

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 pagina Index of /admin/:

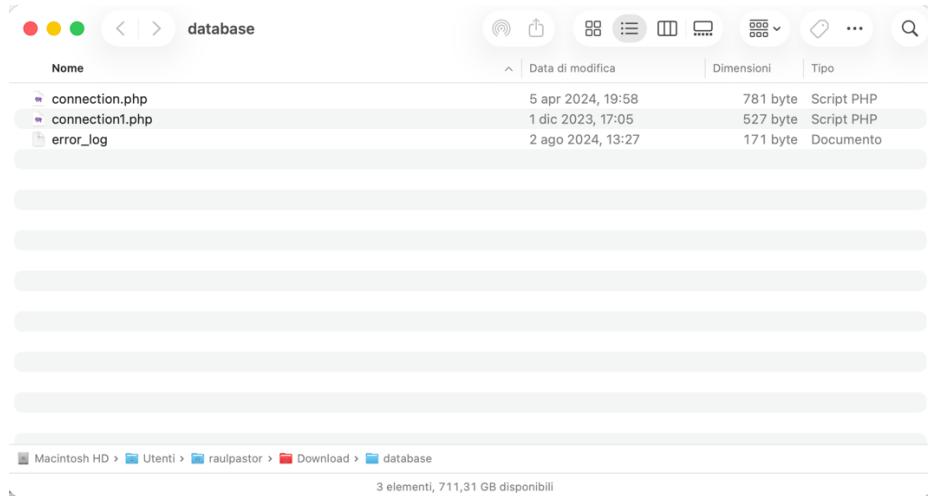
Name	Last Modified	Size
Parent Directory		
certificates	2025-11-15 00:29	-
IDCard	2025-11-15 00:29	-
C:	2024-08-05 21:15	-
database	2024-08-05 20:05	-
request	2024-08-05 20:05	-
MechCert	2024-08-25 12:25	-
asset	2024-06-04 23:18	-
certificatess	2024-05-11 00:28	-
objects	2024-05-10 16:32	-
css	2024-04-01 09:40	-
js	2024-02-28 08:28	-
fonts	2023-08-29 13:46	-
ebonyieprocure.eb.gov.ng	2023-08-21 13:34	-
img	2023-08-21 13:34	-
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

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:



Nome	Data di modifica	Dimensioni	Tipo
connection.php	5 apr 2024, 19:58	781 byte	Script PHP
connection1.php	1 dic 2023, 17:05	527 byte	Script PHP
error_log	2 ago 2024, 13:27	171 byte	Documento

Ecco il contenuto di uno dei due file .php:

```
Server: Localhost / Database: database => connection.php
1 <?php
2 class DBController
3 {
4     public $servername = "localhost";
5     // public $username = "";
6     // public $password = "";
7     public $username = "oyostat1_oyo_commerce";
8     public $password = "Nata@123456";
9     public $conn;
10
11    function __construct()
12    {
13        $this->conn = $this->connectDB();
14    }
15
16    function connectDB()
17    {
18
19        $conn = new PDO("mysql:host=$this->servername;dbname=oyostat1_oyo_commerce", $this->username, $this->password);
20        if($conn){
21            // echo 'successful';
22        }else{
23            // echo 'not successful';
24        }
25        // set the PDO error mode to exception
26        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
27        return $conn;
28    }
29
30
31 }
32
33
34
35 ?>
```

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.