Web Security – IE2062

5. GitLab _Bug Bounty

A D A Ihansa

IT22899606

Web Audit

gitlab.com

Contents

Introduction to Bug Bounty program and audit scope

GitLab.com streamlines software development by offering all the necessary tools in one place. It brings together planning, coding, testing, and deployment in a single platform, making collaboration and efficiency a priority. By using automation and AI, GitLab helps teams develop software faster and at a higher quality. Since it's built on an open-source project and welcomes contributions from developers around the world, GitLab fosters a collaborative development environment.

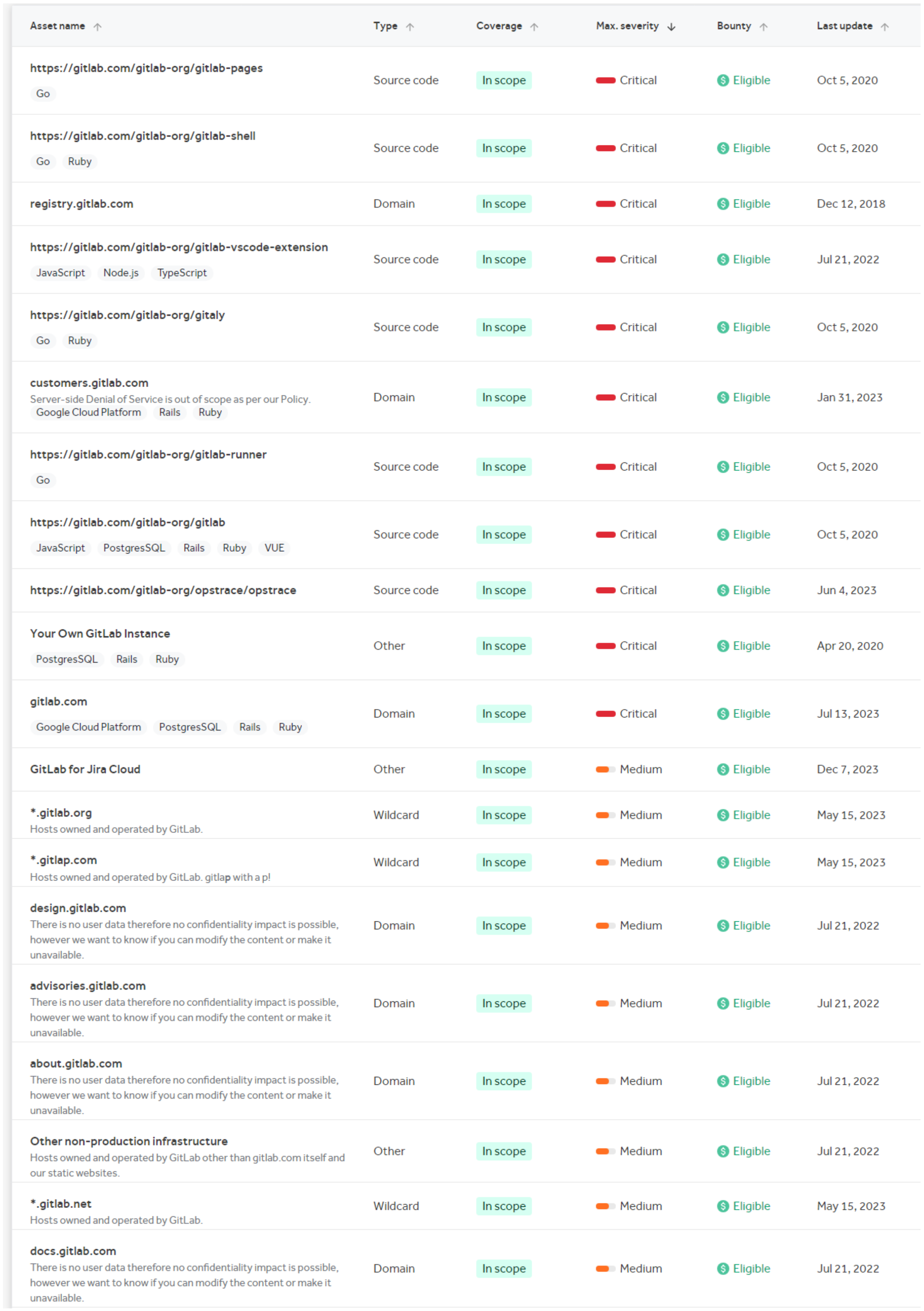
In hackerone bug bounty program, they defined these subdomains (and all inclusive) as valid subdomains for testing.

gitlab.com

registry.gitlab.com

about.gitlab.com

The bug bounty program allows you to report vulnerabilities on any website that's part of the GitLab.com domain. To put it simply, any address that starts with "gitlab.com" (and anything that follows) is considered in scope for the program.



Asset name	Type	Coverage	Max. severity	Bounty	Last update
https://gitlab.com/gitlab-org/gitlab-pages Go	Source code	In scope	Critical	Eligible	Oct 5, 2020
https://gitlab.com/gitlab-org/gitlab-shell Go Ruby	Source code	In scope	Critical	Eligible	Oct 5, 2020
registry.gitlab.com	Domain	In scope	Critical	Eligible	Dec 12, 2018
https://gitlab.com/gitlab-org/gitlab-vscode-extension JavaScript Node.js TypeScript	Source code	In scope	Critical	Eligible	Jul 21, 2022
https://gitlab.com/gitlab-org/gitaly Go Ruby	Source code	In scope	Critical	Eligible	Oct 5, 2020
customers.gitlab.com Server-side Denial of Service is out of scope as per our Policy. Google Cloud Platform Rails Ruby	Domain	In scope	Critical	Eligible	Jan 31, 2023
https://gitlab.com/gitlab-org/gitlab-runner Go	Source code	In scope	Critical	Eligible	Oct 5, 2020
https://gitlab.com/gitlab-org/gitlab JavaScript PostgresSQL Rails Ruby VUE	Source code	In scope	Critical	Eligible	Oct 5, 2020
https://gitlab.com/gitlab-org/opstrace/opstrace	Source code	In scope	Critical	Eligible	Jun 4, 2023
Your Own GitLab Instance PostgresSQL Rails Ruby	Other	In scope	Critical	Eligible	Apr 20, 2020
gitlab.com Google Cloud Platform PostgresSQL Rails Ruby	Domain	In scope	Critical	Eligible	Jul 13, 2023
GitLab for Jira Cloud	Other	In scope	Medium	Eligible	Dec 7, 2023
***.gitlab.org** Hosts owned and operated by GitLab.	Wildcard	In scope	Medium	Eligible	May 15, 2023
***.gitlap.com** Hosts owned and operated by GitLab. gitlap with a p!	Wildcard	In scope	Medium	Eligible	May 15, 2023
design.gitlab.com There is no user data therefore no confidentiality impact is possible, however we want to know if you can modify the content or make it unavailable.	Domain	In scope	Medium	Eligible	Jul 21, 2022
advisories.gitlab.com There is no user data therefore no confidentiality impact is possible, however we want to know if you can modify the content or make it unavailable.	Domain	In scope	Medium	Eligible	Jul 21, 2022
about.gitlab.com There is no user data therefore no confidentiality impact is possible, however we want to know if you can modify the content or make it unavailable.	Domain	In scope	Medium	Eligible	Jul 21, 2022
Other non-production infrastructure Hosts owned and operated by GitLab other than gitlab.com itself and our static websites.	Other	In scope	Medium	Eligible	Jul 21, 2022
***.gitlab.net** Hosts owned and operated by GitLab.	Wildcard	In scope	Medium	Eligible	May 15, 2023
docs.gitlab.com There is no user data therefore no confidentiality impact is possible, however we want to know if you can modify the content or make it unavailable.	Domain	In scope	Medium	Eligible	Jul 21, 2022

Information gathering phase.

The initial phase of information gathering, commonly known as reconnaissance or recon, is crucial for obtaining insights into the nature and behavior of the target. This phase holds significant importance during audits or attacks as it facilitates the identification of potential vulnerabilities by gaining a deeper understanding of the target.

There are two main methods for conducting information gathering scans:

1. Active Scanning: This method involves generating substantial activity on the target system, often resulting in the retrieval of extensive information.

2. Passive Scanning: In contrast to active scanning, this approach minimizes disruption to the target system, albeit typically providing fewer comprehensive results compared to active scanning.

In bug bounty programs, where details about the underlying architecture of systems are usually not disclosed (referred to as black box pentesting), specific tools and techniques are essential for gathering insights into their services, devices, and exposed information. This enables testers to develop a better understanding of the systems they are assessing.

Finding active subdomains and their states

Sublist3r

Sublist3r, a Python-based tool, is designed to discover subdomains associated with a specified target website. Leveraging search engines and online web services, it scours the web for available subdomains linked to the designated target domain. Given the freedom to scrutinize any subdomain under reddit.com, it's prudent to identify additional subdomains for testing purposes.

To install Sublist3r, navigate to its GitHub repository at https://github.com/aboul3la/Sublist3r.git. This repository hosts all the necessary files required for installing the tool. Execute the following command in your shell to download it:

```
git clone https://github.com/aboul3la/Sublist3r.git
```

Please note that Sublist3r necessitates either Python 2.7 or Python 3.4 to operate smoothly.

After downloading the files, go inside the ‘Sublist3r’ directory and install the requirements by entering,

sudo pip install -r requirements.txt

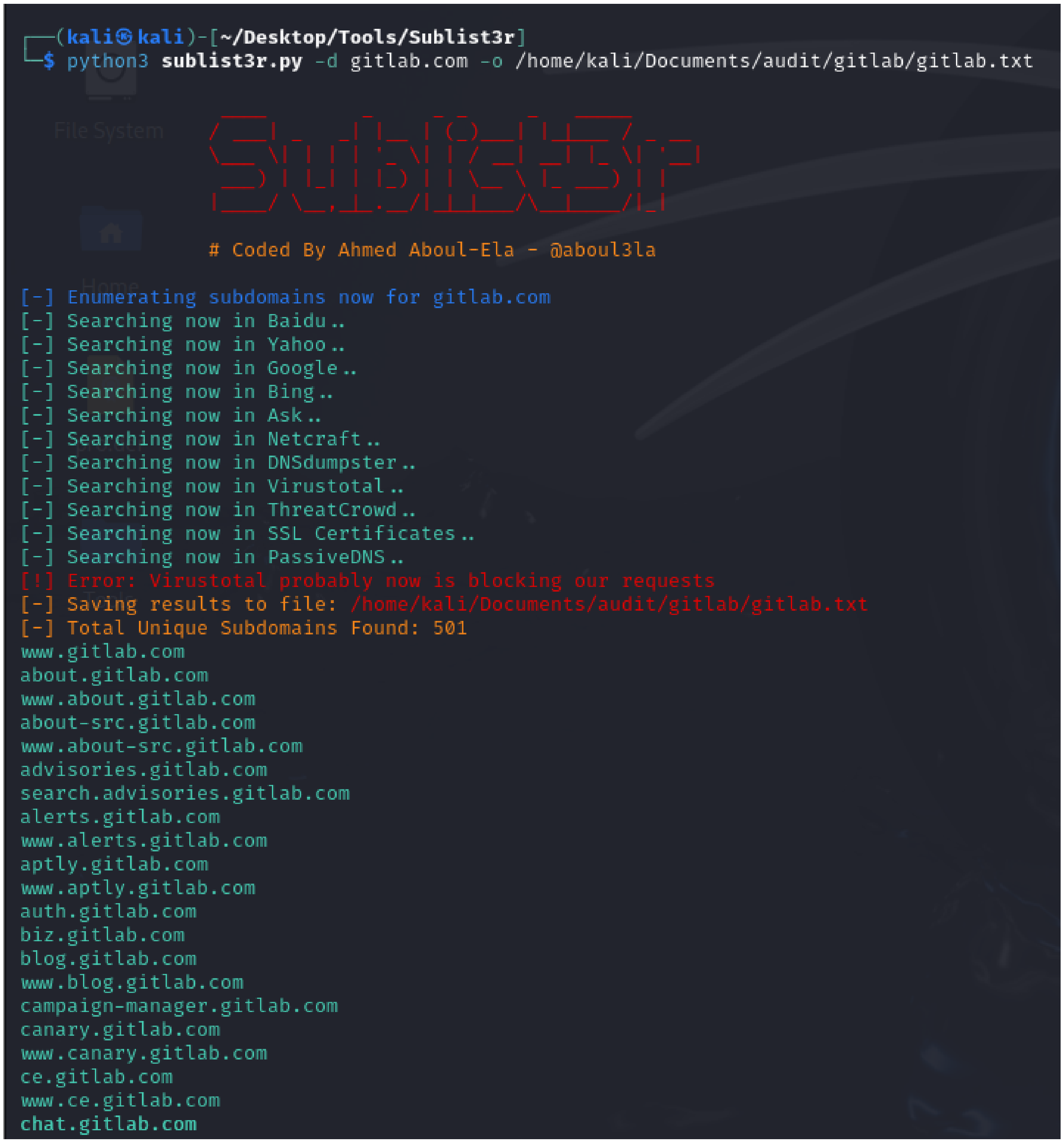
After installing the requirements, enter

python3 sublist3r.py -d <domain_name>

to find subdomains under the mentioned domain.

**In some Linux distributions, there will be an error saying that “[!] Error: Virustotal probably now is blocking our requests”. To avoid this you will need to get the API key from VirusTotal by creating an account. After the API key has been obtained, export it to an environment variable using, export VT_APIKEY=<API key>. This will work most of the time, but this is not a must.*

Since I need to check the subdomains after, I am writing the results to a file using -o switch.

