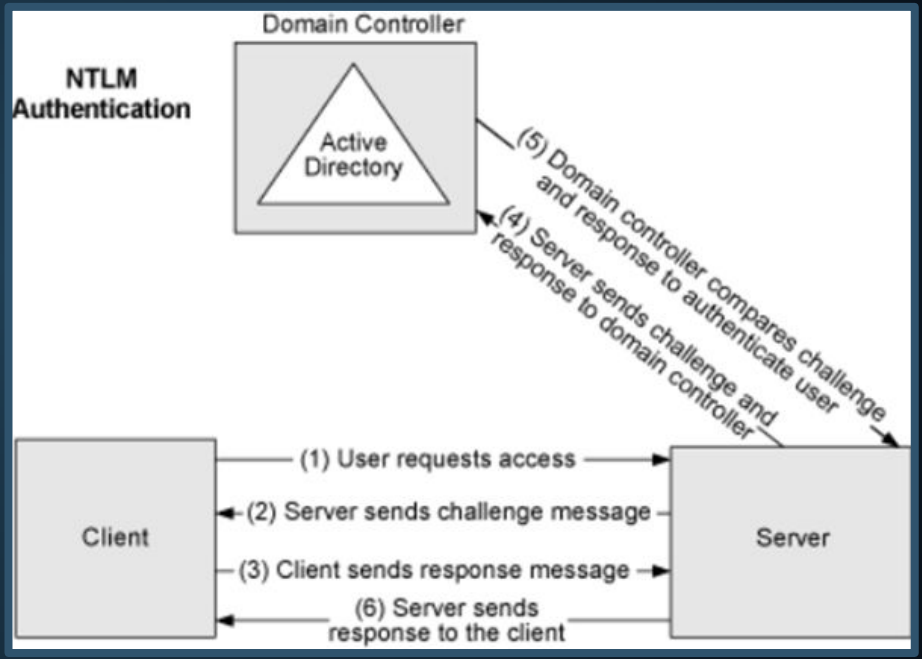
Microsoft proprietary authentication

Uses Windows OS authentication

- Challenge-Response protocol
- NTLM or Kerberos through HTTP(S)
- Authentication model for intranets
  - The server and the client need to be in the same or trusted Active Directory domain

Attacker point of view

- Attacks focused on client machines
- CSRF



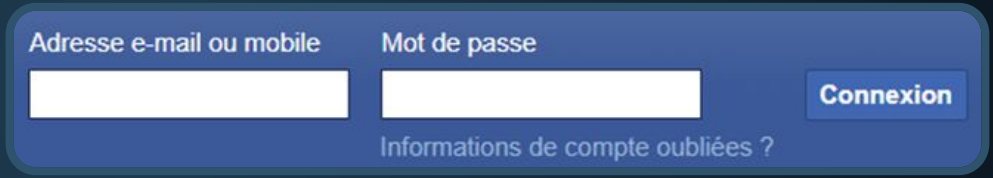
Forms-Based Authentication

Popular authentication model based on HTML forms

- Credentials are transmitted only once in plain text
- But are protected by HTTPS

Back-end authentication

- SQL / NoSQL / LDAP



Attacker point of view

- The security depends on the developer and the frameworks
- SQL / NoSQL / LDAP injections
- XSS: Stealing the authentication token (cookie)

