(2024), 2 (11): 110–121 **Scientica** 3021-8209
Jurnal Ilmiah Sain dan Teknologi

ANALISIS KEAMANAN *WEB* SMAN 1 WANAYASA MENGGUNAKAN SQL*MAP* DENGAN METODE *PENETRATION TESTING EXECUTION STANDARD* (PTES)

Mohamad Noval Rizki Darmawan [1], Yusuf Muhyidin [2], Dayan Singasatia[3]
Program Studi Teknik Informatika S1, Fakultas Teknik, Sekolah Tinggi Teknologi Wastukancana
Jl. Cikopak No.53, Sadang, Purwakarta, Jawa Barat, Indonesia
mohamadnoval74@wastukancana.ac.id

Abstract
Web security is crucial for maintaining the integrity, confidentiality, and availability of systems from threats. This study uses SQLMap and the Penetration Testing Execution Standard (PTES) method to analyze the web security of SMAN 1 Wanayasa. SQLMap detects and exploits SQL injection vulnerabilities, while PTES is a framework for penetration testing. The study did not find SQL injection vulnerabilities but did identify some ports that could potentially be exploited. Recommendations for improvements are provided to enhance web security. Implementing PTES offers a systematic and effective approach to identifying and addressing web security weaknesses.

Article History
Submitted: 19 Juli 2024
Accepted: 24 Juli 2024
Published: 25 Juli 2024

Key Words
Web security, SQLMap, Penetration Testing Execution Standard (PTES), SQL injection, SMAN 1 Wanayasa.

Abstrak
Keamanan *web* sangat penting untuk menjaga integritas, kerahasiaan, dan ketersediaan sistem dari ancaman. Penelitian ini menggunakan SQL*Map* dan metode *Penetration Testing Execution Standard* (PTES) untuk menganalisis keamanan *web* SMAN 1 Wanayasa. SQL*Map* mendeteksi dan mengeksploitasi kerentanan SQL *injection*, sementara PTES adalah kerangka kerja untuk uji penetrasi. Penelitian ini tidak menemukan kerentanan SQL *injection*, namun ada beberapa *port* yang berpotensi diserang. Rekomendasi perbaikan diberikan untuk meningkatkan keamanan *web*. Implementasi PTES memberikan pendekatan sistematis dan efektif dalam mengidentifikasi dan mengatasi kelemahan keamanan *web*

Sejarah Artikel
Submitted: 19 Juli 2024
Accepted: 24 Juli 2024
Published: 25 Juli 2024

Kata Kunci
Keamanan *web*, SQL*Map*, *Penetration Testing Execution Standard* (PTES), SQL *injection*, SMAN 1 Wanayasa..

Pendahuluan

Dengan mengadopsi pendekatan ini, penelitian ini bertujuan untuk melakukan analisis keamanan *web* terhadap situs *web* SMAN 1 Wanayasa menggunakan alat SQL*map*. Melalui analisis ini, diharapkan dapat diidentifikasi potensi kerentanan keamanan yang ada dalam infrastruktur *web* dan merekomendasikan langkah-langkah perbaikan yang sesuai untuk meningkatkan keamanan situs *web* tersebut. Dengan demikian, upaya untuk menjaga keamanan informasi di SMAN 1 Wanayasa dapat ditingkatkan secara signifikan. Dalam era digital yang semakin berkembang, keamanan sistem informasi menjadi hal yang sangat penting untuk dipertimbangkan, terutama dalam konteks aplikasi *web*. Salah satu jenis serangan yang paling umum dan berpotensi merusak adalah SQL *Injection* (SQLi), yang memanfaatkan celah dalam pemrosesan *input* oleh sistem manajemen basis data. Dengan melakukan SQL *Injection*, peretas dapat mengambil alih kontrol atas basis data aplikasi *web*, mengakses informasi sensitif, atau bahkan merusak integritas data (Badan Sandi dan Siber Negara (BSSN), 2023).

Menurut laporan dari *IT Security Agency* (ITSA), SQL *Injection* menjadi salah satu dari lima kerentanan paling kritis pada tahun 2023. Di Indonesia, Badan Siber dan Sandi Negara (BSSN) menerima laporan mengenai penyerangan terhadap sektor pendidikan sebanyak 4

kasus pada tahun yang sama, menyoroti pentingnya keamanan sistem informasi di lembaga pendidikan (Badan Siber dan Sandi Negara (BSSN), 2023).

Tingginya jumlah aduan terkait serangan keamanan pada tahun 2020, sebanyak 1.293 kasus dengan 660 kasus melibatkan laporan dari institusi pemerintah, menunjukkan eskalasi ancaman terhadap keamanan siber di Indonesia. Dari jumlah tersebut, SQL *Injection* mendominasi dengan 265 kasus, menegaskan pentingnya perlindungan terhadap serangan semacam itu (Badan Siber dan Sandi Negara (BSSN), 2023).

Melihat tren sejarah, pada tahun 2017, SQL *Injection* menduduki peringkat pertama dalam daftar OWASP *top* 10 *Application Security Risks*, menunjukkan bahwa serangan ini telah menjadi ancaman yang konsisten dan signifikan dalam lingkungan keamanan aplikasi *web*.

