Group no :7

Name: VUPPU RAKESH RAVINDRA

Process of Bug bounty

**Step 1)** The target for bug hunting is chosen from websites and web applications listed on OpenBugBounty.

The Target system selected from openbugbounty Programs is:

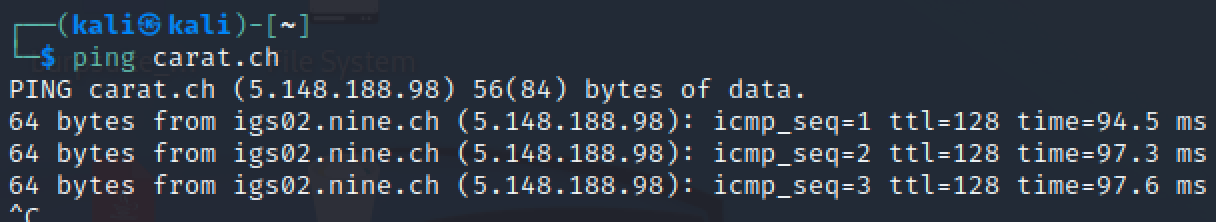
* <https://www.carat.ch>

# Reconnaissance (Information Gathering)

**Step 2)**

Firstly, we verify the connection by sending a ping to the target machine.

And we Found Ip for target machine is 5.148.188.98



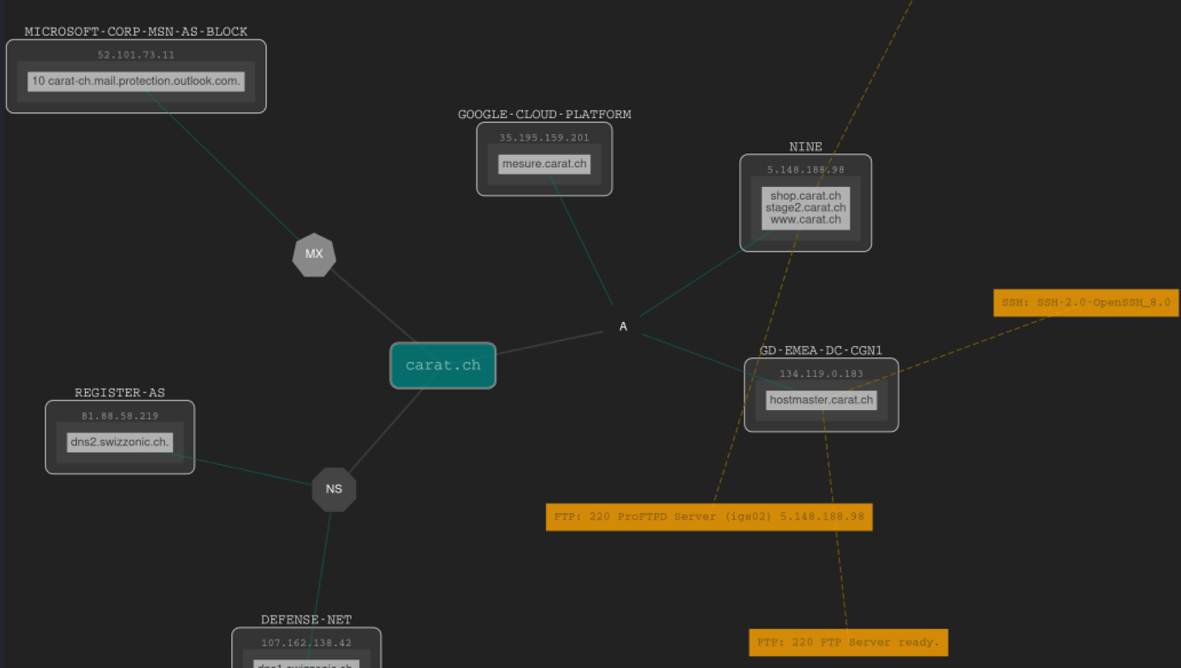
**Step 3)**

DNSDumpster provides a visual representation of the DNS (Domain Name System) information related to a domain. This includes subdomains, associated IP addresses, and other DNS records.

We found these subdomains:

* Shop.carat.ch
* Stage2.carat.ch
* [www.carat.ch](http://www.carat.ch)

