Website Vulnerability Scanner Report

✔ http://87.237.52.246:8080

Summary

Overall risk level:

High

Risk ratings:

High:	1
Medium:	3
Low:	4
Info:	46

Scan information:

Start time:	2022-12-13 01:33:02 UTC+02
Finish time:	2022-12-13 01:42:34 UTC+02
Scan duration:	9 min, 32 sec
Tests performed:	54/54
Scan status:	Finished

Findings

Passwords are submitted unencrypted over the network CONFIRMED

URL	Evidence
http://87.237.52.246:8080/login	Password input detected over unsecure HTTP. Login form: <input id="password" name="password" placeholder="Geben Sie Ihr Passwort ein" required="" type="password"/>

Details

Risk description:

An attacker could intercept the communication between the web browser and the server and he could retrieve the clear-text authentication credentials.

Recommendation:

We recommend you to reconfigure the web server so it uses HTTPS - which encrypts the communication between the web browser and the server. This way, the attacker will not be able to obtain the clear-text passwords, even though he manages to intercept the network communication.

Classification:

CWE : CWE-523
OWASP Top 10 - 2013 : A6 - Sensitive Data Exposure
OWASP Top 10 - 2017 : A3 - Sensitive Data Exposure

Screenshot:

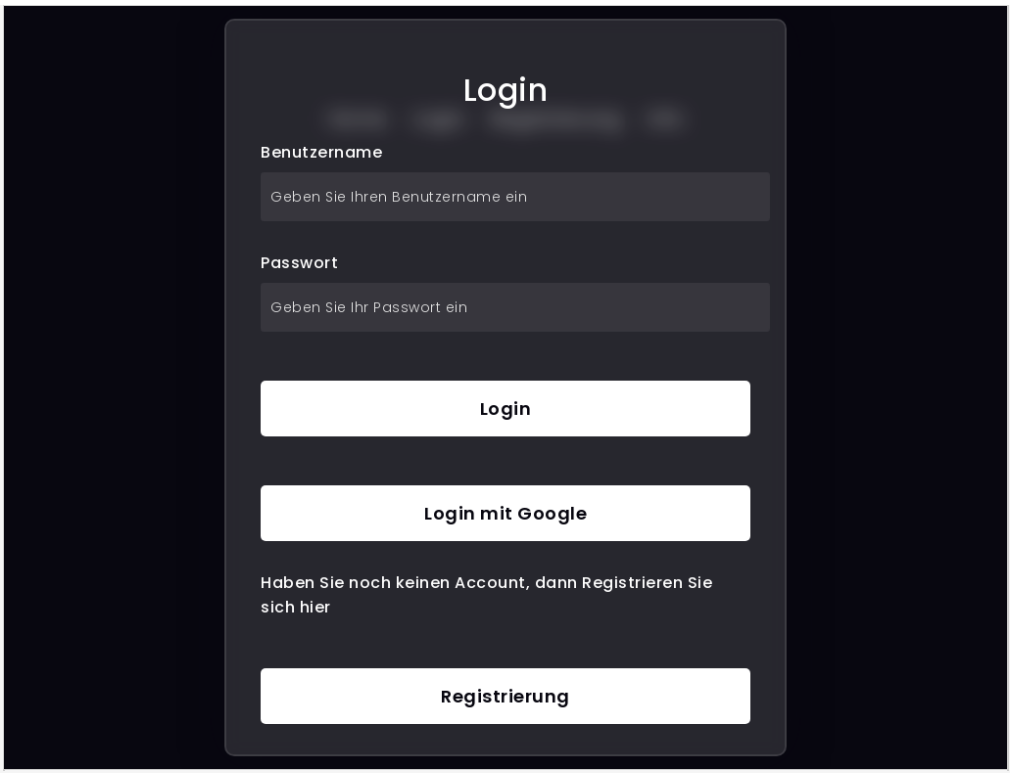


Figure 1. Password field found

Communication is not secure CONFIRMED

URL	Evidence
http://87.237.52.246:8080	Communication is made over unsecure, unencrypted HTTP.