Dalam konteks ini, studi kasus terhadap SMAN 1 Wanayasa sebagai objek penelitian menjadi penting untuk memahami bagaimana keamanan *web* di lembaga pendidikan dapat ditingkatkan. Dengan menggunakan metode Penetration Testing, khususnya dengan alat SQL*Map*, skripsi ini bertujuan untuk menganalisis kelemahan keamanan *web* SMAN 1 Wanayasa terhadap serangan SQL *Injection*, serta memberikan rekomendasi untuk meningkatkan keamanan *web* mereka.

Kajian Pustaka

Website

Website atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau data gambar gerak, data animasi, suara, video dan gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (Andriyan, Septiawan & Aulya., 2020).

Keamanan Web

Keamanan *web* atau *web security* pada dasarnya berarti melindungi situs web atau aplikasi web dengan mendeteksi, mencegah, dan menanggapi ancaman dunia maya (Rifki Mulyawan,. 2024).

Kali Linux

Mengutip dari situs resminya, "*Kali Linux, Advanced Penetration testing Linux distribution used for Penetration testing, Ethical Hacking and network security assessments.*". *Kali Linux* merupakan sistem operasi *open source* yang dapat dimanfaatkan untuk melakukan *Penetration testing* terhadap suatu sistem dan jaringan komputer (Andria, 2020).

SQLmap

SQL*map* merupakan *Tools* untuk penetrasi *open source* yang mampu mengotomatisasi proses deteksi dan eksploitasi kelemahan SQL *Injection* dan juga mampu mengambil alih basis data *web server* (Hermawan, 2021).

Nmap

N*map* (*Network Mapper*) adalah salah satu alat yang paling banyakdigunakan untuk memindai status host target. Dibandingkan dengan alat pemindaian lainnya seperti Hping atau *Scanner*, N*map* menyediakan jenis pemindai yang lebih komprehensif dan metode penghindaran *firewall* (Liao et al., 2020).

(2024), 2 (11): 110–121 **Scientica** 3021-8209
Jurnal Ilmiah Sain dan Teknologi

SQL Injection

SQL *injection* adalah sebuah teknik yang melakukan injeksi kode kepada sebuah aplikasi atau *website* dengan mengincar keamanan *database*. SQL *injection* juga dapat dikatakan sebagai kegiatan yang menipu *query* dari *database* (Ekayanti, Cintiya, Suartana & Pinatih, 2022).

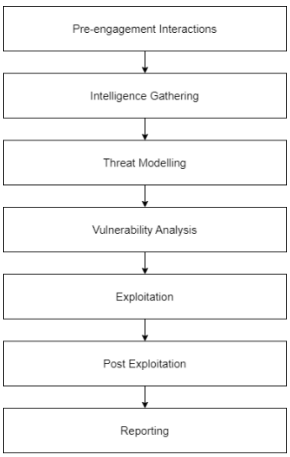
SQL *injection* bisa terjadi karena penyerang menguasai teknik *query* SQL yang sanggup melewati celah keamanan yang ada di SQL pada lapisan basis data suatu aplikasi. Celah ini terjadi karena form input dari pengguna tidak difilter dengan baik terhadap metakarakter dalam pembuatannya menggunakan *form input* (Hermawan, 2021).

Database

Database adalah suatu data yang dimana proses data yang digunakan adalah data secara sistematis yang disimpan melalui sistem data yang telah digunakan dengan data tersebut *database* ini menggunakan secara kalimat, video, gambar dan secara *file*. Biasanya database ini bertujuan untuk mempelancar urusan dan dapat menyimpan *file* data kita secara permanen dengan pemakain yang mampu menyusun dan mempermudah sistem dalam perusahaan, ataupun dalam organisasi lainnya (Pulungan, Febrianti, Lestari, Gurning & Fitriana., 2022).

***Penetration Testing Execution Standard* (PTES)**

Metode *Penetration Testing Execution Standard*s (PTES) dibangun pada tahun 2010 oleh para praktisi information security dengan tujuan menciptakan standar yang akan membantu klien dan penguji dalam menyediakan petunjuk mengenai *Tools* atau peralatan, teknik, dan elemen lainnya yang dilingkupi oleh uji penetrasi secara keseluruhan (Utoro, Nugroho, Meinawati & Widianto., 2020).