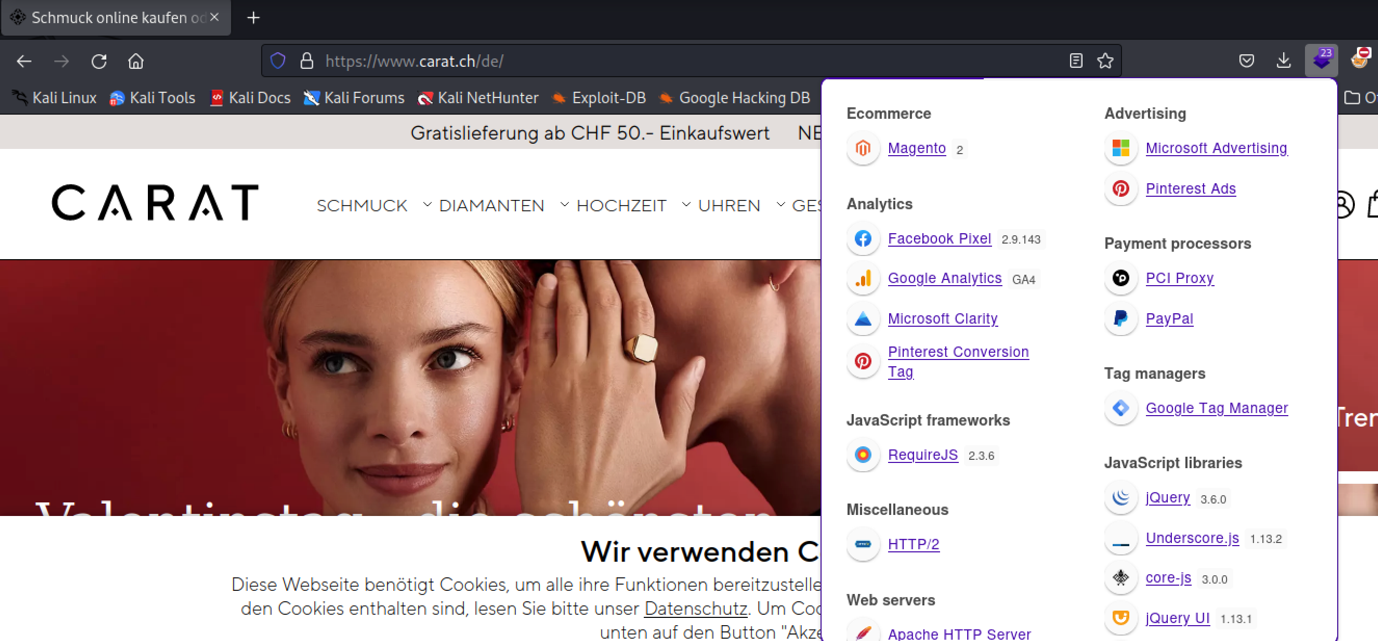
**but we have only** [**www.carat.ch**](http://www.carat.ch) **in scope.**



**Step 4)**

Identify the target application and gather relevant details. we’ve extracted this Information by using tools whois, wappalyzer, Host, builtwith, and curl.

* + Address: 5.148.188.98
  + Identify web server: Apache HTTP Server
  + E-commerce: Megento
  + Database MySQL