Multi-Factor Authentication

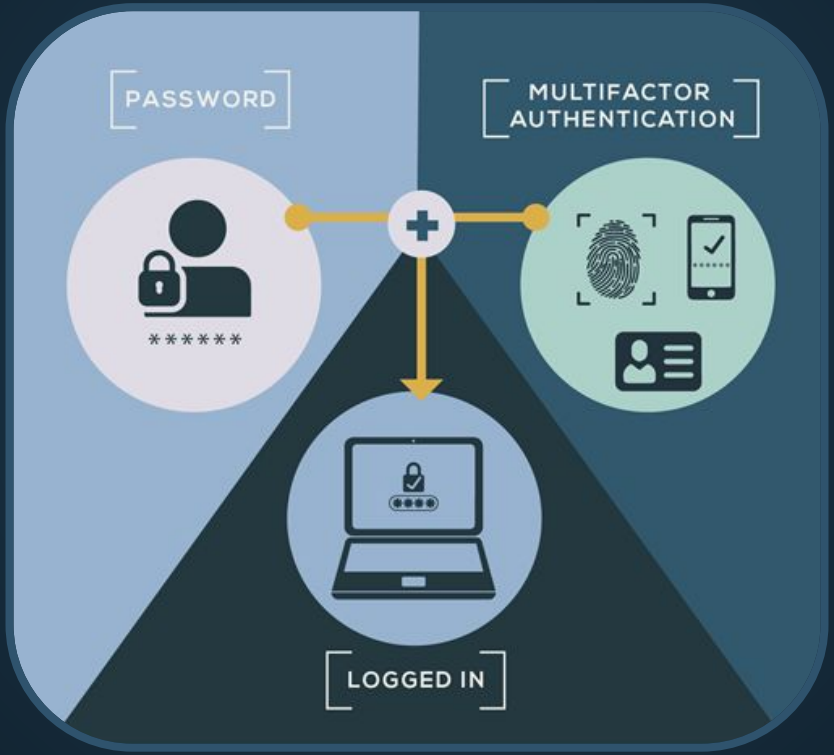
MFA also known as 2FA

- The first factor is your usual password that is standard for any account
- The second factor is a verification code retrieved from an application on a smartphone or computer

Security

- Such authentications reduce drastically the incidence of online identity theft because the password is not enough
- It is still vulnerable to phishing and MiTM

Multi-Factor Authentication



Multi-Factor Authentication

Multi-Factor authentication

What is MFA?

Your passwords can be easily compromised. MFA immediately increases your account security by requiring multiple forms of verification to prove your identity when signing into an application.

Microsoft Authenticator

Approve sign-ins from a mobile app using push notifications, biometrics, or one-time passcodes. Augment or replace passwords with two-step verification and boost the security of your accounts from your mobile phone.

Learn more >

Windows Hello for Business

Replace your passwords with strong two-factor authentication (2FA) on Windows 10 PCs. Use a credential tied to your device along with a PIN, a fingerprint, or facial recognition to protect your accounts.

Learn more >

FIDO2 security keys

Sign in without a username or password using an external USB, near-field communication (NFC), or other external security key that supports Fast Identity Online (FIDO) standards in place of a password.

Learn more >

Hardware tokens

Automatically generate a one-time password (OTP) based on open authentication (OATH) standards from a physical device.

Learn more >

Software tokens

Use the Microsoft Authenticator app or other third-party apps to generate an OATH verification code as a second form of authentication.

Learn more >

SMS and voice

Receive a code on your mobile phone via SMS or voice call to augment the security of your passwords.

Learn more >

Sessions

HTTP is stateless

HTTP needs a state tracking mechanism to track a series of requests and identify users

- Cookies
- URI parameters
- Hidden form fields

Sessions

Cookies

- A cookie is a snippet of data sent from the server to the client
- It is stored in the browser
- It is then sent back to the server
- Option OnlyHTTP
  - Cookie set on server side only
- Option Secure
  - Use of cookie only with HTTPS

Sessions

URI

- Data is place in the URI and sent with an HTTP GET request

```
https://www.mywebsite.com/index.php/sessionid=15316
```

Hidden forms

- Forms are used for user input
- Hidden ones are not displayed to the user

```
<input type="hidden" name="login" value="neo">
```

Sessions

Attacker point of view

- Sessions are major targets
- They can be stolen using various techniques
- It gives access to account and can be used for privilege escalation
  - Also called impersonation → Attacking with someone else's account

Web Server Architecture

Different types of web architecture

- Web servers (standalone)
- Dynamic servers
- Application servers
- Microservices architecture