Gambar 1 Alur PTES

Port Address

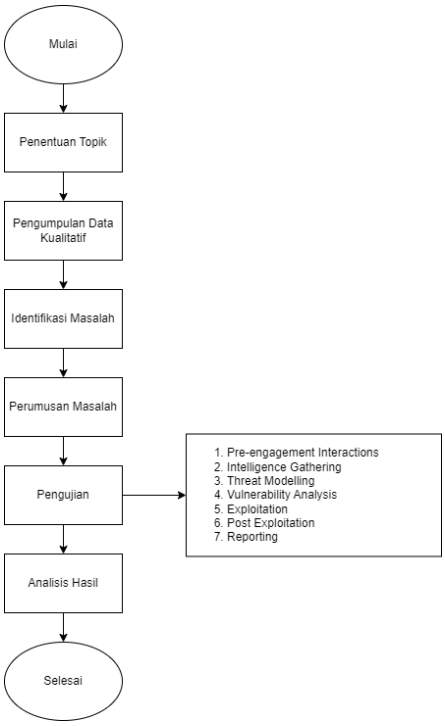
Port komunikasi dalam protokol TCP atau UDP adalah elemen dari lapisan transportasi pada model OSI. *Port-port* ini memungkinkan perangkat berkomunikasi dengan dunia luar dan sebaliknya, memfasilitasi pertukaran informasi yang lancar. Namun, *port* yang terbuka juga dapat menjadi celah keamanan yang bisa dimanfaatkan untuk akses ilegal, menyebabkan potensi gangguan pada jaringan *internal* atau *server* yang terhubung(Riska, Sugiartawan & Wiratama., 2018).

(2024), 2 (11): 110–121 **Scientica** 3021-8209
Jurnal Ilmiah Sain dan Teknologi

Metode

Kerangka Penelitian

Adapun kerangka penelitian yang akan dilaksanakan terdapat di gambar 2 sebagai berikut:



Gambar 2 Kerangka Penelitian

Penentuan Topik

Dalam menentukan topik skripsi tentang analisis keamanan *web* SMAN 1 Wanayasa dengan menggunakan SQL*Map* dan metode *Penetration testing*, langkah-langkah penting perlu diikuti. Pertama, peneliti melakukan penelitian awal tentang keamanan *web* dan teknik-teknik analisisnya. Kemudian, identifikasi kebutuhan khusus dari SMAN 1 Wanayasa terkait dengan keamanan *web* dilakukan, diikuti dengan pengumpulan informasi tentang sistem yang ada. Setelah itu, topik spesifik ditetapkan dengan mempertimbangkan ruang lingkup penelitian yang layak dalam kerangka waktu skripsi. Persetujuan dan validasi dari pihak berwenang diperlukan sebelum melanjutkan ke tahap berikutnya. Dengan langkah-langkah ini, peneliti dapat memastikan bahwa topik skripsi yang dipilih relevan, sesuai dengan kebutuhan, dan memberikan kontribusi yang berarti dalam bidang keamanan *web*.

Pengumpulan Data Kualitatif

Dalam proses pengumpulan data kualitatif ada tiga metode yang dilaksanakan untuk mengumpulkan data data yang diperlukan mengenai penelitian ini. Metode pengumpulan data terdiri sebagai berikut:

1. Wawancara
   Peneliti melakukan pengumpulan data dengan melakukan komunikasi dengan pihak SMAN 1 Wanayasa untuk menanyakan terkait data data yang diperlukan dalam penelitian
2. Observasi

Observasi yang dilakukan peneliti yaitu melakukan pengamatan terhadap *web* SMAN 1 Wanayasa.

3. Studi literatur

Pengumpulan data dengan melihat referensi dari berbagai sumber berupa skripsi, berita, jurnal mengenai keamanan *web*, SQL *injection*, kualitatif dan data pendukung lainnya.

Identifikasi Masalah

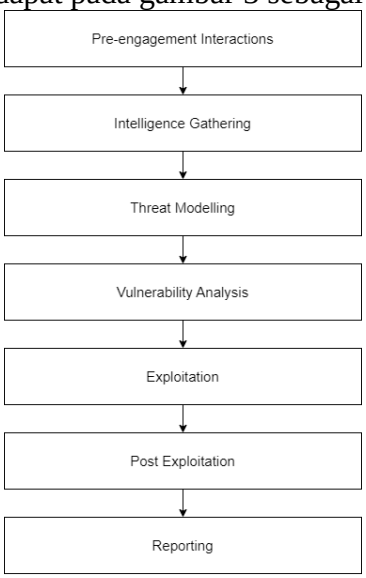
Identifikasi masalah dilakukan setelah mendapatkan data data yang diperlukan dalam proses penelitian. Data data tersebut akan di kumpulkan kemudian akan di simpulkan apa yang menjadi masalah yang terjadi di dalam *web* SMAN 1 Wanayasa.

Perumusan Masalah

Perumusan masalah merupakan proses langkah lanjutan dari identifikasi masalah Ini melibatkan mendefinisikan masalah secara lebih spesifik dan terperinci untuk tujuan analisis dan penelitian. Pada tahap ini, masalah yang telah diidentifikasi dijabarkan dalam bentuk pertanyaan penelitian atau pernyataan yang jelas dan fokus.

Pengujian

Setelah proses perumusan masalah selesai, maka tahap pengujian akan dimulai dengan tahapan tahapan seperti yang terdapat pada gambar 3 sebagai berikut:



Gambar 3 Alur Penyerangan

Pre-engangement Interactions

Tahap *pre-engagement interactions* dimulai dengan menentukan ruang lingkup dan tujuan pengujian keamanan. Hal ini mencakup definisi jelas mengenai apa yang akan diuji dan apa yang diharapkan dari pengujian tersebut. Selanjutnya, mendapatkan izin dan persetujuan dari pihak sekolah untuk melakukan pengujian merupakan langkah penting untuk memastikan legalitas dan etika dari proses pengujian. Setelah persetujuan diperoleh, tahap ini diakhiri dengan menyiapkan alat dan sumber daya yang dibutuhkan, seperti SQL*map* dan N*map*, yang akan digunakan untuk mendeteksi dan mengeksploitasi kerentanan pada *web* sekolah..

