Notes Keeper

- **Author :** Ala Loghmari
- **Difficulty :** **Friendly/Easy**

Description

Notes Keeper is a web application where one can keep his notes, in this challenge, you will need to find your way through the web app by fuzzing the website logging-in/signing up for then to retrieve notes of other users through an IDOR vulnerabiility.

Skills Required

- Web Enumeration

Skills Learned

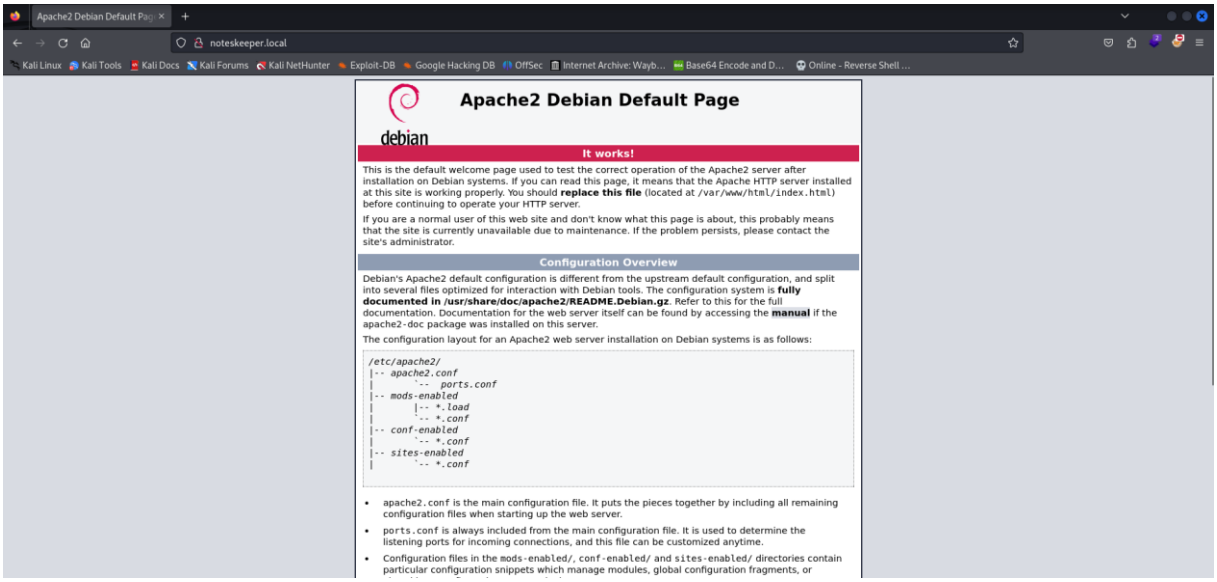
- Web Fuzzing
- Insecure direct object references (IDOR)

References & Guides

- https://portswigger.net/web-security/access-control/idor (Port Swigger)

Enumeration

Accessing the web application, we come across a default apache2 index page.



Nothing seems interesting at first but taking a look at the source code of the page we can see some hidden comments that might help us to move further in our enumeration process.

```
<html xmlns="http://www.w3.org/1999/xhtml">

    <!-- Stucked ? I've got a hint for you in here somewhere :)-->

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
      <tt>/var/www/html/index.html</tt>) before continuing to operate your HTTP server.
</p>

    <!-- Have you checked for any hidden directories?-->

<p>
      If you are a normal user of this web site and don't know what this page is
```