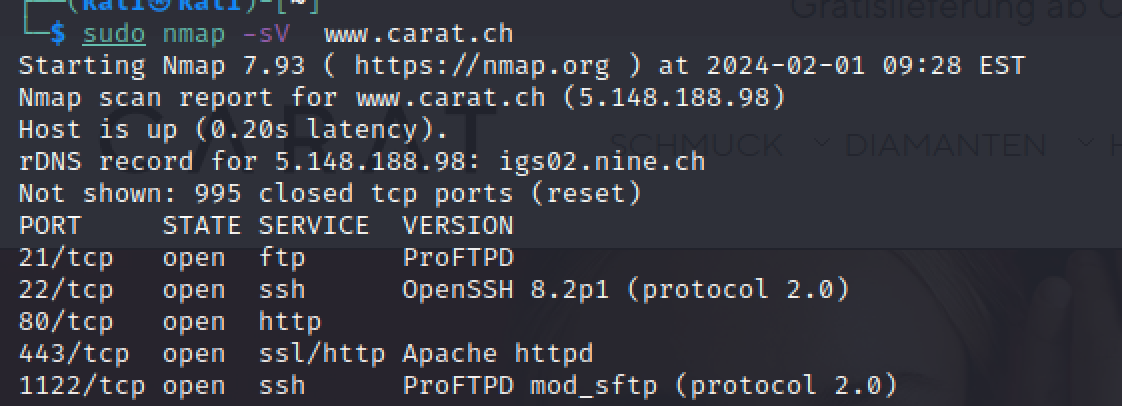


# Scanning

**Step 5)**

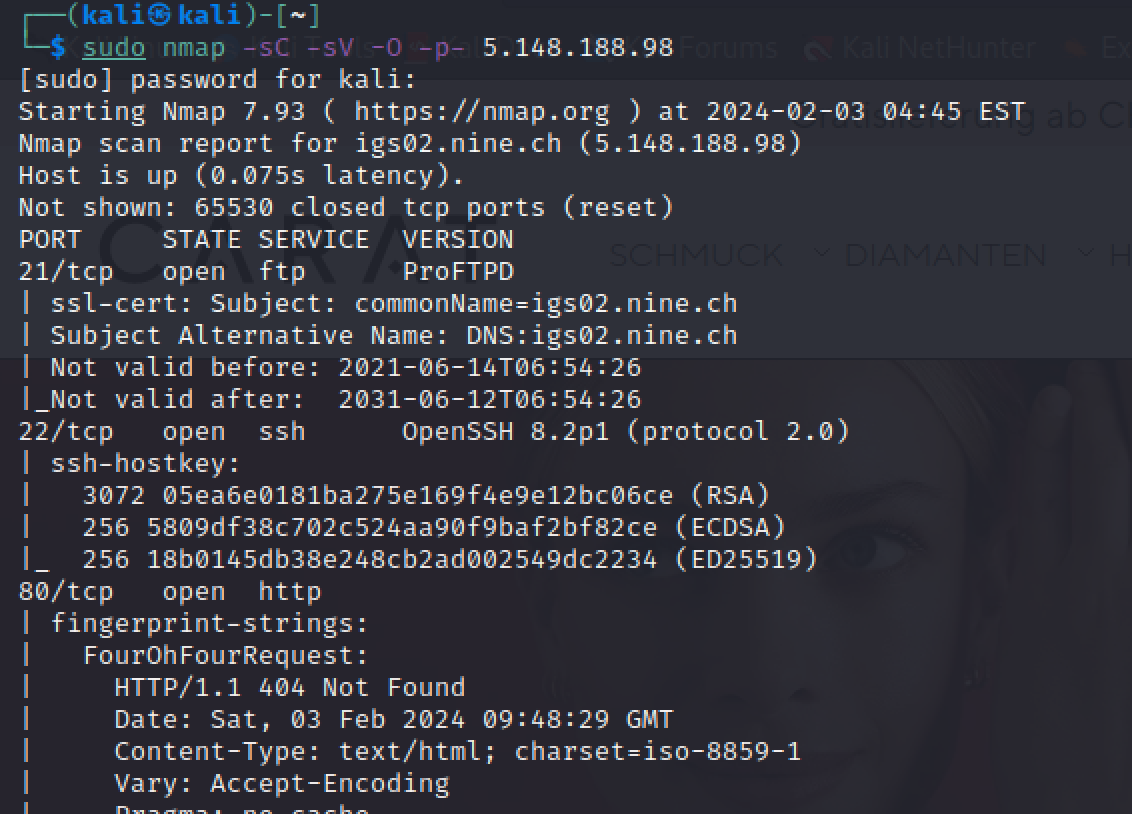
Performing port scan on the target machine using namp.

Open Ports and services running on them listed below :

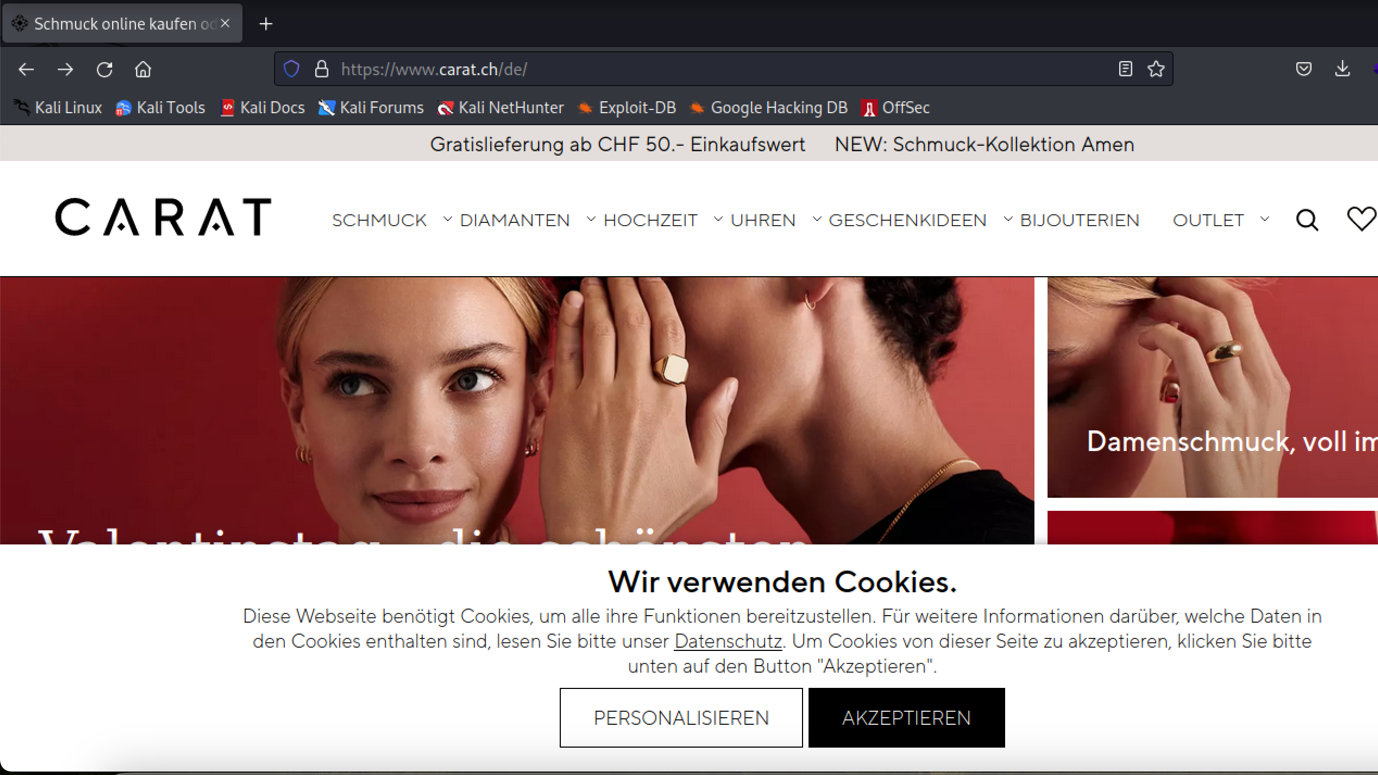


*Nmap of carat.ch scan for open ports & services*

* Port 80(HTTP) is running and an apache server is running on port 443.
* Also, port 1122 is also open for SSH.
* Port 21 is open for ftp
* Port 22 open for ssh