From an offensive perspective, knowing the web server architecture is essential

Web Server (Standalone) Architecture

Most of the time standalone web servers are used to serve static content only

- Not common nowadays

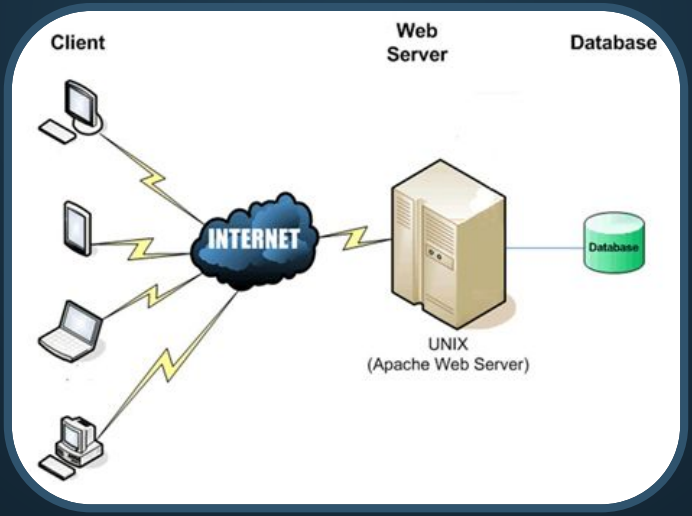
But often used by developers to make POC

- It can be directly on the developer machine, on a VM or a test environment
- Default credentials / misconfigurations / outdated software and framework
- Leads to code execution
- No need to perform lateral movement between the web application server and the database

Dynamic Web Server Architecture

Most common server architecture type

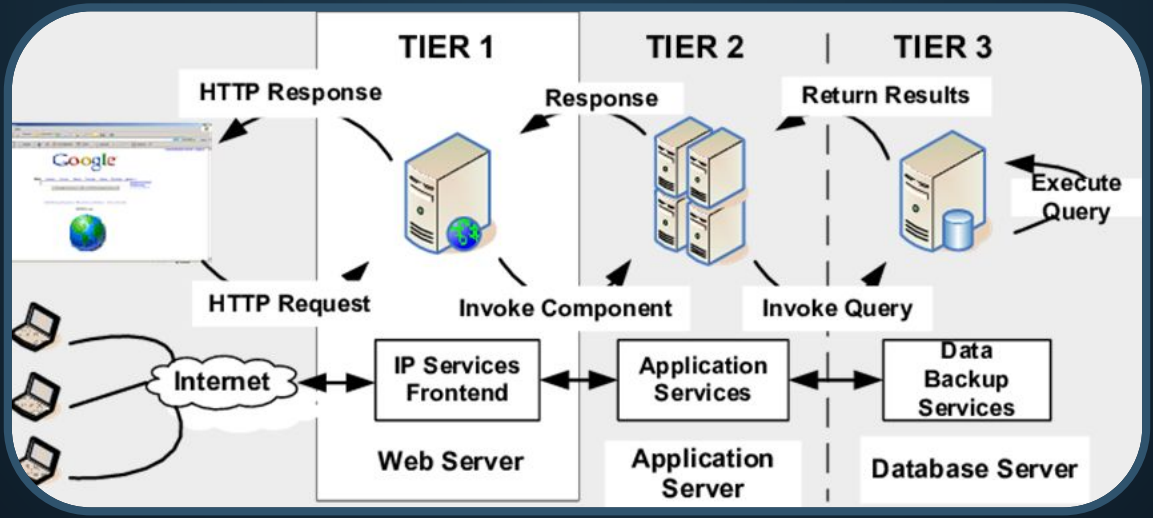
- It serves both static and active content
- The data are separated from the application server



