NASKAH PUBLIKASI (*MANUSCRIPT*)

PENETRATION TESTING TERHADAP CELAH KEAMANAN *DATABASE WEBSITE* MENGGUNAKAN SERANGAN *SQL INJECTION*

PENETRATION TESTING ON WEBSITE DATABASE SECURITY VULNERABILITIES USING SQL INJECTION ATTACKS

Putri Dewi Santika, Faldi

DISUSUN OLEH:

PUTRI DEWI SANTIKA

1911102441062

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

SAMARINDA

2023

Naskah Publikasi (*Manuscript*)

Penetration Testing terhadap Celah Keamanan *Database Website* menggunakan Serangan *SQL Injection*

Penetration Testing on Website Database Security Vulnerabilities using SQL Injection Attacks

Putri Dewi Santika, Faldi

Disusun Oleh:

Putri Dewi Santika

1911102441062

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

SAMARINDA

2023

HALAMAN PENGESAHAN

PENETRATION TESTING TERHADAP CELAH KEAMANAN DATABASE WEBSITE MENGGUNAKAN SERANGAN SQL INJECTION

NASKAH PUBLIKASI

DISUSUN OLEH:

PUTRI DEWI SANTIKA

1911102441062

Pembimbing	Penguji
Faldi, S.Kom., M.TI	Muhammad Taufiq Sumadi, S.Tr.Kom., M.Tr.Kom
NIDN: 1121079101	NIDN: 1111089501

Dekan	Ketua Program Studi
Prof. Ir. Sarjito, M.T., Ph.D., IPM	Asslia Johar Latipah, S.Kom., M.Cs
NIDN: 0610116204	NIDN: 1124098902

Penetration Testing terhadap Celah Keamanan *Database Website* menggunakan Serangan *SQL Injection*

Putri Dewi Santika[1*], Faldi[2], Muhammad Taufiq Sumadi[3]
[1,2,3]*Teknik Informatika, Universitas Muhammadiyah Kalimantan Timur, Indonesia.*

[*]*1911102441062@umkt.ac.id*

Abstract

A vulnerable website will be very dangerous if its users store important information related to personal data when an SQL Injection attack occurs. Therefore, this research was conducted to find vulnerabilities by performing vulnerability detection using nmap and nikto tools as well as penetration testing using sqlmap tools against targets. In this research, penetration testing will be carried out with SQL Injection attacks against website database security holes using the SSDLC method which is composed of the stages of requirements, design, implementation, testing, deployment, and maintenance. The results obtained from the first target have SSL but have SQL Injection vulnerabilities so it is easy to exploit, then the second target does not have SSL which has SQL Injection vulnerabilities but has waf so it fails to exploit, and the third target does not have SSL but does not have SQL Injection vulnerabilities so it fails back to exploitation. Vulnerability detection using the tools nmap and nikto on the first target is about 7 minutes difference, while on the second target the difference is about 8 minutes and on the third target the difference is about 1 hour, 28 minutes. And then the results of the duration of penetration testing using sqlmap on the first target produced quite a long duration, but on the second and third targets it produced a fast duration of only 2 minutes. Nmap is faster at detecting vulnerabilities than nikto, but nikto provides detailed output.

Keywords: Website Database, SSDLC, Penetration Testing, SQL Injection

Abstrak

Website yang rentan akan sangat berbahaya jika penggunanya menyimpan informasi penting terkait data pribadi ketika terjadi serangan *SQL Injection*. Oleh sebab itu, penelitian ini dilakukan untuk menemukan kerentanan dengan melakukan *vulnerability detection* menggunakan *tools nmap* dan *nikto* serta *penetration testing* menggunakan *tools sqlmap* terhadap target. Dalam penelitian ini akan dilakukan *penetration testing* dengan serangan *SQL Injection* terhadap celah keamanan *database website* menggunakan metode *SSDLC* yang tersusun dari tahapan *requirements*, *design*, *implementation*, *testing*, *deployment*, dan *maintenance*. Diperoleh hasil dari target pertama mempunyai *SSL* tetapi terdapat kerentanan *SQL Injection* sehingga mudah di eksploitasi, selanjutnya target kedua tidak mempunyai *SSL* yang terdapat kerentanan *SQL Injection* namun terdapat *waf* sehingga gagal di eksploitasi, dan target ketiga tidak mempunyai *SSL* tetapi tidak terdapat kerentanan *SQL Injection* sehingga gagal kembali melakukan eksploitasi. *Vulnerability detection* menggunakan *tools nmap* dan *nikto* pada target pertama selisih sekitar 7 menit, sedangkan pada target kedua selisih sekitar 8 menit dan pada target ketiga selisih sekitar 1 jam, 28 menit. Dan kemudian hasil durasi *penetration testing* menggunakan *sqlmap* pada target pertama menghasilkan durasi yang cukup lama, namun pada target kedua dan ketiga menghasilkan durasi yang cepat hanya selisih 2 menit. *Nmap* lebih cepat mendeteksi kerentanan daripada *nikto,* namun *nikto* memberikan *output detail*.

Kata kunci: *Database Website, SSDLC, Penetration Testing, SQL Injection*

1. Pendahuluan

1.1. Latar Belakang

Website merupakan salah satu platform media yang menampilkan berbagai informasi penting dan menarik dikarenakan mudah bagi siapapun untuk mengaksesnya serta kapanpun dapat digunakan sesuai kebutuhan selama terhubung ke internet. Namun dengan adanya *website* yang mudah diakses, hal ini dapat menjadi pemicu bagi orang yang tidak bertanggung jawab melakukan *cyber crime* terhadap celah keamanan *website*.

