STRING (Satuan Tulisan Riset dan Inovasi Teknologi) p-ISSN: 2527 - 9661
Vol. 6 No. 2 Desember 2021 e-ISSN: 2549 - 2837

TEKNIK UJI PENETRASI WEB SERVER MENGGUNAKAN SQL INJECTION DENGAN SQLMAP DI KALILINUX

Rudi Hermawan
Program Studi Teknik Informatika, Universitas Indraprasta PGRI Jakarta Indonesia
wowor99@gmail.com

Submitted December 2, 2021; Revised December 4, 2021; Accepted December 4, 2021

Abstrak

Dalam beberapa tahun ini kasus serangan siber yang mengarah pada keamanan *website* semakin meningkat. Ancaman peretasan *website* yang paling banyak digunakan adalah *sql injection*. Dengan memakai *tools SQLmap* yang berjalan di sistem operasi *kalilinux* penyerang dengan mudah mengambil alih data otentifikasi yang sangat penting *user* dengan *passwordnya*. Penyerang hanya menggunakan *script query* sql khusus dengan menggunakan bahasa pemograman *python* akan memaksa *web server* untuk mengeluarkan informasi *database*, tabel, kolom dan isi data. Teknik *sql injection* ini tidaklah sulit dengan mengetahui cara kerja *SQL injection* ini diharapkan menjadi hal yang berguna bagi para admin *web* maupun pengembang aplikasi *web* agar mampu mengamankan akses *user* dari para penyerang. Simulasi penyerangan ini menggunakan *virtual* mesin, dengan membuat dua komputer *virtual* yang diskenariokan sebagai penyerang dan *server* target. Dengan pengujian melalui simulasi ini bisa mengetahui bagaimana proses serangan dan akibat dari serangan yang dilakukan oleh penyerang.
Kata kunci : peretasan, *SQL injection, SQLmap, kalilinux , web, virtual* mesin

Abstract

In recent years, the cyberattacks that lead to website security have increased. The most widely used website hacking threat is sql injection. By using the sqlmap tool that runs on the Kalilinux operating system, attackers can easily take over very important users' authentication data with their passwords. Attackers only use a special SQL query script using the python programming language that will force the web server to release database information, tables, columns and data contents. This sql injection technique is not difficult. Knowing how sql injection works is expected to be useful for web admins and web application developers to be able to secure user access from attackers. The simulation of the attack uses a virtual machine, by creating two virtual computers that are scripted as the attacker and the target server. By testing through this simulation, we can find out the process of the attack and the consequences of attacks by attackers.
Keywords*: hacking, SQL injection, SQLmap, kalilinux , web, virtual machine*

1. PENDAHULUAN

Latar Belakang

Kemajuan teknologi informasi dan digital yang sangat pesat sampai saat ini. Menjadikan informasi sangat mudah diakses melalui berbagai media. Internet *web* dan *mobile* saat ini menjadi primadona dalam menyalurkan informasi ke masyarakat global. Internet menjadikan informasi dapat diakses oleh siapapun tanpa mengenal batas wilayah, batas waktu, dan batas usia. Hal ini memberikan kesempatan bagi pengguna diseluruh dunia bisa mengakses data publik sebagai sumber informasi. Selain data publik yang bisa diakses oleh siapapun. Data juga ada yang mempunyai sifat rahasia untuk kalangan sendiri, Data ini bersifat penting dan tidak peruntukan bebas hanya di peruntukan secara khusus oleh pemilik data.

Dalam perkembangannya keamanan data dan informasi menjadi yang sangat vital

dan penting untuk menjaga integritas dan validitas sebuah data. Sistem, jaringan komunikasi dan data harus dilndungi dari segala serangan dan usaha-usaha penyusupan atau pemindaian oleh orang-orang yang tidak berwenang. Semakin oleh karena itu kebutuhan kualitas keamanan jaringan, *website*, *server* dan *database* harus selalu di pantau dan ditingkatkan secara berkala. Semakin terbukanya pengetahuan tentang *hacking* dan dengan semakin banyaknya *tools* penetrasi yang tersedia di internet. semakin memudahkan para penyusup dan penyerang untuk melakukan aksi penyusupan maupun penyerangan.