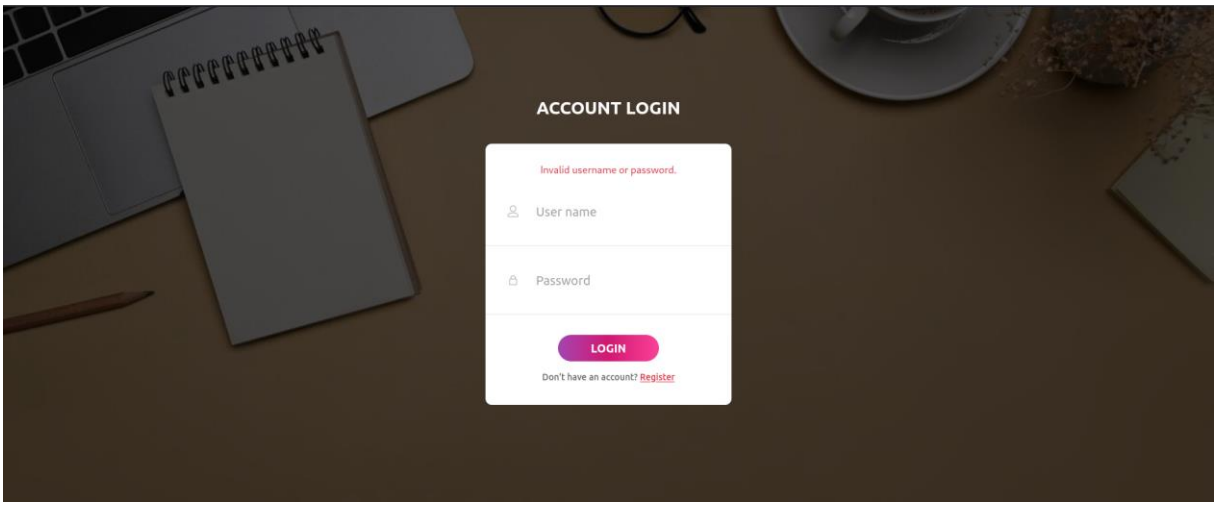
Hidden directories it is, many tools can help us with that such as **Gobuster**, **dirb**, **dirbuster**..

Delving into the source code, we get the tool that we can use as a hint.

```
</div>
            <!-- Have you heard of dirb ?-->

<div class="section_header">
```

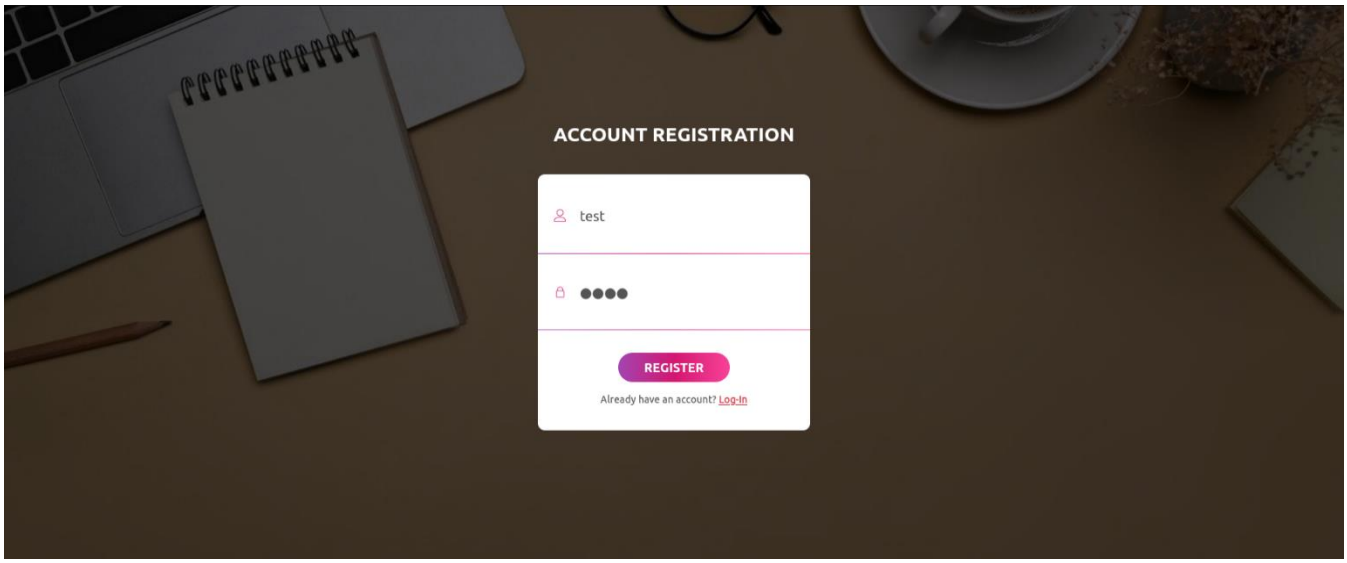
```
┌──(root㉿Voldemort)-[/home/voldemort/Desktop]
└─#  dirb http://noteskeeper.local

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Mar 17 17:20:10 2024
URL_BASE: http://noteskeeper.local
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://noteskeeper.local ----
+ http://noteskeeper.local/index.html (CODE:200|SIZE:11030)

==> DIRECTORY: http://noteskeeper.local/secret/

....


END_TIME: Sun Mar 17 17:20:12 2024
DOWNLOADED: 9224 - FOUND: 3
```

Letting dirb running for some time we come across a hidden directory called secret.
Taking a look at it, we find a login page.