SQL Injection merupakan salah satu serangan yang sangat populer dalam mengeksploitasi celah keamanan *website* dari sisi database. Serangan tersebut mampu

mengeksploitasi celah keamanan dengan melalui *form login* atau *URL address* yang terdapat pada *website* oleh orang yang tidak ingin diketahui identitasnya akan melakukan modifikasi dan menginjeksikan perintah *SQL* yang berbahaya. Sehingga, perintah *SQL* yang telah terinjeksi akan berpotensi masuk dan dieksekusi agar mengirimkan baris perintah ke dalam lapisan *database website* yang dikembangkan oleh *web developer*.

Untuk meminimalisir terjadinya hal yang tidak diinginkan oleh pengguna *website* agar merasa aman, maka perlunya melakukan pengujian secara dini dengan cara *penetration testing* menggunakan serangan *SQL Injection* untuk menemukan kerentanan pada *database* yang terdapat di dalam *website* target dan sebagai bahan evaluasi untuk memperbaiki *website* sehingga dapat terhindar dari aksi *cyber crime* oleh pelaku yang tidak ingin menanggung kesalahan atas apa yang terjadi.

1.2. Landasan Teori

1.2.1. Cyber Crime

Salah satu dampak negatif yang digunakan sebagai sarana untuk melakukan kejahatan disebut *cyber crime* atau kejahatan dunia maya. Selain disebut sebagai *cyber criminal*, istilah ini juga disebut sebagai *computer crime*, yaitu suatu jenis kejahatan oleh manusia yang dapat terjadi di internet dengan bantuan alat berupa komputer untuk menghasilkan uang. [1]. Pada saat yang sama, aplikasi web menghadapi tantangan lain. Seperti yang dilaporkan oleh organisasi standar keamanan OWASP, serangan injeksi termasuk di antara sepuluh kerentanan teratas pada tahun 2013 dan 2017, serta serangan *SQL Injection* merupakan jenis serangan injeksi yang terkenal utama [2].

1.2.2. Vulnerability

Untuk menghindari celah keamanan yang terdapat pada suatu sistem maka dapat dilakukan *vulnerability assessment* secara berkala agar cepat mengetahui kerentanan yang paling berbahaya sebelum terjadi peretasan. *Vulnerability assessment* adalah metode yang menguji keamanan aplikasi interaktif seperti *e-banking*, siaran berita, dan web belanja online [3].

1.2.3. Website

Website adalah kumpulan yang berisi sejumlah informasi dalam web yang ditemukan pada nama *domain website* yang menampilkan sejumlah data [4]. Secara teknis, *website* dikenal juga dengan kumpulan halaman web terkait yang dikelompokkan berdasarkan nama atau alamat web secara unik untuk mengidentifikasi *web server*.

1.2.4. Database

Data sangat penting karena berisi informasi pribadi. Data juga dapat berupa gambar, *file*, *pdf*, dan lain-lain. *Database* adalah sekumpulan tabel data yang berisi keterangan terkait dan *database* dapat terbentuk dari satu atau lebih banyak tabel [5].

1.2.5. SQL Injection

SQL Injection merupakan kegiatan meretas yang ditujukan dalam aplikasi *client* dengan mengubah perintah *SQL* dalam *database* aplikasi klien, untuk melakukan teknik yang dieksploitasi oleh aplikasi yang mendasarinya, di mana sistem menggunakan *database* untuk menyimpan data [6]. Terdapat beberapa metode untuk melakukan serangan *SQL Injection* terhadap kerentanannya antara lain *tautologies*, selain itu *union queries*, kemudian *error-based*, berikutnya *boolean-based*, dan terdapat *time-based*, selanjutnya *out of band exploitation technique*, terdapat o*ut of band*, serta *piggy – backed queries attacks*, dan juga *stored procedure injection*, *encoding attacks* [7].

Terdapat beberapa ancaman dari serangan *SQL Injection* seperti *identify spoofing*, mengubah data asli, memodifikasi data, mendapatkan akses secara penuh, penolakan layanan, mendapatkan akses atas informasi yang sangat sensitif [8].

1.2.6. Penetration Testing

Penetration testing (juga dikenal dengan *pentest*) adalah proses yang secara terstruktur untuk menguji keseluruhan dari bagian sistem komputasi yang mencari kerentanan seperti konfigurasi sistem, *bug software* dan

hardware, serta proses operasionalnya dalam mengidentifikasi kelemahan tersebut [9].

Penetration testing dilakukan dengan *tools* seperti *Nikto*, *SQLMap*, dan *XSStrike*. Selain itu dalam melakukan *penetration testing* diperlukan tahapan ataupun proses dalam mencari kerentanan suatu sistem [10]. Menurut [11], tahapan pada *penetration testing* meliputi satu *scope*, kedua *reconnaissance*, ketiga *vulnerability detection*, keempat *information analysis and planning*, kelima *penetration testing*, keenam *privilege escalation*, ketujuh *result analysis*, kedelapan *reporting*, dan kesembilan *clean-up*. Adapun *tools* dalam *vulnerability detection* yang akan digunakan, yaitu:

a. Nmap

Nmap adalah singkatan dari "*Network Mapper*". Nmap adalah alat penemuan jaringan dan audit keamanan *open source* gratis. Nmap mempunyai paket IP *raw* untuk mencari *host* yang tersedia di jaringan. Ini digunakan untuk pemindaian *port* untuk menemukan *port* terbuka di *host*. Menggunakan skrip *nmap* menyediakan *vulnerability detection*, *services detection*, dan fitur lainnya [12].

b. Nikto

Nikto dapat digunakan secara gratis dan berfungsi sebagai *vulnerability scanner* yang bersifat *open source*. *Nikto* memindai server web untuk mencari potensi yang menimbulkan masalah berbahaya dan kerentanan keamanan seperti kesalahan dari konfigurasi server dan *software*, program dan *file default*, *file* dan program yang tidak aman, serta server dan program yang telah habis masa berlakunya [13].

Selain itu, *tools* dalam *penetration testing* yang akan digunakan, yaitu:

a. SQLMap

Tools ini merupakan *software* yang gratis untuk melakukan pengujian penetrasi yang mampu mendeteksi secara efektif untuk menemukan informasi dan mengakses sistem *record* untuk menjalankan urutannya [14].