Intelligence Gathering

Tahap *Intelligence Gathering* bertujuan untuk mengumpulkan informasi detail tentang target *web*, termasuk struktur dan teknologi yang digunakan. Langkah ini juga melibatkan identifikasi parameter input yang dapat diinput oleh pengguna. Informasi ini akan dikumpulkan menggunakan *Tools* seperti N*map*, yang membantu dalam pemetaan jaringan dan identifikasi layanan yang berjalan di server *web*. Dalam konteks *web* SMAN 1 Wanayasa, parameter *input* yang diidentifikasi terdapat pada form login dan form pencarian di bagian halaman siswa. Dengan mengumpulkan informasi ini, kita dapat memahami lebih baik potensi kerentanan dan titik-titik yang mungkin dieksploitasi dalam tahap pengujian berikutnya.

Threat Modelling

Pada tahap *threat modelling*, langkah pertama adalah menganalisis data yang dikumpulkan untuk memahami ancaman potensial yang mungkin dihadapi oleh *web* SMAN 1 Wanayasa. Proses ini melibatkan identifikasi aset-aset penting, seperti halaman siswa dan guru yang berisi data pribadi milik guru dan siswa. Setelah aset-aset ini diidentifikasi, evaluasi risiko yang terkait dengan masing-masing aset dilakukan untuk menentukan tingkat kerentanan dan dampak potensial dari serangan. Untuk mengidentifikasi celah keamanan, akan dilakukan pencarian celah dengan *scanning port* menggunakan N*map* dan juga mencari celah keamanan SQL *Injection* dengan menggunakan SQL*map*. Dengan cara ini, analisis mendalam terhadap ancaman dapat dilakukan, dan langkah-langkah mitigasi yang tepat dapat dirumuskan untuk melindungi data sensitif tersebut..

Vulnerability Analysis

Melakukan pemeriksaan untuk menemukan kerentanan dalam sistem dengan mgnggunakan alat otomatis berupa N*map* yang akan digunakan dengan perintah:

n*map* (alamat ip *target*)

Exploitation

Pada tahap ini, peneliti mencoba mengeksploitasi kerentanan yang ditemukan untuk memahami sejauh mana kerentanan ini dapat digunakan untuk mendapatkan akses tidak sah atau menyebabkan kerusakan. Berikut ini adalah jenis jenis serangan yang akan dilakukan oleh peneliti:

1. *Error-Based* SQL *Injection*

Serangan ini memanfaatkan pesan kesalahan yang dihasilkan oleh basis data untuk mengekstrak informasi. Contohnya adalah menambahkan *payload* SQL seperti ' OR 1=1-- untuk menghasilkan kesalahan yang berisi informasi tentang struktur database.

Contoh *Payload*: **http://target.com/index.php?id=1'**

2. *Union-Based* SQL *Injection*

Teknik ini menggunakan klausa UNION SQL untuk menggabungkan hasil dari dua atau lebih *query* SELECT. Penyerang dapat menggabungkan *query* berbahaya dengan query yang valid untuk mengambil data dari tabel lain.

Contoh *Payload***: ' OR 0=0 UNION SELECT NULL, VERSION()#** yang digunakan untuk menggabungkan hasil query berbahaya dengan *query* yang valid untuk mengambil data, seperti versi database.

3. *Automated-Based* SQL *Injection*

Jenis serangan ini menggunakan alat otomatis seperti SQL*Map* untuk mengotomatisasi serangan SQL *Injection*. Alat ini dapat mengidentifikasi dan mengeksploitasi berbagai jenis SQL *Injection* dengan efisien.

Contoh *Payload*: **sql*map* -u (*link target*) -dbs –level 3 –risk 3 –tamper=luanginx.py -v 4** , di mana perintah ini digunakan untuk mengeksplorasi basis data target secara mendalam dengan berbagai tingkat risiko dan kompleksitas.

Post Exploitation

Pada tahap *post exploitation*, langkah pertama adalah menganalisis dampak dari eksploitasi dan mengevaluasi sejauh mana celah tersebut dapat membahayakan sistem. Setelah memahami dampak dan potensi risiko, langkah selanjutnya adalah mengidentifikasi langkah-langkah perbaikan yang perlu diambil untuk menutup celah keamanan. Hal ini penting untuk memastikan bahwa sistem *web* SMAN 1 Wanayasa dapat kembali beroperasi dengan aman dan terlindungi dari serangan serupa di masa mendatang..

Reporting

Tahap *Reporting* dalam pengujian keamanan *web* melibatkan mendokumentasikan seluruh temuan dan hasil pengujian, termasuk deskripsi kerentanan, dampak potensial, dan rekomendasi perbaikan. Hasil dokumentasi ini kemudian disusun menjadi laporan pengujian yang komprehensif dan disampaikan kepada pihak sekolah untuk tindakan lebih lanjut.