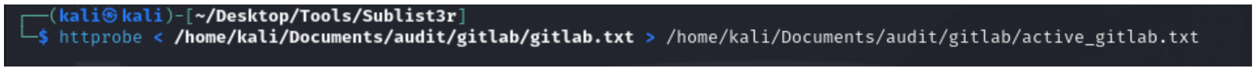


```
┌──(kali㉿kali)-[~/Desktop/Tools/Sublist3r]
└─$ python3 sublist3r.py -d gitlab.com -o /home/kali/Documents/audit/gitlab/gitlab.txt

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for gitlab.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Saving results to file: /home/kali/Documents/audit/gitlab/gitlab.txt
[-] Total Unique Subdomains Found: 501
www.gitlab.com
about.gitlab.com
www.about.gitlab.com
about-src.gitlab.com
www.about-src.gitlab.com
advisories.gitlab.com
search.advisories.gitlab.com
alerts.gitlab.com
www.alerts.gitlab.com
aptly.gitlab.com
www.aptly.gitlab.com
auth.gitlab.com
biz.gitlab.com
blog.gitlab.com
www.blog.gitlab.com
campaign-manager.gitlab.com
canary.gitlab.com
www.canary.gitlab.com
ce.gitlab.com
www.ce.gitlab.com
chat.gitlab.com
```

After identifying accessible subdomains, the subsequent step involves determining which ones are operational. This can be achieved by utilizing an additional tool known as 'httprobe'.

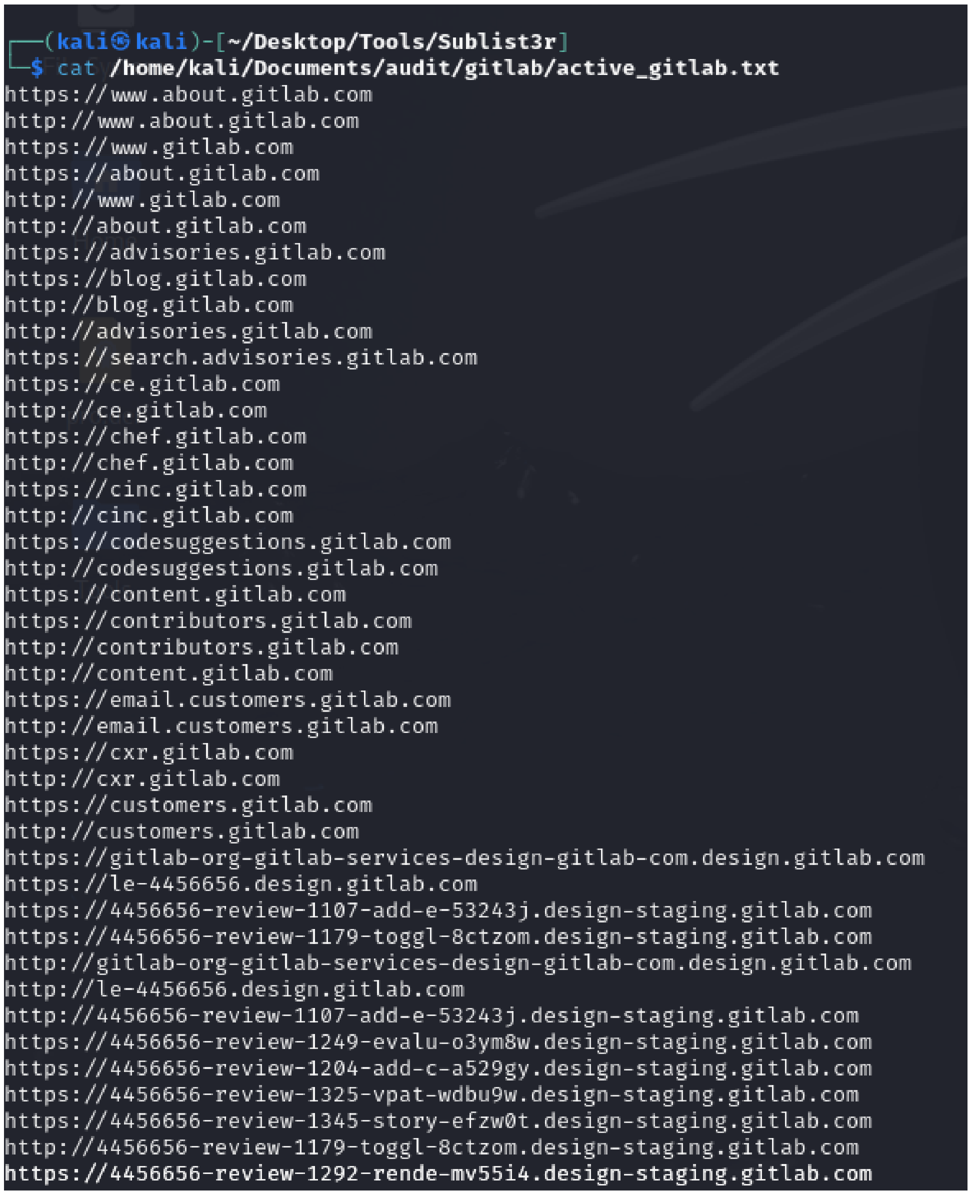
HTTProbe

This tool can identify active domains that are operational. To discover active subdomains under this site, I'm utilizing the text file previously generated by Sublist3r and writing the active subdomains to a new file.



```
┌──(kali㉿kali)-[~/Desktop/Tools/Sublist3r]
└─$ httprobe < /home/kali/Documents/audit/gitlab/gitlab.txt > /home/kali/Documents/audit/gitlab/active_gitlab.txt
```

Following the completion of the scan, the findings reveal that the majority of the subdomains are indeed active.

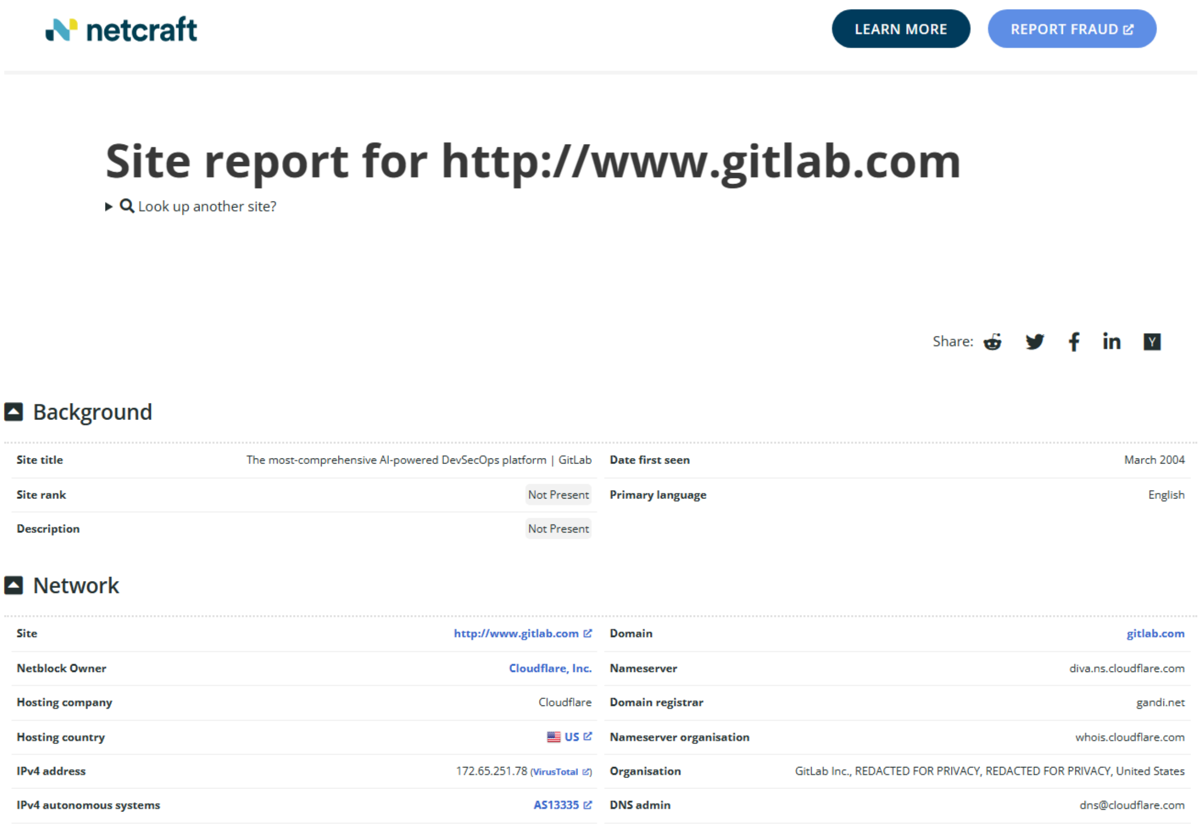


```
┌──(kali㉿kali)-[~/Desktop/Tools/Sublist3r]
└─$ cat /home/kali/Documents/audit/gitlab/active_gitlab.txt
https://www.about.gitlab.com
http://www.about.gitlab.com
https://www.gitlab.com
https://about.gitlab.com
http://www.gitlab.com
http://about.gitlab.com
https://advisories.gitlab.com
https://blog.gitlab.com
http://blog.gitlab.com
http://advisories.gitlab.com
https://search.advisories.gitlab.com
https://ce.gitlab.com
http://ce.gitlab.com
https://chef.gitlab.com
http://chef.gitlab.com
https://cinc.gitlab.com
http://cinc.gitlab.com
https://codesuggestions.gitlab.com
http://codesuggestions.gitlab.com
https://content.gitlab.com
https://contributors.gitlab.com
http://contributors.gitlab.com
http://content.gitlab.com
https://email.customers.gitlab.com
http://email.customers.gitlab.com
https://cxr.gitlab.com
http://cxr.gitlab.com
https://customers.gitlab.com
http://customers.gitlab.com
https://gitlab-org-gitlab-services-design-gitlab-com.design.gitlab.com
https://le-4456656.design.gitlab.com
https://4456656-review-1107-add-e-53243j.design-staging.gitlab.com
https://4456656-review-1179-toggl-8ctzom.design-staging.gitlab.com
http://gitlab-org-gitlab-services-design-gitlab-com.design.gitlab.com
http://le-4456656.design.gitlab.com
http://4456656-review-1107-add-e-53243j.design-staging.gitlab.com
https://4456656-review-1249-evalu-o3ym8w.design-staging.gitlab.com
https://4456656-review-1204-add-c-a529gy.design-staging.gitlab.com
https://4456656-review-1325-vpat-wdbu9w.design-staging.gitlab.com
https://4456656-review-1345-story-efzw0t.design-staging.gitlab.com
http://4456656-review-1179-toggl-8ctzom.design-staging.gitlab.com
https://4456656-review-1292-rende-mv55i4.design-staging.gitlab.com
```

Netcraft

Netcraft, a leader in internet security solutions, provides a cloud-based platform for automated vulnerability scanning and application security. This user-friendly approach eliminates the need for downloads or complex installations.

Leveraging Netcraft's website search functionality, users can effortlessly obtain valuable insights about any web property, including its ranking on the internet, IP address, SSL/TLS encryption details, hosting location, and the hosting provider.



Site report for http://www.gitlab.com

▸ Look up another site?

Background

Site title	The most-comprehensive AI-powered DevSecOps platform | GitLab	Date first seen	March 2004
Site rank	Not Present	Primary language	English
Description	Not Present		

Network

Site	http://www.gitlab.com	Domain	gitlab.com
Netblock Owner	Cloudflare, Inc.	Nameserver	diva.ns.cloudflare.com
Hosting company	Cloudflare	Domain registrar	gandi.net
Hosting country	US	Nameserver organisation	whois.cloudflare.com
IPv4 address	172.65.251.78 (VirusTotal)	Organisation	GitLab Inc., REDACTED FOR PRIVACY, REDACTED FOR PRIVACY, United States
IPv4 autonomous systems	AS13335	DNS admin	dns@cloudflare.com

For full site report: [Site report for http://www.gitlab.com | Netcraft]

While the target system's data is publicly accessible, uncovering specific technical details can be facilitated by leveraging publicly available resources like Netcraft. Due to the platform's popularity and the potential for overlapping IP addresses with other targets, I will initially focus on obtaining a Netcraft report for this primary domain. However, it's important to remember that reports for the remaining subdomains can also be readily retrieved through Netcraft.

