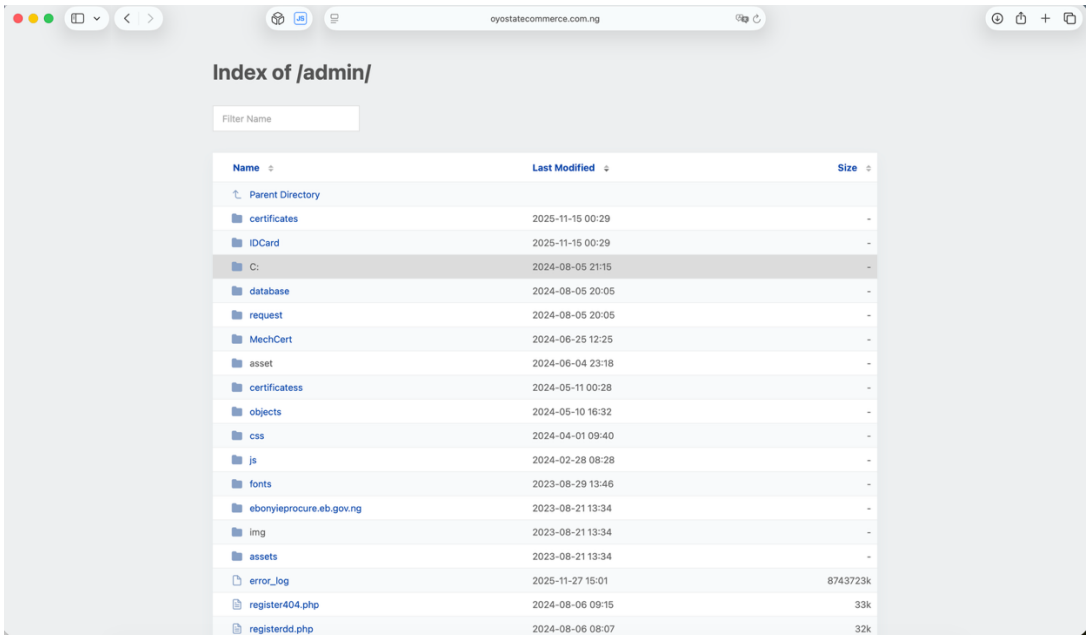
Raul Pastor

Google Hacking

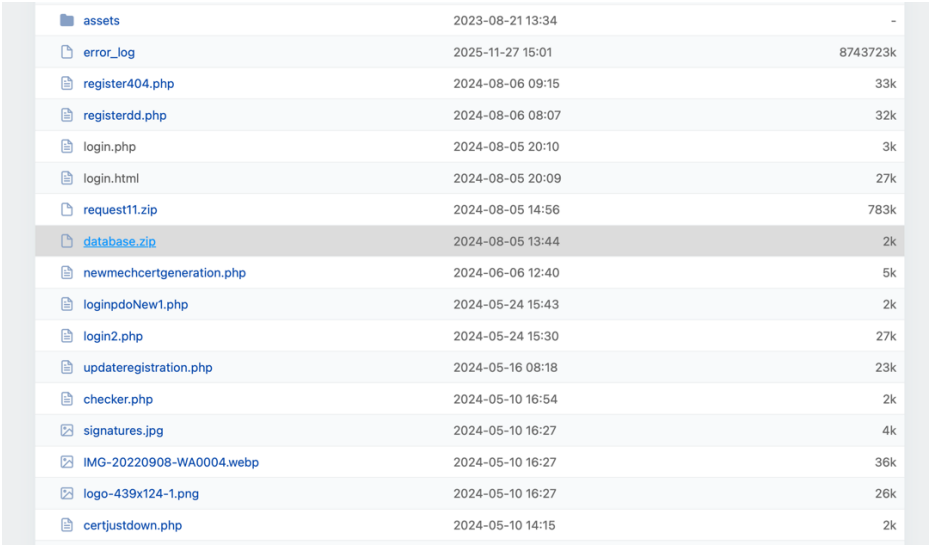
L’esercizio di oggi riguarda l’utilizzo di Google Hacking e in particolare dei comandi di filtraggio di Google per accedere a determinati domini, subdomini o persino ad alcuni tipi di file.
Il sito vulnerabile che ho trovato è ‘https://oyostatecommerce.com.ng’ che dichiara di essere un sito istituzionale del governo dello stato di Oyo State, in Nigeria.
Ho fatto un po di ricerche usando i comandi ‘site’, ‘inurl’ e ‘intext’, dopo aver trovato questo sito ho utilizzato ‘intitle:index.of inurl:admin intext:oyostatecommerce’ e ho veramente trovato tantissime informazioni sensibili tra cui: presenza di carte d’identità, certificati vuoti ma precompilati, firme digitali e soprattutto la presenza di connessioni al db con server, username e password direttamente accessibili attraverso un file .php.

