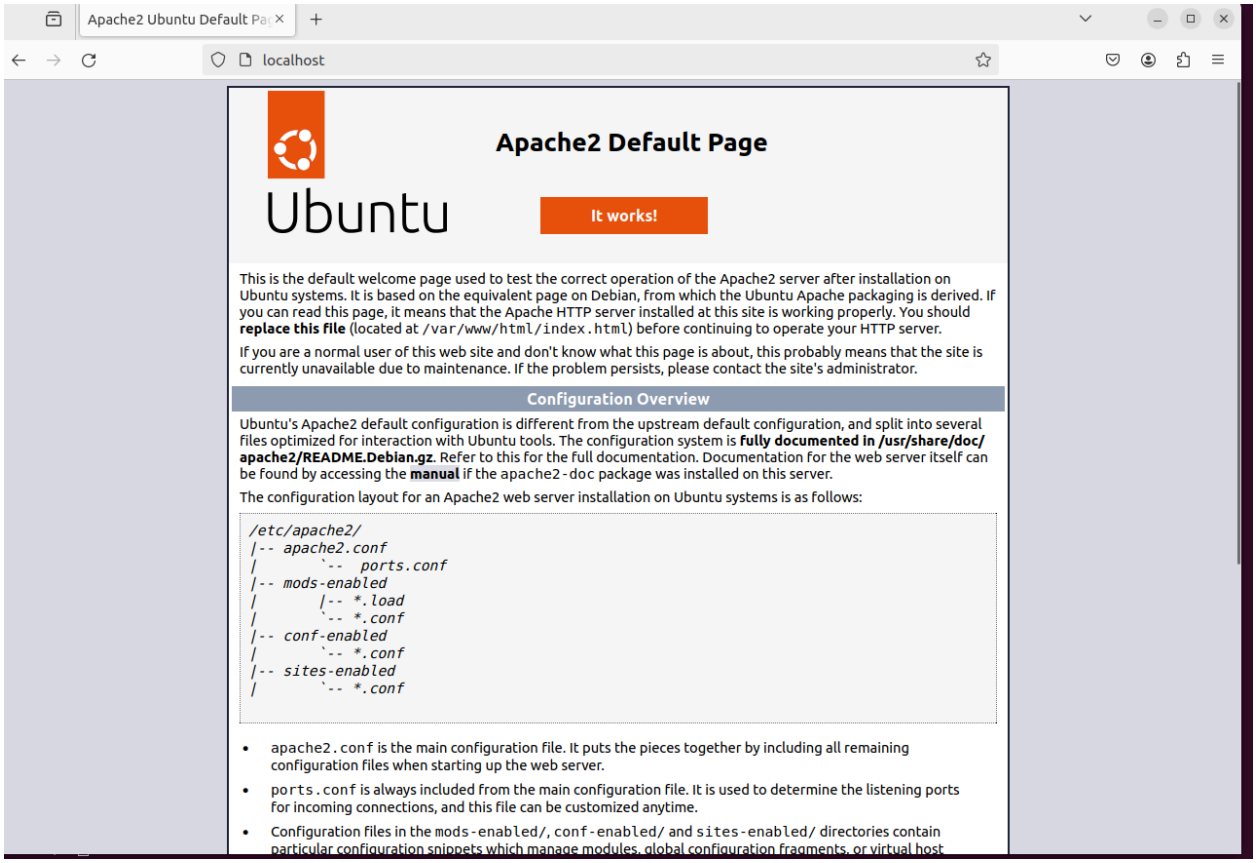
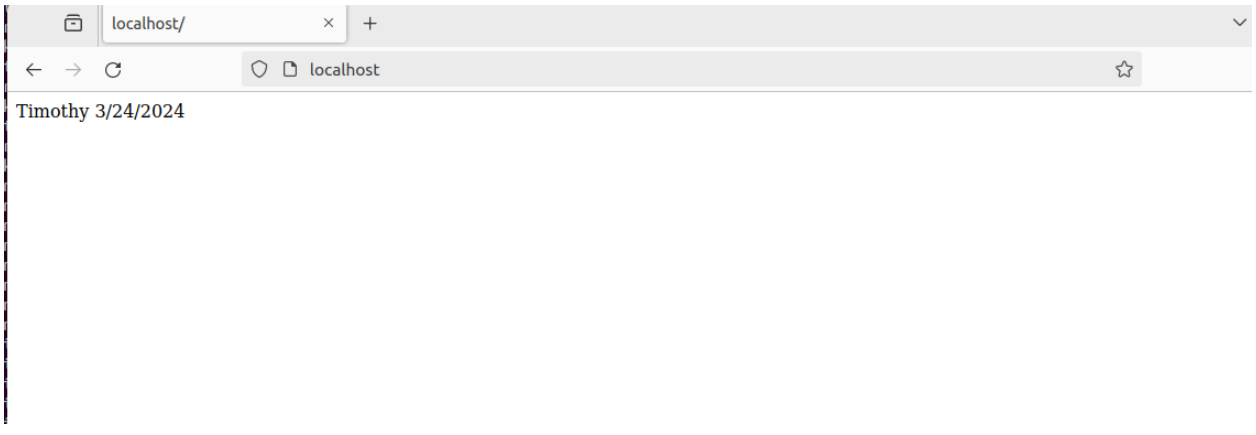
Task 1:



In this step, I have downloaded the apache webserver, started it and can view it in my web browser.



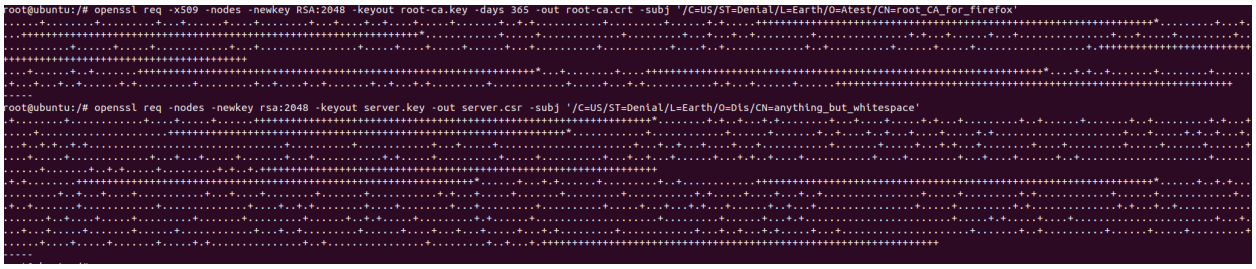
I have updated the default html page with my name and date.

```
root@ubuntu:/# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

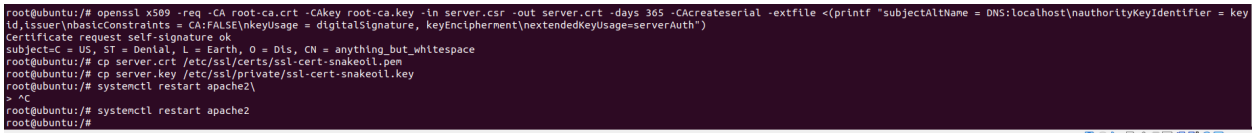
In this step, I am enabling SSL.