1.2.7. SSDLC

SSDLC (*Secure Software Development Lifecycle*) adalah model proses yang digunakan oleh organisasi atau perusahaan untuk membuat aplikasi yang aman, sehingga proses *SSDLC* menentukan bagaimana cara menyatukan keamanan ke dalam proses pengembangan *software* [15].

2. Metode Penelitian

Website yang akan dijadikan simulasi akan dibangun menggunakan metode *SSDLC* (*Secure Software Development Lifecycle*). Tahapan dari *SSDLC* yang akan dilakukan dalam penelitian ini yaitu *requirements, design, implementation, testing, deployment*, dan *maintenance*. Adapun penjelasan secara singkat dari metode *SSDLC* tersebut akan diuraikan sebagai berikut:

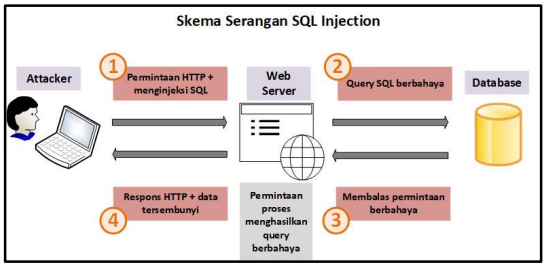
2.1. Requirements

Pada tahap *requirements* ini akan dilakukan pemeriksaan kebutuhan yang akan digunakan dalam membangun sebuah *website*. Adapun *software* dan *hardware* sebagai teknologi yang akan digunakan dalam melakukan penetration testing dengan serangan *SQL Injection* antara lain perangkat keras (*hardware*) yaitu *laptop* dengan 64 *bit-operating system*, *Ram* 8.00 GB, dan *Processor Intell® Core™* i7-8565U dan perangkat lunak (*software*) yaitu *Oracle VM Virtualbox*, *Kali Linux*, *Xampp*, *Google Chrome*, *Nikto & Nmap*, *SQLmap*, dan *Wordpress* versi 6.2.2.

2.2. Design

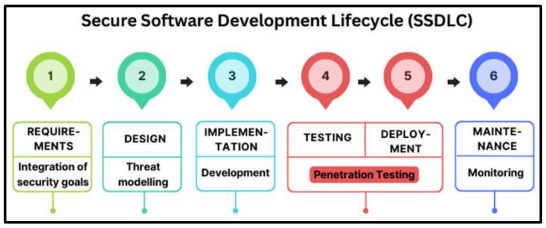
Pada tahap design ini akan dilakukan perancangan. Adapun hasil perancangan desain *prototype* ini berupa skema serangan *SQL Injection*, alur tahapan metode *SSDLC* (*Secure Software Development Lifecycle*) dan alur tahapan penetration testing.

Pada gambar 2.1 dibawah ini yang menunjukkan desain dari skema serangan *SQL Injection*, dimana terdapat 3 komponen utama yang mempunyai tugas sesuai perannya, komponen tersebut yaitu *attacker*, *web server*, dan *database*.



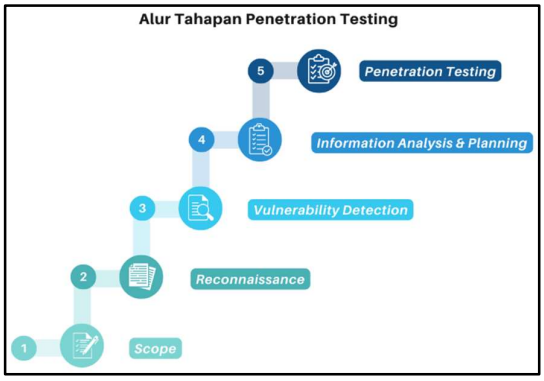
Gambar 2.1 Skema Serangan SQL Injection

Pada gambar 2.2 dibawah ini yang menunjukkan desain dari alur tahapan metode *SSDLC*.



Gambar 2.2 Alur Tahapan Metode SSDLC

Pada gambar 2.3 dibawah ini yang menunjukkan desain dari alur tahapan penetration testing yang akan dilakukan.



Gambar 2.3 Alur Tahapan Penetration Testing

2.3. Implementation

Pada tahap ini akan mengimplementasikan rancangan dari *prototype* yang telah dibuat dengan melakukan pembuatan *website* yang akan diuji. *Website* akan dibuat menggunakan *web server* secara *local* yaitu *xampp* dan menggunakan wordpress versi 6.2.2. *Website* yang telah dibuat pada tahap selanjutnya akan dilakukan *testing*.

2.4. Testing & Deployment

Pada tahap testing ini akan dilakukan pengujian yaitu melakukan *penetration testing* menggunakan dua *tools* yaitu SQLMap dan Nikto dengan serangan *SQL Injection*. Melakukan *penetration testing* sesuai dengan tahapan yang akan dilakukan untuk mengidentifikasi dan mengeksploitasi celah keamanan *website*.

2.5. Maintenance

Pada tahap ini akan diberikan solusi untuk perbaikan terhadap celah keamanan dalam *database website* dengan tetap melakukan *monitoring* terhadap *website* setelah tahap pengujian yang dilakukan.

3. Hasil Penelitian

3.1. Hasil

3.1.1. Scope

Menentukan *scope* merupakan langkah awal dalam melakukan tahapan penetration testing. Dalam menentukan *scope*, terdapat tiga *website* yang akan digunakan sebagai target yaitu *website* yang mempunyai dan tidak mempunyai keamanan *SSL* (*Secure Socket Layer*).

3.1.2. Reconnaissance

Pada tahap kedua dilakukan *reconnaissance* dalam proses *penetration testing*. Proses *reconnaissance* dilakukan dengan menggunakan beberapa *tools* seperti *host*, *nslookup*, *traceroute*, *dnsrecon*, *wafw00f*, *dig*, *whois*, *whatweb* dan *gobuster*.