Analisis Hasil

Setelah hasil percobaan dilakukan apabila terdapat kerentanan yang terjadi ketika melakukan percobaan penyerangan, maka peneliti akan memberikan rekomendasi pada pihak terkait mengenai langkah untuk mengurangi kerentanan yang ditemukan.

Hasil dan Pembahasan

Penentuan Topik

Topik penelitian ini ditentukan berdasarkan kebutuhan untuk memastikan keamanan aplikasi *web* yang digunakan oleh SMAN 1 Wanayasa. Keamanan *web* menjadi sangat penting karena aplikasi *web* sering menjadi sasaran serangan siber, seperti SQL *Injection*. Dengan demikian, penelitian ini bertujuan untuk menganalisis keamanan *web* sekolah dengan menggunakan alat otomatisasi seperti SQL*map*, yang akan membantu mengidentifikasi potensi kerentanan terhadap serangan SQL *Injection*.

Pengumpulan Data Kualitatif

Setelah dilakukan penentuan topik dari penelitian ini, peneliti melakukan melakukan tahapan selanjutnya dengan mengumpulkan data data yang dibuthkan untuk melakukan pengujian keamanan *web* SMAN 1 Wanayasa yaitu dengan melakukan wawancara dengan pihak pengelola data di SMAN 1 Wanayasa, yaitu dengan pihak *staff* Tata Usaha.

Identifikasi Masalah

Setelah mengumpulkan data kualitatif, langkah selanjutnya adalah mengidentifikasi masalah keamanan yang mungkin ada pada *web* SMAN 1 Wanayasa. Dari hasil pengumpulan data, ditemukan beberapa masalah utama yang perlu diatasi:

1. Kurangnya pengujian keamanan
2. Kebijakan Keamanan yang Lemah

Berdasarkan analisis awal, beberapa potensi kerentanan diidentifikasi, termasuk kemungkinan adanya celah SQL *Injection* yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab.

(2024), 2 (11): 110–121 **Scientica** 3021-8209
Jurnal Ilmiah Sain dan Teknologi

Perumusan Masalah

Masalah utama yang difokuskan dalam penelitian ini adalah bagaimana mengidentifikasi dan menganalisis kerentanan SQL *injection* pada aplikasi *web* SMAN 1 Wanayasa menggunakan SQL*map*, dan bagaimana menerapkan Metode *Penetration Testing Execution Standard* (PTES) untuk menguji dan meningkatkan keamanan *web* tersebut.

Pengujian

Tahap pengujian akan menggunakan alur penyerangan *penetration testing execution standard* (PTES) sebagai berikut:

Pre-engagement Interactions

Pada tahap ini, perizinan dijukan oleh peneliti kepada pihak SMAN 1 Wanayasa pada pengujian keamanan *web* yang akan datang di SMAN 1 Wanyasa. Setelah izin diperoleh, peneliti mengatur alat pengujian yang diperlukan seperti SQL*map* dan N*map*.

Intelligence Gathering

Tahap *intelligence gathering* bertujuan untuk mengumpulkan informasi detail tentang target *web*, termasuk struktur dan teknologi yang digunakan. Langkah ini juga melibatkan identifikasi parameter *input* yang dapat diinput oleh pengguna. Informasi ini akan dikumpulkan menggunakan *Tools* seperti N*map*, yang membantu dalam pemetaan jaringan dan identifikasi layanan yang berjalan di *server web*. Dalam konteks *web* SMAN 1 Wanayasa, parameter *input* yang diidentifikasi terdapat pada *form login* dan *form* pencarian di bagian halaman siswa. Dengan mengumpulkan informasi ini, kita dapat memahami lebih baik potensi kerentanan dan titik-titik yang mungkin dieksploitasi dalam tahap pengujian berikutnya.

Threat Modelling

Pada tahap *threat modelling*, langkah pertama adalah menganalisis data yang dikumpulkan untuk memahami ancaman potensial yang mungkin dihadapi oleh *web* SMAN 1 Wanayasa. Proses ini melibatkan identifikasi aset-aset penting, seperti halaman siswa dan guru yang berisi data pribadi milik guru dan siswa. Setelah aset-aset ini diidentifikasi, evaluasi risiko yang terkait dengan masing-masing aset dilakukan untuk menentukan tingkat kerentanan dan dampak potensial dari serangan. Untuk mengidentifikasi celah keamanan, pemindaian akan dilakukan dengan menggunakan N*map*, sementara pemindaian kerentanan SQL *Injection* akan dilakukan dengan menggunakan SQL*Map*.