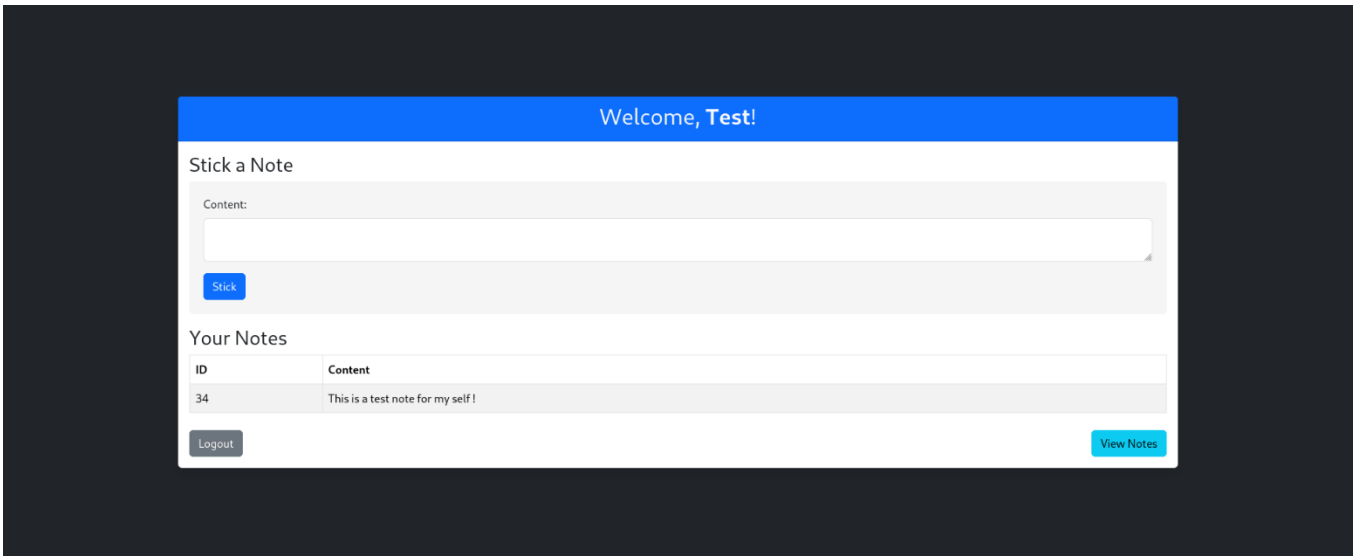


Trying some of the most known combinations (admin:admin, root:root, user:user…) doesn't seem to work as well.

From here, and in order to go further, we decide to create an account by clicking the Register link redirecting us to /register.php.



Once the account is created we get redirected to /home.php where we can, from what it seems, write notes to keep for ourselves.



Trying to create a note we notice something odd, the id of the note does not begin at 1, for instance, the one we created has been assigned to it the id “34”.

Keeping that in mind, we checked the source code of the page looking for a hint or something that could help us with our enumeration.

Hitting CTRL U to view the source code we come across a new hidden message/comment.

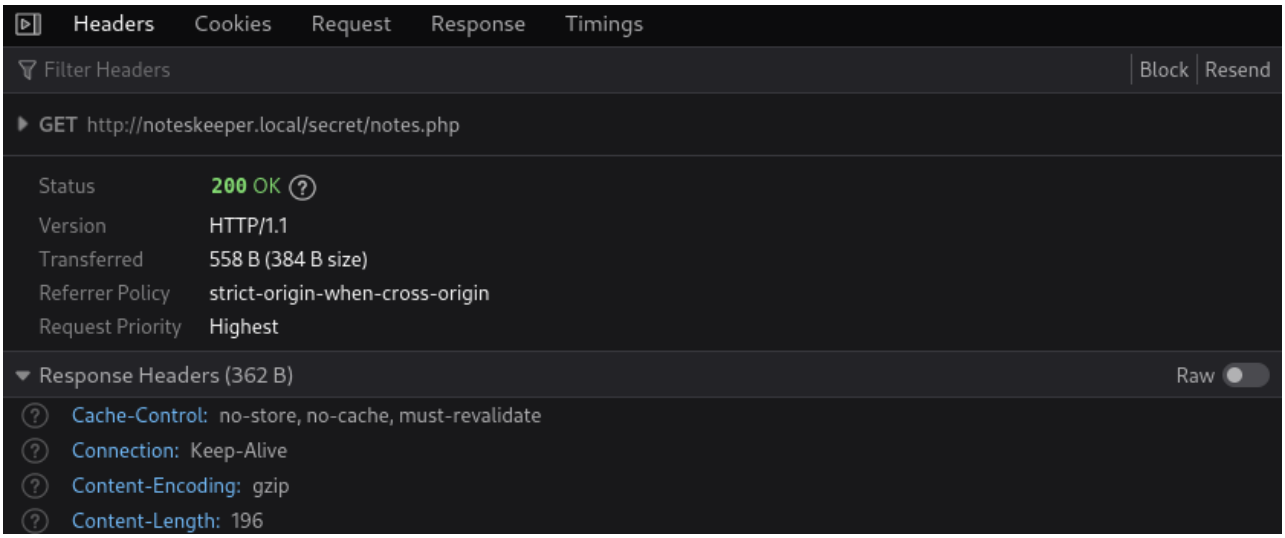
```
" Hey Sam, if you're reading this, I've left for you a way into my notes
that you need to read, you can check "view notes", I know you can find your
way from there! "
```

```
  </style>
</head>
<body class="bg-dark">
    <!-- Hey Sam, if you're reading this, I've left for you a way into my notes that you need to read, you can check "view notes", I know you can find your way from there!-->
  <div class="container-fluid d-flex flex-column min-vh-100 justify-content-center align-items-center">
    <div class="card shadow border-0 w-75">
      <div class="card-header bg-primary text-light">
        <h2 style="text-align: center">Welcome, <b>Test</b>!</h2>
      </div>
      <div class="card-body">
```