3.1.3. Vulnerability Detection

Vulnerability detection dilakukan dengan menngunakan dua *tools* yaitu *nmap* dan *nikto* untuk melakukan deteksi kerentanan terhadap ketiga target.

1. Nmap:

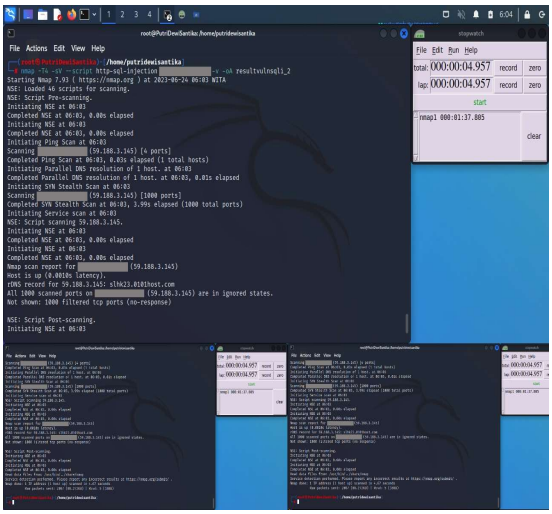
Vulnerability detection dengan menggunakan *tools nmap* pada ketiga target ditunjukkan dalam gambar 3.1 – 3.3 yang terlampir.

a. Pada tampilan gambar 3.1, menampilkan hasil dari *vulnerability detection* menggunakan *tools namp* dan *stopwatch* pada target pertama. *Tools nmap* yang digunakan mampu mendeteksi kerentanan *SQL Injection* dengan memberikan hasil *port scanning* yang menemukan *port-port* terbuka, hal ini menunjukkan adanya celah kerentanan

terhadap *database mysql* yang ditunjukkan pada *port* 3306 dari domain target pertama. Dan diperoleh hasil durasi *vulnerability detection* menggunakan *tools nmap* sekitar 1 jam, 37 menit, 805 milidetik pada target pertama.

Gambar 3.1 *Vulnerability detection SQLi – Nmap* (1)

b. Pada gambar 3.2, menampilkan hasil dari *vulnerability detection* menggunakan *tools namp* pada target kedua. *Tools nmap* yang digunakan tidak terdeteksi kerentanan *SQL Injection* sehingga tidak menampilkan *port-port* yang terdapat pada target kedua, hal ini menunjukkan tidak mempunyai celah kerentanan terhadap *database mysql* dari domain target kedua. Dan diperoleh hasil durasi *vulnerability detection* menggunakan *tools nmap* sekitar 4 detik, 957 milidetik pada target kedua.



Gambar 3.2 *Vulnerability detection SQLi - Nmap* (2)

c. Pada gambar 3.3, menampilkan hasil dari *vulnerability detection* menggunakan *tools namp* pada target ketiga. Dalam melakukan *scanning vulnerability detection* menggunakan *tools nmap* pada target ketiga bahwa tidak terdeteksi kerentanan *SQL Injection* sehingga tidak menampilkan port-port yang terdapat pada target ketiga. Hal ini menunjukkan domain target ketiga tidak mempunyai celah kerentanan terhadap *database mysql*. Dan diperoleh hasil durasi *vulnerability detection* menggunakan *tools nmap* sekitar 2 detik, 677 milidetik pada target ketiga.

Gambar 3.3 *Vulnerability detection SQLi - Nmap* (3)

2. Nikto:

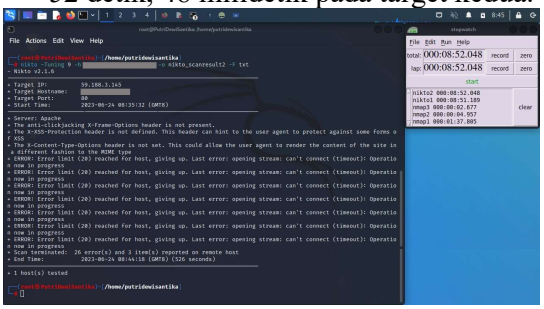
Vulnerability detection dengan menggunakan *tools nmap* pada ketiga target ditunjukkan dalam gambar 3.4 – 3.6 yang terlampir.

a. Pada tampilan gambar 3.4, menampilkan hasil dari *vulnerability detection* menggunakan *tools nikto* pada target pertama. *Tools nikto* yang digunakan mampu mendeteksi server berupa *nginx* dan *control panel* yang digunakan yaitu *plesklin* yang mendukung *database mysql* dan terdeteksi kerentanan selain

SQL Injection seperti *XSS* pada target pertama. Hal ini menunjukkan adanya kerentanan terhadap *database mysql* pada target pertama meskipun tidak terdeteksi secara spesifik kerentanan *SQL Injection*. Dan diperoleh hasil durasi *vulnerability detection* menggunakan *tools nikto* sekitar 8 menit, 51 detik, 189 milidetik pada target pertama.

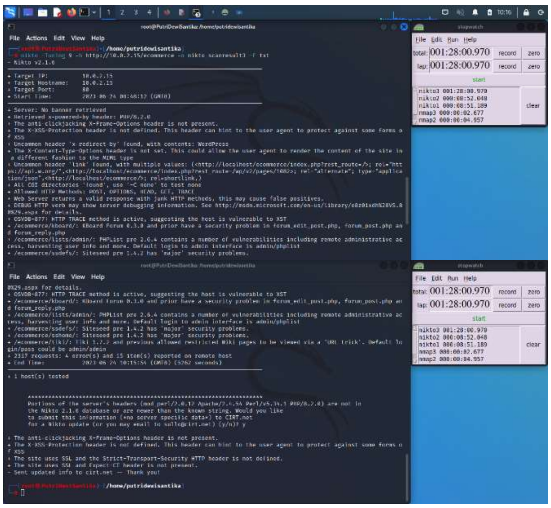
Gambar 3.4 *Vulnerability detection SQLi – Nikto* (1)

b. Pada gambar 3.5, menampilkan hasil dari *vulnerability detection* menggunakan *tools nikto* pada target kedua. *Tools nikto* yang digunakan mampu mendeteksi server berupa *apache* yang mendukung *database mysql* yang artinya terdapat kerentanan *SQL Injection* dan selain itu terdapat kerentanan *clickjacking* dan *XSS* pada target kedua. Hal ini menunjukkan adanya kerentanan terhadap *database mysql* pada target kedua. Dan diperoleh hasil durasi *vulnerability detection* menggunakan *tools nikto* sekitar 8 menit, 52 detik, 48 milidetik pada target kedua.