```
root@ubuntu:/# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

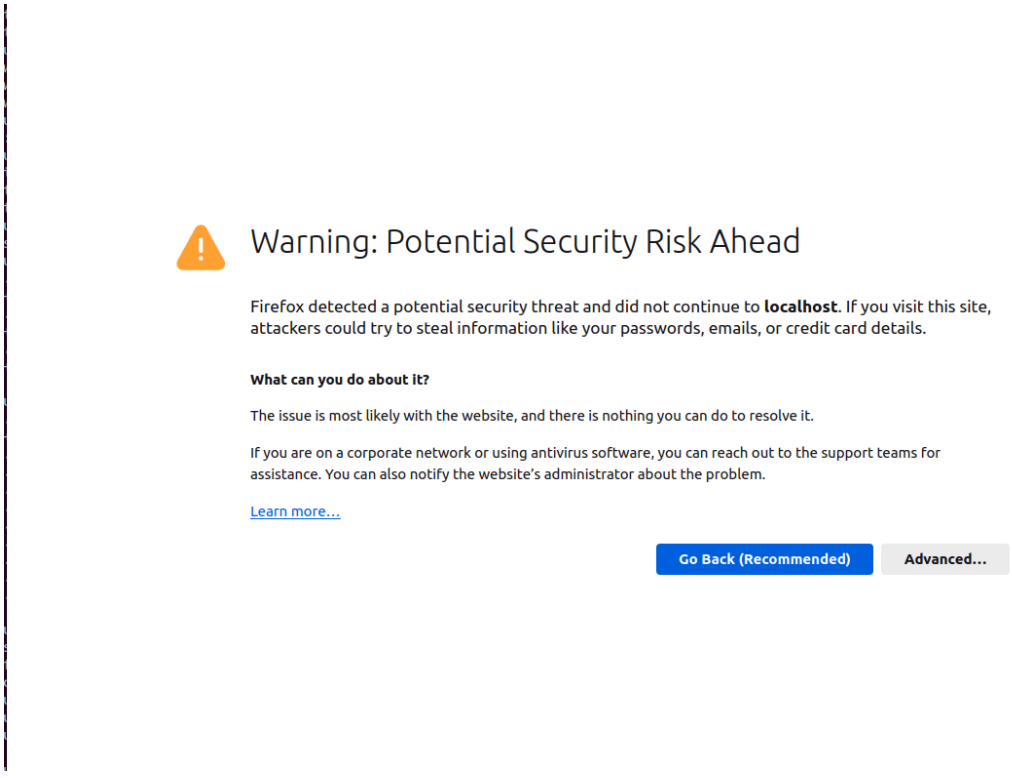
In this step, I am enabling the default SSL site.



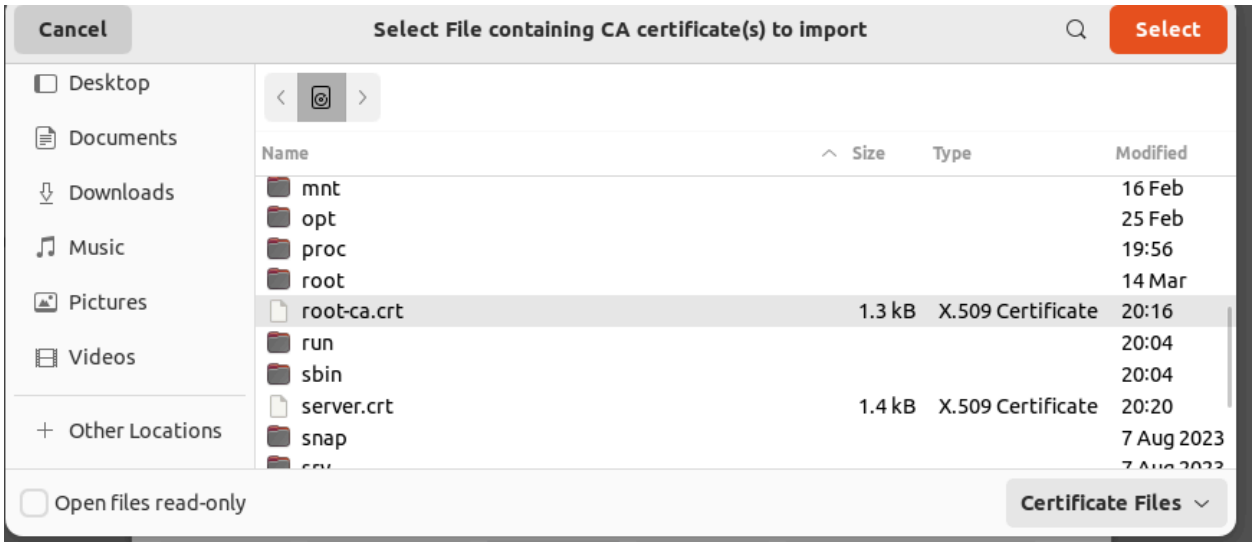
Here I am creating a private certificate authority and creating a private key with a certificate signing request.



In this step, I am first creating a TLS self-signed certificate and then replacing the default certificate and key for our Apache site. I then am restarting the apache server.