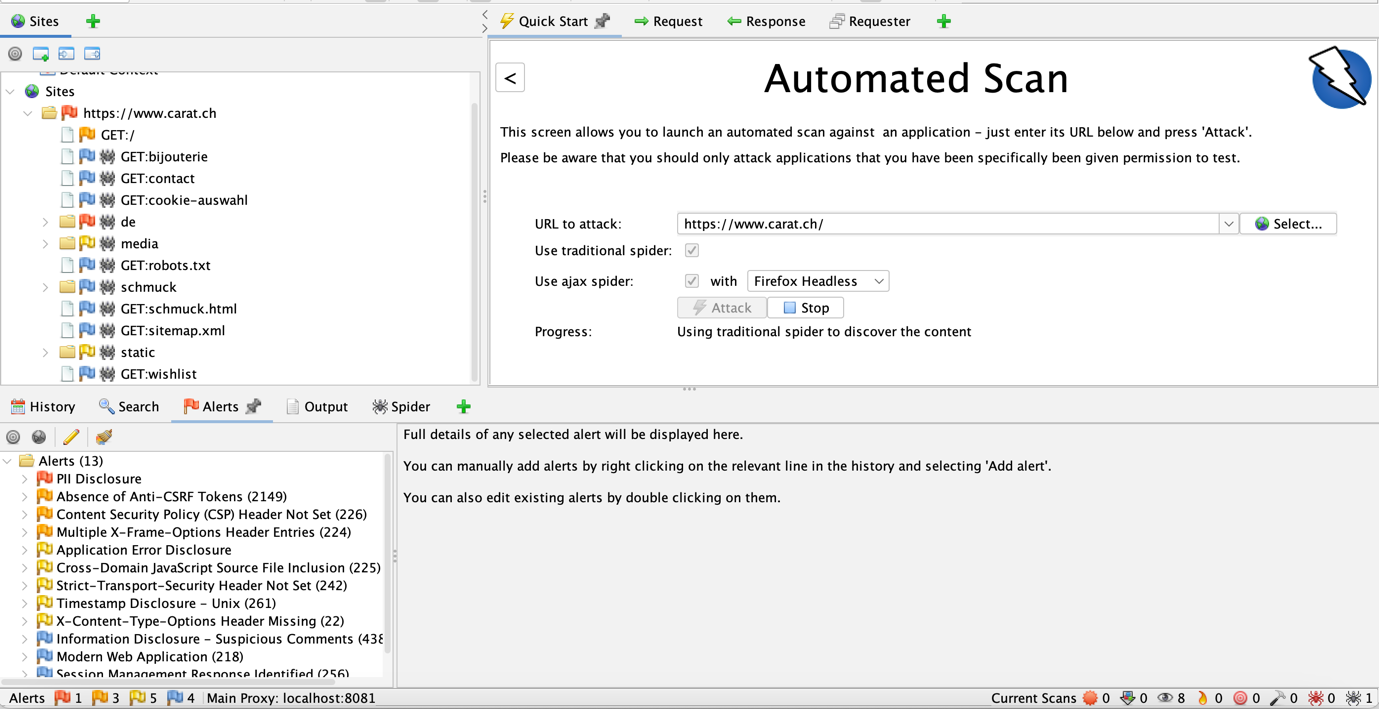
****

**Step 6)** Checking the web service running on target machine

****

Upon inspecting the page, we can find “Search” input fields on the website.

**Step 7)** We will scan the website using OWASP ZAP to look for potential vulnerabilities.



Upon scanning, we can discover these vulnerabilities.

* PII Disclosure                        HIGH
* Absence of Anti-CSRF Tokens MEDIUM
* Information Disclosure - Suspicious Comment                 MEDIUM
* Content Security Policy (CSP) Header Not Set MEDIUM

