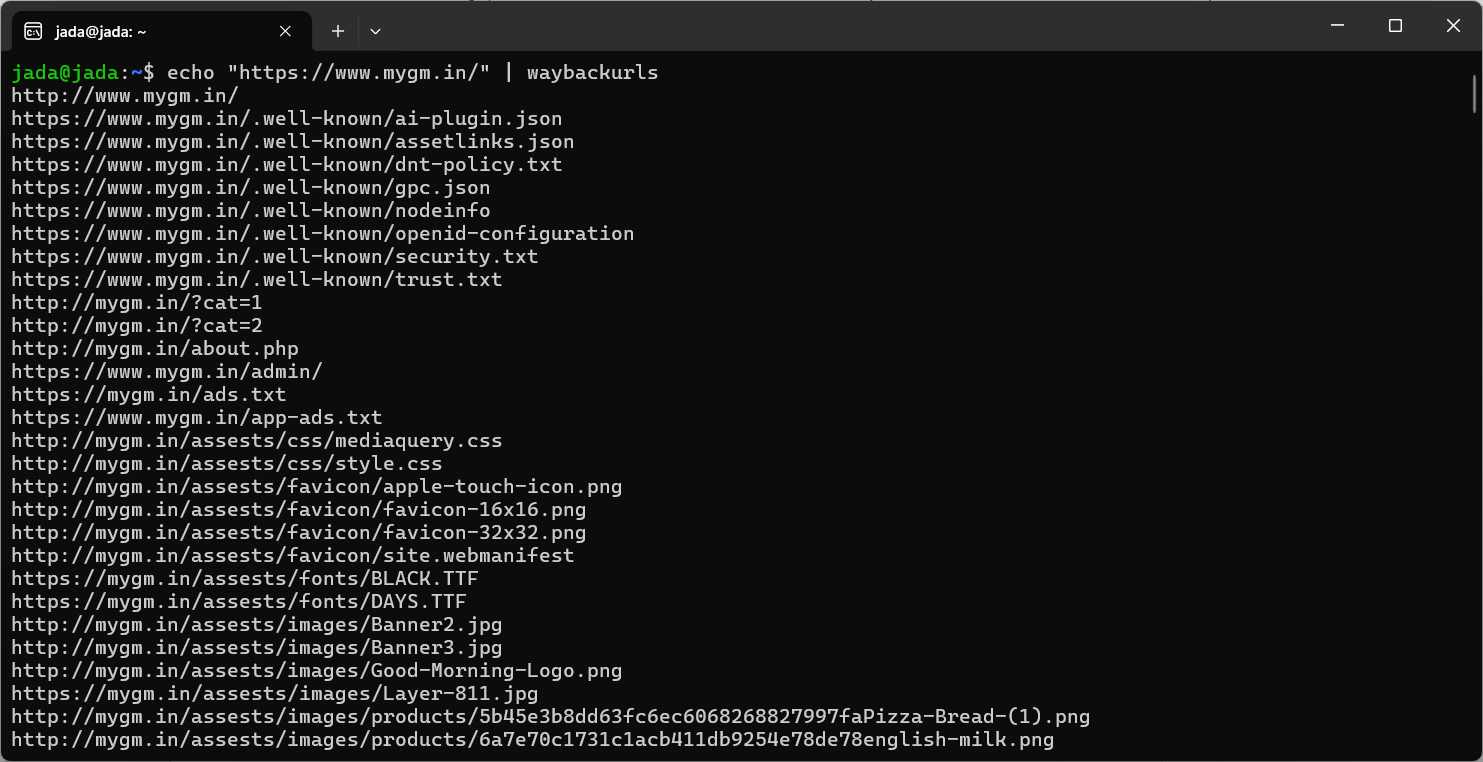
**21BCE0555  
JADA AVINASH**

**Step 1: URL Enumeration (Waybackurls)**

The first screenshot shows the use of the waybackurls command to gather a list of URLs for the target domain (mygm.in). This tool retrieves URLs archived by services like the Wayback Machine. The output shows a variety of URLs, including:

* Paths to administrative pages (e.g., /admin/)
* Asset files (e.g., .css, .js, images)
* Potential API endpoints or PHP scripts (products.php)