Here we can now see that there is a warning when we go back to the browser.



In this step, I am downloading the certificate.

Now, I can open the web page and see that it is loading with no errors. It also has the lock icon in the search bar, meaning that it is TLS secured.

```
root@ubuntu:/# apt install libapache2-mod-security2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  liblua5.1-0 modsecurity-crs
Suggested packages:
  lua geoip-database-contrib ruby python
The following NEW packages will be installed:
  libapache2-mod-security2 liblua5.1-0 modsecurity-crs
0 upgraded, 3 newly installed, 0 to remove and 107 not upgraded.
Need to get 504 kB of archives.
After this operation, 2,376 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 liblua5.1-0 amd64 5.1.5-8.1build4 [99.9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libapache2-mod-security2 amd64 2.9.5-1 [265 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 modsecurity-crs all 3.3.2-1 [139 kB]
Fetched 504 kB in 1s (558 kB/s)
Selecting previously unselected package liblua5.1-0:amd64.
(Reading database ... 256085 files and directories currently installed.)
Preparing to unpack .../liblua5.1-0_5.1.5-8.1build4_amd64.deb ...
Unpacking liblua5.1-0:amd64 (5.1.5-8.1build4) ...
Selecting previously unselected package libapache2-mod-security2.
Preparing to unpack .../libapache2-mod-security2_2.9.5-1_amd64.deb ...
Unpacking libapache2-mod-security2 (2.9.5-1) ...
Selecting previously unselected package modsecurity-crs.
Preparing to unpack .../modsecurity-crs_3.3.2-1_all.deb ...
Unpacking modsecurity-crs (3.3.2-1) ...
Setting up modsecurity-crs (3.3.2-1) ...
Setting up liblua5.1-0:amd64 (5.1.5-8.1build4) ...
Setting up libapache2-mod-security2 (2.9.5-1) ...
apache2_invoke: Enable module security2
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
root@ubuntu:/# mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
root@ubuntu:/# sed -i 's/SecRuleEngine DetectionOnly/SecRuleEngine On/g' /etc/modsecurity/modsecurity.conf
root@ubuntu:/# systemctl restart apache2
^[[D^[[C^[[D^C
root@ubuntu:/# systemctl restart apache2
root@ubuntu:/#
```

In this step, I am installing ModSecurity and setting up its config file. I then am turning on its blocking mode and restarting the apache server.

Forbidden

You don't have permission to access this resource.

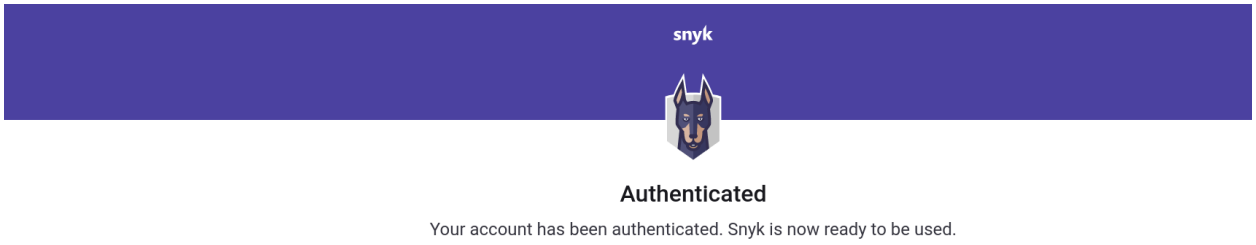
Apache/2.4.52 (Ubuntu) Server at localhost Port 443

I have navigated to https://localhost/?%3Cscript%3Ealert(%27xss%27)%3C/script%3E and see the forbidden response.