Gambar 3.5 *Vulnerability detection SQLi – Nikto* (2)

c. Pada gambar 3.6, menampilkan hasil dari *vulnerability detection* menggunakan *tools nikto* pada target ketiga. *Tools nikto* yang melakukan scanning dan mendapatkan hasil bahwa tidak terdeteksi adanya server, dan tidak terdeteksi kerentanan *SQL Injection*, tetapi kerentanan lain terdeteksi seperti *clickjacking* dan *XSS* pada target ketiga. Hal ini menunjukkan tidak terdeteksinya kerentanan terhadap *database mysql* pada target ketiga. Dan diperoleh hasil durasi *vulnerability detection* menggunakan *tools nikto* sekitar 1 jam, 28 menit, 970 milidetik pada target ketiga.



Gambar 3.6 *Vulnerability detection SQLi – Nikto* (3)

3.1.4. Information Analysis & Planning

Setelah melakukan *vulnerability detection* yang berhasil menemukan kerentanan menggunakan *tools nmap* dan *nikto* pada ketiga target, didapatkan hasil deteksi berupa celah keamanan dari database target. Celah keamanan pada target ditampilkan dalam tabel 4.1 dibawah ini.

Tabel 4.1 Celah keamanan yang terdapat pada target

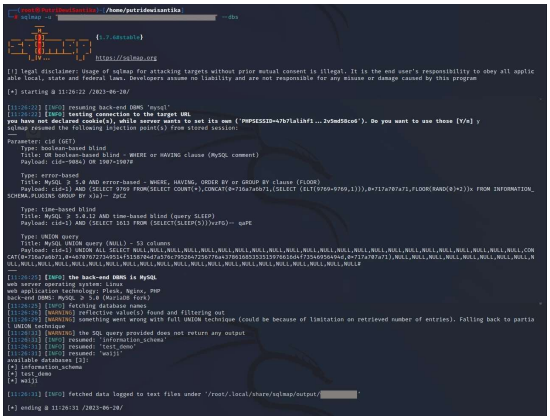
No.	Target	Jenis Kerentanan
1	Pertama	SQL Injection, Clickjacking, XSS
2	Kedua	SQL Injection, Clickjacking, XSS
3	Ketiga	Clickjacking, XSS

Pada tahap *information analysis & planning* akan menggunakan celah keamanan yang terdapat pada tabel 4.1 yaitu target pertama, kedua dan ketiga sebagai bahan dalam melakukan *penetration testing* dengan serangan *SQL Injection* menggunakan *tools* secara otomatis yaitu *sqlmap*. Meskipun target kedua dan ketiga tidak terdeteksi adanya celah keamanan pada *database* target, akan tetapi akan dilakukan pengujian terhadap ketiga target agar dapat melihat perbandingannya dari hasil dan durasinya saat melakukan *penetration testing* menggunakan *tools sqlmap*.

3.1.5. Penetration Testing

Adapun spesifikasi perangkat yang digunakan dalam melakukan *penetration testing* yaitu menggunakan *Oracle VM VirtualBOx* 7.0, Sistem Operasi *Kali Linux*, *Memory* 2 *GB*, *Processor* 2 *CPU*. Dengan spesifikasi perangkat tersebut akan dilakukan *penetration testing* terhadap ketiga target menggunakan *tools sqlmap*.

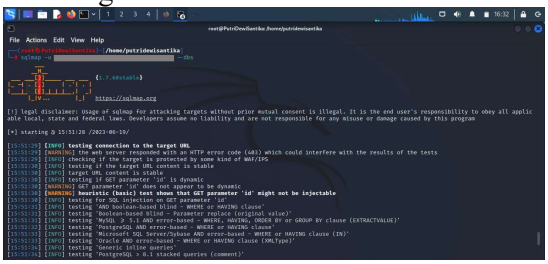
1. Target Pertama



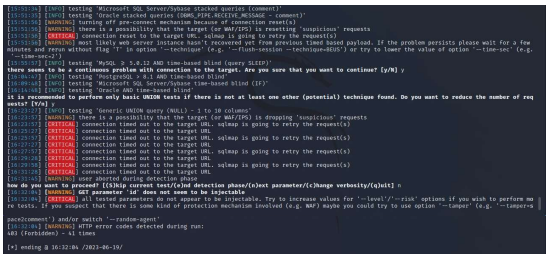
Gambar 3.7 *Penetration testing SQLi - sqlmap* t1

Pada gambar 3.7, menampilkan hasil dari penetration testing menggunakan *tools sqlmap* pada target pertama, dengan menggunakan perintah *--dbs* dan menghasilkan informasi database yaitu *back-end target* pertama dari *BDMS* yaitu *MySQL*. Selain itu, target pertama mempunyai 3 database antara lain, *information_schema*, *test_demo* dan *waiji* yang terdapat pada target pertama.