3-Tier Web Server Architecture

Most common server architecture type

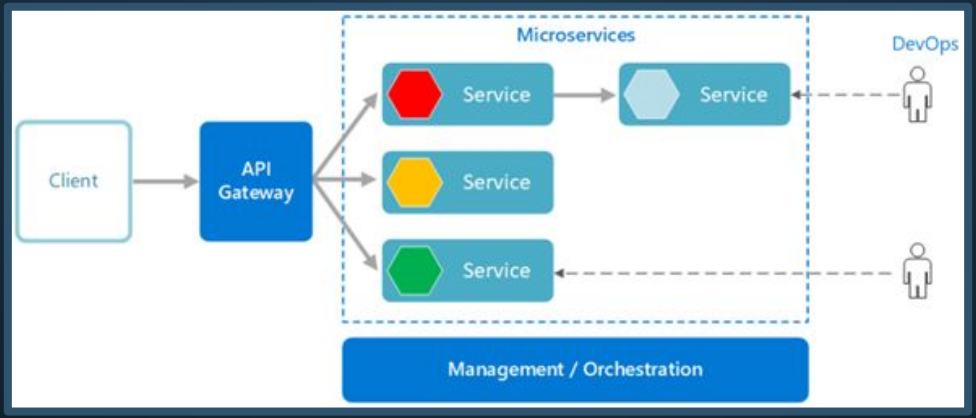
- Each tier runs on its own infrastructure (Presentation / Application / Data)
- Each tier can be developed simultaneously by a separate development team, and can be updated or scaled as needed without impacting the other tiers



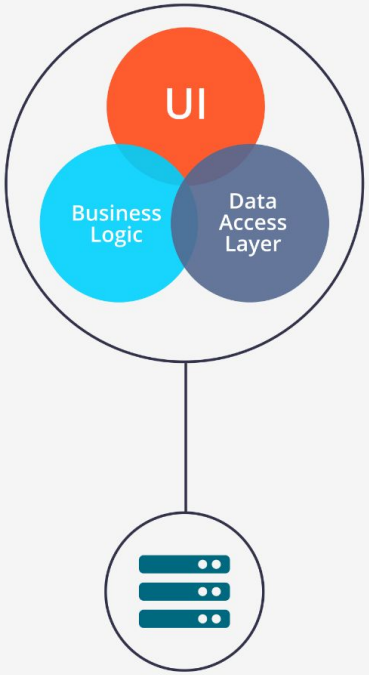
Microservices Architecture

Microservices structure an application as a collection of services that are

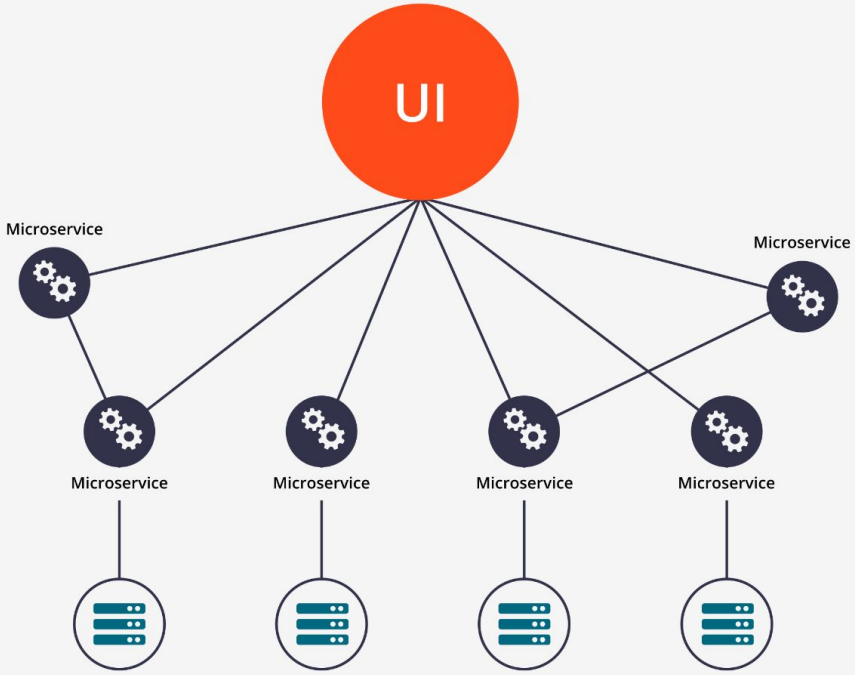
- Highly maintainable and testable
- Independently deployable
- Organized around specific uses



