

Nizar Session

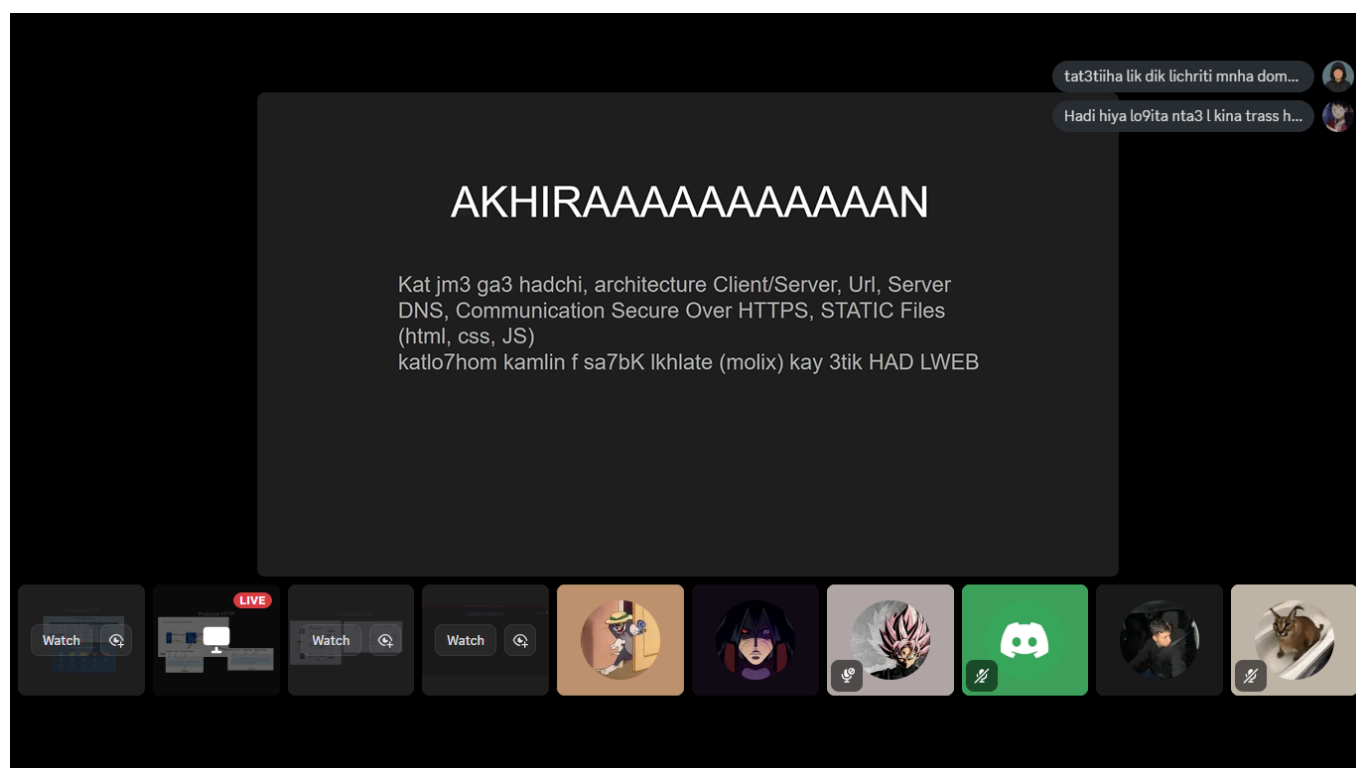
DNS

- In a URL, we have TLD (Domaine de premier niveau) (Top Level domain), it is important. (It is usually the final part of the url (.org, .ma, .com, ...))
- You can see all DNS with this website : <https://securitytrails.com/>

HTTPS

- Fun Fact, HTTPS = HTTP + SSL Encryption.
- SSL is used with a certification, SSL certification.

