Google Dorking Report: Exposed File

1. Introduction

The purpose of this exercise was to demonstrate how Google Dorking can be used to identify exposed sensitive data on a website. During this learning process, we focused on searching for publicly available `.env` files, which often contain sensitive information such as database credentials, API keys, and other configuration details.

Dorking Tool Used: Google Search
Search Operator Used: `filetype:env "DB_PASSWORD"`
Target Website: `https://docharakat.com/`
Date of Search: [Insert Date]

2. Methodology

- **Google Dorking**: I used the `filetype:env "DB_PASSWORD"` operator in Google Search to find `.env` files that might expose sensitive data like database credentials. The search results led to the discovery of an exposed `.env` file on the target website.
- **Dorking Details**: The `filetype:env` operator targets `.env` files, which are commonly used in web development, especially in PHP and Laravel applications, to store environment variables securely.

3. Findings

Exposed .env File Found

- **File URL**: https://docharakat.com/.env
- **Exposed Information**:
  The `.env` file contains several critical pieces of information, including:
  - **Application Credentials**: The application's environment and key are publicly visible (APP_KEY, APP_ENV, etc.).
  - **Database Credentials**:
    - **DB_HOST**: 127.0.0.1
    - **DB_USERNAME**: `u917243327_root`
    - **DB_PASSWORD**: `wtL7[0O&tO1?`
  - **Mail Server Information**:
    - **MAIL_HOST**: `smtp.hostinger.com`
    - **MAIL_USERNAME**: `contact@docharakat.com`
    - **MAIL_PASSWORD**: `wtL7[0O&tO1?`

- **Other Information**: There are keys for Redis, session, queue configuration, and API credentials (e.g., AWS, Pusher, etc.).

Risks Associated with Exposed Data:

1. **Database Credentials**:
   The `.env` file includes the MySQL database username, password, and database name. This is **highly sensitive information** that can be exploited by attackers to gain unauthorized access to the website's database.
2. **Email Credentials**:
   The file reveals the SMTP username and password for the `contact@docharakat.com` email account. If this information is misused, it could lead to spam, phishing, or unauthorized access to the email account.
3. **Unencrypted Secrets**:
   The AWS and Pusher keys are empty, but if they were populated, they could provide direct access to cloud services, enabling attackers to take control of critical infrastructure.

4. Impact Analysis

- **High Risk**: The exposure of database credentials is particularly severe. With access to the database username and password, an attacker could execute SQL queries, delete or modify sensitive data, or compromise the integrity of the site's backend.
- **Moderate Risk**: Exposed email credentials could lead to abuse, such as spam, phishing attacks, or unauthorized access to email communication from the `contact@docharakat.com` email address.
- **Low Risk**: The unpopulated AWS and Pusher keys don't present an immediate risk, but if filled out, they could grant access to cloud resources, which could potentially lead to unauthorized data access or service manipulation.

5. Conclusion

This exercise highlights the importance of securing environment files like `.env` and ensuring that sensitive credentials are not exposed to the public. With the credentials exposed on `https://docharakat.com/`, there is a significant security risk, including unauthorized access to the website's database and email systems.

It is highly recommended that the website administrators review their security practices, update sensitive credentials, and prevent further exposure of critical data.