Monolithic vs Microservices Architecture



Monolithic Architecture



Microservice Architecture

Reverse Proxy and Load Balancer

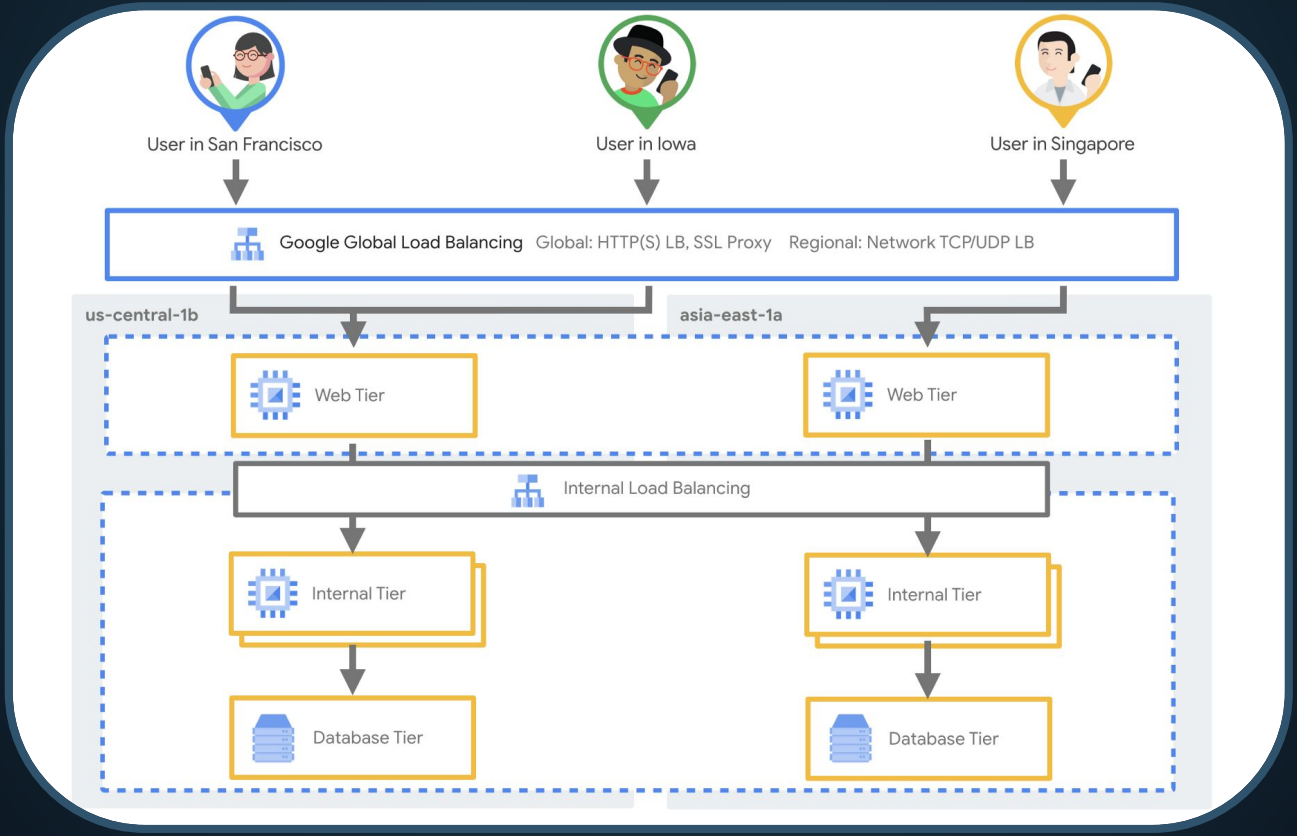
Reverse proxy

- Accepts a request from a client, forwards it to a server that can fulfill it, and returns the server's response to the client

Load balancer

- Distributes incoming client requests among a group of servers, in each case returning the response from the selected server to the appropriate client