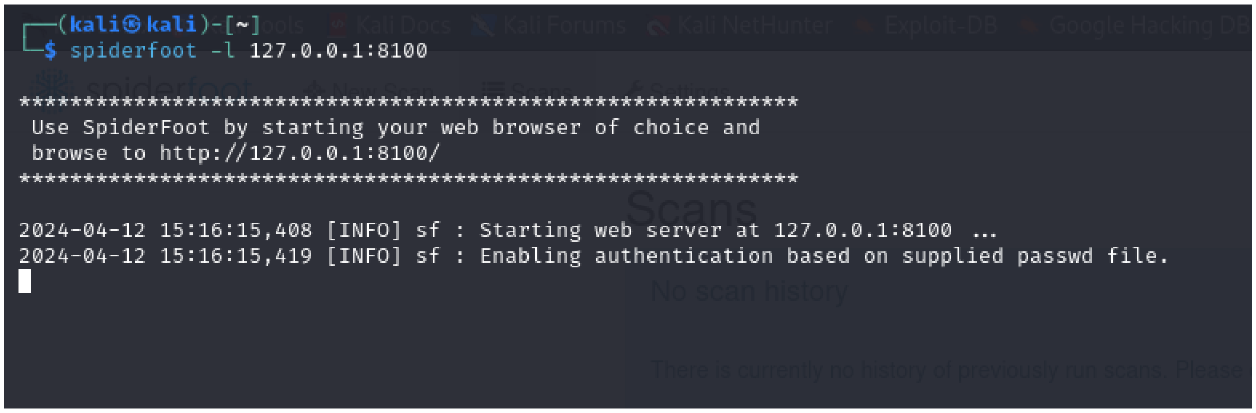
Spiderfoot

Spiderfoot is an Open-Source Intelligence (OSINT) tool proficient in combing through target websites to reveal scattered information across the web. It consolidates these discoveries thoroughly, rendering it indispensable during the information-gathering process. With the ability to detect disclosed directories, usernames, and email addresses within a website, Spiderfoot also traces their links to other platforms. These observations can be exploited for social engineering attacks against owners of disclosed accounts or emails, as the tool monitors their activity across multiple platforms.

Using spiderfoot

It must be setup, before using this tool.

Spiderfoot -l 127.0.0.1:8100

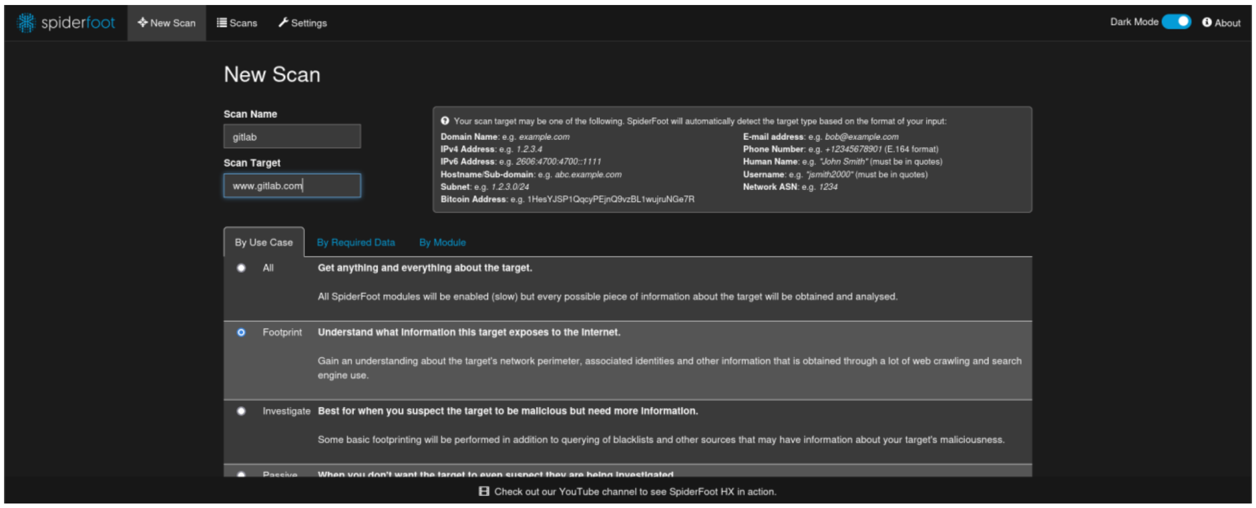


```
┌──(kali㉿kali)-[~]
└─$ spiderfoot -l 127.0.0.1:8100

*************************************************************
 Use SpiderFoot by starting your web browser of choice and
 browse to http://127.0.0.1:8100/
*************************************************************

2024-04-12 15:16:15,408 [INFO] sf : Starting web server at 127.0.0.1:8100 ...
2024-04-12 15:16:15,419 [INFO] sf : Enabling authentication based on supplied passwd file.
```

To utilize the Spiderfoot tool, which is hosted on localhost (127.0.0.1) at port 8100, just launch a web browser and enter http://127.0.0.1:8100 in the address bar.

After the scanner loads, proceed to "New scan" and tailor your scan type according to the scope of your investigation. There are various modules at your disposal that can be activated or deactivated based on your permissions. Since you're engaging in a passive information gathering phase, opt for the 'footprint' option to crawl and collect information about the website.

Spiderfoot results



New Scan

Scan Name: gitlab

Scan Target: www.gitlab.com

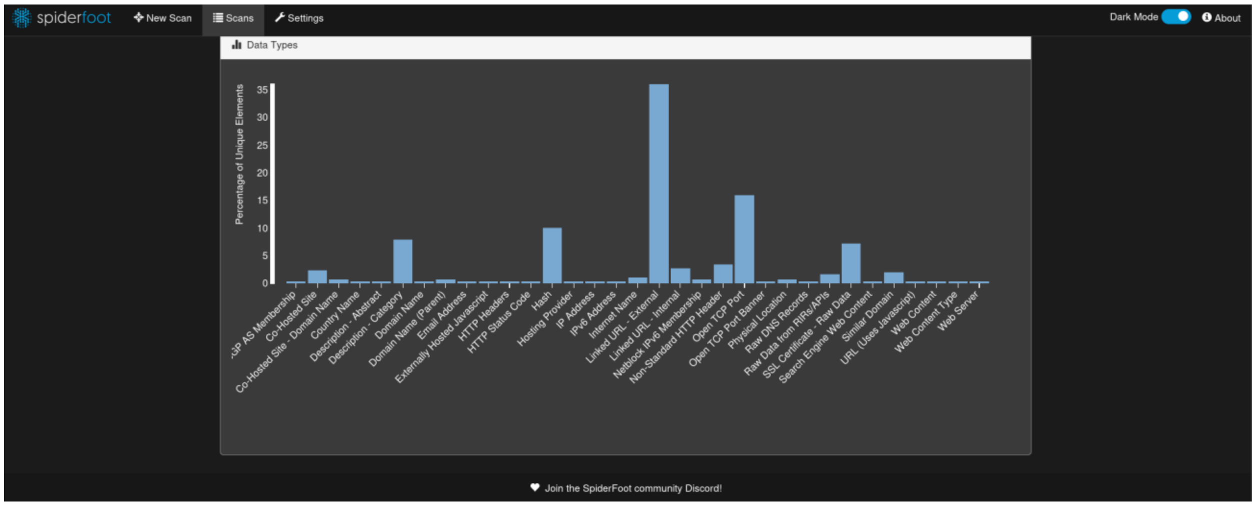
Your scan target may be one of the following. SpiderFoot will automatically detect the target type based on the format of your input:

Domain Name: e.g. *example.com*
IPv4 Address: e.g. *1.2.3.4*
IPv6 Address: e.g. *2606:4700:4700::1111*
Hostname/Sub-domain: e.g. *abc.example.com*
Subnet: e.g. *1.2.3.0/24*
Bitcoin Address: e.g. 1HesYJSP1QqcyPEjnQ9vzBL1wujruNGe7R
E-mail address: e.g. *bob@example.com*
Phone Number: e.g. *+12345678901* (E.164 format)
Human Name: e.g. *"John Smith"* (must be in quotes)
Username: e.g. *"jsmith2000"* (must be in quotes)
Network ASN: e.g. *1234*

By Use Case | By Required Data | By Module

●	All	**Get anything and everything about the target.** All SpiderFoot modules will be enabled (slow) but every possible piece of information about the target will be obtained and analysed.
[✓]	Footprint	**Understand what information this target exposes to the Internet.** Gain an understanding about the target's network perimeter, associated identities and other information that is obtained through a lot of web crawling and search engine use.
●	Investigate	**Best for when you suspect the target to be malicious but need more information.** Some basic footprinting will be performed in addition to querying of blacklists and other sources that may have information about your target's maliciousness.
●	Passive	**When you don't want the target to even suspect they are being investigated.**

Check out our YouTube channel to see SpiderFoot HX in action.

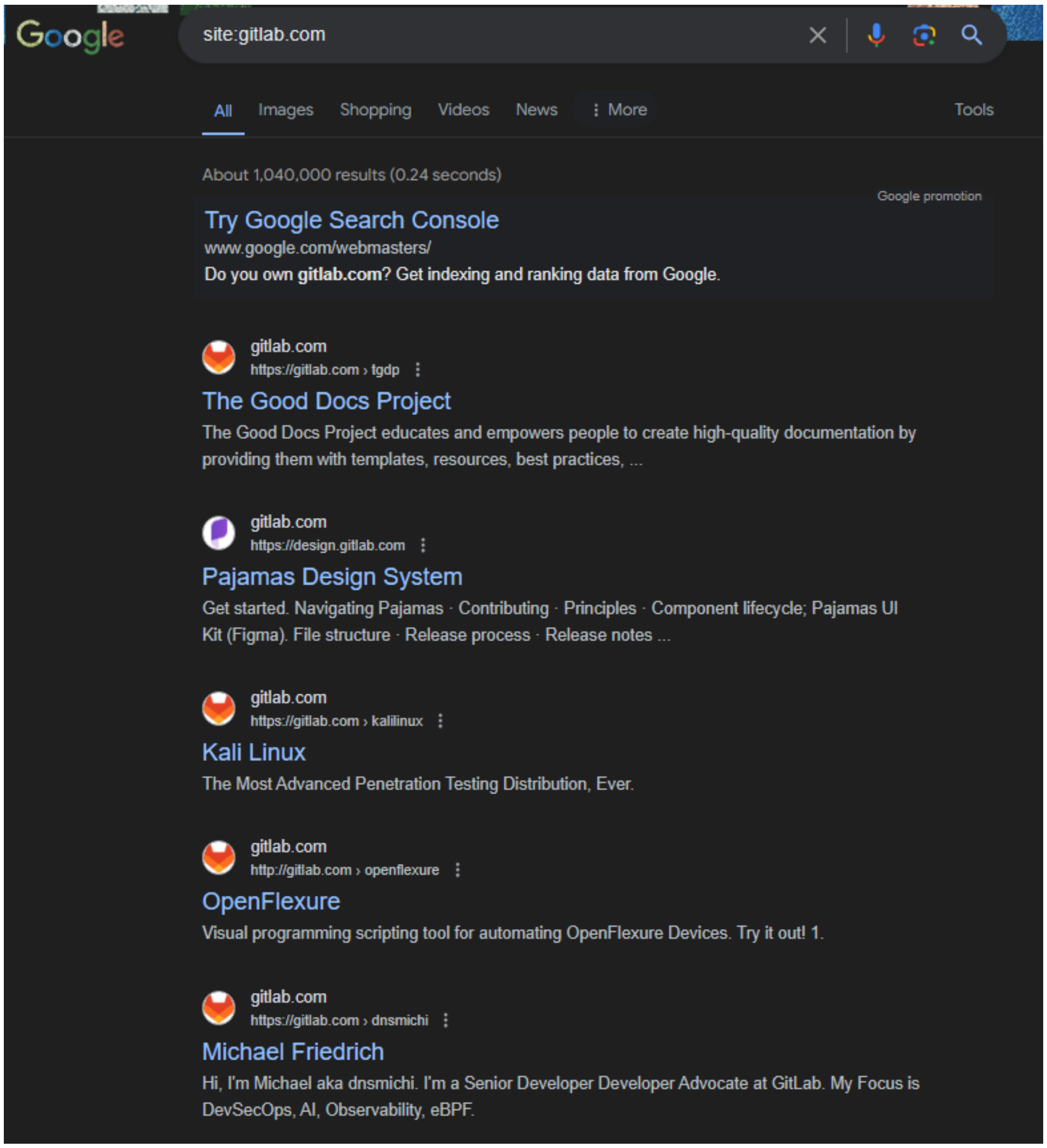


The scan has produced noteworthy findings, such as usernames, SSL certificates, and physical addresses. A significant portion of this data seems to be publicly accessible information and links leading to external websites. However, it's essential to highlight those usernames, especially when coupled with their corresponding email addresses, could potentially become avenues for social engineering or spear phishing attacks. Nevertheless, it's important to acknowledge that addressing such concerns lies beyond the boundaries of this assessment.

Google Dorks

Google Dorking offers an alternative approach to pinpointing potential vulnerabilities or revealing confidential web pages, documents, or services within a server or web application. By utilizing tailored search queries, individuals can prompt the Google search engine to reveal any misconfigurations or unsecured information services. This method involves providing explicit instructions to the search engine regarding what to search for and where to look, enabling the identification of sensitive information that might be inadvertently exposed online.

site:gitlab.com operator searches for websites that has "**gitlab.com**" in their domains.



About 1,040,000 results (0.24 seconds)

Google promotion

Try Google Search Console
www.google.com/webmasters/
Do you own **gitlab.com**? Get indexing and ranking data from Google.

gitlab.com
https://gitlab.com › tgdp
The Good Docs Project
The Good Docs Project educates and empowers people to create high-quality documentation by providing them with templates, resources, best practices, ...

gitlab.com
https://design.gitlab.com
Pajamas Design System
Get started. Navigating Pajamas · Contributing · Principles · Component lifecycle; Pajamas UI Kit (Figma). File structure · Release process · Release notes ...