Questo è la pagine Index of /admin/:



Raul Pastor

Tra le varie informazioni sensibili ho trovato questo database.zip:

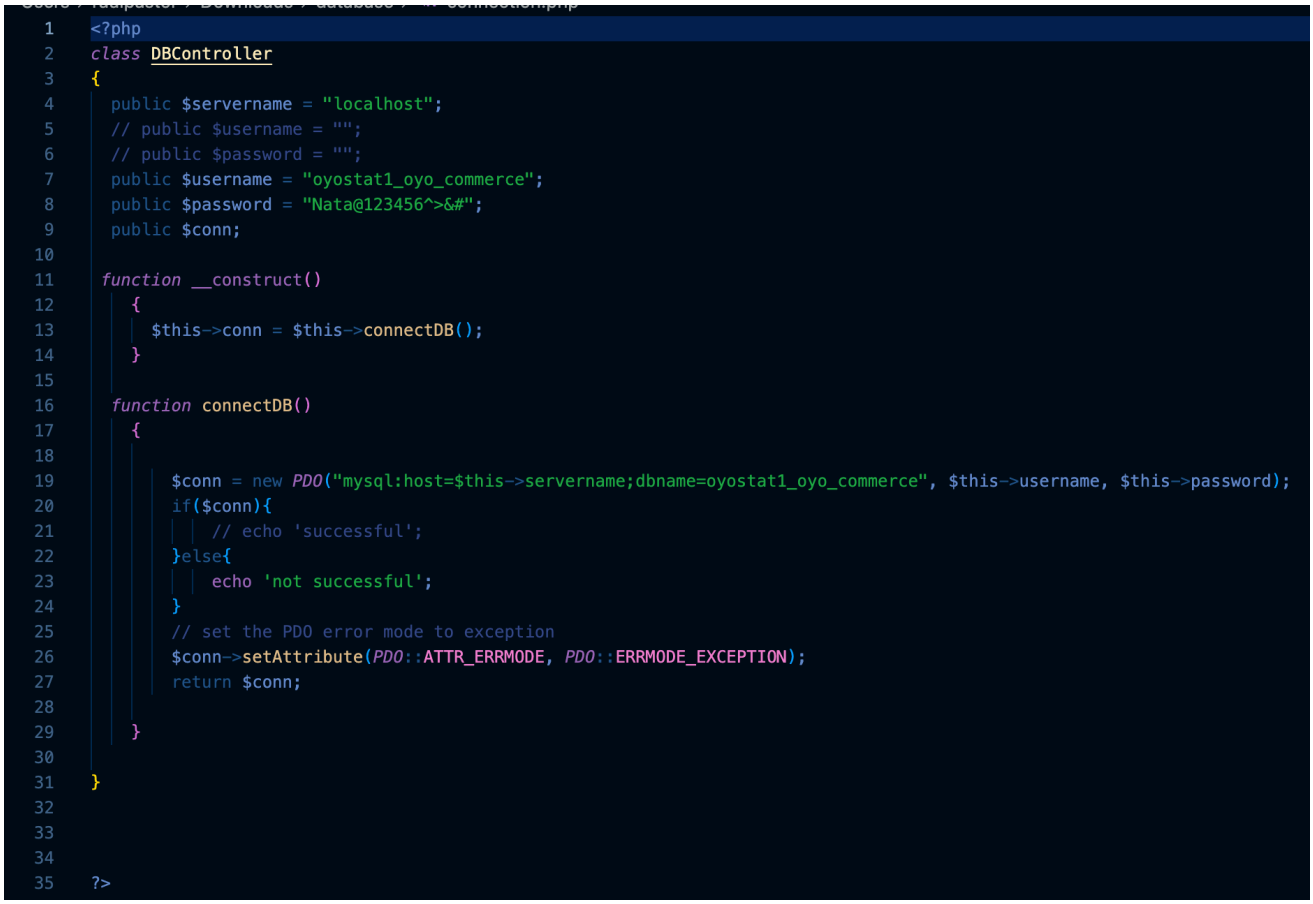


assets	2023-08-21 13:34	-
error_log	2025-11-27 15:01	8743723k
register404.php	2024-08-06 09:15	33k
registerdd.php	2024-08-06 08:07	32k
login.php	2024-08-05 20:10	3k
login.html	2024-08-05 20:09	27k
request11.zip	2024-08-05 14:56	783k
database.zip	2024-08-05 13:44	2k
newmechcertgeneration.php	2024-06-06 12:40	5k
loginpdoNew1.php	2024-05-24 15:43	2k
login2.php	2024-05-24 15:30	27k
updateregistration.php	2024-05-16 08:18	23k
checker.php	2024-05-10 16:54	2k
signatures.jpg	2024-05-10 16:27	4k
IMG-20220908-WA0004.webp	2024-05-10 16:27	36k
logo-439x124-1.png	2024-05-10 16:27	26k
certjustdown.php	2024-05-10 14:15	2k

Una volta scaricato lo zip erano presenti due file .php di connessioni al db:

Raul Pastor

Ecco il contenuto di uno dei due file .php:



```php
<?php
class DBController
{
  public $servername = "localhost";
  // public $username = "";
  // public $password = "";
  public $username = "oyostat1_oyo_commerce";
  public $password = "Nata@123456^>&#";
  public $conn;

 function __construct()
   {
     $this->conn = $this->connectDB();
   }

  function connectDB()
   {

       $conn = new PDO("mysql:host=$this->servername;dbname=oyostat1_oyo_commerce", $this->username, $this->password);
       if($conn){
           // echo 'successful';
       }else{
           echo 'not successful';
       }
       // set the PDO error mode to exception
       $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
       return $conn;

   }

}



?>
```

Sicuramente una lista di cose da applicare per evitare questa debolezza potrebbe essere:

- Disattivare il directory listing (da Apache: Options -Indexes).
- Non inserire mai file sensibili in directory pubbliche.
- Controllare i permessi su filesystem.
- Inserire un’autenticazione a livello di server.