Web server menjadi salah satu yang paling sering dan rentan menjadi sasaran dan target para penyerang. Serangan pada *web server* menjadi langkah awal untuk melanjutkan serangan ke *database*. *Web server* merupakan *backbone* dari *world wide web*, yang mempunyai fungsi melayani koneksi *transfer* data melalui *protokol Hyper Text Transfer Protocol (HTTP*) yang dikirim oleh pengguna melalui *web browser* untuk ditampilkan pada halaman *website*. Umumnya dokumen *web* dibuat menggunakan bahasa *php/html* [1].

Dalam beberapa tahun terakhir ini kasus serangan siber yang mengarah pada keamanan *website* semakin meningkat. Ancaman terbesar peretasan website yang menduduki peringkat pertama sampai saat ini adalah *SQL injection.* Sudah lebih dari 20 tahun sejak serangan *SQL injection* diungkapkan ke publik sampai saat ini masih sering sekali ditemui menjadi serangan paling banyak. lalu apa itu *SQL injection* dan bagaimana cara mengatasinya?

Gambar 1. Penjelasan Pengertian *SQL injection*

Aplikasi berbasis *web* terhubung dengan database menggunakan perintah *query sql*. Serangan *sql injection* terjadi krn adanya celah keamanan pada aplikasi web. Celah keamanan ini di manfaatkan oleh penyerang dengan menggunakan *script query* SQL khusus untuk mengelabui *web server* meminta *database* untuk mendapatkan tujuannya. Setelah mendapatkan otentifikasi penyerang bisa masuk kedatabase dan dapat dengan mudah mengambil dan memanipulasi *database* korban.

SQL Injection

SQL injection merupakan salah satu teknik yang digemari oleh para penyerang, karena sampai saat ini masih banyak *website* yang kurang memperhatikan celah keamanan sistemnya yang bisa dimanfaatkan oleh pengguna yang tidak bertanggung jawab. *SQL injection* bisa terjadi karena penyerang menguasai teknik *query SQL* yang sanggup melewati celah keamanan yang ada di *SQL* pada lapisan basis data suatu aplikasi. Celah ini terjadi karena *form input* dari pengguna tidak *difilter* dengan baik terhadap metakarakter dalam pembuatannya menggunakan form input. Jadi sampai saat ini *sql injection* masih menjadi favorit penyerang dalam melakukan serangan pada *website*. Apalagi sekarang ini *hacking* melalui jaringan internet sudah tidak sesulit seperti zaman dulu. Sekali lagi umumnya serangan *SQL injection* terjadi karena murni kelalaian *programmer* (pengembang aplikasi) tidak mengimplementasikan pembatas filter terhadap metakarakter (&, ;, `, ', \, “, |, *, ?, ~, <, >, ^, (,), [,], {, }, $, \n, \r) yang digunakan dalam sintaks sql pada form input aplikasi, sehingga penyerang dapat menginput metakarakter tersebut dengan kombinasi *script query* untuk bisa melakukan aksi liar dalam menembus otentifikasi data. Jika sebuah aplikasi *web* tidak menerapkan *filter* terhadap form input maka penyerang dapat melancarkan

serangan dengan cara memasukkan username dengan menambahkan '#', misalnya 'rudz#'. Hal ini menyebabkan karakter selanjutnya tidak dianggap sebagai kode SQL, alhasil *username* "rudz" tidak perlu memasukkan *password* agar bisa masuk kedalam sistem. Masih banyak teknik SQL injection yang dapat dicari melalui mesin pencari [2].