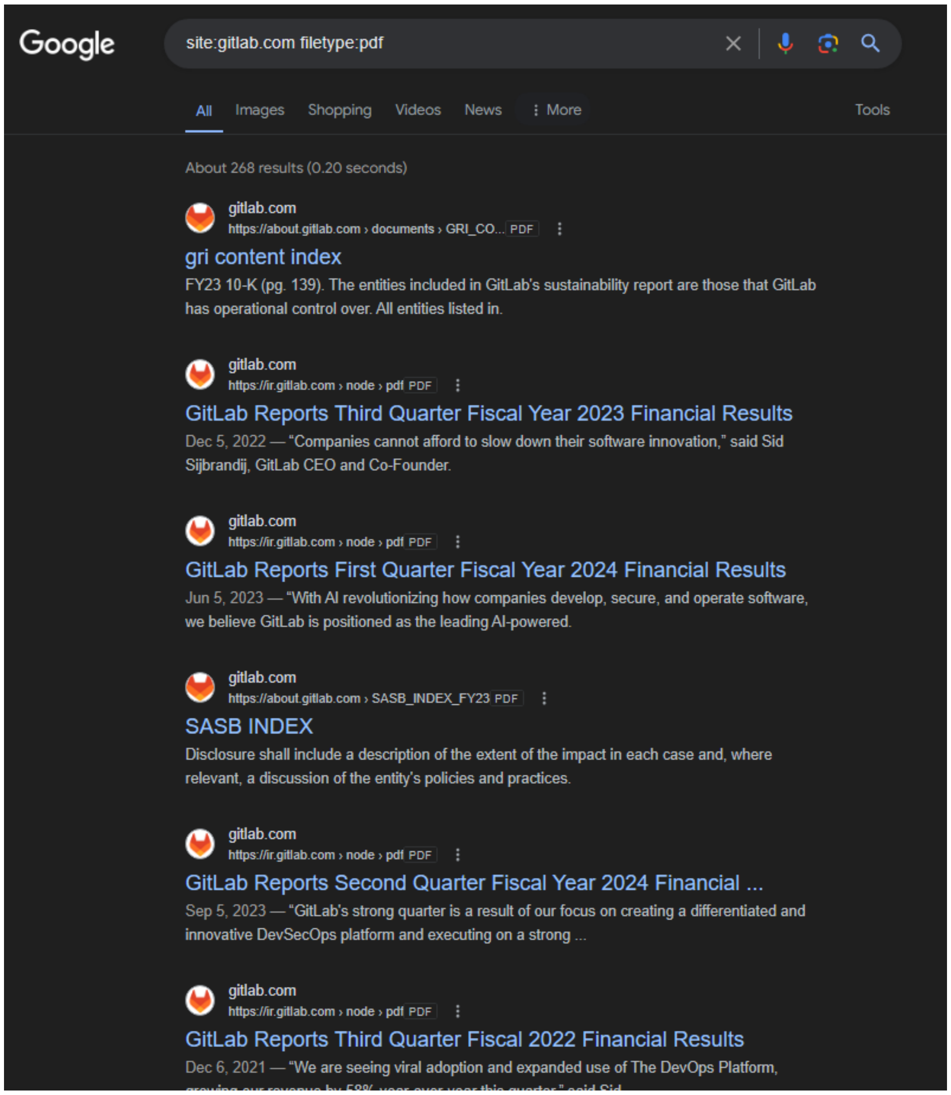
gitlab.com
https://gitlab.com › kalilinux
Kali Linux
The Most Advanced Penetration Testing Distribution, Ever.

gitlab.com
http://gitlab.com › openflexure
OpenFlexure
Visual programming scripting tool for automating OpenFlexure Devices. Try it out! 1.

gitlab.com
https://gitlab.com › dnsmichi
Michael Friedrich
Hi, I'm Michael aka dnsmichi. I'm a Senior Developer Developer Advocate at GitLab. My Focus is DevSecOps, AI, Observability, eBPF.

Certainly, the appearance of subdomains in search results illustrates a common utilization of Google Dorking. This method facilitates the discovery of different file types and pages that are exposed by a specific website on the internet, whether it's deliberate or unintentional. Through the refinement of search queries, users can unearth particular information and potentially pinpoint vulnerabilities or confidential data that might be accessible online.

During a search for various file types exposed on the internet, I came across some intriguing and possibly concerning PDFs within this subdomain.



site:gitlab.com filetype:pdf

About 268 results (0.20 seconds)

gitlab.com
https://about.gitlab.com › documents › GRI_CO... PDF
gri content index
FY23 10-K (pg. 139). The entities included in GitLab's sustainability report are those that GitLab has operational control over. All entities listed in.

gitlab.com
https://ir.gitlab.com › node › pdf PDF
GitLab Reports Third Quarter Fiscal Year 2023 Financial Results
Dec 5, 2022 — "Companies cannot afford to slow down their software innovation," said Sid Sijbrandij, GitLab CEO and Co-Founder.

gitlab.com
https://ir.gitlab.com › node › pdf PDF
GitLab Reports First Quarter Fiscal Year 2024 Financial Results
Jun 5, 2023 — "With AI revolutionizing how companies develop, secure, and operate software, we believe GitLab is positioned as the leading AI-powered.

gitlab.com
https://about.gitlab.com › SASB_INDEX_FY23 PDF
SASB INDEX
Disclosure shall include a description of the extent of the impact in each case and, where relevant, a discussion of the entity's policies and practices.

gitlab.com
https://ir.gitlab.com › node › pdf PDF
GitLab Reports Second Quarter Fiscal Year 2024 Financial ...
Sep 5, 2023 — "GitLab's strong quarter is a result of our focus on creating a differentiated and innovative DevSecOps platform and executing on a strong ...

gitlab.com
https://ir.gitlab.com › node › pdf PDF
GitLab Reports Third Quarter Fiscal 2022 Financial Results
Dec 6, 2021 — "We are seeing viral adoption and expanded use of The DevOps Platform,

The management of information and files within this subdomain appears to be efficient, as no additional significant files were uncovered apart from the previously mentioned PDFs.

Directory and services enumeration

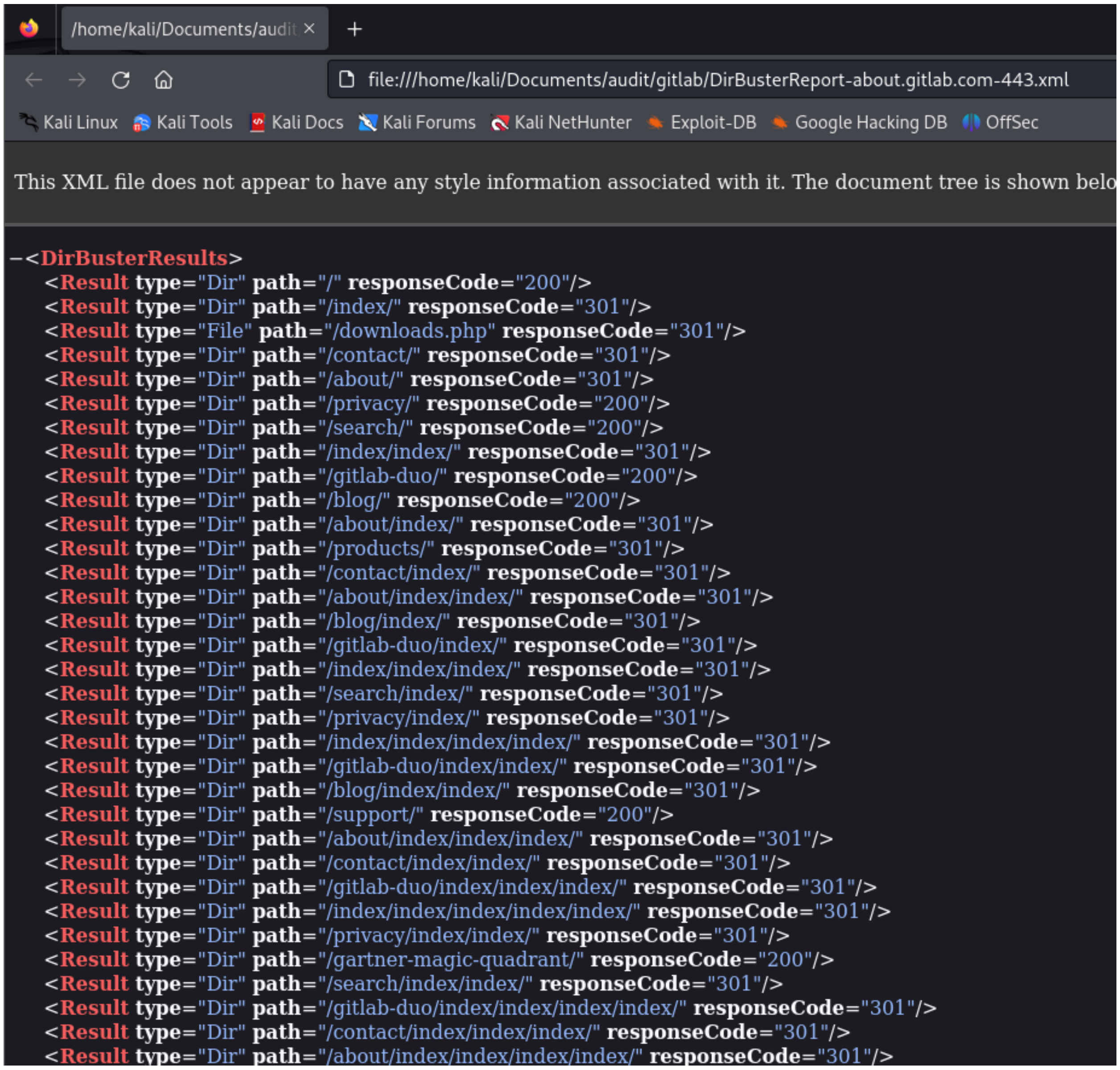
Dirbuster

DirBuster, a web content scanner developed by OWASP, utilizes brute force methods to uncover different directories within a target website. By scrutinizing HTTP responses and their associated response codes, the tool detects concealed or referenced directories. Built in Java, DirBuster supports multi-threading to expedite directory scanning and produce a comprehensive file and folder structure of the target site.

Employing this tool facilitates the identification of directories or files that might be accessible yet not overtly exposed. Furthermore, it offers a glimpse into the server's file and folder arrangement, assisting in comprehending its structure and potential vulnerabilities.

Domain: www.gitlab.com

I managed to gain a general understanding of the folder structure within the web server through Dirbuster exploration and without any errors.


/home/kali/Documents/audit
file:///home/kali/Documents/audit/gitlab/DirBusterReport-about.gitlab.com-443.xml
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec

This XML file does not appear to have any style information associated with it. The document tree is shown belo

```
-<DirBusterResults>
    <Result type="Dir" path="/" responseCode="200"/>
    <Result type="Dir" path="/index/" responseCode="301"/>
    <Result type="File" path="/downloads.php" responseCode="301"/>
    <Result type="Dir" path="/contact/" responseCode="301"/>
    <Result type="Dir" path="/about/" responseCode="301"/>
    <Result type="Dir" path="/privacy/" responseCode="200"/>
    <Result type="Dir" path="/search/" responseCode="200"/>
    <Result type="Dir" path="/index/index/" responseCode="301"/>
    <Result type="Dir" path="/gitlab-duo/" responseCode="200"/>
    <Result type="Dir" path="/blog/" responseCode="200"/>
    <Result type="Dir" path="/about/index/" responseCode="301"/>
    <Result type="Dir" path="/products/" responseCode="301"/>
    <Result type="Dir" path="/contact/index/" responseCode="301"/>
    <Result type="Dir" path="/about/index/index/" responseCode="301"/>
    <Result type="Dir" path="/blog/index/" responseCode="301"/>
    <Result type="Dir" path="/gitlab-duo/index/" responseCode="301"/>
    <Result type="Dir" path="/index/index/index/" responseCode="301"/>
    <Result type="Dir" path="/search/index/" responseCode="301"/>
    <Result type="Dir" path="/privacy/index/" responseCode="301"/>
    <Result type="Dir" path="/index/index/index/index/" responseCode="301"/>
    <Result type="Dir" path="/gitlab-duo/index/index/" responseCode="301"/>
    <Result type="Dir" path="/blog/index/index/" responseCode="301"/>
    <Result type="Dir" path="/support/" responseCode="200"/>
    <Result type="Dir" path="/about/index/index/index/" responseCode="301"/>
    <Result type="Dir" path="/contact/index/index/" responseCode="301"/>
    <Result type="Dir" path="/gitlab-duo/index/index/index/" responseCode="301"/>
    <Result type="Dir" path="/index/index/index/index/index/" responseCode="301"/>
    <Result type="Dir" path="/privacy/index/index/" responseCode="301"/>
    <Result type="Dir" path="/gartner-magic-quadrant/" responseCode="200"/>
    <Result type="Dir" path="/search/index/index/" responseCode="301"/>
    <Result type="Dir" path="/gitlab-duo/index/index/index/index/" responseCode="301"/>
    <Result type="Dir" path="/contact/index/index/index/" responseCode="301"/>
    <Result type="Dir" path="/about/index/index/index/index/" responseCode="301"/>
```

I checked every result manually and not found that is suspicious or flaws.

Nmap

Nmap, also known as Network Mapper, is a flexible port scanner engineered to probe hosts and reveal their open ports, associated services, port statuses, running service versions, and the operating system in use, among other details. This open-source tool serves various purposes, offering valuable insights into network configurations and potential vulnerabilities. Additionally, Nmap supports the execution of scripts on target systems to exploit vulnerabilities or gather additional information.

Upon installation, you can access the available options by typing "nmap -h" in your command line interface. For a more detailed understanding of how the tool operates, you can consult the manual page by entering "man nmap" in your command line interface.*Note that some options may require administrator / super user privileges. *Note that some options may require administrator / super user privileges.