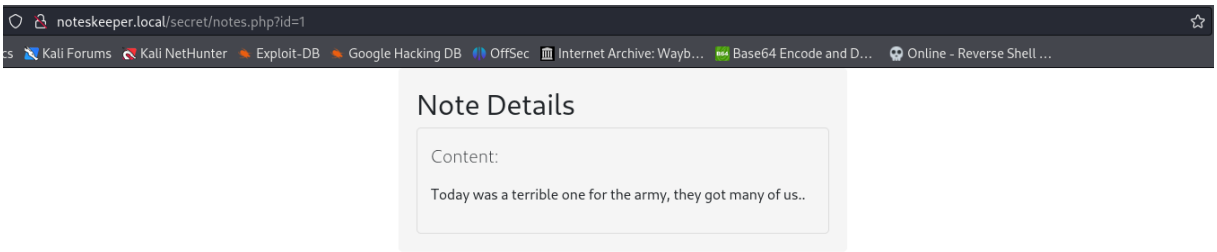
Doing what the message say we get a redirection to /notes.php with an error at our screen.

With an “Invalid Note Id” error being shown we might need to specify a correct id somehow for it to retrieve a note.

Checking the Network tab we see that the page is sending a GET request which confirms that the page is waiting for a specific parameter to be specified.



As an intuition, we find out that web application is waiting an **id** parameter, adding the parameter to the url we get a note that’s not ours!



Exploitation

Time for the exploitation phase, after discovering the correct GET parameter that the web app is requesting we figure out that the application is vulnerable to an IDOR.

Going through the different IDs we get different existing notes listed in the table below :

ID	Content
1	Today was a terrible one for the army, they got many of us..
2	How could someone imagine some unarmed guys could set such traps and kill this many.. things are getting serious!
3	Note to Self: Don't forget to call Sam at 2PM
4	URGENT: WE NEED MORE BACKUP !!!!!
5	SAM if you're seeing this, tell my family that i love them this might be my last day! these Palestinians are stronger than we thought..
6	I almost forgot, here's the key to my house Sam : **Securinets{N0T3S_L34KED}** get in there and open the book I left on the table, check page number 32!
...	
32	R3JlYXQgam9iIGNoYW1waW9uLCBpbXByZXNzaXZlIGhvdyB5b3UgZ290IGhlcmUsIHlvdSBoYXZlIHNvbWUgc2tpbGxzIG91dCB0aGVyZSEgRG9uJ3QgZm9yZ2V0IHRvIHByYXkgZm9yIG91ciBicm90aGVycyBhbmQgc2lzdGVycywgVklWQSBQQUxFU1RJTkEh

As we go through the IDs one by one, we can retrieve our flag ! but, the challenge is not over yet as the 6th note is talking about a book and a page number, could it be a note id?

Indeed, after supplying the id we get some sort of an encrypted text.
Different tools on the web can help us decode the text, one of the most known is, **Cyberchef**.

Giving it the text we got, we'll see a magic wand that we can click on to decrypt the text and it's a base64 encoded text.