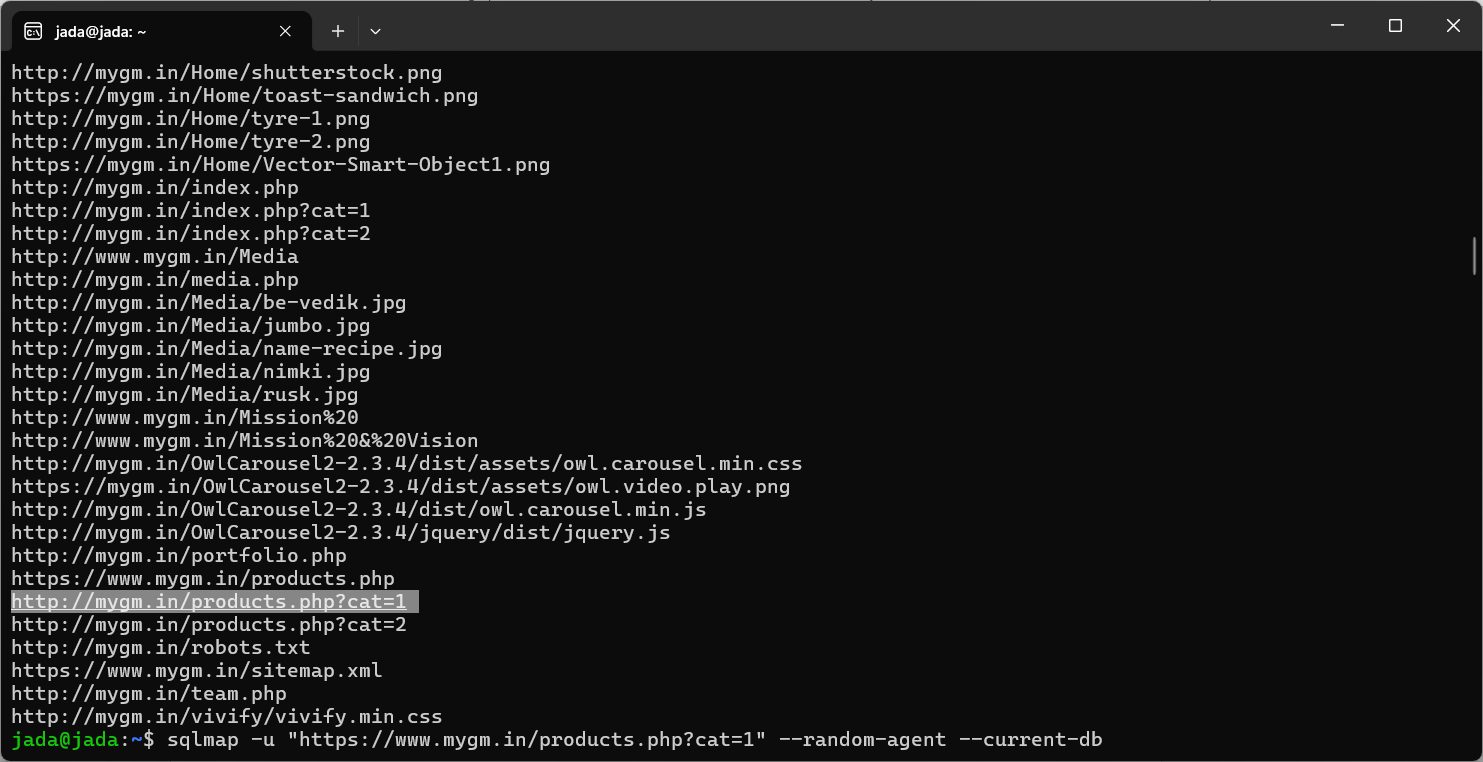


**Step 2: Identifying Vulnerable Parameters**

The next image highlights a **specific URL**, https://mygm.in/products.php?cat=1, which is a dynamic parameter (cat=1). This parameter is selected as a candidate for SQL injection testing using SQLmap. The command:

sqlmap -u "https://mygm.in/products.php?cat=1" --random-agent --current-db

* **-u** specifies the target URL.
* **--random-agent** uses a random user-agent to evade detection.
* **--current-db** aims to identify the current database in use.