I am using the following scan options for this assessment.

sudo nmap <host name> -sS -sV -O -oN <filename>

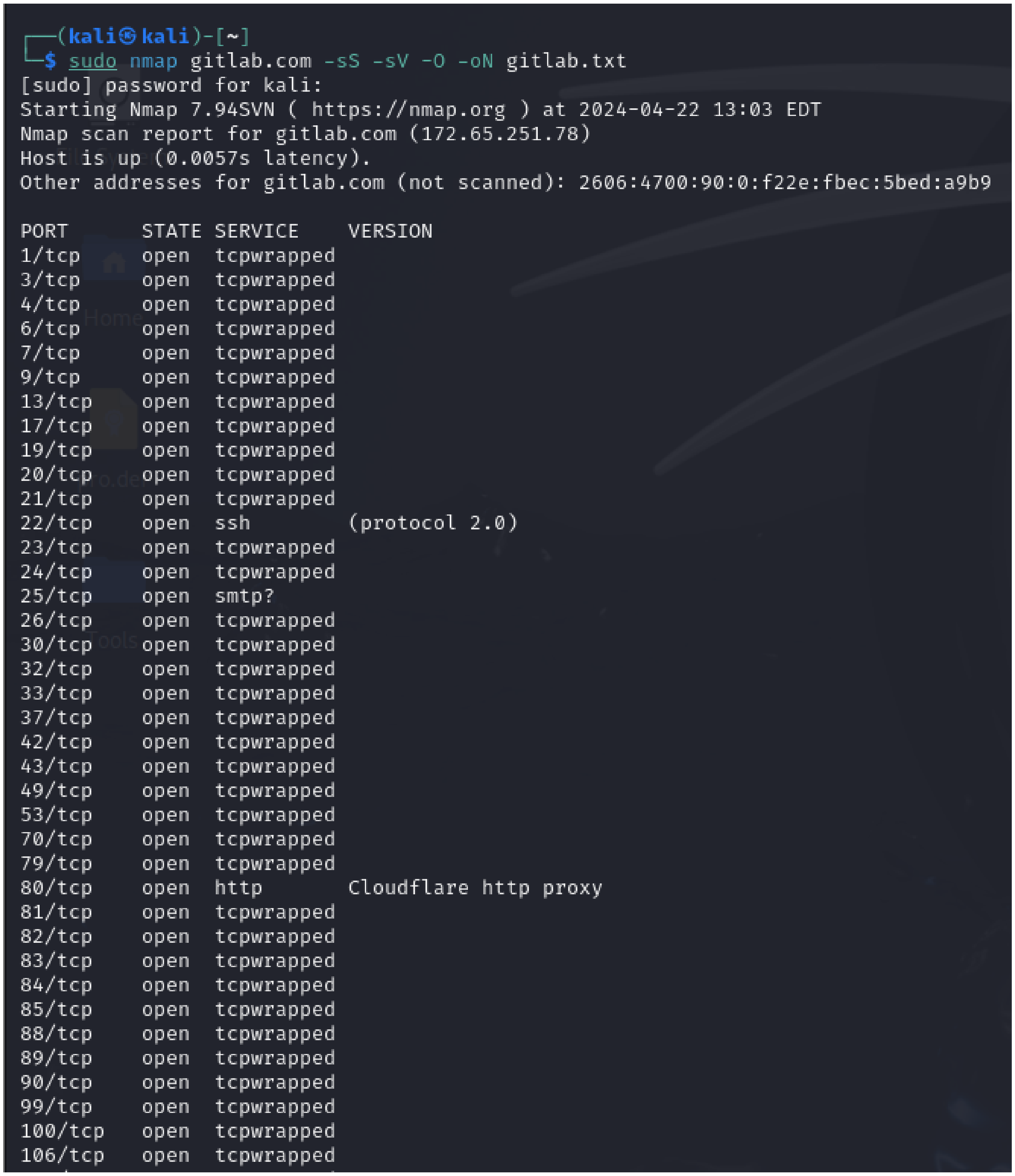
-sS: Enables SYN scan (also known as Stealth scan).

-sV: Enables version detection. It tries to detect the version of the service running in that port.

-O: Enables Operating System detection.

-oN : Outputs the scan results to text file

Scanned results for https://www.gitlab.com/



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap gitlab.com -sS -sV -O -oN gitlab.txt
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-22 13:03 EDT
Nmap scan report for gitlab.com (172.65.251.78)
Host is up (0.0057s latency).
Other addresses for gitlab.com (not scanned): 2606:4700:90:0:f22e:fbec:5bed:a9b9

PORT      STATE SERVICE    VERSION
1/tcp     open  tcpwrapped
3/tcp     open  tcpwrapped
4/tcp     open  tcpwrapped
6/tcp     open  tcpwrapped
7/tcp     open  tcpwrapped
9/tcp     open  tcpwrapped
13/tcp    open  tcpwrapped
17/tcp    open  tcpwrapped
19/tcp    open  tcpwrapped
20/tcp    open  tcpwrapped
21/tcp    open  tcpwrapped
22/tcp    open  ssh        (protocol 2.0)
23/tcp    open  tcpwrapped
24/tcp    open  tcpwrapped
25/tcp    open  smtp?
26/tcp    open  tcpwrapped
30/tcp    open  tcpwrapped
32/tcp    open  tcpwrapped
33/tcp    open  tcpwrapped
37/tcp    open  tcpwrapped
42/tcp    open  tcpwrapped
43/tcp    open  tcpwrapped
49/tcp    open  tcpwrapped
53/tcp    open  tcpwrapped
70/tcp    open  tcpwrapped
79/tcp    open  tcpwrapped
80/tcp    open  http       Cloudflare http proxy
81/tcp    open  tcpwrapped
82/tcp    open  tcpwrapped
83/tcp    open  tcpwrapped
84/tcp    open  tcpwrapped
85/tcp    open  tcpwrapped
88/tcp    open  tcpwrapped
89/tcp    open  tcpwrapped
90/tcp    open  tcpwrapped
99/tcp    open  tcpwrapped
100/tcp   open  tcpwrapped
106/tcp   open  tcpwrapped
```

Automated Testing

For automated testing, I've selected OWASP ZAP widely used tool within the industry.

OWASP ZAP

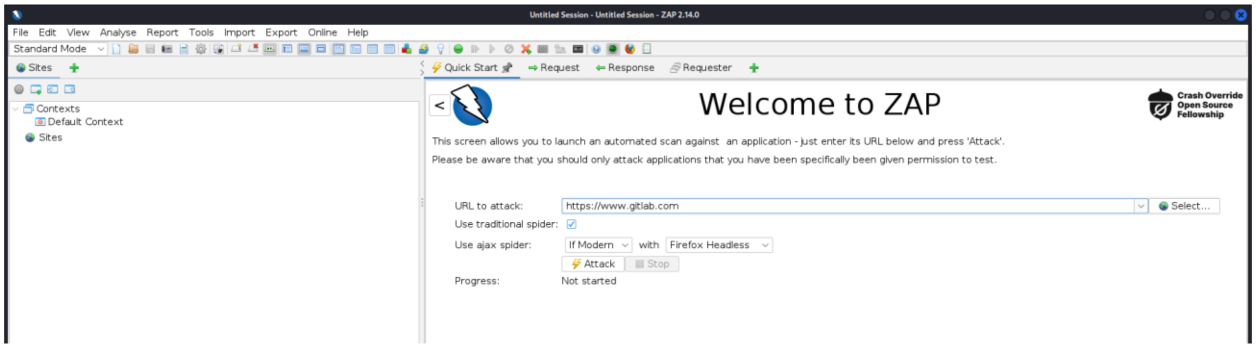
The Open Web Application Security Project Zed Attack Proxy (OWASP ZAP) is a well-known open-source vulnerability scanner recognized for its ability to operate as a Man-in-the-Middle (MITM) proxy. It evaluates various vulnerabilities by examining responses from the web application or server. OWASP ZAP is notably user-friendly and offers customization options through the installation of modules, allowing for efficient management of results.

Within this proxy, there are primarily two types of scans available:

1. Automated Scan: Users input the target URL and initiate the attack. The behavior can be customized by selecting the ZAP mode, triggering all scripts against the target to detect vulnerabilities and generate reports accordingly.

2. Manual Explore: Users can navigate to the target web application and begin exploration. During manual exploration, ZAP HUD (Heads Up Display) captures each page, while the ZAP proxy records responses.

For this assessment, I am running ZAP on automated mode.


Untitled Session - Untitled Session - ZAP 2.14.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
Crash Override Open Source Fellowship
Select...
Attack
Stop

Welcome to ZAP

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: https://www.gitlab.com

Use traditional spider: ✓

Use ajax spider: If Modern with Firefox Headless

Progress: Not started

After entering the target URL in the designated textbox, simply click on "Attack" to begin the scanning process. Once finished, you can generate a detailed report of the findings by clicking on "Report."

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	4 (26.7%)	1 (6.7%)	0 (0.0%)	5 (33.3%)
	Low	0 (0.0%)	1 (6.7%)	3 (20.0%)	1 (6.7%)	5 (33.3%)
	Informational	0 (0.0%)	0 (0.0%)	2 (13.3%)	3 (20.0%)	5 (33.3%)
	Total	0 (0.0%)	5 (33.3%)	6 (40.0%)	4 (26.7%)	15 (100%)

Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
CSP: Wildcard Directive	Medium	1 (6.7%)
CSP: script-src unsafe-eval	Medium	1 (6.7%)
CSP: script-src unsafe-inline	Medium	1 (6.7%)
CSP: style-src unsafe-inline	Medium	1 (6.7%)
Missing Anti-clickjacking Header	Medium	1 (6.7%)
Cookie with SameSite Attribute None	Low	3 (20.0%)
Cross-Domain JavaScript Source File Inclusion	Low	6 (40.0%)
Strict-Transport-Security Header Not Set	Low	1 (6.7%)
Timestamp Disclosure - Unix	Low	2 (13.3%)
X-Content-Type-Options Header Missing	Low	1 (6.7%)
Information Disclosure - Suspicious Comments	Informational	1 (6.7%)
Loosely Scoped Cookie	Informational	3 (20.0%)
Re-examine Cache-control Directives	Informational	1 (6.7%)
Retrieved from Cache	Informational	1 (6.7%)
Session Management Response Identified	Informational	3 (20.0%)
Total		15

*Note that these vulnerabilities are rated according to the OWASP risk rating methodology which can be found in this link. OWASP Risk Rating Methodology.

Here are the vulnerabilities detected within this domain, along with their impacts. The insights provided, including screenshots and remedies, are sourced from the report generated by OWASP ZAP and the Netsparker website. [https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities]) https://www.gitlab.com (4)

CSP: Wildcard Directive (1)

▾ GET https://www.gitlab.com

Alert tags	▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Other info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: script-src, style-src, img-src, connect-src, frame-src, font-src, media-src, object-src, worker-src, form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
Request	▸ Request line and header section (229 bytes) ▸ Request body (0 bytes)
Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Parameter	`content-security-policy`
Evidence	`default-src 'self' https: http:; script-src 'self' 'unsafe-inline' 'unsafe-eval' https: http: *.googletagmanager.com; style-src 'self' 'unsafe-inline' https: http: https://fonts.googleapis.com; object-src https: http:; base-uri 'self'; connect-src 'self' https: http: wss: ws: *.google-analytics.com *.analytics.google.com *.googletagmanager.com; frame-src 'self' https: http:; img-src 'self' https: http: data: *.google-analytics.com *.googletagmanager.com; manifest-src 'self'; media-src 'self' https: http:; child-src 'self' blob: https: http:; font-src 'self' https: http: data: https://fonts.gstatic.com;`
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

https://www.gitlab.com (4)

CSP: Wildcard Directive (1)

▸ GET https://www.gitlab.com

CSP: script-src unsafe-eval (1)

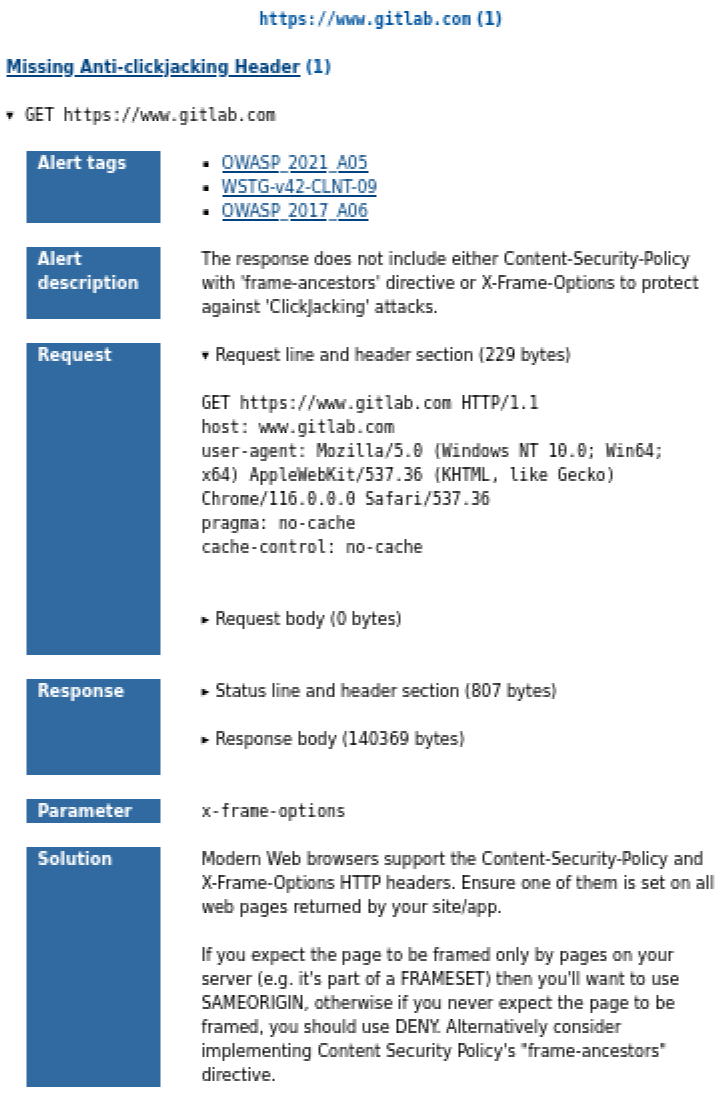
▾ GET https://www.gitlab.com

Alert tags	▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Other info	script-src includes unsafe-eval.
Request	▸ Request line and header section (229 bytes) ▸ Request body (0 bytes)
Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Parameter	`content-security-policy`
Evidence	`default-src 'self' https: http:; script-src 'self' 'unsafe-inline' 'unsafe-eval' https: http: *.googletagmanager.com; style-src 'self' 'unsafe-inline' https: http: https://fonts.googleapis.com; object-src https: http:; base-uri 'self'; connect-src 'self' https: http: wss: ws: *.google-analytics.com *.analytics.google.com *.googletagmanager.com; frame-src 'self' https: http:; img-src 'self' https: http: data: *.google-analytics.com *.googletagmanager.com; manifest-src 'self'; media-src 'self' https: http:; child-src 'self' blob: https: http:; font-src 'self' https: http: data: https://fonts.gstatic.com;`
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

CSP: style-src unsafe-inline (1)

▾ `GET https://www.gitlab.com`

Alert tags	▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Other info	style-src includes unsafe-inline.
Request	▸ Request line and header section (229 bytes) ▸ Request body (0 bytes)
Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Parameter	`Content-Security-Policy`
Evidence	`default-src 'self' https: http:; script-src 'self' 'unsafe-inline' 'unsafe-eval' https: http: *.googletagmanager.com; style-src 'self' 'unsafe-inline' https: http: https://fonts.googleapis.com; object-src https: http:; base-uri 'self'; connect-src 'self' https: http: wss: ws: *.google-analytics.com *.analytics.google.com *.googletagmanager.com; frame-src 'self' https: http:; img-src 'self' https: http: data: *.google-analytics.com *.googletagmanager.com; manifest-src 'self'; media-src 'self' https: http:; child-src 'self' blob: https: http:; font-src 'self' https: http: data: https://fonts.gstatic.com;`
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.



https://www.gitlab.com (1)

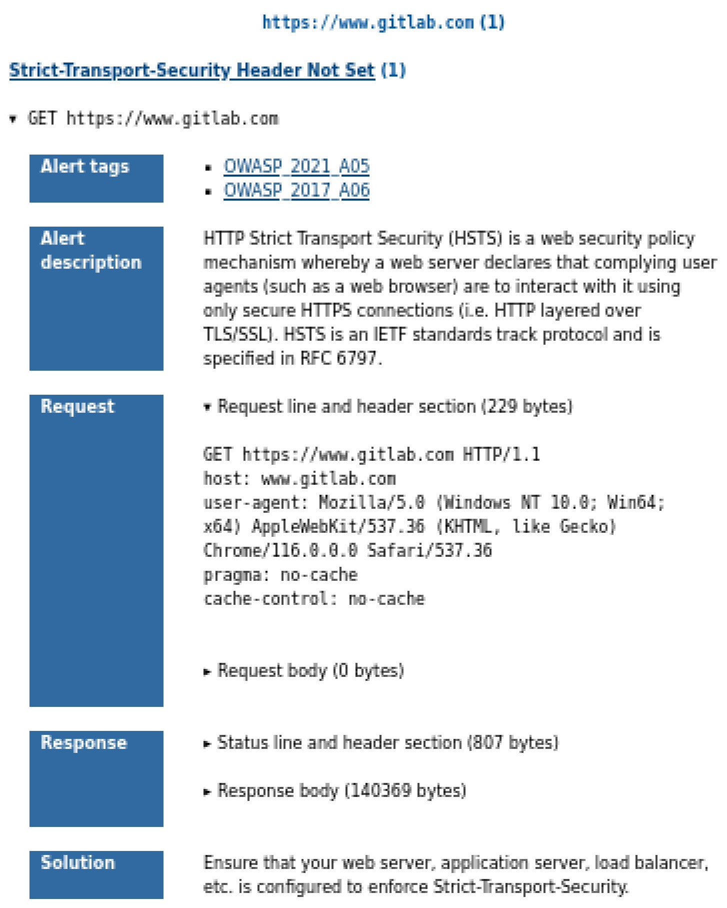
Missing Anti-clickjacking Header (1)

▾ GET https://www.gitlab.com

Alert tags	▪ OWASP_2021_A05 ▪ WSTG-v42-CLNT-09 ▪ OWASP_2017_A06
Alert description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

Request	▾ Request line and header section (229 bytes)
	
	`GET https://www.gitlab.com HTTP/1.1`
	`host: www.gitlab.com`
	`user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36`
	`pragma: no-cache`
	`cache-control: no-cache`
	
	▸ Request body (0 bytes)

Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Parameter	`x-frame-options`
Solution	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.



https://www.gitlab.com (1)

Strict-Transport-Security Header Not Set (1)

▾ GET https://www.gitlab.com

Alert tags

- OWASP_2021_A05
- OWASP_2017_A06

Alert description

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Request

▾ Request line and header section (229 bytes)