Details

Risk description:
The communication between the web browser and the server is done using the HTTP protocol, which transmits data unencrypted over the network. Thus, an attacker who manages to intercept the communication at the network level is able to read and modify the data transmitted (including passwords, secret tokens, credit card information and other sensitive data).

Recommendation:
We recommend you to reconfigure the web server to use HTTPS - which encrypts the communication between the web browser and the server.

Classification:
CWE : CWE-311
OWASP Top 10 - 2013 : A6 - Sensitive Data Exposure
OWASP Top 10 - 2017 : A3 - Sensitive Data Exposure

Insecure cookie setting: missing HttpOnly flag CONFIRMED

URL	Cookie Name	Evidence
http://87.237.52.246:8080/login	csrftoken	Set-Cookie: csrftoken=B0j2vl5vJqxa8iIW3OTIcBQUSx1ZMb07; expires=Mon, 11 Dec 2023 23:33:35 GMT; Max-Age=31449600; Path=/; SameSite=Lax

Details

Risk description:
A cookie has been set without the `HttpOnly` flag, which means that it can be accessed by the JavaScript code running inside the web page. If an attacker manages to inject malicious JavaScript code on the page (e.g. by using an XSS attack) then the cookie will be

accessible and it can be transmitted to another site. In case of a session cookie, this could lead to session hijacking.

Recommendation:
Ensure that the HttpOnly flag is set for all cookies.

References:
https://owasp.org/www-community/HttpOnly

Classification:
CWE : CWE-1004
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Insecure cookie setting: missing Secure flag CONFIRMED

URL	Cookie Name	Evidence
http://87.237.52.246:8080/login	csrftoken	Set-Cookie: csrftoken=B0j2vI5vJqxa8iIW3OTIcBQUSx1ZMb07; expires=Mon, 11 Dec 2023 23:33:35 GMT; Max-Age=31449600; Path=/; SameSite=Lax

Details

Risk description:
Since the `Secure` flag is not set on the cookie, the browser will send it over an unencrypted channel (plain HTTP) if such a request is made. Thus, the risk exists that an attacker will intercept the clear-text communication between the browser and the server and he will steal the cookie of the user. If this is a session cookie, the attacker could gain unauthorized access to the victim's web session.

Recommendation:
Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.

References:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html

Classification:
CWE : CWE-614
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: Content-Security-Policy CONFIRMED

URL	Evidence
http://87.237.52.246:8080	Response headers do not include the HTTP Content-Security-Policy security header

Details

Risk description:
The Content-Security-Policy (CSP) header activates a protection mechanism implemented in web browsers which prevents exploitation of Cross-Site Scripting vulnerabilities (XSS). If the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Recommendation:
Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

References:
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

Classification:

CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-XSS-Protection CONFIRMED

URL	Evidence
http://87.237.52.246:8080	Response headers do not include the HTTP X-XSS-Protection security header

Details

Risk description:
The `X-XSS-Protection` HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

Recommendation:
We recommend setting the X-XSS-Protection header to `X-XSS-Protection: 1; mode=block` .

References:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Internal Server Error Found CONFIRMED

URL	Method	Parameters	Evidence
http://87.237.52.246:8080/myshare	POST	Body: delete=True id=9033785474003477892	Response has an internal server error status code: 500