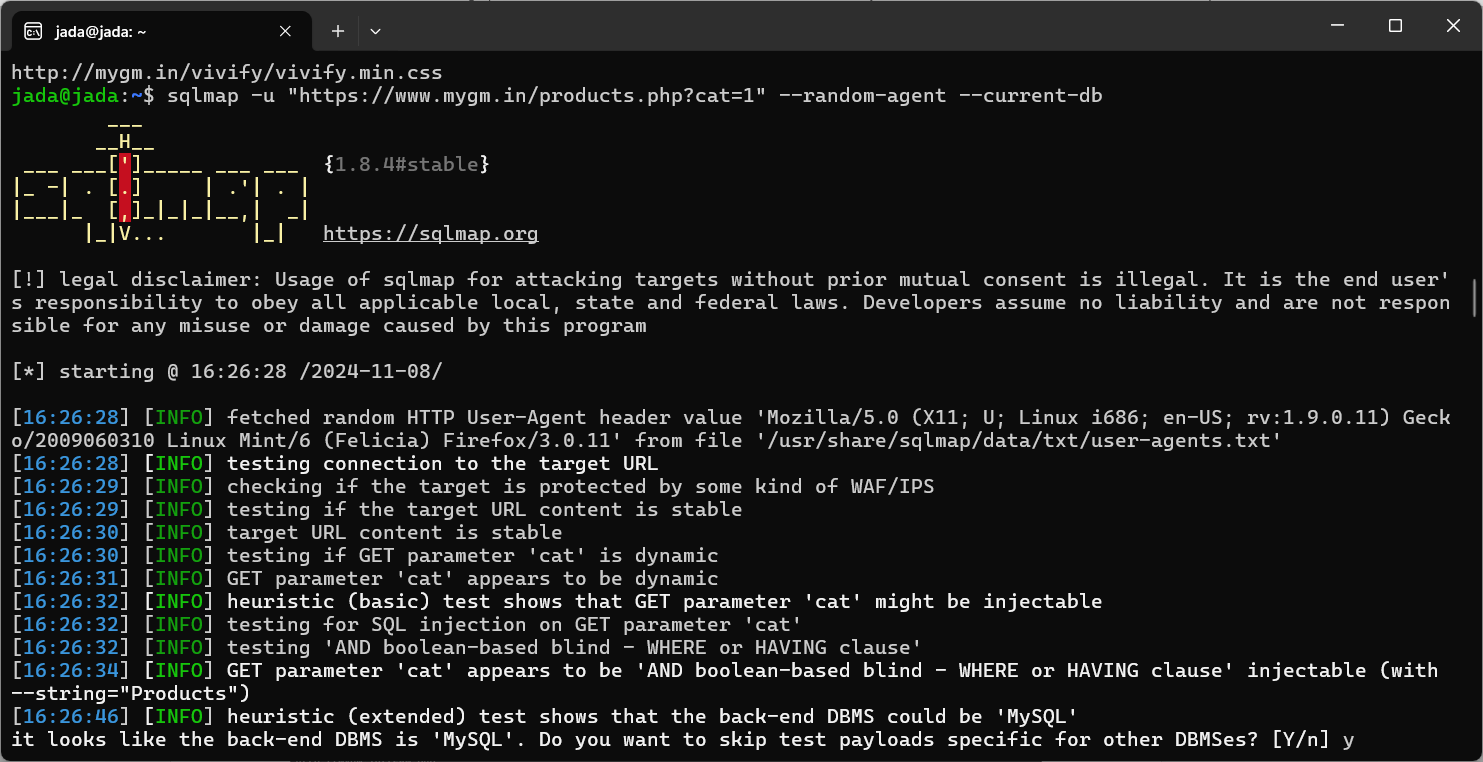


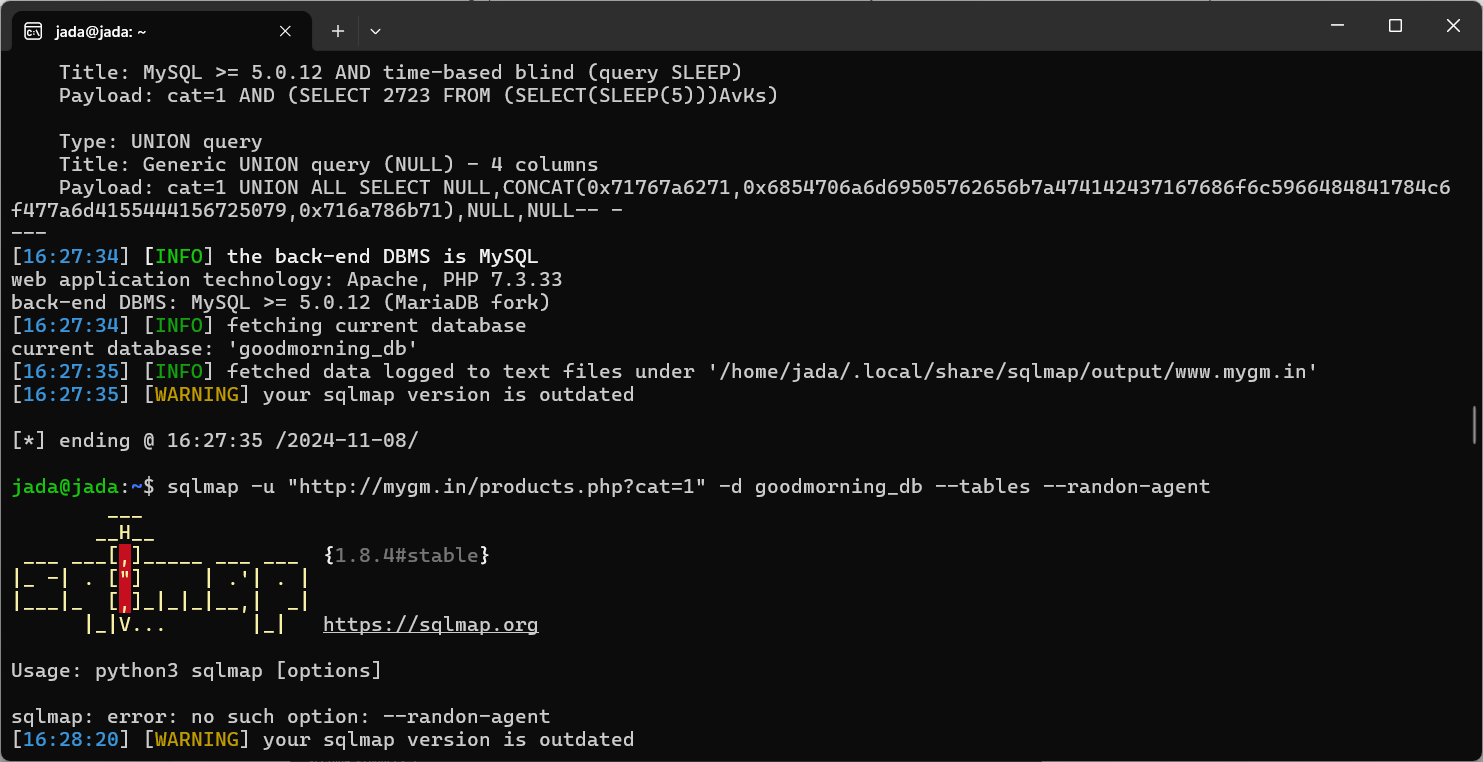
**Step 3: SQL Injection Testing**

SQLmap begins testing the parameter cat for SQL injection vulnerabilities. The output indicates:

* The parameter is **dynamic and injectable**.
* The backend database is **MySQL**.
* A specific type of injection (boolean-based blind) is possible.

After confirming the vulnerability, SQLmap proceeds to exploit it to retrieve database information.





**Step 4: Fetching Database Information**

SQLmap successfully retrieves the **current database name**, which is goodmorning\_db.