2. Target Kedua



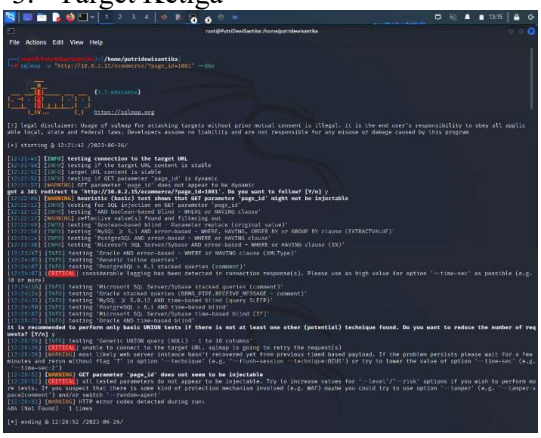
Gambar 3.8 *Penetration testing SQLi - sqlmap* t2



Gambar 3.9 *Penetration testing SQLi - sqlmap* t2 - lanjutan

Pada gambar 3.8 dan 3.9, menampilkan hasil dari *penetration testing* menggunakan *tools sqlmap* pada target kedua, dengan menggunakan perintah *–dbs* yang akan mendeteksi *database* dari target kedua. Tetapi terlihat pada hasil *output* yang diberikan tidak menampilkan *database* target kedua. Setelah memberikan hasil *output*, terlihat tidak terdeteksi adanya *database mysql* pada target kedua, hal ini terjadi dikarenakan adanya perlindungan dari *WAF* yang mampu memblokir serangan *SQL Injection* terhadap *database* target kedua.

3. Target Ketiga



Gambar 3.10 *Penetration testing SQLi - sqlmap* t3

Pada gambar 3.10, menampilkan hasil dari penetration testing menggunakan *tools sqlmap* dengan menggunakan perintah *–dbs* yang akan mengidentifikasi *database* pada target ketiga. Namun terlihat pada hasil yang diberikan bahwa tidak terbacanya database dari target ketiga. Hal ini dapat disebabkan karena kegagalan dalam melakukan serangan SQL Injection pada target ketiga. Hal ini menunjukkan bahwa target ketiga mempunyai *WAF* yang mampu melindungi *database website* dari serangan *SQL Injection*.

3.2. Pembahasan

Melakukan *penetration testing* untuk setiap target akan memerlukan waktu. Sehingga untuk mengetahui waktunya dengan akurat maka digunakan alat pengukur waktu yaitu *stopwatch* yang digunakan selama melakukan vulnerability detection dan penetration testing terhadap target pertama, kedua dan ketiga.

3.2.1. Durasi Vulnerability Detection Dengan Tools Nmap dan Nikto

Dari hasil durasi vulnerability detection menggunakan tools nmap dan nikto saat melakukan scanning kerentanan terhadap target pertama, kedua dan ketiga didapatkan hasil kecepatan durasi yang berbeda dari masing-masing target yang ditampilkan pada tabel 4.2 berikut.

Tabel 4. 2 Hasil Durasi *Vulnerability Detection - tools Nmap* & *Nikto*

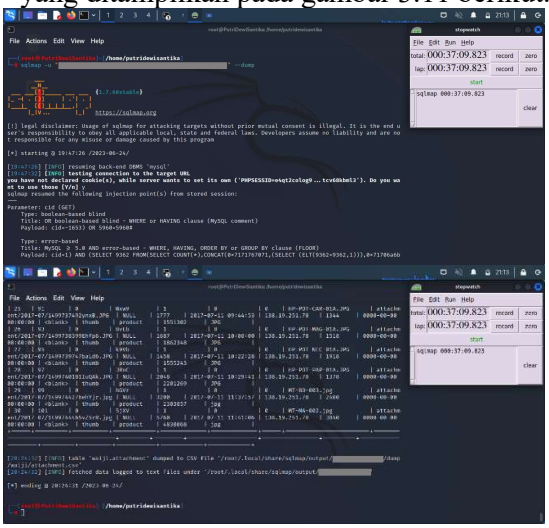
Tools Vuln	**Target**	**Durasi**
Nmap	Pertama	000:01:37.805
	Kedua	000:00:04.957
	Ketiga	000:00:02.677
Nikto	Pertama	000:08:51.189
	Kedua	000:08:52.048
	Ketiga	001:28:00.970

Pada tabel 4.2 terlihat perbandingan hasil durasi *tools nmap* dan *nikto* pada target pertama hasil durasi *scanning vulnerability detection* selisih sekitar 7 menit, sedangkan target kedua selisih sekitar 8 menit dan target ketiga selisih sekitar 1 jam, 28 menit. Hasil *scanning* ini menunjukkan *tools nmap* lebih cepat dalam mendeteksi kerentanan daripada *nikto*. Hasil *scanning* ini menunjukkan *tools nmap* lebih cepat dalam mendeteksi kerentanan daripada *nikto*, namun nikto memberikan hasil output yang lebih detail daripada *nmap*.

3.2.2. Durasi Penetration Testing Dengan Tools SQLMap

1. Durasi Penetration Testing Pada Target Pertama:

Penetration testing menggunakan *tools sqlmap* pada target pertama menggunakan *stopwatch* sebagai pengujian durasi seberapa cepat *tools sqlmap* dalam mengeksploitasi kerentanan *SQL Injection* yang ditampilkan pada gambar 3.11 berikut.

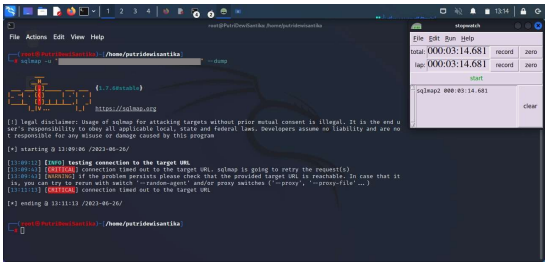


Gambar 3.11 *Penetration testing SQLi - sqlmap* t1 (*stopwatch*)

Diperoleh hasil dari *stopwatch* untuk durasi *penetration testing* menggunakan *sqlmap* pada target pertama yaitu sekitar 37 menit, 9 detik, 823 milidetik. Hasil durasi tersebut menunjukkan durasi selama melakukan *penetration testing* terhadap *database* pada target pertama.