Reverse Proxy and Load Balancer



Firewall and IDS / IPS

Firewall

- Blocks / filters ports
- Local and network firewalls should be configured

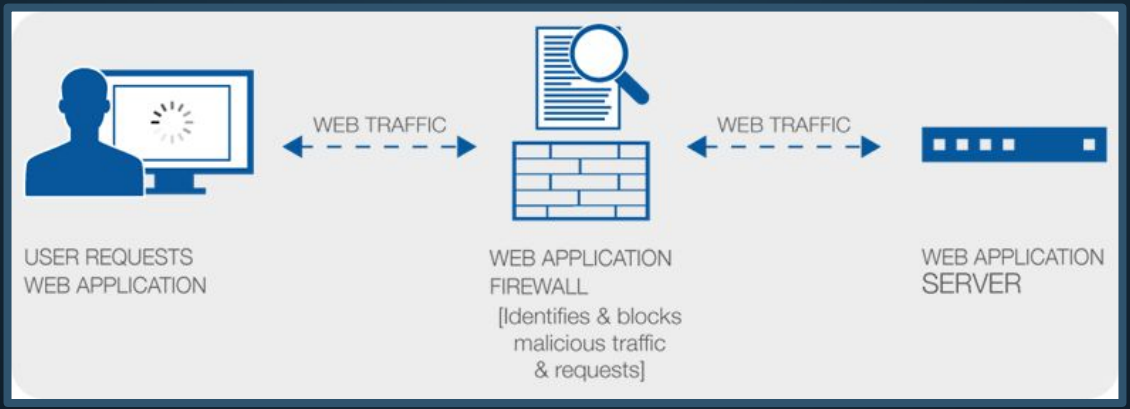
Intrusion and Detection / Prevention System (IDS / IPS)

- Detects (and blocks) network traffic
- Based on signatures
- Sensors can make SSL / TLS interception
- Some network firewalls have IDS / IPS module

Web Application Firewall

WAF

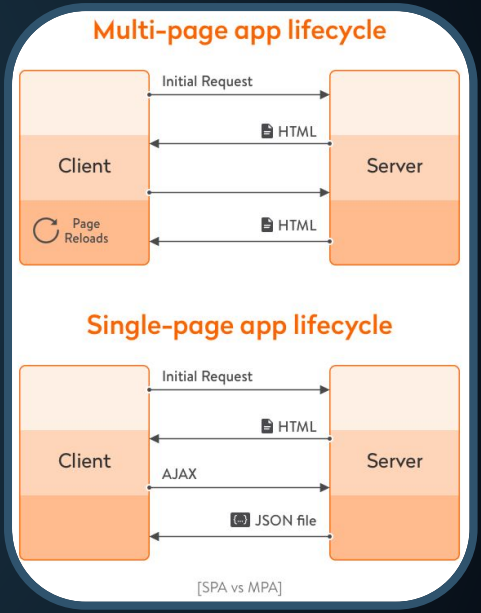
- Identifies and blocks malicious requests
- Throttle requests traffic (spidering)
- Both WAF and application must add security checks



Single Page Application

SPA re-renders its content in response for navigation actions without making a request to the server to fetch new HTML

- JavaScript frameworks
- Complex dynamic clients (JS libraries / HTML5)
- Backend Rest APIs specific to SPA



Attacker point of view

- Automatic scans will struggle
  - No spidering → Manually search and test API routes