In general

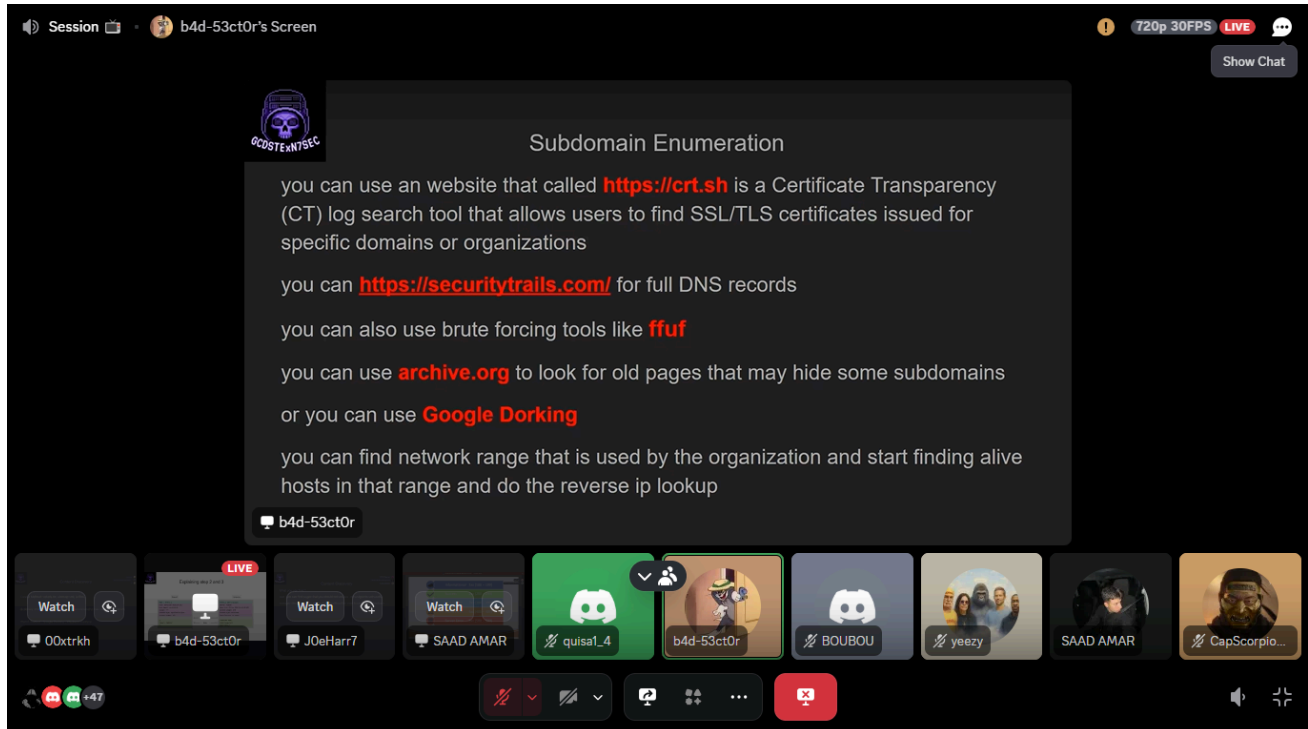


OSINT

- Google dorking,

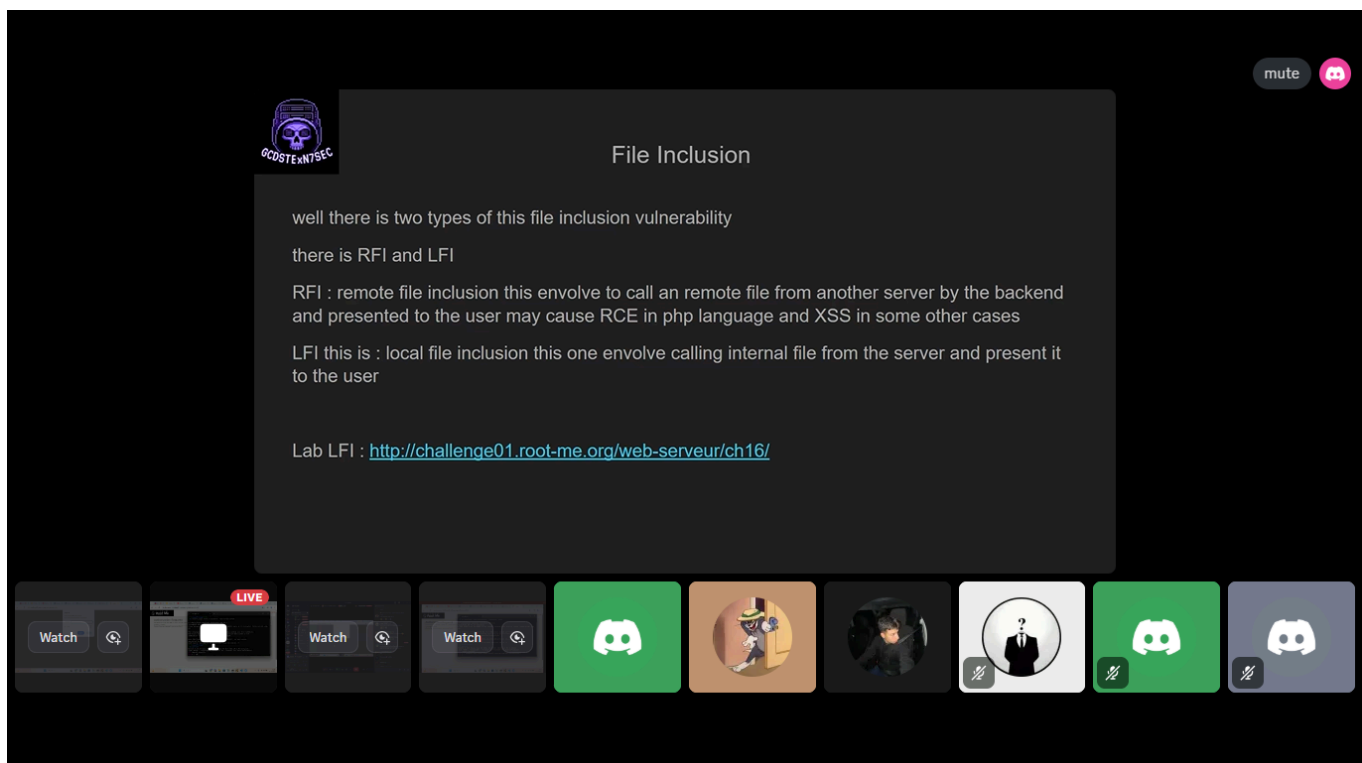
also known as Google hacking, is a technique that uses advanced Google Search queries to find security vulnerabilities in websites. It involves using specific keywords and operators to