Vulnerability Analysis

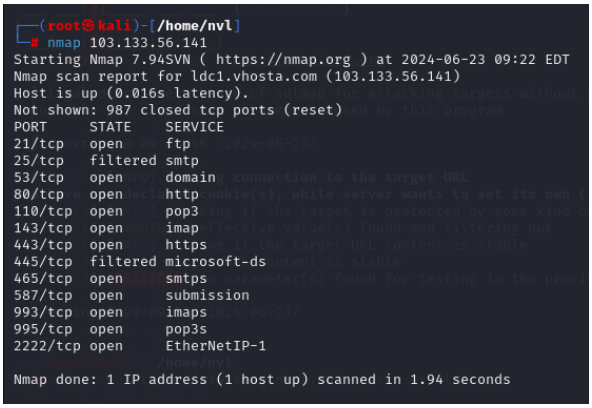
Pada tahap *vulnerability analysis*, dilakukan *scanning port* secara menyeluruh dengan menggunakan *Tools* n*map* dengan hasil yang diharapkan bisa mendapatkan informasi dari *port* yang terbuka pada *web* tersebut.

Di tahap ini, peneliti menggunakan *Tools* n*map* dengan perintah sebagai berikut:

n*map* 103.133.56.141

Hasil dari Nmap bisa dilihat pada gambar 4 sebagai berikut:

(2024), 2 (11): 110–121 **Scientica** 3021-8209
Jurnal Ilmiah Sain dan Teknologi



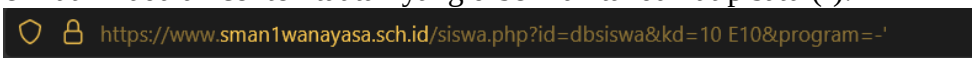
Gambar 4 Hasil N*map*

Exploitation

Pada tahapan *exploitation*, peneliti menggunakan 3 cara untuk bisa mendapatkan celah SQL *injection* pada *web* SMAN 1 Wanayasa. Berikut ini adalh penjelasan nya:

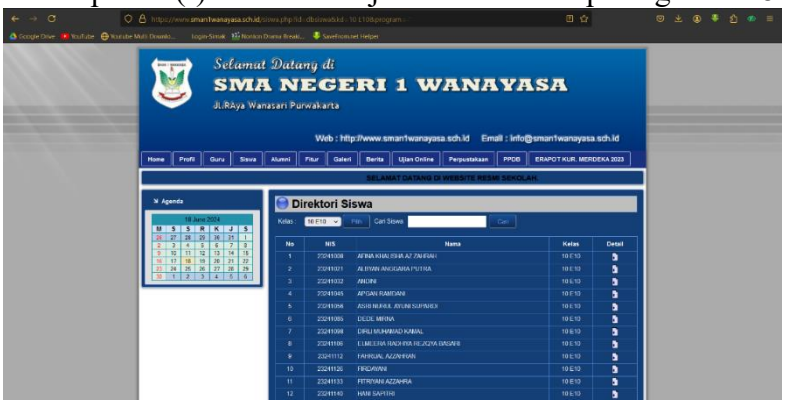
1. *Error Based Injection*

Cara pertama yang ditempuh yaitu dengan menggunakan cara *error based injection* atau mencari celah *error* dari tiap halaman pada *web* SMAN 1 Wanayasa, dengan cara menambahkan tanda kutip satu (‘) pada akhir tautan di tiap halaman di *web* SMAN 1 Wanayasa. Berikut ini adalah contoh tautan yang diberikan tanda kutip satu (‘):



Gambar 5 Perintah *Error Based Injection*

Lalu hasil yang didapatkan adalah *web* tersebut kembali ke halaman awal sebelum ditambahkan tanda kutip satu (‘). Hasil lebih jelas bisa dilihat pada gambar 6 sebagai berikut:



Gambar 6 Hasil *Error Based Injection*

2. UNION *Based Injection*

Untuk cara kedua, peneliti menggunakan cara *union based injection*, yaitu peneliti menggunakan operator UNION untuk menggabungkan pernyataan SQL yang tidak berbahaya dengan pernyataan yang berbahaya. Lalu perintah itu dimasukkan kedalam *form input* yang terdapat pada *web* tersebut. Contoh perintah union yang akan digunakan pada perintah penyerangan ini adalah dengan menggunakan perintah sebagai berikut:

' or 0=0 union select null, version()#

Berikut adalah penjelasan rinci mengenai setiap bagian perintah tersebut:

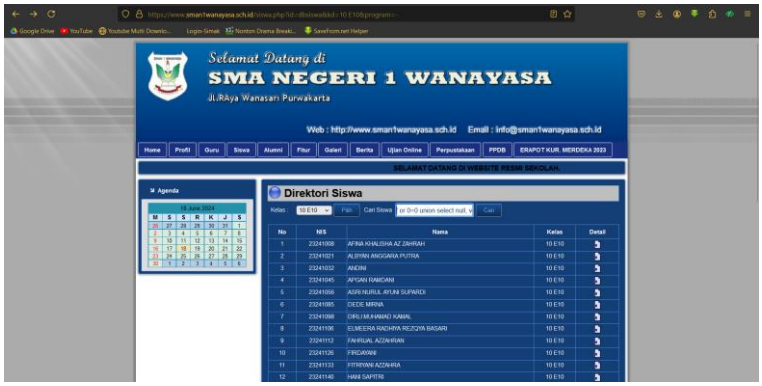
1. **' or 0=0**: Bagian ini adalah bagian dari serangan injeksi SQL yang mencoba untuk mengubah logika kueri asli.

a. (') menutup string literal yang ada dalam kueri asli.

b. (**or 0=0**) adalah kondisi yang selalu benar karena 0 memang sama dengan 0. Ini membuat semua baris data terpilih karena kondisi WHERE selalu benar.