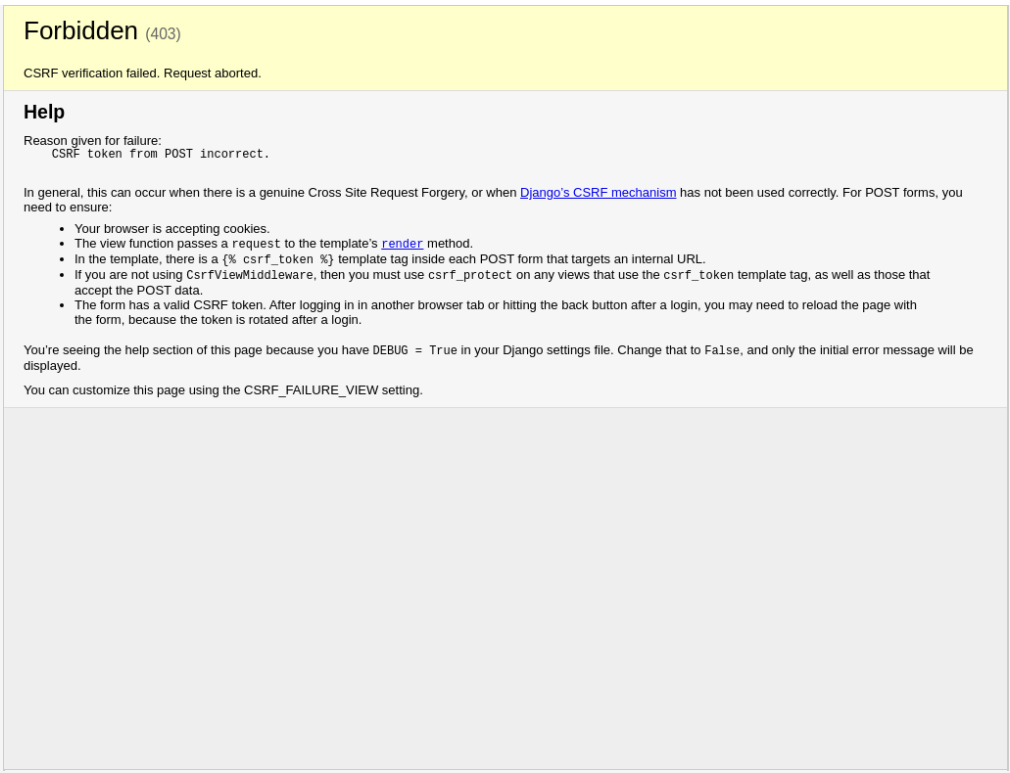
Details

Risk description:
The website does not handle or incorrectly handles an exceptional condition. An attacker may use the contents of error messages to help launch another, more focused attack. For example, an attempt to exploit a path traversal weakness (CWE-22) might yield the full pathname of the installed application.

Recommendation:
Ensure that error messages only contain minimal details that are useful to the intended audience, and nobody else. The messages need to strike the balance between being too cryptic and not being cryptic enough. They should not necessarily reveal the methods that were used to determine the error. Such detailed information can be used to refine the original attack to increase the chances of success. If errors must be tracked in some detail, capture them in log messages - but consider what could occur if the log messages can be viewed by attackers. Avoid recording highly sensitive information such as passwords in any form. Avoid inconsistent messaging that might accidentally tip off an attacker about internal state, such as whether a username is valid or not.

Classification:
CWE : CWE-209
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Screenshot:



Forbidden (403)

CSRF verification failed. Request aborted.

Help

Reason given for failure:
```
CSRF token from POST incorrect.
```

In general, this can occur when there is a genuine Cross Site Request Forgery, or when Django's CSRF mechanism has not been used correctly. For POST forms, you need to ensure:

- Your browser is accepting cookies.
- The view function passes a `request` to the template's `render` method.
- In the template, there is a `{% csrf_token %}` template tag inside each POST form that targets an internal URL.
- If you are not using `CsrfViewMiddleware`, then you must use `csrf_protect` on any views that use the `csrf_token` template tag, as well as those that accept the POST data.
- The form has a valid CSRF token. After logging in in another browser tab or hitting the back button after a login, you may need to reload the page with the form, because the token is rotated after a login.

You're seeing the help section of this page because you have `DEBUG = True` in your Django settings file. Change that to `False`, and only the initial error message will be displayed.

You can customize this page using the CSRF_FAILURE_VIEW setting.