locate exposed information or misconfigured systems, such as finding sensitive files or login pages that were not intended to be public.



It's here where we use the Authentication ByPass

- File Inclusion:



RFI and LFI are both **file inclusion vulnerabilities** in web applications that occur due to improper input validation, but they differ in the **location of the files** an attacker can include.

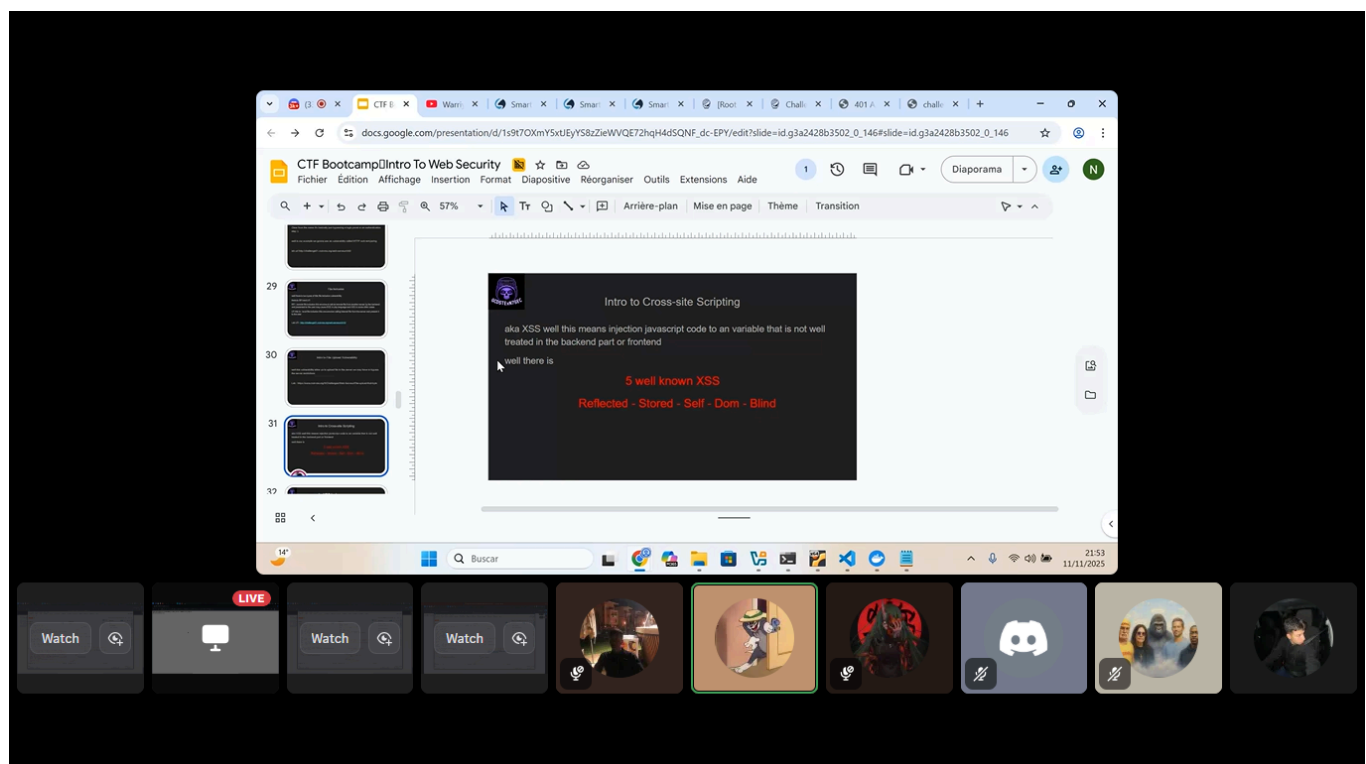
RFI: Remote File Inclusion

- **Definition:** RFI allows an attacker to include files hosted on a **remote server** (an external source controlled by the attacker) into the vulnerable web application.
- **Mechanism:** The web application fetches and executes the attacker's remote malicious code, typically specified via a URL in a user-supplied input parameter.
- **Severity:** RFI is generally considered more severe than LFI because it can immediately lead to **remote code execution (RCE)** on the target server, allowing the attacker to run arbitrary commands and potentially compromise the entire system.
- **Prerequisites:** A successful RFI attack often requires specific server configurations, such as the `allow_url_include` setting being enabled in PHP.