2. Durasi Penetration Testing Pada Target Kedua:

Penetration testing menggunakan *tools sqlmap* pada target kedua menggunakan *stopwatch* dapat dilihat pada gambar 3.12 berikut.



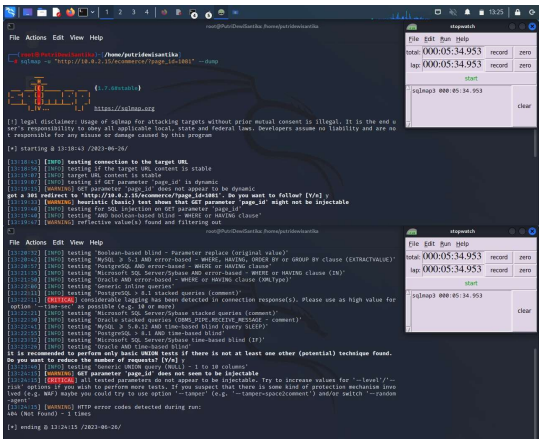
Gambar 3.12 *Penetration testing SQLi - sqlmap* t2 (*stopwatch*)

Diperoleh hasil dari *stopwatch* untuk durasi *penetration testing* menggunakan *sqlmap* pada target kedua yaitu sekitar 3 menit, 14 detik, 681 milidetik. Hasil durasi tersebut menunjukkan durasi selama

melakukan *penetration testing* pada target kedua. Berdasarkan hasil durasi pada target kedua berjalan lebih cepat dibandingkan pada target pertama, hal ini disebabkan karena saat *tools sqlmap* mengekploitasi kerentanan pada target kedua terjadi kegagalan dan tidak berhasil hingga akhirnya durasi eksploitasi akan berhenti ketika tidak menemukan *database* yang terdapat pada target kedua.

3. Durasi Penetration Testing Pada Target Ketiga:

Penetration testing menggunakan *tools sqlmap* pada target ketiga menggunakan *stopwatch* sebagai pengujian durasi seberapa cepat *tools sqlmap* dalam mengeksploitasi kerentanan *SQL Injection* yang dapat dilihat pada gambar 3.13 berikut.



Gambar 3.13 *Penetration testing SQLi - sqlmap* t3 (*stopwatch*)

Diperoleh hasil dari *stopwatch* untuk durasi *penetration testing* menggunakan *sqlmap* pada target ketiga yaitu sekitar 5 menit, 34 detik, 953 milidetik. Hasil durasi tersebut menunjukkan durasi selama melakukan *penetration testing* pada target ketiga. Berdasarkan hasil durasi pada target ketiga berjalan lebih cepat dibandingkan pada target ketiga, hal ini disebabkan karena saat *tools sqlmap* mengekploitasi kerentanan pada target ketiga terjadi kegagalan dan tidak berhasil hingga akhirnya durasi eksploitasi akan berhenti ketika tidak menemukan *database* yang terdapat pada target ketiga.

4. Kesimpulan

Setelah melakukan penelitian *penetration testing* terhadap celah keamanan *database website* menggunakan serangan *SQL Injection* menggunakan *tools sqlmap* diperoleh kesimpulan yaitu:

a. Hasil durasi *scanning vulnerability detection* yang memberikan selisih sekitar 7 menit, sedangkan pada target kedua selisih sekitar 8 menit dan pada target ketiga selisih sekitar 1 jam, 28 menit. Hal ini menunjukkan *tools nmap* lebih cepat mendeteksi kerentanan daripada nikto, namun *nikto* memberikan hasil deteksi yang lebih detail daripada *nmap*.

b. Target pertama mempunyai kerentanan terhadap serangan *SQL Injection*, *Clickjacking*, dan *XSS*. Selain itu pada target kedua mempunyai kerentanan yang sama dengan target pertama. Dan target ketiga tidak mempunyai kerentanan *SQL Injection* tetapi rentan terhadap serangan *Clickjacking* dan *XSS*.

c. Pada saat melakukan *penetration testing* menggunakan *tools sqlmap* ditemukan *database* pada target pertama yang meskipun mempunyai keamanan *SSL* namun tidak mempunyai *web application firewall*. Namun pada target kedua dan ketiga saat melakukan serangan *SQL Injection* menggunakan *tools sqlmap* hasilnya tidak menemukan *database* yang meskipun tidak mempunyai keamanan *SSL* tetapi mempunyai *web application firewall* sehingga target kedua dan ketiga terlindungi dari serangan *SQL Injection*.

d. Dari hasil durasi pada target pertama menghasilkan durasi yang cukup lama disebabkan karena *tools sqlmap* berhasil mengekploitasi kerentanan *SQL Injection*. Sedangkan pada target kedua dan ketiga menghasilkan durasi yang cepat hanya selisih 2 menit.

5. Saran

Berdasarkan analisis dan dari hasil kesimpulan tersebut, diperoleh saran yaitu:

a. Perlunya melakukan *vulnerability detection* menggunakan *tools* selain *nmap* dan *nikto*.

b. Perlunya memasang dan meningkatkan keamanan *website* dalam perlindungan data pada *database website*. Selain menggunakan *SSL* yang belum tentu aman dari serangan *SQL Injection* tetapi dibutuhkan juga teknologi keamanan website lain seperti *WAF* yang mampu mencegah dan memblokir serangan *SQL Injection*.

c. Perlunya menggunakan *tools* selain *sqlmap* seperti *havij* dalam melakukan *penetration testing*.

6. Daftar Pustaka

Jurnal

[1] I. Koto, "Cyber Crime According to the ITE Law," *Int. J. Reglem. Soc. (IJRS*, no. August, pp. 103–110, 2021, doi: 10.55357/ijrs.v2i2.124.