Types of Flaw - Information Exposure

Found in response headers / stack trace / HTML code

Infrastructure information

- OS / DB / software / framework versions

Pathing

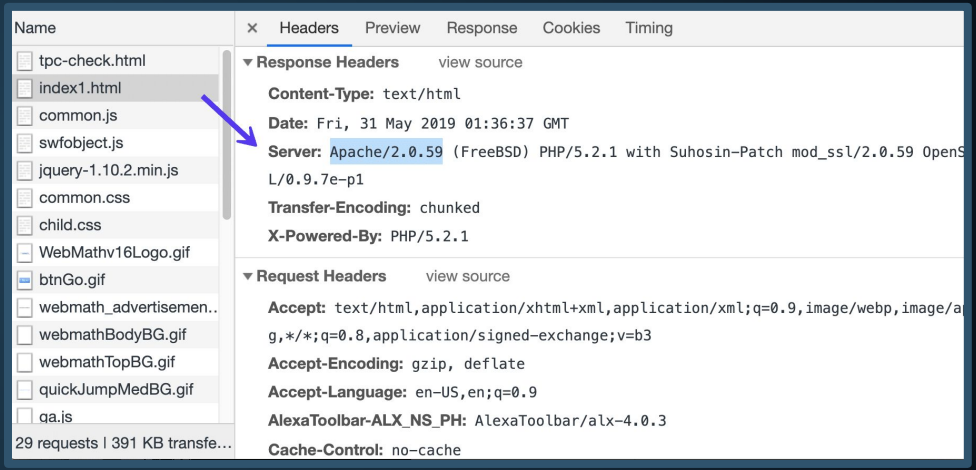
- Location of files on the server

Code and data

- Source code dump / data and data format

Credentials

- Reuse them



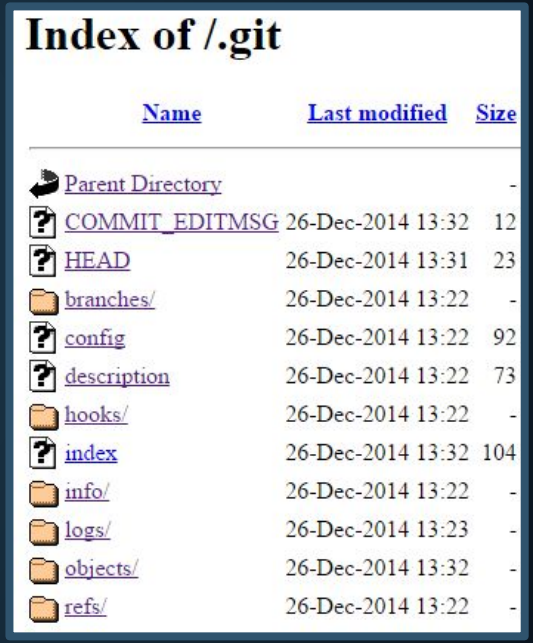
Types of Flaw - Configuration

Common oversight

- Default admin pages / configuration files / credentials
- Folders and files automatically created (temporary files)
- Application runs with inadequate privileges

Consequences

- Information and vulnerability exposure
- Presence of backdoors on the application or the OS
  - Admin panel not protected by a password (developer / debug mode)
  - Direct system compromise (SSH default credentials)



Index of /.git

Name	Last modified	Size
Parent Directory		-
COMMIT_EDITMSG	26-Dec-2014 13:32	12
HEAD	26-Dec-2014 13:31	23
branches/	26-Dec-2014 13:22	-
config	26-Dec-2014 13:22	92
description	26-Dec-2014 13:22	73
hooks/	26-Dec-2014 13:22	-
index	26-Dec-2014 13:32	104
info/	26-Dec-2014 13:22	-
logs/	26-Dec-2014 13:23	-
objects/	26-Dec-2014 13:32	-
refs/	26-Dec-2014 13:22	-

Types of Flaw - Bypass

Authentication bypass

- Interaction with the application without credentials

Authorization bypass

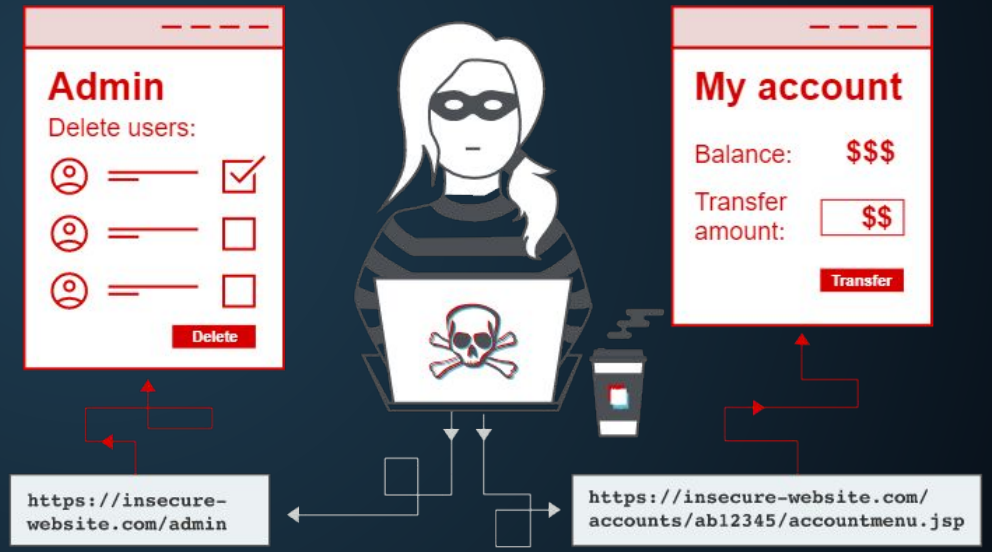
- Access to restricted ressources
- Privilege escalation