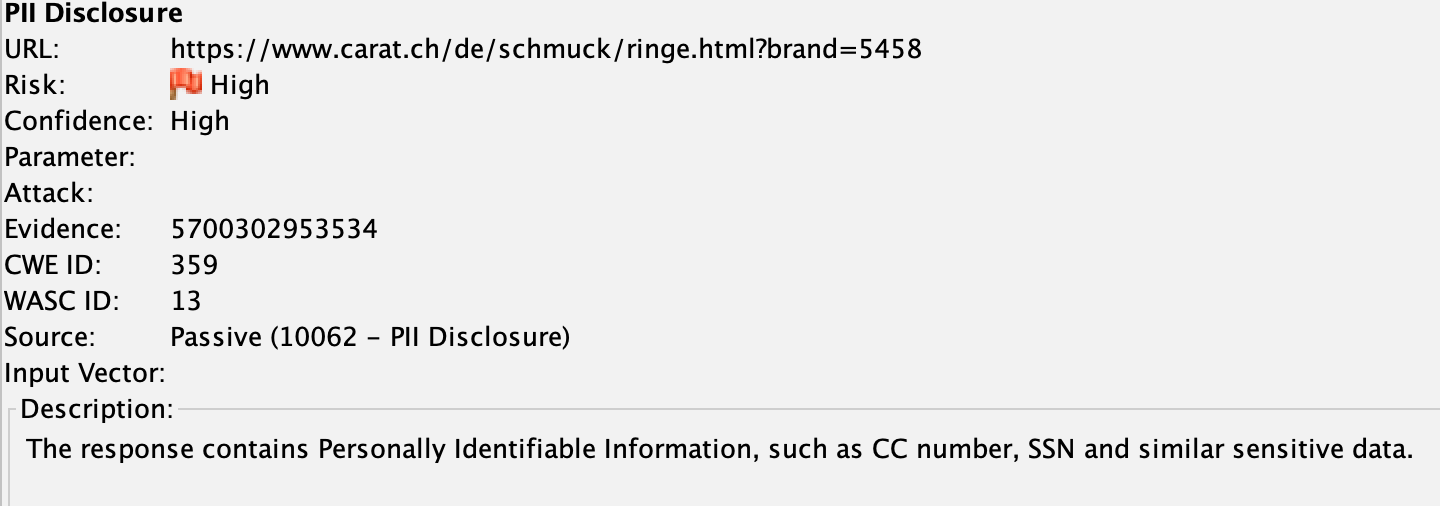
# PII Disclosure: HIGH

**Description:**

PII (Personally Identifiable Information) disclosure occurs when sensitive personal information about individuals is exposed or improperly handled, posing a risk to their privacy and security. This can include details such as names, addresses, social security numbers, financial data, and other information that can be used to identify or target individuals.

POC:





## Remediation:

Minimize data collected, employ robust encryption, and enforce strict access controls to mitigate PII disclosure risks.

# Absence of Anti-CSRF Tokens MEDIUM

The form on the website doesn't have a security feature called Anti-CSRF tokens. These tokens act like safety checks to make sure that actions on the website are done by you and not by someone else trying to trick you.

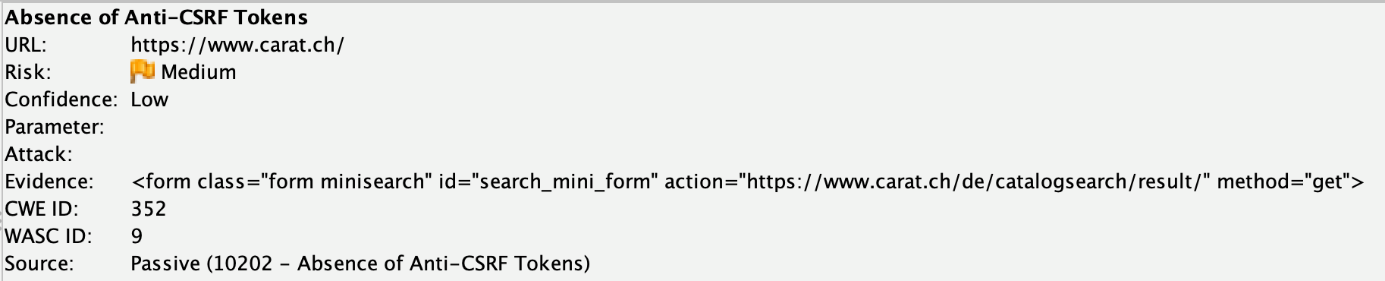
Without these tokens, there's a risk that someone could make you do things on the website without you knowing.

This is a common problem, especially if you're logged in, authenticated, or on the same network as the website.

Remediation:

To stay safe, websites should use Anti-CSRF tokens to protect users from these sneaky tricks.

POC:





# Information Disclosure - Suspicious Comment

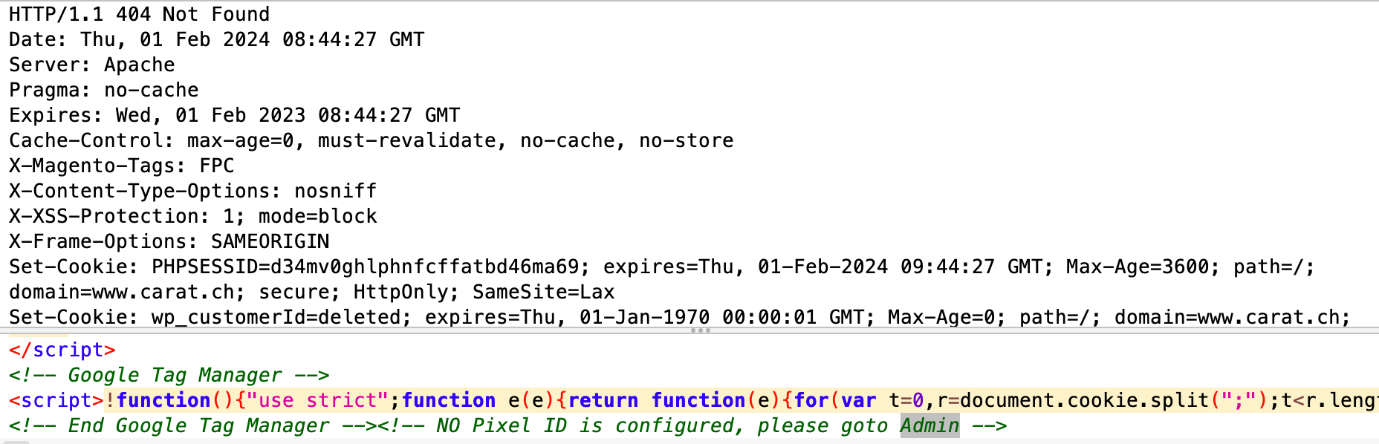
The response appears to contain suspicious comments which may help an attacker.