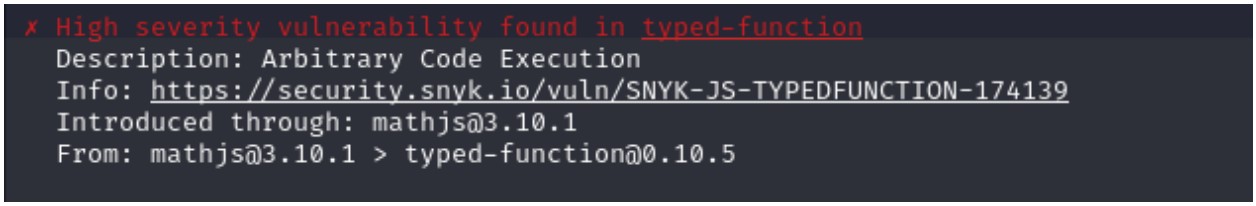
Task 2:

```
┌──(timothyd㉿kali)-[~]
└─$ wget https://static.snyk.io/cli/latest/snyk-linux
--2024-03-25 21:04:36--  https://static.snyk.io/cli/latest/snyk-linux
Resolving static.snyk.io (static.snyk.io)... 2600:1406:3c:396::ecd, 2600:140
6:3c:38a::ecd, 23.202.197.137
Connecting to static.snyk.io (static.snyk.io)|2600:1406:3c:396::ecd|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [binary/octet-stream]
Saving to: 'snyk-linux'

snyk-linux                                                  [

2024-03-25 21:04:57 (4.84 MB/s) - 'snyk-linux' saved [102850560]


┌──(timothyd㉿kali)-[~]
└─$ git clone https://github.com/appsecco/dvna
Cloning into 'dvna'...
remote: Enumerating objects: 645, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 645 (delta 1), reused 2 (delta 0), pack-reused 639
Receiving objects: 100% (645/645), 3.18 MiB | 6.83 MiB/s, done.
Resolving deltas: 100% (279/279), done.

┌──(timothyd㉿kali)-[~]
└─$ chmod +x snyk-linux

┌──(timothyd㉿kali)-[~]
└─$ sudo mv snyk-linux /usr/local/bin/snyk
[sudo] password for timothyd:

┌──(timothyd㉿kali)-[~]
└─$
```

In this step, I am cloning the git repo and installing the Snyk linux binary. I then am configuring its use.

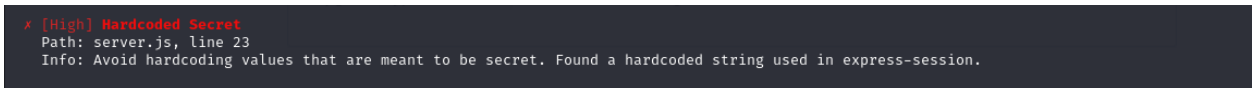


I have ran synk auth and logged in.



After running the test command, a list of vulnerabilities were returned and the image above is the one that interests me the most.

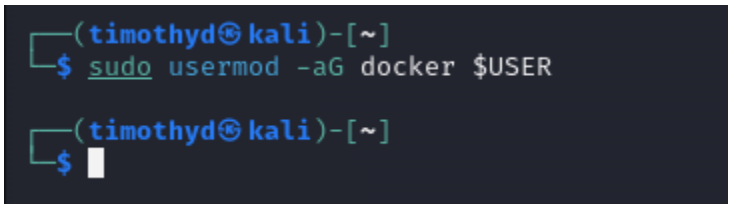
This vulnerability means that the typed-function library needs to be upgraded. This library is used for type checking of Javascript functions. The impact of this not being upgraded is that it may be vulnerable to arbitrary code execution.



In this step, I ran the “snyk code test” command and it returned some more vulnerabilities. One that stood out to me is the one that is pictured above.

This vulnerability means that somebody hardcoded something that shouldn’t be hardcoded. This could be something like a password or important information. Having something hardcoded is a security risk because if a hacker can get into your code repository, then your important information is just sitting right there for them to steal.