[2] Q. Li, W. Li, J. Wang, and M. Cheng, "A SQL Injection Detection Method Based on Adaptive Deep Forest," *IEEE Access*, vol. 7, pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.

[3] S. Nagpure and S. Kurkure, "Vulnerability Assessment and Penetration Testing of Web Application," *2017 Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2017*, pp. 1–6, 2018, doi: 10.1109/ICCUBEA.2017.8463920.

[4] J. Asmara, "Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala)," *J. Pendidik. Teknol. Inf.*, vol. 2, no. 1, pp. 1–7, 2019.

[6] A. Bastian, H. Sujadi, and L. Abror, "Analisis Keamanan Aplikasi Data Pokok Pendidikan (DAPODIK) Menggunakan Penetration Testing Dan SQL Injection," *INFOTECH J.*, vol. 6, no. 2, pp. 65–70, 2020.

[7] J. Hu, W. Zhao, and Y. Cui, "A Survey on SQL Injection Attacks, Detection and Prevention," *ACM Int. Conf. Proceeding Ser.*, pp. 483–488, 2020, doi: 10.1145/3383972.3384028.

[8] S. Lika, R. D. P. Halim, and I. Verdian, "Analisa Serangan Sql Injeksi Menggunakan Sqlmap," *POSITIF J. Sist. dan Teknol. Inf.*, vol. 4, no. 2, p. 88, 2018.

[9] H. M. Z. Al Shebli and B. D. Beheshti, "A study on penetration testing process and tools," *2018 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2018*, pp. 1–7, 2018, doi: 10.1109/LISAT.2018.8378035.

[10] S. Rawat, T. Bhatia, and E. Chopra, "Web Application Vulnerability Exploitation using Penetration Testing scripts," *Int. J. Sci. Res. Eng. Trends*, vol. 6, no. 1, pp. 2395–566, 2020, [Online]. Available: www.google.com

[11] Y. Khera, D. Kumar, S. Sujay, and N. Garg, "Analysis and Impact of Vulnerability Assessment and Penetration Testing," *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 525–530, 2019, doi: 10.1109/COMITCon.2019.8862224.

[12] K. Chhillar and S. Shrivastava, "University Computer Network Vulnerability Management using Nmap and Nexpose," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 10, no. 6, pp. 3084–3090, 2021, doi: 10.30534/ijatcse/2021/021062021.

[13] K. Siva Prasad, D. K. Raja Sekhar, and D. P. Rajarajeswari, "An Integrated Approach Towards Vulnerability Assessment & Penetration Testing for a Web Application," *Int. J. Eng. Technol.*, vol. 7, no. 2.32, p. 432, 2018, doi: 10.14419/ijet.v7i2.32.15733.

[14] B. S. Samantha and M. V Phanindra, "an Overview on the Utilization of Kali Linux Tools," *IJRAR-International J. Res. Anal. Rev.*, vol. 5, no. 2, 2018, [Online]. Available: http://ijrar.com/

[15] L. Y. Banowosari and B. A. Gifari, "System Analysis and Design Using Secure Software Development Life Cycle Based On ISO 31000 and STRIDE. Case Study Mutiara Ban Workshop," *Proc. 2019 4th Int. Conf. Informatics Comput. ICIC 2019*, pp. 1–6, 2019, doi: 10.1109/ICIC47613.2019.8985938.

Buku

[5] E. Setyawati, Sarwani, H. Wijoyo, and N. Soeharmoko, *RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)*, Cetakan Pe. Banyumas, Jawa Tengah: Penerbit CV. Pena Persada, 2020. doi: 10.1145/800194.805904.

Naspub: Penetration Testing Terhadap Celah Keamanan Database Website Menggunakan Serangan SQL Injection

by Putri Dewi Santika

Submission date: 24-Jul-2023 08:49AM (UTC+0800)
Submission ID: 2135659235
File name: enggunakan_Serangan_SQL_Injection_Putri_Dewi_Santika_-_rev3.docx (12.32M)
Word count: 3956
Character count: 24963

Naspub: Penetration Testing Terhadap Celah Keamanan Database Website Menggunakan Serangan SQL Injection

ORIGINALITY REPORT

6%	5%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	isindexing.com Internet Source	2%
2	Submitted to Indiana University Student Paper	1%
3	doku.pub Internet Source	1%
4	www.coursehero.com Internet Source	1%
5	iwayanjatiyasatumingal.blogspot.com Internet Source	<1%
6	eprints.radenfatah.ac.id Internet Source	<1%
7	erepository.uwks.ac.id Internet Source	<1%
8	pusdatin.kemkes.go.id Internet Source	<1%
9	www.scribd.com Internet Source	<1%

UMKT
UNIVERSITAS MUHAMMADIYAH
Kalimantan Timur

Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda
Kampus 2 : Jl. Pelita, Pesona Mahakam, Samarinda
Telp. 0541-748511 Fax.0541-766832

بسم الله الرحمن الرحيم

SURAT KETERANGAN ARTIKEL PUBLIKASI

Assalamu'alaikum Warahmatullahi wabarakatuh

Saya yang bertanda tangan dibawah ini:

Nama	:	Faldi, S.Kom., M.TI
NIDN	:	1121079101
Nama	:	Putri Dewi Santika
NIM	:	1911102441062
Fakultas	:	Sains dan Teknologi
Progam Studi	:	S1 Teknik Informatika

Manyatakan bahwa artikel ilmiah yang berjudul "Penetration Testing Terhadap Celah Keamanan Database Website Menggunakan Serangan SQL Injection" telah di submit pada Media Teknologi Informasi dan Komputer Jurnal (METIK JURNAL) pada tahun 2023.

https://journal.universitasmulia.ac.id/index.php/metik

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Warahmatullahi wabarakatuh

Samarinda, Senin 24 Juli 2023

Mahasiswa	Pembimbing
Putri Dewi Santika	Faldi, S.Kom., M.TI
NIM. 1911102441062	**NIDN.** 1121079101