**Step 5: Enumerating Tables in the Database**

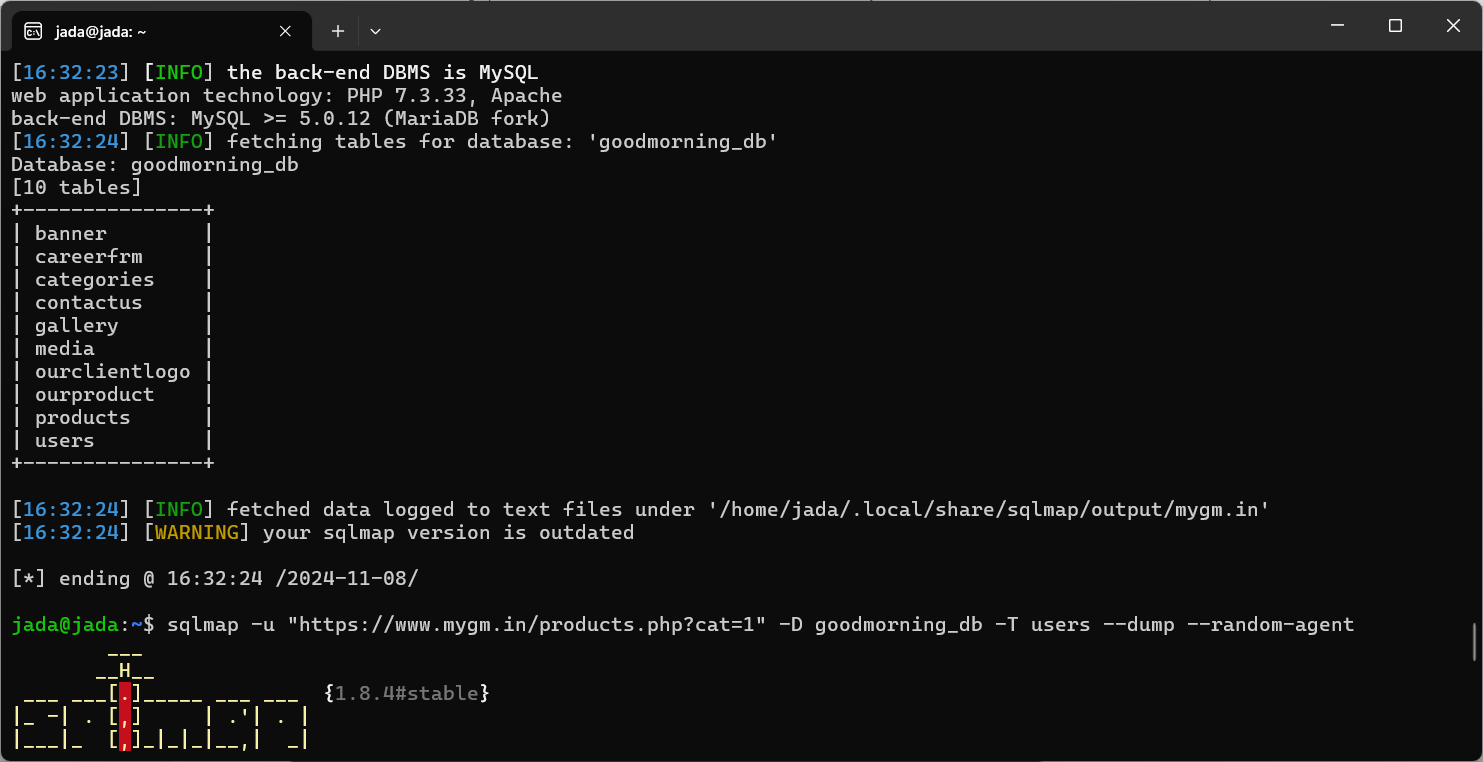
The next command:

sqlmap -u "https://mygm.in/products.php?cat=1" -D goodmorning\_db --tables --random-agent

* Specifies the database (goodmorning\_db) using -D.
* Lists all tables in the database (--tables).

The output reveals several tables, including:

* users (potentially containing sensitive information)
* Other tables like banner, categories, media.



**Step 6: Extracting Data from the users Table**

The final command:

sqlmap -u "https://mygm.in/products.php?cat=1" -D goodmorning\_db -T users --dump --random-agent

* Specifies the users table with -T users.
* Dumps its contents (--dump).

SQLmap extracts:

* A single user entry (admin).
* The password hash for the admin user.

The dumped data is saved to a CSV file, which can be further analyzed.