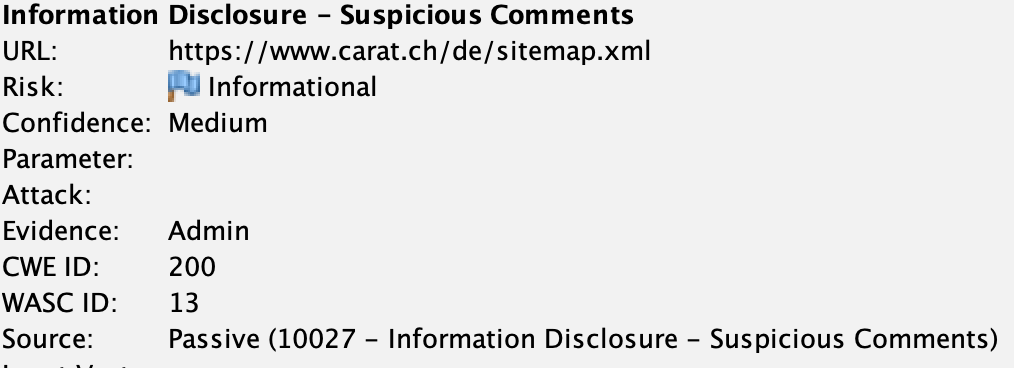
Description:

Information Disclosure occurs when a system unintentionally reveals sensitive data to unauthorized individuals. This can involve exposing details about the system's configuration, user accounts, or other confidential information.

Remediation:

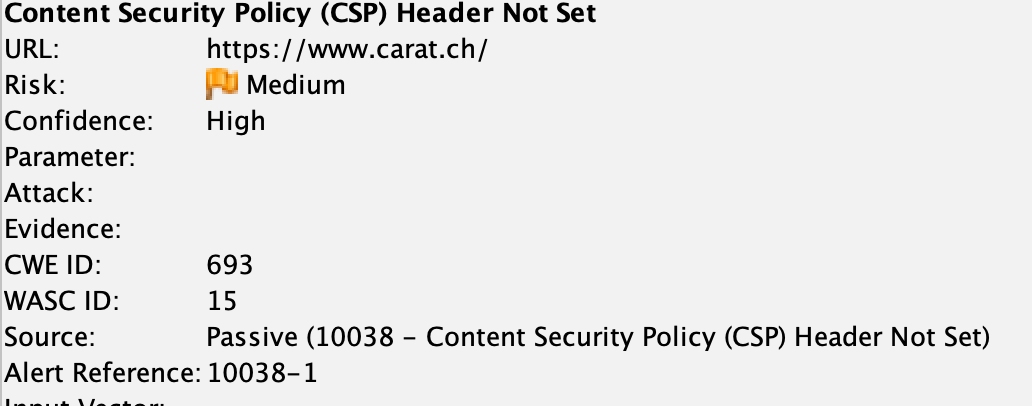
Mitigating Information Disclosure involves managing error messages, configuring access controls, and regularly auditing systems to prevent the exposure of sensitive information.





# Content Security Policy (CSP) Header Not Set

Content Security Policy (CSP) is like a protecting shield for websites. CSP acts like a rulebook for the browser, telling it where it's okay to get stuff like pictures, videos, or special web powers (JavaScript). This ensures the browser only listens to trusted sources, keeping your website safe from troublemakers.