Sqlmap

Ada banyak cara untuk melakukan penyerangan suatu website, salah satunya dengan menggunakan teknik SQL injection. *Tools* apa yang banyak digunakan untuk melakukan SQL injection salah satunya adalah ***sqlmap***". Jadi SQLmap merupakan *tools* untuk penetrasi *open source* yang mampu mengotomatisasi proses deteksi dan eksploitasi kelemahan injeksi *SQL* dan juga mampu mengambil alih basis data *web server* [3]. Dengan kata lain *SQLmap* ini merupakan *tools* yang dapat mendeteksi dan melakukan *exploit* pada bug *SQL injection* secara otomatis. dengan melakukan serangan *SQL injection* seorang penyerang dapat mengambil alih serta memanipulasi sebuah database di dalam sebuah *web server*.

Fitur *SQLmap* mendukung penuh untuk sistem database seperti MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, Informix, MariaDB, MemSQL, TiDB, CockroachDB, HSQLDB, H2, MonetDB, Apache Derby, Amazon Redshift, Vertica, Mckoi, Presto, Altibase, MimerSQL, CrateDB, Greenplum, Gerimis, Apache Ignite, Cubrid, Cache InterSystems, IRIS, eXtremeDB, FrontBase, Raima Database Manager, YugabyteDB dan sistem manajemen basis data Virtuoso. *SQLmap* juga mendukung untuk mencari nama *database* tertentu, tabel tertentu di semua *database* atau kolom tertentu di semua tabel *database* . Hal Ini menjadi sangat berguna untuk mengidentifikasi *database*, tabel, yang relevan berisi *string* seperti id, *username* dan *pass*. Agar bisa menggunakan dan menjalankan *sqlmap* harus menginstall *python* karena untuk mengeksekusi *SQLmap* menggunakan bahasa pemrograman *python* [4].

Kalilinux

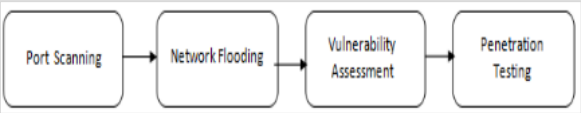
Kalilinux merupakan sistem operasi berbasis *linux debian* yang di kembangkan oleh *Offensive Security. Kalilinux* memiliki tampilan sederhana dan tidak terlalu mencolok dan penggunaanya pun tergolong cukup mudah, sehingga kalilinux sangat baik untuk para pemula yang sedang belajar dalam melakukan penetrasi pada sistem, jaringan dan aplikasi. Selain terdapat di PC, *Kalilinux* juga memiliki versi yang terdapat di *Android* yang disebut *KaliNethunter* yang memiliki fungsi yang sama. *Kalilinux* merupakan reinkarnasi dari sistem operasi legenda *BackTrack*, salah satu distri *Linux* yang diciptakan secara khusus supaya bisa memenuhi keperluan dalam *penetration* juga testing di sebuah system serta keamanan pada komputer. Dengan pengembangan *Kalilinux* diharapkan akan lebih stabil serta *powerful* dari generasi sebelumnya "*BackTrack*". [5]

2. METODE PENELITIAN

Penelitian ini menggunakan *metode Research and Development* (R&D). Metode R&D merupakan suatu proses atau langkah-langkah untuk mengembangkan dan menyempurnakan produk yang telah ada, yang dapat dipertanggung jawabkan [6]. Obyek penelitian ini menggunakan situs *web* yang dirancang khusus untuk keperluan simulasi pengujian keamanan sistem informasi berbasis *web* yang beralamatkan. Tahap-tahap yang dilakukan untuk menemukan celah keamanan pada aplikasi berbasis *web meliputi scope,*

reconnaissance, vulnerability detection, information analysis & planning dan penetration testing. Pada proses *vulnerability detection* di dalamnya terdapat metode DAST (*Dynamic Application Security Testing)* dalam menemukan celah keamanan pada *website* dengan bantuan aplikasi, misalnya *Acunetic* [7].