```
GET https://www.gitlab.com HTTP/1.1
host: www.gitlab.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/116.0.0.0 Safari/537.36
pragma: no-cache
cache-control: no-cache
```

▸ Request body (0 bytes)

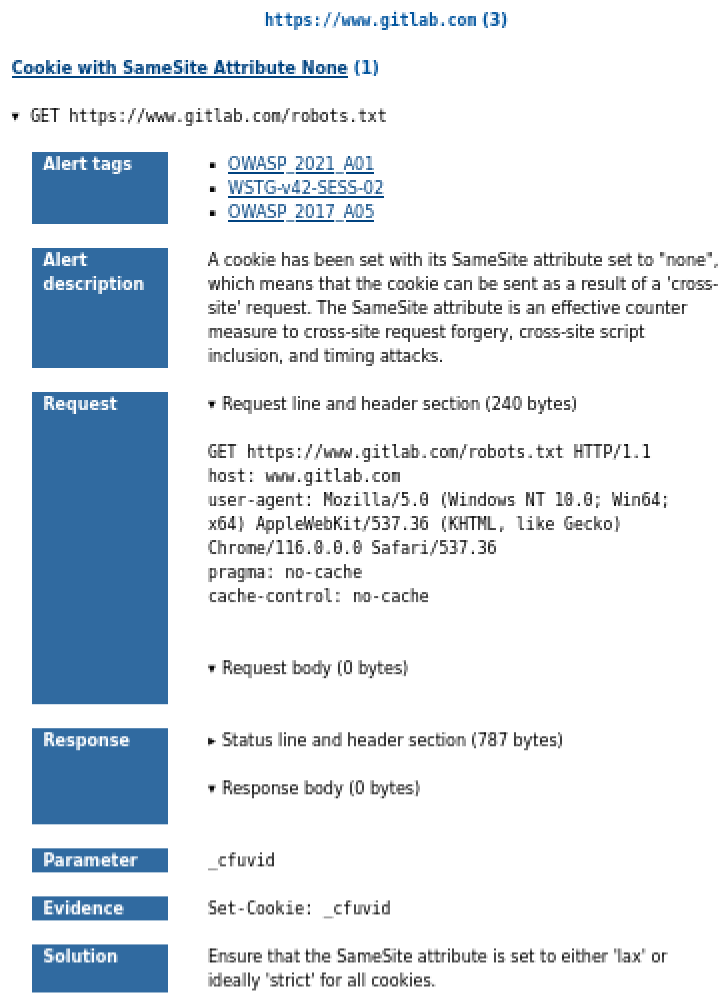
Response

▸ Status line and header section (807 bytes)

▸ Response body (140369 bytes)

Solution

Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.



https://www.gitlab.com (3)

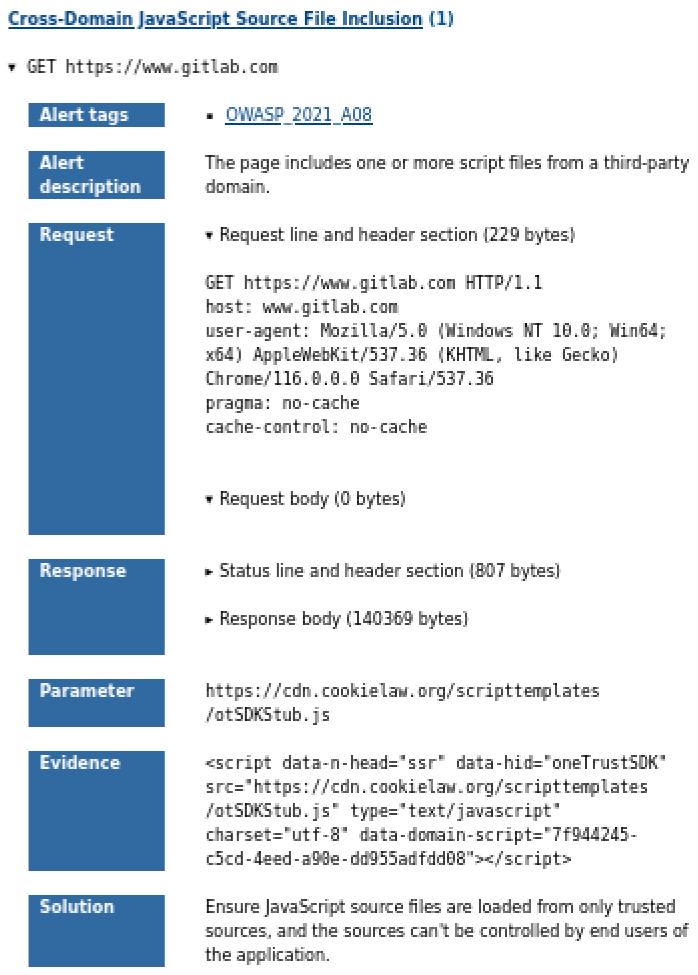
Cookie with SameSite Attribute None (1)

▾ GET https://www.gitlab.com/robots.txt

Alert tags	• OWASP_2021_A01 • WSTG-v42-SESS-02 • OWASP_2017_A05
Alert description	A cookie has been set with its SameSite attribute set to "none", which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Request	▾ Request line and header section (240 bytes)
	
	`GET https://www.gitlab.com/robots.txt HTTP/1.1`
	`host: www.gitlab.com`
	`user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36`
	`pragma: no-cache`
	`cache-control: no-cache`
	
	▾ Request body (0 bytes)

Response	▸ Status line and header section (787 bytes) ▾ Response body (0 bytes)
Parameter	`_cfuvid`
Evidence	`Set-Cookie: _cfuvid`
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.



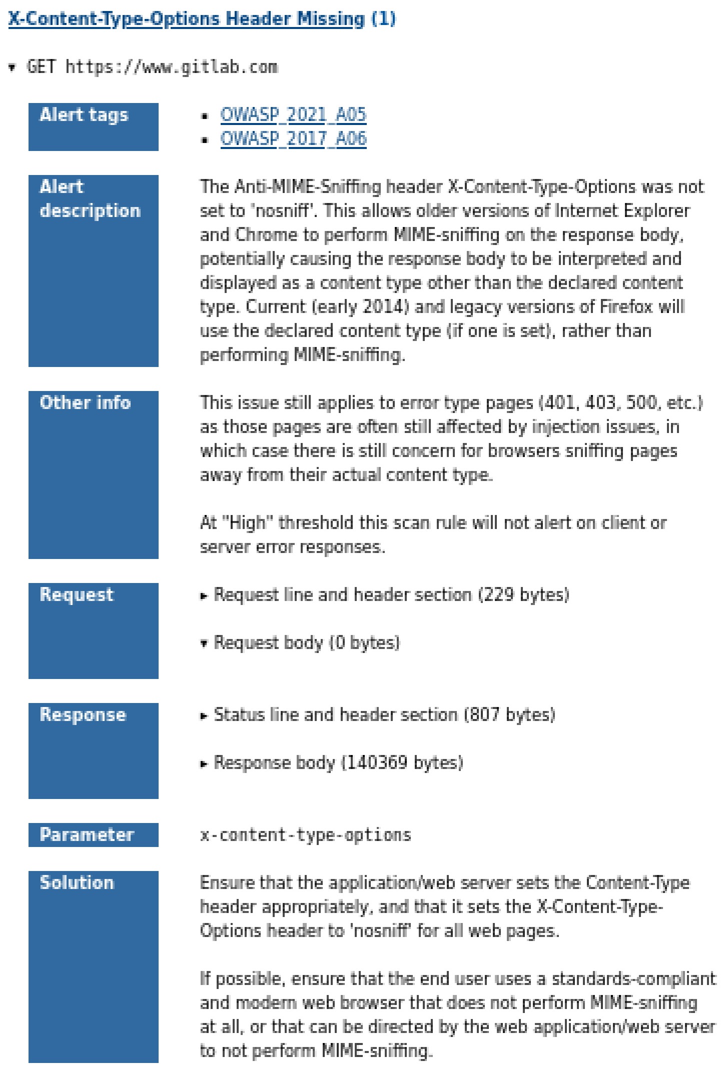
Cross-Domain JavaScript Source File Inclusion (1)

▾ `GET https://www.gitlab.com`

Alert tags	▪ OWASP_2021_A08
Alert description	The page includes one or more script files from a third-party domain.

Request	▾ Request line and header section (229 bytes)
	`GET https://www.gitlab.com HTTP/1.1`
	`host: www.gitlab.com`
	`user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36`
	`pragma: no-cache`
	`cache-control: no-cache`
	▾ Request body (0 bytes)

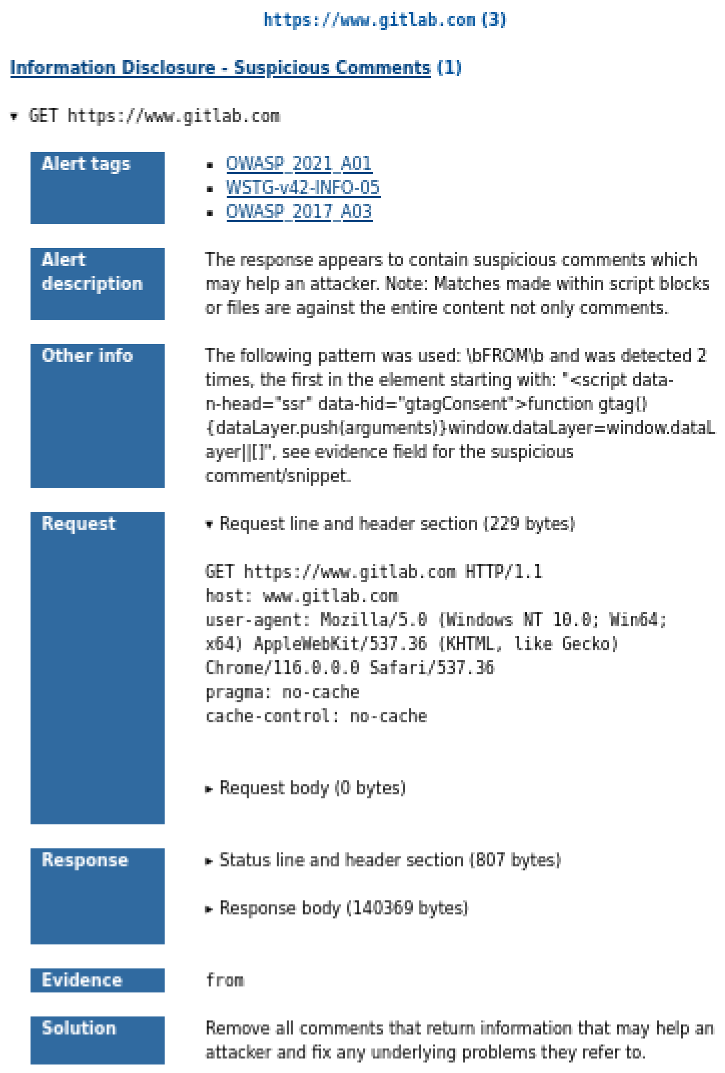
Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Parameter	`https://cdn.cookielaw.org/scripttemplates/otSDKStub.js`
Evidence	`<script data-n-head="ssr" data-hid="oneTrustSDK" src="https://cdn.cookielaw.org/scripttemplates/otSDKStub.js" type="text/javascript" charset="utf-8" data-domain-script="7f944245-c5cd-4eed-a90e-dd955adfdd08"></script>`
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.



X-Content-Type-Options Header Missing (1)

▾ `GET https://www.gitlab.com`

Alert tags	▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
Other info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
Request	▸ Request line and header section (229 bytes) ▾ Request body (0 bytes)
Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Parameter	`x-content-type-options`
Solution	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.



https://www.gitlab.com (3)

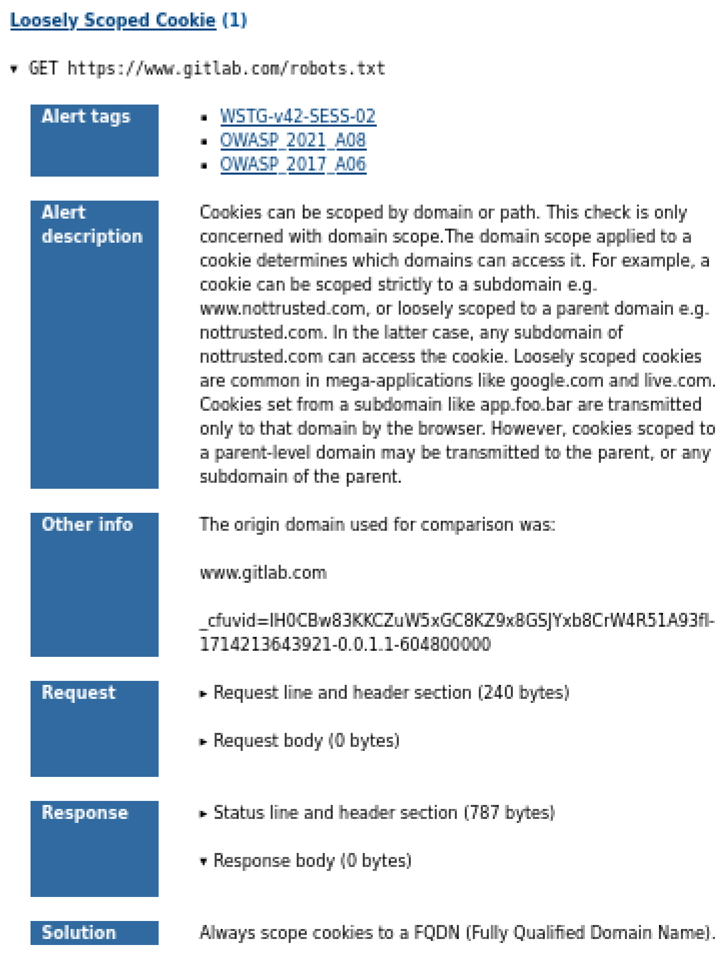
Information Disclosure - Suspicious Comments (1)

▾ GET https://www.gitlab.com

Alert tags	▪ OWASP_2021_A01 ▪ WSTG-v42-INFO-05 ▪ OWASP_2017_A03
Alert description	The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
Other info	The following pattern was used: \bFROM\b and was detected 2 times, the first in the element starting with: "<script data-n-head="ssr" data-hid="gtagConsent">function gtag(){dataLayer.push(arguments)}window.dataLayer=window.dataLayer||[]", see evidence field for the suspicious comment/snippet.

Request	▾ Request line and header section (229 bytes)
	
	`GET https://www.gitlab.com HTTP/1.1`
	`host: www.gitlab.com`
	`user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36`
	`pragma: no-cache`
	`cache-control: no-cache`
	
	▸ Request body (0 bytes)

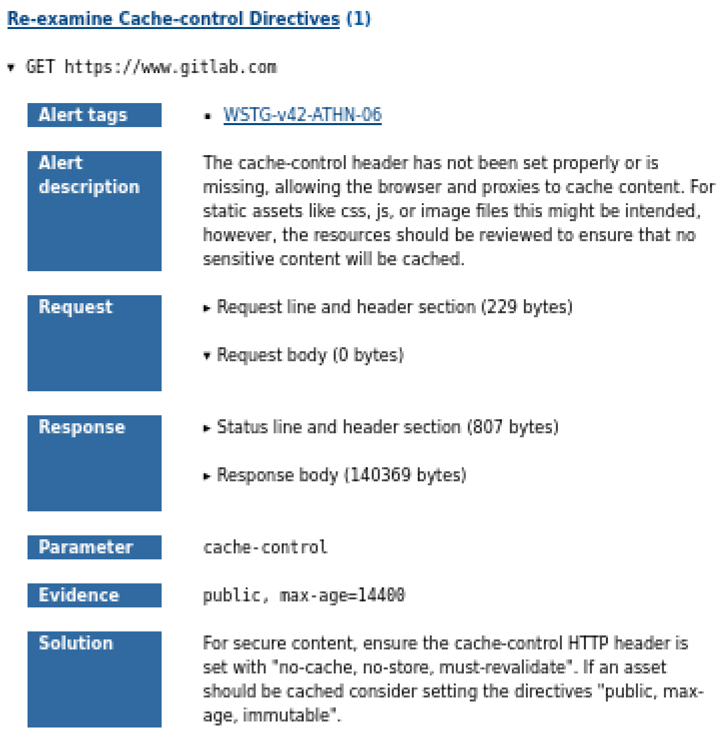
Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Evidence	`from`
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.



Loosely Scoped Cookie (1)

▾ `GET https://www.gitlab.com/robots.txt`

Alert tags	▪ WSTG-v42-SESS-02 ▪ OWASP_2021_A08 ▪ OWASP_2017_A06
Alert description	Cookies can be scoped by domain or path. This check is only concerned with domain scope.The domain scope applied to a cookie determines which domains can access it. For example, a cookie can be scoped strictly to a subdomain e.g. www.nottrusted.com, or loosely scoped to a parent domain e.g. nottrusted.com. In the latter case, any subdomain of nottrusted.com can access the cookie. Loosely scoped cookies are common in mega-applications like google.com and live.com. Cookies set from a subdomain like app.foo.bar are transmitted only to that domain by the browser. However, cookies scoped to a parent-level domain may be transmitted to the parent, or any subdomain of the parent.
Other info	The origin domain used for comparison was: www.gitlab.com _cfuvid=IH0CBw83KKCZuW5xGC8KZ9x8G5JYxb8CrW4R51A93fl-1714213643921-0.0.1.1-604800000
Request	▸ Request line and header section (240 bytes) ▸ Request body (0 bytes)
Response	▸ Status line and header section (787 bytes) ▾ Response body (0 bytes)
Solution	Always scope cookies to a FQDN (Fully Qualified Domain Name).



Re-examine Cache-control Directives (1)

▾ GET https://www.gitlab.com

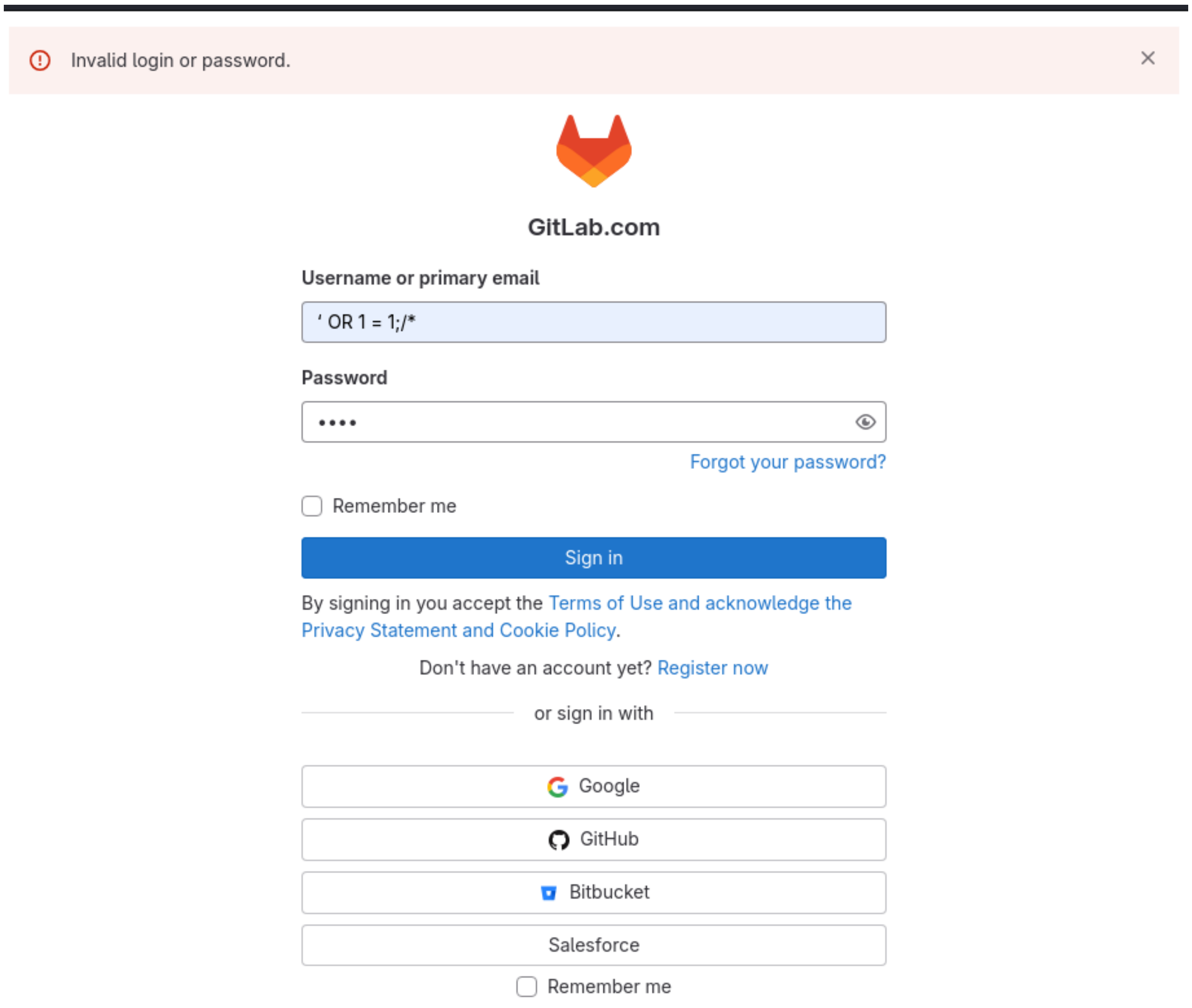
Alert tags	▪ WSTG-v42-ATHN-06
Alert description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.
Request	▸ Request line and header section (229 bytes) ▾ Request body (0 bytes)
Response	▸ Status line and header section (807 bytes) ▸ Response body (140369 bytes)
Parameter	`cache-control`
Evidence	`public, max-age=14400`
Solution	For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".

Manual Testing

SQL injection

If this page is vulnerable to SQL injection attack the Query will be executed as:

SELECT * FROM users WHERE email address = ‘ OR 1 = 1;/* AND password = */--



Invalid login or password.

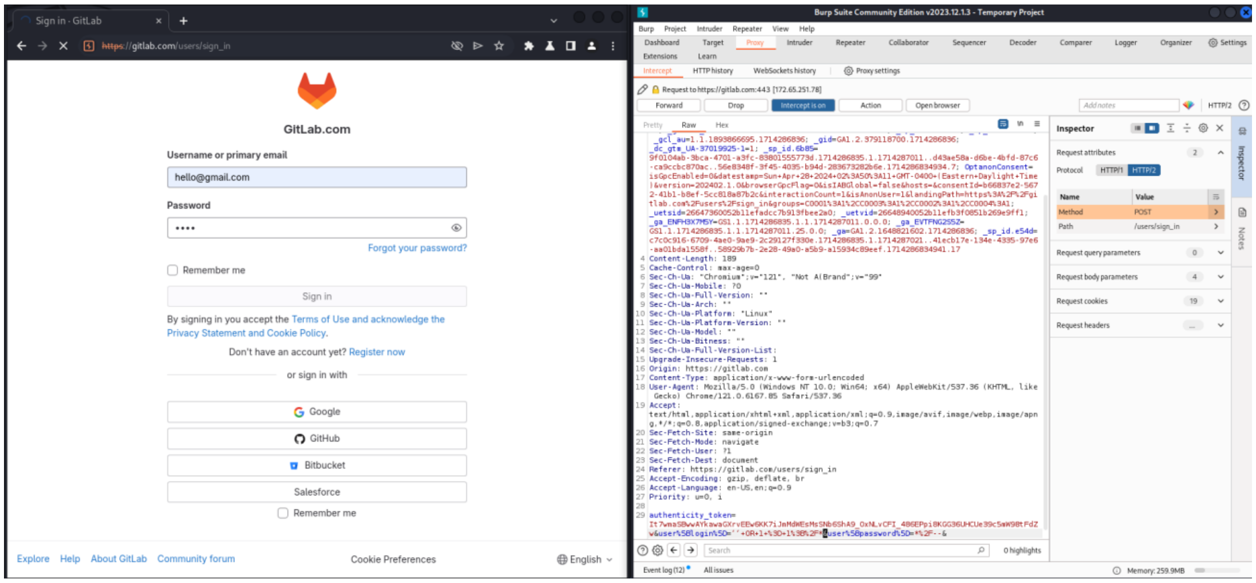
GitLab.com

Username or primary email: ‘ OR 1 = 1;/*

Password: ••••

By signing in you accept the Terms of Use and acknowledge the Privacy Statement and Cookie Policy.

Don't have an account yet? Register now


Username or primary email: hello@gmail.com

Request to https://gitlab.com:443 [172.65.251.78]