Figure 2. Internal Error

Possible Broken Authentication UNCONFIRMED

URL	Method	Parameters
http://87.237.52.246:8080/crshare	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxlQG
http://87.237.52.246:8080/info	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhklFQnafQXZh4mQUP
http://87.237.52.246:8080/shshare	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxlQG

Details

Risk description:
We found that the application might be vulnerable to Broken Authentication. Broken Authentication happens when an endpoint that should only be accessible using an authorization mechanism (generally HTTP Headers or Cookies) is accessible without providing any credentials. At the very least, this allows a malicious user to read confidential data. Depending on the functionality of the page, other sensitive actions might be possible. Please note that an automated scanner cannot know how sensitive the information on a page is. As a result, this type of vulnerability needs to be manually validated.

Recommendation:
We recommend that, on the server-side, the first thing you do is check that the expected set of credentials came with the HTTP request. Do this before any other action to avoid giving access to functionality or data that should be protected by authentication. Don't limit the check only to the presence of an HTTP Header. Always check that the received value is the one you expect. If this condition is not fulfilled, immediately return an error response without proceeding further.

Classification:
CWE : CWE-287
OWASP Top 10 - 2013 : A2 - Broken Authentication and Session Management

OWASP Top 10 - 2017 : A2 - Broken Authentication

Login Interface Found CONFIRMED

URL	Evidence
http://87.237.52.246:8080/login	<input id="username" name="username" placeholder="Geben Sie Ihren Benutzername ein" required="" type="text"/> <input id="password" name="password" placeholder="Geben Sie Ihr Passwort ein" required="" type="password"/> <button type="submit">Login </button>
http://87.237.52.246:8080/register	<input id="username" name="username" placeholder="Geben Sie Ihren Benutzername ein" required="" type="text"/> <input id="password" name="password" placeholder="Geben Sie Ihr Passwort ein" required="" type="password"/> <button type="submit">Registrieren </button>

Details

Risk description:
An attacker could use this interface to mount brute force attacks against known passwords and usernames combinations leaked throughout the web.

Recommendation:
Ensure each interface is not bypassable using common knowledge of the application or leaked credentials using occasional password audits.

References:
https://pentest-tools.com/network-vulnerability-scanning/password-auditor
http://capec.mitre.org/data/definitions/16.html

Screenshot:

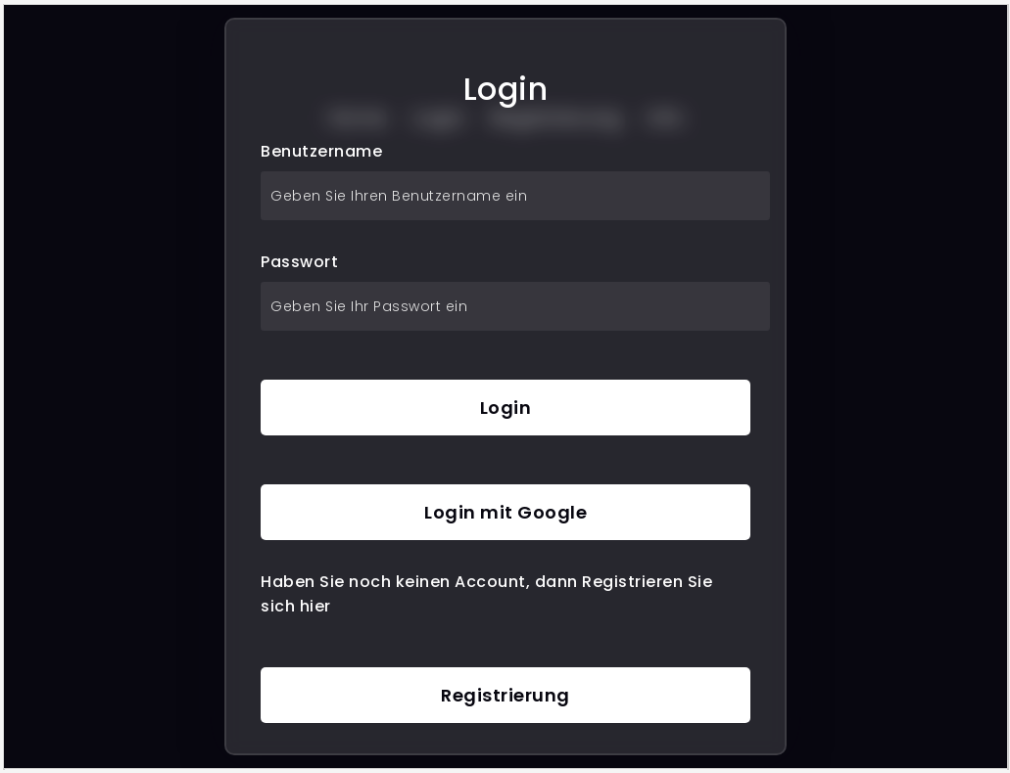


Figure 3. Login Interface

Security.txt file is missing CONFIRMED

URL

Missing: http://87.237.52.246:8080/.well-known/security.txt

Details

Risk description:
We have detected that the server is missing the security.txt file. There is no particular risk in not creating a valid Security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

Recommendation:
We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.

References:
https://securitytxt.org/

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Authentication complete: Automatic method.

URL
http://87.237.52.246:8080/login

Details

Screenshot:

Figure 4. Authentication sequence result

Spider results

URL	Method	Parameters

http://87.237.52.246:8080/	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhkIFQnafQXZh4mQUP
http://87.237.52.246:8080/crshare	POST	Body: docname=1d3d2d231d2dd4 document=This is a file Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhkIFQnafQXZh4mQUP
http://87.237.52.246:8080/crshare	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxIQG
http://87.237.52.246:8080/info	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhkIFQnafQXZh4mQUP
http://87.237.52.246:8080/login	POST	Body: password=Secure123456$ username=1d3d2d231d2dd4 Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhkIFQnafQXZh4mQUP sessionid=9l3e5iu55yh2jldxadc30fvxabt98o1l
http://87.237.52.246:8080/login	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxIQG
http://87.237.52.246:8080/media/5/	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhkIFQnafQXZh4mQUP sessionid=9l3e5iu55yh2jldxadc30fvxabt98o1l
http://87.237.52.246:8080/myshare	POST	Body: delete=True id=9033785474003477892 Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhkIFQnafQXZh4mQUP
http://87.237.52.246:8080/myshare	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxIQG
http://87.237.52.246:8080/register	POST	Body: email=example_email%40example.com firstname=1d3d2d231d2dd4 lastname=1d3d2d231d2dd4 password=Secure123456%24 passwordcon=Secure123456%24 username=1d3d2d231d2dd4

http://87.237.52.246:8080/register	POST	Body: email=example_email@example.com firstname=1d3d2d231d2dd4 lastname=1d3d2d231d2dd4 password=Secure123456$ passwordcon=Secure123456$ username=1d3d2d231d2dd4 Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies:...
http://87.237.52.246:8080/register	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxIQG
http://87.237.52.246:8080/shshare	POST	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=14cf80W0d0741wdhkIFQnafQXZh4mQUP
http://87.237.52.246:8080/shshare	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxIQG
http://87.237.52.246:8080/static/	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxIQG
http://87.237.52.246:8080/static	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Cookies: csrftoken=BoqVLTGmJ1fJNNpQGagq4pdq9EHUxIQG
http://87.237.52.246:8080	GET	Headers: User-Agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36

Website is accessible.

Nothing was found for website technologies.

Nothing was found for vulnerabilities of server-side software.

Nothing was found for client access policies.

Nothing was found for robots.txt file.

Nothing was found for outdated JavaScript libraries.