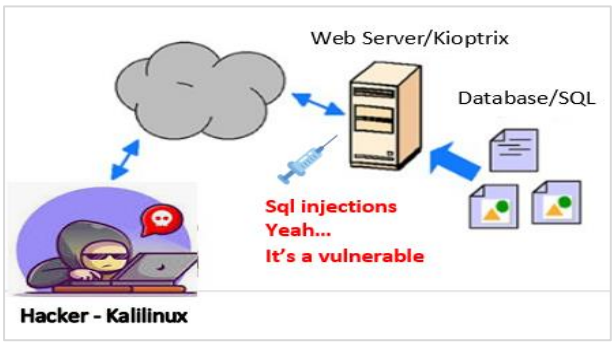
Pada metode ini peneliti juga melakukan *Penetration Testing* atau Uji Penetrasi suatu simulasi serangan terkontrol yang membantu mengidentifikasi kerawanan terhadap aplikasi. Uji penetrasi ini untuk menganalisis seluruh kerentanan pada sistem yang potensial, termasuk sistem konfigurasi yang lemah dan tidak sesuai, celah pada perangkat lunak atau perangkat keras, serta kelemahan operasi di bagian proses atau penanganan teknisnya. Dengan memindai informasi yang dibutuhkan dalam mencari kerentanan pada sistem [8]. Alur metodologi untuk lebih jelasnya ditunjukkan pada Gambar1 sbb :



Gambar 2. Alur Tes Metodologi

3. HASIL DAN PEMBAHASAN

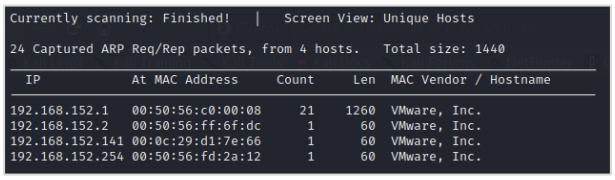
Dalam melakukan uji penetrasi yang perlu dilakukan yaitu menggunakan aplikasi *Vmware* 15 yang berfungsi sebagai aplikasi pembuat mesin *virtual* komputer. Selanjutnya membuat skenario simulasi penyerangan dengan menggunakan dua komputer *virtual* di *VMWare*. Komputer pertama diskenariokan menjadi komputer penyerang dengan menggunakan sistem operasi *Kalilinux* , selanjutnya komputer kedua difungsikan sebagai *web server* yang akan menjadi target penyerangan ini dengan sistem operasi *Kioptrix Server* . Teknik serangan menggunakan teknik Injeksi SQL dengan menggunakan tools *sqlmap*.



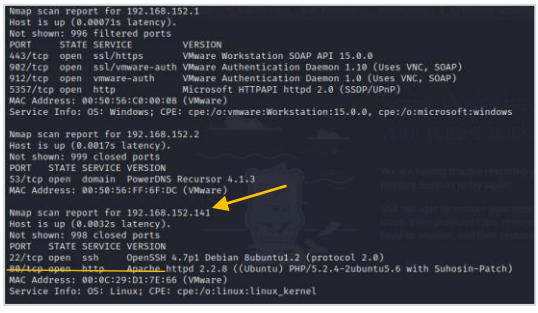
Gambar 3. Skenario Dua Komputer Virtual Penyerang dan Target Korban

Fase Pemindaian Target Korban/ *information gathering*

Pada fase ini penyerang melakukan aktifitas pengintaian untuk menentukan target yang akan dijadikan sasaran serangan. Dalam demo fase simulasi ini dilakukan secara lokal area. Pengintaian dilakukan secara 2 tahap yakni tahap pemetaan jaringan menggunakan perintah *netdiscover* dan pemindaian port akses menggunakan perintah nmap.