```
_gcl_au=1.1.1893866695.1714286836; _gid=GA1.2.379118700.1714286836;
_dc_gtm_UA-37019925-1=1; _sp_id.6b85=
9f0104ab-3bca-4701-a3fc-83801555773d.1714286835.1.1714287011..d43ae58a-d6be-4bfd-87c6
-ca9ccbc870ac..56e8348f-3f45-4035-b94d-283673282b6e.1714286834934.7; OptanonConsent=
isGpcEnabled=0&datestamp=Sun+Apr+28+2024+02%3A50%3A11+GMT-0400+(Eastern+Daylight+Time
)&version=202402.1.0&browserGpcFlag=0&isIABGlobal=false&hosts=&consentId=b66837e2-567
2-41b1-b8ef-5cc818a87b2c&interactionCount=1&isAnonUser=1&landingPath=https%3A%2F%2Fgi
tlab.com%2Fusers%2Fsign_in&groups=C0001%3A1%2CC0003%3A1%2CC0002%3A1%2CC0004%3A1;
_uetsid=26647360052b11efadcc7b913fbee2a0; _uetvid=26648940052b11efb3f0851b269e9ff1;
_ga_ENFH3X7M5Y=GS1.1.1714286835.1.1.1714287011.0.0.0; _ga_EVTFNG2S5Z=
GS1.1.1714286835.1.1.1714287011.25.0.0; _ga=GA1.2.1648821602.1714286836; _sp_id.e54d=
c7c0c916-6709-4ae0-9ae9-2c29127f330e.1714286835.1.1714287021..41ecb17e-134e-4335-97e6
-aa01bda1558f..58929b7b-2e28-49a0-a5b9-a15934c89eef.1714286834941.17
Content-Length: 189
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Full-Version: ""
Sec-Ch-Ua-Arch: ""
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Platform-Version: ""
Sec-Ch-Ua-Model: ""
Sec-Ch-Ua-Bitness: ""
Sec-Ch-Ua-Full-Version-List:
Upgrade-Insecure-Requests: 1
Origin: https://gitlab.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/121.0.6167.85 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://gitlab.com/users/sign_in
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