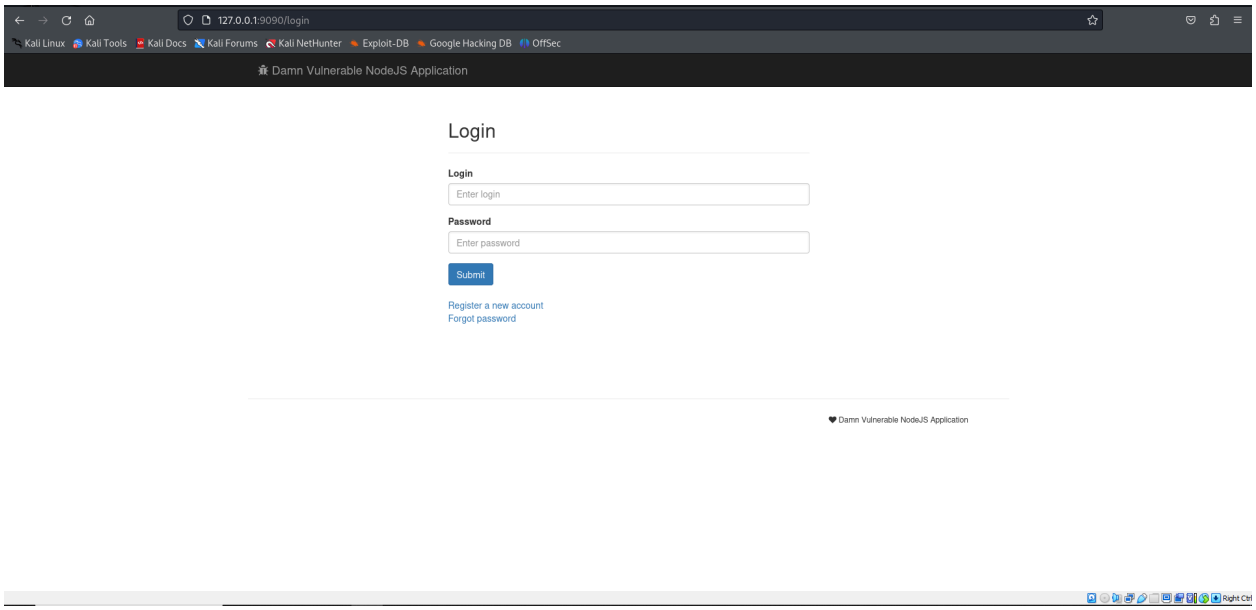
Task 3:



In this step, I have downloaded Docker and am adding my Kali VM to the user group.

```
┌──(timothyd㉿kali)-[~]
└─$ docker run --name dvna -p 9090:9090 -d appsecco/dvna:sqlite
Unable to find image 'appsecco/dvna:sqlite' locally
sqlite: Pulling from appsecco/dvna
57936531d1ee: Pull complete
b186cf19f9ed: Pull complete
eadbf8312262: Pull complete
cf528b18b6ce: Pull complete
075c4f074e90: Pull complete
d0562d9451f1: Pull complete
48671e1607ad: Pull complete
4879e9b180ec: Pull complete
4bcad28e8244: Pull complete
Digest: sha256:2657051b73ac8f879d3a3b419b4618d98e4596a0fe27f8e91582c403541ef1b0
Status: Downloaded newer image for appsecco/dvna:sqlite
1c29364e6b2b42ac783e922b95f35f91e5ee86d0b1e28b55886c12e3dec50900

┌──(timothyd㉿kali)-[~]
└─$
```

In this step, I am running the dvna docker container.



Here we can see the dvna is up and running.

```
2024-03-28 00:28:06 INFO  dastardly.ScanManager - Scan Finished, exiting
2024-03-28 00:28:06 ERROR dastardly.ScanFinishedHandler - Failing build as scanner identified issue(s) with severity higher than "INFO":
2024-03-28 00:28:06 ERROR dastardly.ScanFinishedHandler - Path: /forgotpw Issue Type: Vulnerable JavaScript dependency Severity: LOW
2024-03-28 00:28:06 ERROR dastardly.ScanFinishedHandler - Path: /login Issue Type: Vulnerable JavaScript dependency Severity: LOW
2024-03-28 00:28:06 ERROR dastardly.ScanFinishedHandler - Path: /assets/jquery-3.2.1.min.js Issue Type: Vulnerable JavaScript dependency Severity: LOW
2024-03-28 00:28:06 ERROR dastardly.ScanFinishedHandler - Path: /register Issue Type: Vulnerable JavaScript dependency Severity: LOW

┌──(timothyd㉿kali)-[~]
```

In this step, I have launched a Dastardly container targeting the local DVNA server. Here I can see the scan is completed and the severity levels are low for these vulnerabilities.