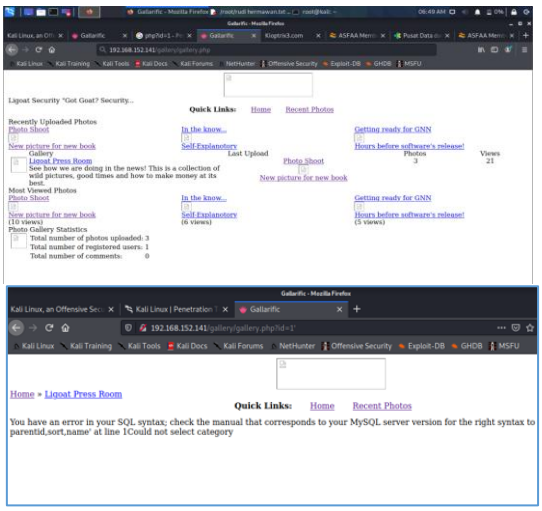


Gambar 4. Pemetaan Host pada Jaringan Lokal



Gambar 5. Pengumpulan Informasi Port dari Target

Fase Uji Celah keamanan / *Vulnerablitiy*
Didalam fase ini penyerang melakukan aksi pemindaian menggunakan browser untuk mencek *web server* dari target. Alamat target menggunakan alamat ip address 192.168.152.141 sesuai alamat ip pada gambar 5. Dalam pemindaian celah keamanan di ekspose alamat yang mengandung data dari *server*, alamat tersebut sebagai titik injeksi *SQL*. Dalam proses ini berhasil dipindai alamat *gallery.php* berisi data yang terkoneksi database dengan *server*.

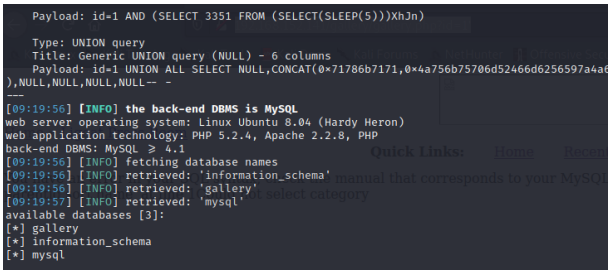


Gambar 6. Pengujian Celah Keamanan Melalui *url* Menggunakan *web browser*

Selanjutnya diuji dengan menggunakan *script query* menggunakan kombinasi meta karakter, jika hasilnya menjadi *error* mengindikasikan ada celah keamanan yang bisa diinjeksi, tapi jika hasilnya tidak ada perubahan sama dengan tampilan sebelumnya berarti *form input* sudah di terproteksi dengan baik dengan berhasil memfilter *meta* karakter.

Fase Penyerangan Dengan *Sqlmap*
Dalam fase ini menjalankan file *eksekusi python sqlmap* dengan menentukan alamat *host target* dengan memaksa untuk menampilkan *database*, tabel, kolom dari *database*. Dengan menggunakan kombinasi *SQLmap* dan *query SQL* penyerang mencoba mendapatkan informasi data *user* dengan *password* otentifikasinya. Dari banyaknya *database*, tabel dan kolom yang ditemukan penyerang harus mempunyai keahlian dan kemampuan untuk mengendus posisi dimana data *user* dan *password* tersimpan. Penyerang menggunakan perintah *SQLmap* dengan *querynya* memaksa *server* target untuk menampilkan *database* yang dimiliki oleh *web server*. Hasilnya terlihat ada tiga *database* pada server target seperti terlihat pada gambar 7 dan 8 dibawah ini.

Gambar 7. Menjalankan Tools SQLmap



Gambar 8. Menampilkan Database dari Hasil Injeksi SQL

Dengan perintah *sqlmap* menggunakan spesial querynya akan memaksa *database* untuk menampilkan nama tabel-tabel yang dimiliki oleh *database gallery*. Hasilnya terlihat ada tujuh tabel pada *database gallery* seperti terlihat pada gambar 9 dibawah ini

Gambar 9. SQLmap menampilkan isi tabel

Selanjutnya perintah *SQLmap* dengan kombinasi *querynya* mencoba untuk membuka kolom-kolom yang ada pada tabel *dev_accounts*. Hasilnya terlihat ada tiga kolom pada tabel *dev_account* seperti terlihat pada gambar 10 dibawah ini.