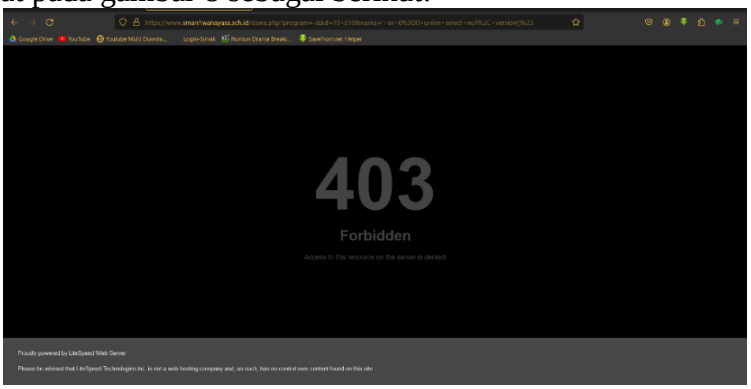
2. **union select null, version()#**: Bagian ini digunakan untuk menggabungkan hasil dari kueri asli dengan hasil dari kueri yang ditentukan penyerang.
   a. (**union**) adalah operator SQL yang menggabungkan hasil dari dua kueri.
   b. (**select null, version())** adalah kueri kedua yang akan digabungkan dengan hasil kueri asli.
   c. (**null**) adalah *placeholder* yang mungkin digunakan untuk mencocokkan jumlah kolom dari kueri asli.
   d. (**version()**) adalah fungsi SQL yang mengembalikan versi dari sistem database yang digunakan.
   e. (#) adalah komentar dalam SQL, yang membuat segala sesuatu setelahnya diabaikan oleh *database server*.

Secara keseluruhan, perintah ini berusaha untuk melakukan serangan SQL *injection* dengan tujuan mendapatkan informasi tentang versi *database* yang digunakan oleh sistem. Jika berhasil, hasil dari fungsi (version()) akan ditampilkan bersama dengan hasil dari kueri asli, memberikan penyerang informasi yang berharga mengenai *database* yang mungkin bisa dieksploitasi lebih lanjut. Berikut ini adalah contoh penyerangan dengan UNION *based injection*:



Gambar 7 Perintah UNION *Based Injection*

Hasil yang didapatkan setelah mencoba UNINON *based injection* adalah *web* menampilkan *error* 403 (*Forbidden*) atau perintah yang diminta ditolak oleh *server*. Hasil tersebut bisa dilihat pada gambar 8 sebagai berikut:



Gambar 8 Hasil UNION *Based Injection*

Hal ini bisa terjadi karena kemungkinan kemungkinan sebagai berikut:
   a. Akses ditolak
   b. IP diblokir, atau
   c. Konfigurasi *server* yang membatasi akses

3. *Automated Based Injection*

(2024), 2 (11): 110–121 **Scientica** 3021-8209
Jurnal Ilmiah Sain dan Teknologi

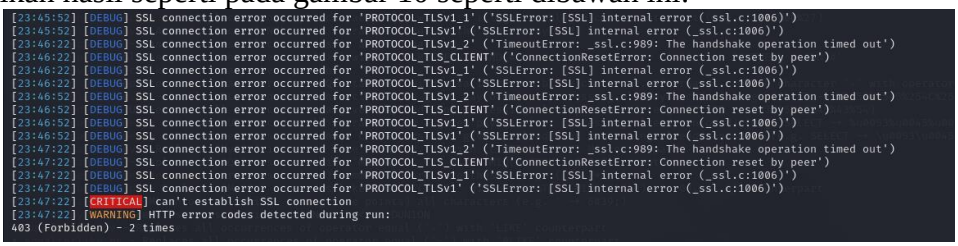
Selanjutnya peneliti menggunakan cara *automated based SQL injection* atau menggunakan alat yang bisa melakukan SQL *injection* secara otomatis contohnya seperti SQL*map*. Untuk menjalankan alat SQL*map* diperlukan perintah sebagai berikut:

sql*map* -u (link target) -dbs –level 3 –risk 3 –tamper=luanginx.py -v 4

Perintah ini akan menjalankan sql*map* untuk menguji URL target yang diberikan untuk kerentanan SQL *Injection* dengan hasil yang diharapkan adalah berhasil menemukan *database* dari *web* tersebut. Dengan menggunakan tingkat pengujian agresif (--level 3) dan risiko tinggi (--risk 3), alat ini akan mencoba berbagai *payload* untuk mengeksploitasi potensi kerentanan. Selain itu, menggunakan skrip tamper luanginx.py akan membantu menghindari mekanisme deteksi keamanan yang mungkin ada. Output dari sql*map* akan cukup rinci karena tingkat verbositas yang tinggi (-v 4) seperti yang terdapat pada gambar 9 berikut ini:

Gambar 9 Perintah SQL*map*

Setelah menunggu beberapa saat untuk menunggu proses yang dilakukan sqlmap selesai, lalu diberikan hasil seperti pada gambar 10 seperti dibawah ini:



Gambar 10 Hasil SQL*map*

Hasil yang didapat, yaitu *web* memunculkan notifikasi *error* dengan kode 403 (*Forbidden*) seperti yang terjadi saat mencoba UNION *based injection.*

Post-Exploitation

Hasil yang ditemukan dari pengujian keamanan pada *web* SMAN 1 Wanayasa adalah *web* ini sudah dilindungi oleh *firewall* yang mumpuni. Sehingga, pihak pengelola *web* hanya perlu terus mengembangkan keamanan untuk mencegah jenis serangan yang lebih berbahaya dari SQL *injection*.