Front-end bypass

- Direct back-end interaction

File control bypass

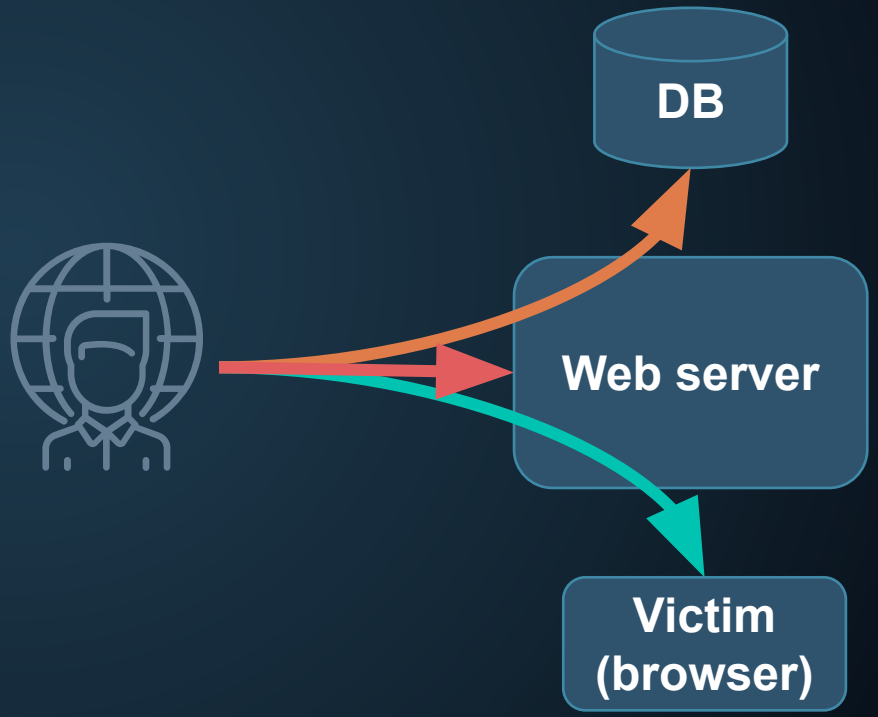
- Access to resources that are not linked to the application



Types of Flaw - Injection

Injection vulnerabilities are due to an improper input validation

- Command injection
- Code injection
- SQLi / NoSQLi / LDAPi
- Cross Site Scripting (XSS)
- Cross Site Request Forgery (CSRF)



Tooling

A consultant must always keep the control on the tools

- The tool does not replace the consultant's knowledge
- It is important to be able to reproduce any vulnerability manually

Pick the tool that suits your need

- It is more efficient and it speeds up the assessment

Aggressive vulnerability scanners can lead to DoS

- DoS is out of scope

Proof of Concept

Do not reinvent the wheel for each PoC

- There are many great tools, do not waste your time

Sometimes you will need to a make a script to prove the point

- Python / Bash / Java...

Some attacks are hard / impossible to automate

- Due to the vulnerability itself
  - Authorization tests (Authorization matrix)
  - Custom business features
- Due to the need of too many requests (request throttling)

How to Skill Up as a Student?

By practicing on platforms

- With course and write-up
  - Portswigger
  - Tryhackme
- Without write-up
  - Root-Me
  - Hack The Box

By understanding attacks

- Web course
  - Portswigger
- Blogs / articles / Twitter
- Podcasts / Conferences
  - Nolimitsecu / YouTube

Conclusion

1

Class Presentation

What is this class about?

2

General Overview

Why a web security course?

3

Penetration Test

What is penetration test?

4

Web

How the web works?

Questions

THANKS!