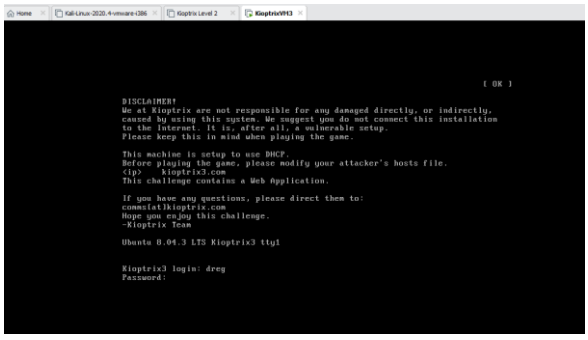
Gambar 10. SQLmap menampilkan isi kolom

Selanjutnya *sqlmap* dengan *querynya* mencoba untuk membuka isi data *user* beserta *pasword* otentifikasinya. Jika sudah dapat menampilkan isi data *user server* target sudah siap diambil alih oleh penyerang. Terlihat pada gambar 11 dibawah ini

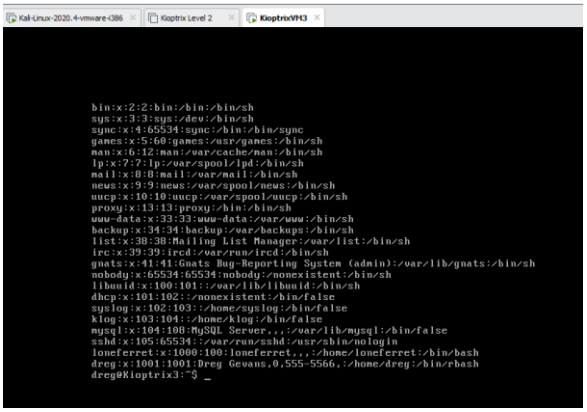
Gambar 11. *sqlmap* menampilkan isi data *user*

Fase Ambil Alih Target

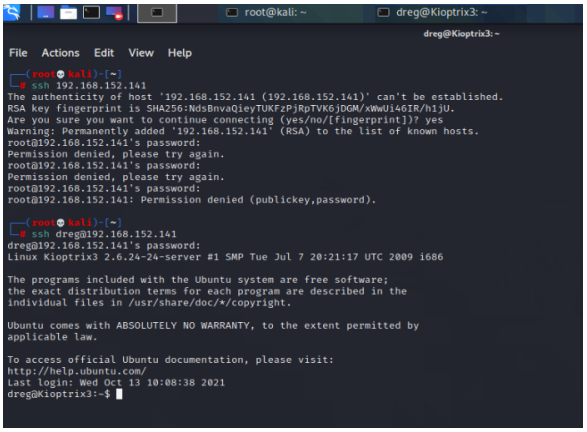
Setelah berhasil mendapatkan *username* dan *password*. Saatnya coba *login* pada *servertarget*.



Gambar 12 Login di *server* Target



Gambar 13. Menjalankan perintah di *server* target



Gambar 14. *Login server* Target dengan ssh

Pengambil alihan *server* target berhasil dilakukan oleh penyerang. Dari hasil simulasi ini menunjukkan penyerang akan dengan mudah menguasai *server* setelah berhasi mendapatkan informasi data user melalui cara *sql injection*.

4. SIMPULAN

Simpulan

Dalam jurnal ini menjelaskan bagaimana cara mengeksploitasi kerentanan dalam aplikasi *web server* menggunakan *SQLinjection*. Untuk menguasai teknik ini tidaklah sulit. Oleh karena itu sangatlah penting bagi *administrator* maupun pengembang aplikasi untuk lebih memperhatikan dan menguatkan keamanannya.