- Nothing was found for CORS misconfiguration.
- Nothing was found for use of untrusted certificates.
- Nothing was found for enabled HTTP debug methods.
- Nothing was found for sensitive files.
- Nothing was found for administration consoles.
- Nothing was found for interesting files.
- Nothing was found for information disclosure.
- Nothing was found for software identification.
- Nothing was found for directory listing.
- Nothing was found for Cross-Site Scripting.
- Nothing was found for SQL Injection.
- Nothing was found for Local File Inclusion.
- Nothing was found for OS Command Injection.
- Nothing was found for error messages.
- Nothing was found for debug messages.
- Nothing was found for code comments.
- Nothing was found for missing HTTP header - Strict-Transport-Security.

- Nothing was found for missing HTTP header - X-Frame-Options.
- Nothing was found for missing HTTP header - X-Content-Type-Options.
- Nothing was found for missing HTTP header - Referrer.
- Nothing was found for missing HTTP header - Feature.
- Nothing was found for domain too loose set for cookies.
- Nothing was found for mixed content between HTTP and HTTPS.
- Nothing was found for cross domain file inclusion.
- Nothing was found for secure password submission.
- Nothing was found for sensitive data.
- Nothing was found for Server Side Request Forgery.
- Nothing was found for Open Redirect.
- Nothing was found for PHP Code Injection.
- Nothing was found for JavaScript Code Injection.
- Nothing was found for Ruby Code Injection.
- Nothing was found for Python Code Injection.
- Nothing was found for Perl Code Injection.
- Nothing was found for Remote Code Execution through Log4j.

Nothing was found for Server Side Template Injection.

Nothing was found for Remote Code Execution through VIEWSTATE.

Scan coverage information

List of tests performed (54/54)

- Checking for website accessibility...
- Trying to authenticate...
- Checking for secure communication...
- Checking for missing HTTP header - Content Security Policy...
- Checking for missing HTTP header - X-XSS-Protection...
- Checking for HttpOnly flag of cookie...
- Checking for login interfaces...
- Checking for passwords submitted unencrypted...
- Checking for Secure flag of cookie...
- Spidering target...
- Checking for website technologies...
- Checking for vulnerabilities of server-side software...
- Checking for client access policies...
- Checking for robots.txt file...
- Checking for absence of the security.txt file...
- Checking for outdated JavaScript libraries...
- Checking for CORS misconfiguration...
- Checking for use of untrusted certificates...
- Checking for enabled HTTP debug methods...
- Checking for sensitive files...
- Checking for administration consoles...
- Checking for internal error code...
- Checking for interesting files... (this might take a few hours)
- Checking for Broken Authentication...
- Checking for information disclosure... (this might take a few hours)
- Checking for software identification...
- Checking for directory listing...
- Checking for Cross-Site Scripting...
- Checking for SQL Injection...
- Checking for Local File Inclusion...
- Checking for OS Command Injection...
- Checking for error messages...
- Checking for debug messages...
- Checking for code comments...
- Checking for missing HTTP header - Strict-Transport-Security...
- Checking for missing HTTP header - X-Frame-Options...
- Checking for missing HTTP header - X-Content-Type-Options...
- Checking for missing HTTP header - Referrer...
- Checking for missing HTTP header - Feature...
- Checking for domain too loose set for cookies...
- Checking for mixed content between HTTP and HTTPS...
- Checking for cross domain file inclusion...
- Checking for secure password submission...
- Checking for sensitive data...
- Checking for Server Side Request Forgery...
- Checking for Open Redirect...
- Checking for PHP Code Injection...
- Checking for JavaScript Code Injection...
- Checking for Ruby Code Injection...
- Checking for Python Code Injection...
- Checking for Perl Code Injection...
- Checking for Remote Code Execution through Log4j...
- Checking for Server Side Template Injection...

✔ Checking for Remote Code Execution through VIEWSTATE...

Scan parameters

Website URL:	http://87.237.52.246:8080
Scan type:	Full_scan_default
Authentication:	True

Scan stats

Unique Injection Points Detected:	17
URLs spidered:	24
Total number of HTTP requests:	17899
Average time until a response was received:	40ms
Total number of HTTP request errors:	34