LFI: Local File Inclusion

- **Definition:** LFI allows an attacker to include and access files that are already present on the **local server** where the application is running.
- **Mechanism:** Attackers manipulate file paths using techniques like directory traversal (e.g., `../`) in user input to access sensitive local files (like configuration files, source code, or system logs) that were not intended to be exposed.
- **Severity:** While LFI primarily leads to **sensitive information disclosure**, it can sometimes be escalated to RCE if the attacker can find a way to place controlled data (like through a log file or an insecure upload function) on the server and then include it.

Cross-site scripting (XSS):



why XSS is dangerous

Stealing Information: Attackers can steal any information your users type into the site. This includes usernames, passwords, credit card numbers, and personal details like names and addresses.

Taking Over User Accounts: An attacker can "hijack" a user's login session. This means they can act as that user, change their password, access their private messages, make purchases, or transfer money without the user ever knowing.

Spreading Malware: The attacker can use your website to show pop-ups that trick users into downloading viruses, spyware, or ransomware onto their own computers.

Tricking Your Users (Phishing): An attacker can change parts of your website to show a fake login form. When a user tries to log in, their password is sent directly to the attacker.

Changing Your Website (Defacement): Attackers can change what your website looks like. They could add offensive content, delete pages, or post their own messages, making your site look unprofessional or broken.

The slide is part of a presentation with a navigation bar at the bottom showing various icons and a 'LIVE' indicator.

- It can be used to steal cookies and stuff

XSS (Stored)

The attack is permanently saved on the website for all visitors to see.

How it works: An attacker posts a malicious comment on a blog post or a product review: "Nice article! `<script>stealEveryoneCookies()</script>`"

What happens: The website saves this dangerous comment to its database. Now, every single person (including admins) who views that blog post will have the malicious script run in their browser, potentially stealing everyone's login session.

The slide is part of a presentation with a navigation bar at the bottom showing various icons and a 'LIVE' indicator. A chat bubble in the top right corner shows a message: "z3ma bə9i had xss".

Session b4d-53ct0r's Screen 720p 30FPS LIVE

XSS (blind)

Explanation: This is when an attacker injects a script that runs on a private, back-end system (like an admin's control panel) that the attacker themselves cannot see.

b4d-53ct0r

Watch 1SM41L b4d-53ct0r JOeHarr7 SAAD AMAR b4d-53ct0r irfanWH GouloJk yeezy SAAD AMAR Bushiway

- Command Injection

Session b4d-53ct0r's Screen 720p 30FPS LIVE

Command Injection

```
<?php
// Get an IP address from the user (e.g., ping.php?ip=8.8.8.8)
$ip = $_GET['ip'];

// VULNERABLE: The user's input is put directly into the command!
$output = shell_exec("ping -c 1 " . $ip);

// Show the output to the user
echo "<pre>" . $output . "</pre>";
?>
```


b4d-53ct0r

Watch 00xtrkh 1SM41L b4d-53ct0r JOeHarr7 SAAD AMAR b4d-53ct0r mess3oudd yeezy GouloJk SAAD AMAR

kmal Good dz

- SQL Injection

Session - b4d-53ct0r's Screen 720p 30FPS LIVE



SQL Injection

```
<?php
// Get user input (BAD: not sanitized)
$user = $_POST['username'];
$pass = $_POST['password'];

// VULNERABLE: Building the query by concatenating strings
$sql = "SELECT * FROM users WHERE username = '$user' AND password = '$pass'";
// ...then the code runs this $sql query against the database...
?>
```

b4d-53ct0r

Watch 00xtrkh

Watch 1Sm41L

LIVE b4d-53ct0r

Watch JOeHarr7

Watch SAAD AMAR

b4d-53ct0r





MR. KATANA Mr.katana


mess3oudd



GouloJk

SAAD AMAR

+29







✓ Showing rows 0 - 0 (1 total, Query took 0.0011 seconds.)

```
SELECT * FROM users WHERE username='kiki' union select null,"nono" as username,"fofo" as password , null;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

id	username	password	reg_date
NULL	nono	fofo	NULL

- onlinephp.io/password-hash