Akses terhadap *database* jika sudah diambil alih oleh penyerang, maka penyerang mempunya kendali dan kontrol penuh terhadap

sistem. Hal tersebut jika terjadi menjadi bencana besar karena data menjadi sangat rentan di manipulasi. Dengan jatuhnya server ke tangan penyerang akibatnya akan sangat fatal bagi pemilik *server*. Penyerang dapat dengan mudahnya mencuri data, modifikasi dan manipulasi data, merusak data.

Saran

Dunia *hacking* dan siber sekuriti semakin terus berkembang, *Administrator* dan pengembang aplikasi sebaiknya selalu mengikuti perkembangan dunia *security*. Hal ini menjadi bekal pengetahuan dalam mengantisipasi dan mengontrol data dan aplikasi dari pengguna yang tidak berhak. Jika memungkinkan batasi panjang *input box*, dengan membatasinya di kode program, jadi si penyerang akan kesulitan melihat *input box* nya tidak bisa diinject dengan perintah *query* yang panjang.

Data *user* harus difilter dalam *form inputan*, misalnya membatasi *string* yang hanya dibolehkan dalam pengisian *form*.

Matikan atau sembunyikan pesan-pesan *error* yang keluar dari *SQL Server* yang berjalan. Hal ini untuk menyulitkan bagi penyerang dalam proses memindai target serangan.

Bila memungkinan *administrator* baiknya menggunakan *firewall* untuk aplikasi *webnya* yang fungsinya sebagai *filter* terhadap query-*query* yang terdeteksi mempunyai potensi bahaya bagi keamanan *web*.

DAFTAR PUSTAKA

[1] M. Ula, “Evaluasi Kinerja Software Web Penetration Testing,” *TECHSI - J. Tek. Inform.*, vol. 11, no. 3, p. 336, Oct. 2019, doi: 10.29103/TECHSI.V11I3.1996.

[2] A. RICO AGARTA, “Analisa Keamanan Website Pada Universitas Gunadarma Terhadap Serangan Sql Injection,” Apr. 2021, Accessed: Oct. 06, 2021. [Online]. Available: https://www.binadarma.ac.id/.

[3] B. Bin Halib, E. Budiman, and H. J. Setyadi, “Teknik Hacking Web Server Dengan Sqlmap Di Kali Linux,” *J. Rekayasa Teknol. Inf.*, vol. 1, no. 1, pp. 67–72, Jun. 2017, doi: 10.30872/JURTI.V1I1.642.

[4] R. U. Putri and J. E. Istiyanto, “Analisis Forensik Jaringan Studi Kasus Serangan Sql injection pada Server Universitas Gadjah Mada,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 6, no. 2, pp. 101–112, Jul. 2013, doi: 10.22146/IJCCS.2157.

[5] “Kalilinux Penetration Testing Bible - Google Books.” https://www.google.co.id/books/edition/Kali_Linux_Penetration_Testing_Bible/0EkrEAAAQBAJ?hl=en&gbpv=1&dq=kali+linux&printsec=frontcover (accessed Oct. 12, 2021).

[6] S. S. Ardiansyah, S. Raharjo, and J. Triyono, “Analisis Keamanan Serangan Sql Injection Berdasarkan Metode Koneksi Database,” *J. Scr.*, vol. 4, no. 2, pp. 72–80, Dec. 2016, Accessed: Oct. 06, 2021. [Online]. Available: https://journal.akprind.ac.id/index.php/script/article/view/742.

[7] S. Utoro *et al.*, “Analisis Keamanan Website E-Learning SMKN 1 Cibatu Menggunakan Metode Penetration Testing Execution Standard.”

[8] D. Kurnia, “Analisis Forensik Serangan Sql injection dan DoS Menggunakan Instrution Detection System Pada Server Berbasis Lokal,” *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 4, no. 2, pp. 208–212, Apr. 2020, doi: 10.30743/INFOTEKJAR.V4I2.2420