authenticity_token=
It7wmaSBwwAYkawaGXrvEEw6KK7iJmMdWEsMsSNb6ShA9_OxNLvCFI_486EPpi8KGG36UHCUe39c5aW98tFdZ
w&user%5Blogin%5D=''+OR+1+%3D+1%3B%2F*&user%5Bpassword%5D=*%2F--&
```

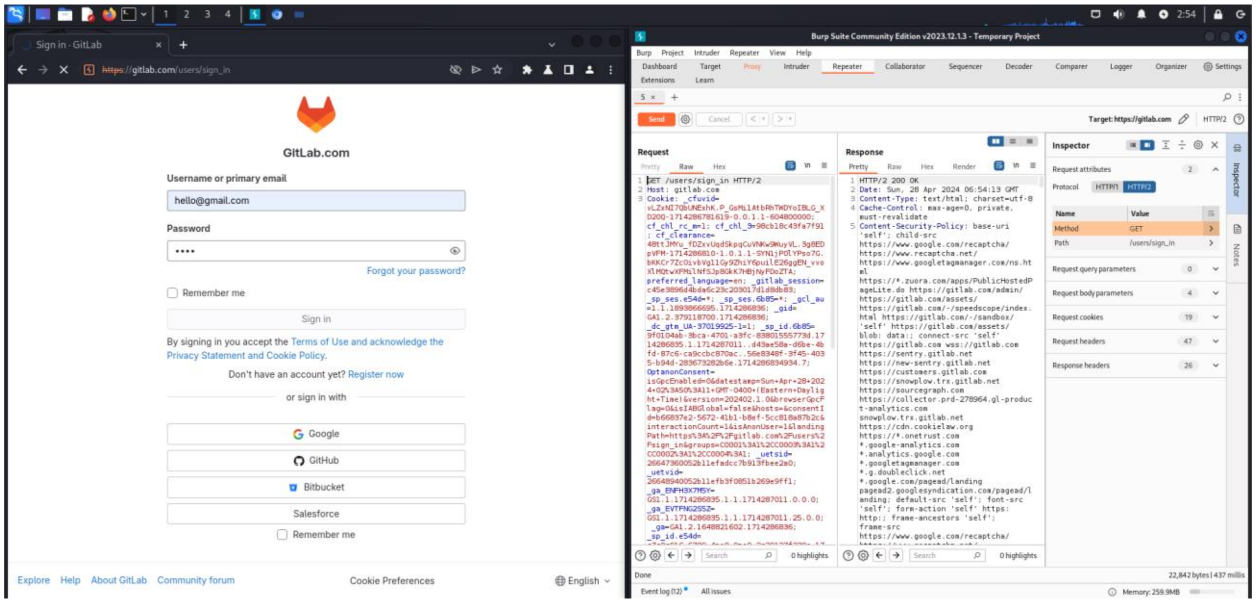
Name	Value
Method	POST
Path	/users/sign_in

They have implemented password policies, so I attempted to intercept the request and conduct SQL injection, but I was unsuccessful. The system is well secured against SQL injection.

Checking for Insecure HTTP methods

Specific HTTP methods, particularly DELETE and PUT, pose security threats to the server. DELETE has the ability to eliminate resources from the server, while PUT can upload and execute files. However, these methods should be used with caution as they could potentially jeopardize the Confidentiality, Integrity, and Availability of the server and its users.

To determine the supported HTTP methods, I used a Burp proxy to capture requests and a Repeater to alter the request method. This allowed me to send the adjusted requests to the server and analyze the responses.



```
GET /users/sign_in HTTP/2
Host: gitlab.com
Cookie: _cfuvid=
vLZxNI7QbUNExhK.P_GsMilAtbRhTWDYoIBLG_X
D20Q-1714286781619-0.0.1.1-604800000;
cf_chl_rc_m=1; cf_chl_9=98cb18c43fa7f91
; cf_clearance=
48ttJMYu_fDZxvUqdSkpqCuVNKw9MuyVL.3g8ED
pVFM-1714286810-1.0.1.1-SYN1jPOlYPso7G.
bKKCr7ZcOivbVgl1Gy9ZhiY6puilE26ggEN_vvo
XlMQtwXFMilNfSJp8GkK7HBjNyFDoZTA;
preferred_language=en; _gitlab_session=
c45e3896d4bda6c23c203017d1d8db83;
_sp_ses.e54d=*; _sp_ses.6b85=*; _gcl_au
=1.1.1893866695.1714286836; _gid=
GA1.2.379118700.1714286836;
_dc_gtm_UA-37019925-1=1; _sp_id.6b85=
9f0104ab-3bca-4701-a3fc-83801555773d.17
14286835.1.1714287011..d43ae58a-d6be-4b
fd-87c6-ca9ccbc870ac..56e8348f-3f45-403
5-b94d-283673282b6e.1714286834934.7;
OptanonConsent=
isGpcEnabled=0&datestamp=Sun+Apr+28+202
4+02%3A50%3A11+GMT-0400+(Eastern+Daylig
ht+Time)&version=202402.1.0&browserGpcF
lag=0&isIABGlobal=false&hosts=&consentI
d=b66837e2-5672-41b1-b8ef-5cc818a87b2c&
interactionCount=1&isAnonUser=1&landing
Path=https%3A%2F%2Fgitlab.com%2Fusers%2
Fsign_in&groups=C0001%3A1%2CC0003%3A1%2
CC0002%3A1%2CC0004%3A1; _uetsid=
26647360052b11efadcc7b913fbee2a0;
_uetvid=
26648940052b11efb3f0851b269e9ff1;
_ga_ENFH3X7M5Y=
GS1.1.1714286835.1.1.1714287011.0.0.0;
_ga_EVTFNG2S5Z=
GS1.1.1714286835.1.1.1714287011.25.0.0;
 _ga=GA1.2.1648821602.1714286836;
_sp_id.e54d=
```

```
HTTP/2 200 OK
Date: Sun, 28 Apr 2024 06:54:19 GMT
Content-Type: text/html; charset=utf-8
Cache-Control: max-age=0, private,
must-revalidate
Content-Security-Policy: base-uri
'self'; child-src
https://www.google.com/recaptcha/
https://www.recaptcha.net/
https://www.googletagmanager.com/ns.ht
ml
https://*.zuora.com/apps/PublicHostedP
ageLite.do https://gitlab.com/admin/
https://gitlab.com/assets/
https://gitlab.com/-/speedscope/index.
html https://gitlab.com/-/sandbox/
'self' https://gitlab.com/assets/
blob: data:; connect-src 'self'
https://gitlab.com wss://gitlab.com
https://sentry.gitlab.net
https://new-sentry.gitlab.net
https://customers.gitlab.com
https://snowplow.trx.gitlab.net
https://sourcegraph.com
https://collector.prd-278964.gl-produc
t-analytics.com
snowplow.trx.gitlab.net
https://cdn.cookielaw.org
https://*.onetrust.com
*.google-analytics.com
*.analytics.google.com
*.googletagmanager.com
*.g.doubleclick.net
*.google.com/pagead/landing
pagead2.googlesyndication.com/pagead/l
anding; default-src 'self'; font-src
'self'; form-action 'self' https:
http:; frame-ancestors 'self';
frame-src
https://www.google.com/recaptcha/
```

Username or primary email: hello@gmail.com
Password: ••••

The server only supports the POST method. Therefore, no insecure methods were utilized on the server.

Conclusion

From a security perspective, the web application and its associated authorization portal are well-maintained and managed. Despite instances of information disclosure, these can be effectively managed, and the associated risk is deemed low. It's important to highlight that all detected vulnerabilities fell into the categories of either Informational or Low levels. Furthermore, the vulnerabilities classified as Low are not easily exploitable, as previously demonstrated.

References

OWASP Top Ten | OWASP Foundation

WSTG - Stable | OWASP Foundation

ZAP (zaproxy.org)

GitHub - aboul3la/Sublist3r: Fast subdomains enumeration tool for penetration testers

GitHub - tomnomnom/httprobe: Take a list of domains and probe for working HTTP and HTTPS servers

Kali Tools | Kali Linux Tools

Netcraft | Leader in Phishing Detection, Cybercrime Disruption and Website Takedown

Nmap: the Network Mapper - Free Security Scanner