Reporting

Dari hasil pengujian yang telah dilakukan pada *web* SMAN 1 Wanayasa, pihak pengembang telah berhasil untuk mengamakan *web* dari serangan SQL *injection.*

Analisis Hasil

Hasil yang bisa didapat dari penelitian ini adalah pihak pengembang *web* dari SMAN 1 Wanayasa telah berhasil untuk mengamankan *web* milik mereka dari serangan SQL *injection.* Akan tetapi pada saat pemindaian celah keamanan, masih ditemukan beberapa *port* terbuka yang bisa menjadi celah bagi jenis serangan yang lain.

Kesimpulan

Pada penelitian ini peneliti berhasil mengumpulkan hasil hasil yang dibutuhkan untuk menyimpulkan hasil dari penelitian di *web* SMAN 1 Wanayasa. Berikut ini adalah hasil yang ditemukan oleh peneliti pada saat fase pengujian:

1. Pada saat pengujian serangan SQL *injection* terhadap *web* SMAN 1 Wanayasa, *web* telah masuk ke dalam kategori aman. Karena *web* telah melindungi diri nya dengan WAF (*web application firewall*) berupa Imunify360 yang telah berhasil mencegah serangan atau kegiatan yang mencurigakan pada *web.*
2. Pada tahap pemindaian celah, *web* ini masih memiliki beberapa *port* yang terbuka, hal ini bisa dimanfaatkan oleh penyerang untuk melakukan eksploitasi untuk mengambil, memanipulasi atau merusak data yang terdapat pada *web* SMAN 1 Wanayasa.

Jadi pada intinya, *web* ini telah berhasil melindungi diri terhadap serangan SQL *injection.* Akan tetapi, masih terdapat celah celah lain dalam *web* SMAN 1 Wanayasa yang memungkinkan untuk dilakukan eksploitasi.

Daftar Pustaka

Andria. (2020). Analisis Celah Keamanan Website Menggunakan Tools WEBPWN3R di Kali Linux. *Juli 2020 Generation Journal*, *4*(2). http://www.starrybyte.com

Andriyan, W., Septiawan, S., & Aulya, A. (2020). PERANCANGAN WEBSITE SEBAGAI MEDIA INFORMASI DAN PENINGKATAN CITRA PADA SMK DEWI SARTIKA TANGERANG. *Jurnal Teknologi Terpadu*, *6*, 79–88. https://journal.nurulfikri.ac.id/index.php/JTT

Badan Siber dan Sandi Negara (BSSN). (2023). *LANSKAP KEAMANAN SIBER INDONESIA.*

Ekayanti, M. A. G., Cintiya, D. A. D., Suartana, P. Y., & Pinatih, P. N. R. (2022). PERBANDINGAN TOOLS SQL SUS, SQL NINJA, DAN THE MOLE DALAM PENERAPAN SQL INJECTION. In *JINTEKS* (Vol. 4, Issue 4).

Hermawan, R. (2021). *TEKNIK UJI PENETRASI WEB SERVER MENGGUNAKAN SQL INJECTION DENGAN SQLMAP DI KALILINUX*.

Liao, S., Zhou, C., Zhao, Y., Zhang, Z., Zhang, C., Gao, Y., & Zhong, G. (2020). A Comprehensive Detection Approach of Nmap: Principles, Rules and Experiments. *Proceedings - 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2020*, 64–71. https://doi.org/10.1109/CyberC49757.2020.00020

Pulungan, M. S., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2022). Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database. *Jurnal Ekonomi Manajemen Dan Bisnis (JEMB)*, *01*(2), 143–147. https://doi.org/10.47233/jemb.v2i1.533

Riska, P., Sugiartawan, P., & Wiratama, I. (2018). Sistem Keamanan Jaringan Komputer Dan Data Dengan Menggunakan Metode Port Knocking. *Jurnal Sistem Informasi Dan Komputer Terapan Indonesia (JSIKTI)*, *1*(2), 53–64. https://doi.org/10.33173/jsikti.12

Utoro, S., Nugroho, A. B., Meinawati, & Widianto, R. S. (2020). *Analisis Keamanan Website E-Learning SMKN 1 Cibatu Menggunakan Metode Penetration Testing Execution Standard*.

Mulyawan R. (2024). Web Security: Apa itu Keamanan Website? Tujuan dan Fungsi, Jenis, Macam, Strategi serta Kenapa Itu Penting! https://rifqimulyawan.com/blog/pengertian-web-security/