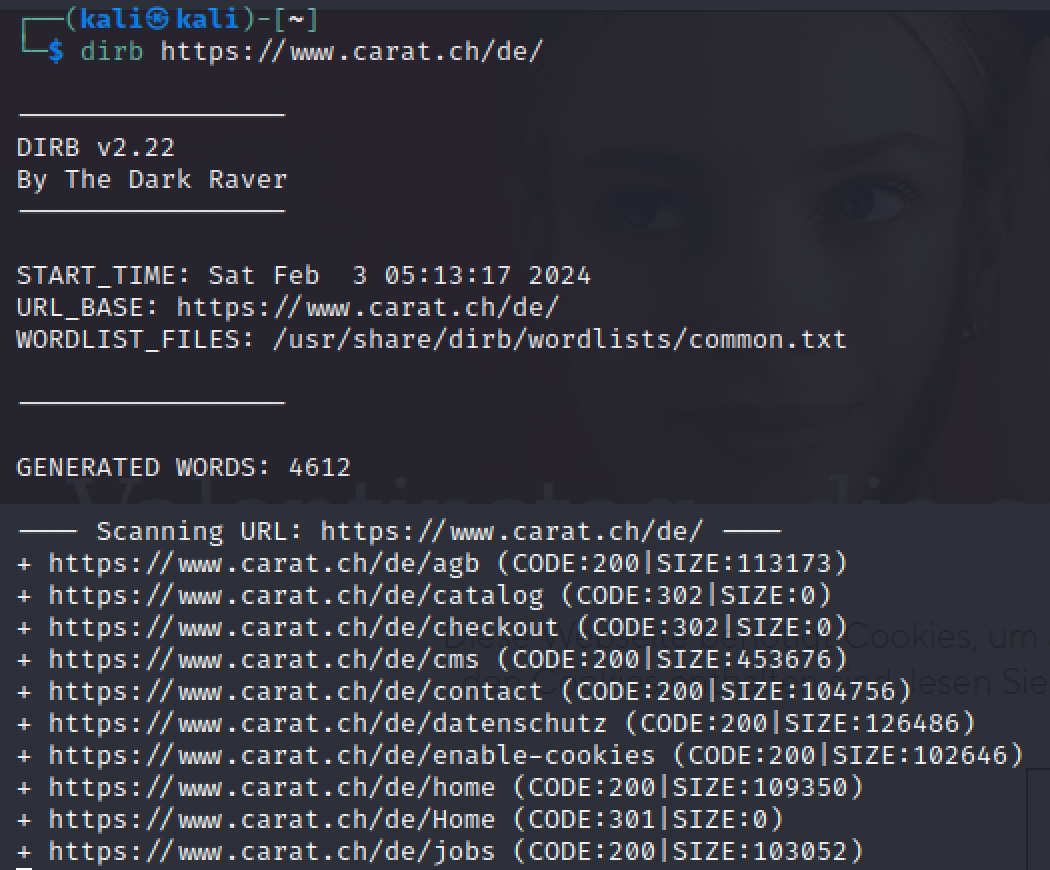
# Manual Testing

* Directories Enumeration
* Insufficient server-side Input sanitization
* No rate of limit implemented on search(fuzzing Applied)

**Step 8)** Checking for /hidden/ directory

Here we can discover three user based on this message

* <https://www.carat.ch/de/robots.txt>
* https://www.carat.ch/de/agb
* https://www.carat.ch/de/cms
* https://www.carat.ch/de/contact
* https://www.carat.ch/de/datenschutz
* https://www.carat.ch/de/enable-cookies
* https://www.carat.ch/de/home
* https://www.carat.ch/de/jobs

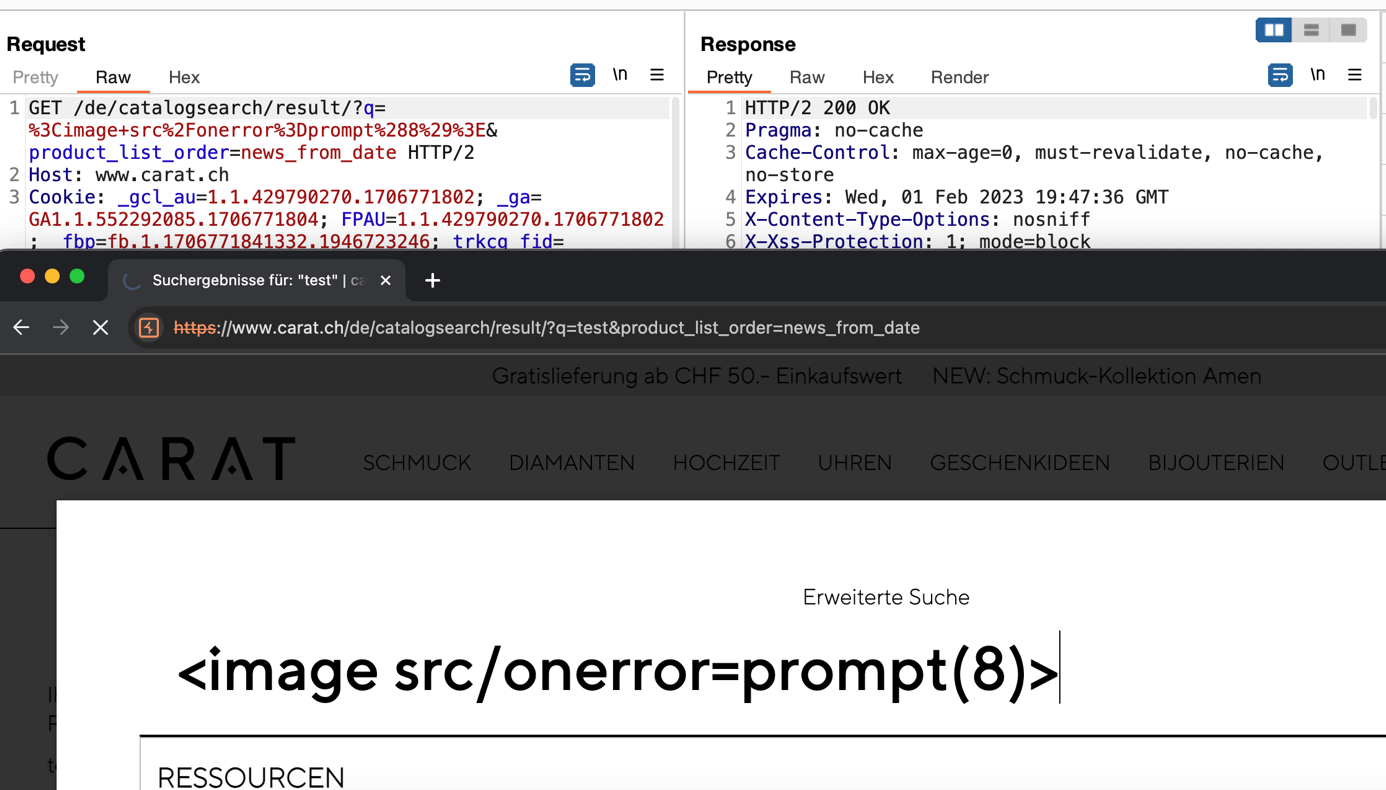


# **Step 9)** Looking for vulnerabilities by Manual testing:-

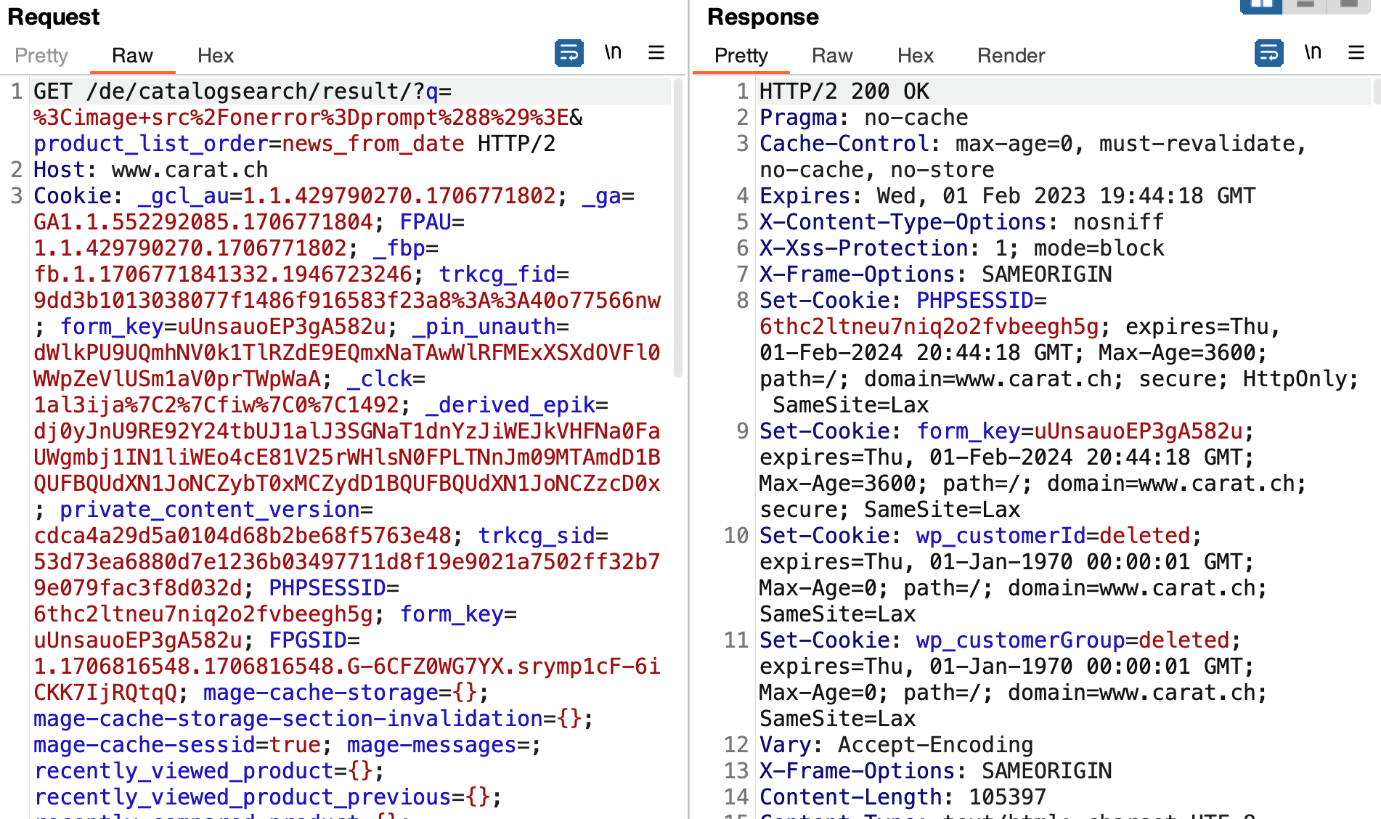
# Insufficient server-side Input sanitization

Improper validation and sanitization of input on the server side pose a security vulnerability in web applications. When a web application fails to thoroughly validate and cleanse user-provided data, it opens the door to potential cyber threats such as injection attacks, cross-site scripting (XSS), and unauthorized data access.

POC



*Insufficient server-side input sanitization*



## Recommendations:

* Implement thorough input validation and sanitization.
* Use parameterized queries or prepared statements for database interactions.
* Apply output encoding to prevent XSS attacks.

**Step 10)**

# No Rate Limiting (fuzzing Applied on search bar)

The "Unlimited Search Requests" vulnerability in a web application indicates a lack of restrictions on the number of search requests a user can make within a set timeframe.

This gap in security can result in misuse, including brute force attacks and resource exhaustion, potentially causing performance issues and impacting the application's availability.

## Recommendation:

Establish strong rate-limiting mechanisms to manage the number of search requests a user can initiate, preventing misuse and bolstering overall security. Regularly monitor and analyze traffic patterns, and perform security audits to verify the efficacy of rate-limiting controls.