Analisis Keamanan Website *XYZ* Dengan Teknik *Footprinting* Dan *Vulnerability Scanning*

Ni Ketut Rida Pratiwi[1)], Roy Rudolf Huizen[2)], I Made Ari Santosa[3)]
Teknologi Informasi[1)], Sistem Informasi[2)], Sistem Komputer[3)]
Institut Teknologi dan Bisnis STIKOM Bali
Denpasar, Indonesia
e-mail: 210040222@stikom-bali.ac.id[1)] , roy@stikom-bali.ac.id[2)], arisantosamade@gmail.com[3)]

Abstrak

Keamanan website menjadi aspek krusial dalam era digital untuk melindungi data sensitif dari serangan siber yang semakin kompleks. Penelitian ini mengevaluasi tingkat keamanan website XYZ dengan menerapkan teknik footprinting dan vulnerability scanning guna mengidentifikasi celah kerentanan yang dapat dieksploitasi oleh peretas. Metode Vulnerability Assessment digunakan dengan memanfaatkan alat seperti Nmap, SQLMap, dan OpenVAS untuk mengukur tingkat risiko keamanan. Hasil pengujian menunjukkan adanya kerentanan berisiko tinggi (High Risk), terutama pada serangan SQL Injection dan XSS (Cross-Site Scripting), serta kerentanan berisiko rendah (Low Risk) pada ICMP Timestamp Reply, yang dapat menyebabkan kebocoran data atau manipulasi informasi dalam sistem. Berdasarkan temuan ini, direkomendasikan penerapan kebijakan keamanan yang lebih ketat, seperti pembatasan hak akses, enkripsi data sensitif, penerapan Web Application Firewall (WAF), serta audit keamanan berkala. Selain itu, implementasi Content Security Policy (CSP) dan pembaruan sistem secara rutin menjadi langkah strategis dalam meningkatkan ketahanan terhadap ancaman siber. Kesimpulannya, penelitian ini memberikan wawasan penting bagi pengelola website dalam meningkatkan perlindungan terhadap serangan siber yang terus berkembang dan memastikan integritas serta keamanan sistem informasi.

Kata kunci: *Keamanan Website, Footprinting, Vulnerability Scanning, SQL Injection, OpenVAS.*

1. Pendahuluan

Era digitalisasi saat ini mengalami transformasi yang sangat pesat, yang ditandai dengan perkembangan teknologi membawa perubahan sebagai media penyebaran informasi dan komunikasi [1]. Salah satu implementasi nyata teknologi informasi mampu memenuhi kebutuhan masyarakat dalam memberikan akses informasi yang cepat untuk mendukung penelitian, pendidikan, dan pengambilan keputusan berbasis data yakni adanya *website*.

Website adalah kumpulan halaman yang berisi informasi berupa aset data penting sehingga perlu diperhatikan keamanannya guna menghindari kemungkinan adanya pencurian data ataupun mengubah isi dari tampilan *website* tersebut [1],[2]. Untuk memastikan keamanan website tersebut, diperlukan aplikasi web yang mampu memberikan dedikasi terhadap proses terjadinya peretasan data dalam kerentanan web, atau yang sering kita temui yaitu *badstore.*

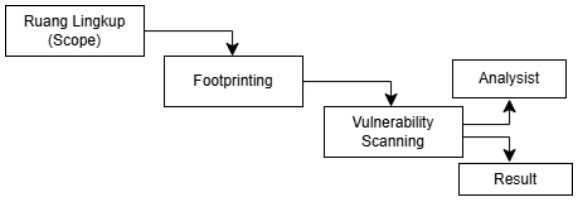
Badstore merupakan aplikasi web yang memang dirancang memiliki kerentanan untuk dilakukannya pengujian sebagai media pembelajaran. Server badstore berada pada *virtual machine*, yang dimana untuk menjalankannya menggunakan perintah *ifconfig* untuk mendapatkan alamat IP kemudian dipanggil menggunakan *browser* untuk diuji kerentanannya melalui *Kali Linux*. Adapun kerentanan yang akan diuji seperti *XSS (Cross-Site Scripting), SQL Injection* dan lain-lain [3], [4], [5]. Alasan menguji dua jenis kerentanan ini, karena menurut daftar *OWASP* Top 10, termasuk serangan paling umum dan berbahaya.

Pengujian *vulnerability* pada penelitian ini, menggunakan *Vulnerability Assessment*, proses mengidentifikasi dan memprioritaskan kerentanan dalam sistem komputer, aplikasi dan infrastruktur jaringan, berupa penilaian latar belakang risiko yang melibatkan alat pemindaian otomatis dan diharapkan dapat memberikan rekomendasi dan solusi dalam meningkatkan performa keamanan website [6], [7].

2. Metode Penelitian

Penelitian ini menggunakan metode *Vulnerability Assessment*, yakni metode untuk mencari celah kerentanan dalam sebuah website berupa penilaian pengukuran skor *high, medium, low*, dan informational yang didasarkan atas sistem CVSS (*Common Vulnerability Scoring System*) [8].

Dalam pengujian dan evaluasi keamanan sistem, diperlukannya teknik *Footprinting* dan *Vulnerability scanning*, yang terbagi atas beberapa tahapan: ruang lingkup (*scope*), *footprinting, vulnerability scanning, analysist* dan *result*, tahapan penelitian ini ditunjukkan seperti pada Gambar 1, sebagai berikut:



Gambar 1. Tahapan Penelitian

2.1 Ruang Lingkup (Scope)
Tahap awal yang dilakukan dengan memperhatikan lingkungan uji coba, vulnerabilitas, deteksi mitigasi serangan dan pemeliharaan pengembangan website target yang akan diuji tanpa melakukan eksploitasi terhadap sistem [9].

2.2 Footprinting
Tahap pengumpulan informasi Badstore menggunakan sistem operasi *Kali Linux* dalam melakukan *penetration testing*, menggunakan *Nmap* dan menguji *SQLMap* untuk memperoleh hasil *SQL Injection* dan skor *high, medium, low*, dan informational [5].

2.3 Vulnerability Scanning
Tahapan yang memanfaatkan tools network scanning *OpenVAS* untuk memperoleh informasi terkait kerentanan yang ada [3]. Dalam menjalankan *OpenVas gvm-setup gvm-start*. *OpenVas* dapat diakses melalui browser dengan https://127.0.0.1:9392 [10].

2.4 Analysist
Tahap menganalisis informasi-informasi vulnerability yang ditemukan setelah dilakukannya scanning serta memberikan rekomendasi perbaikan dan pengembangan [9], [11]

2.5 Result
Tahap akhir yang dilakukan untuk mencatat dan mendokumentasikan hasil analisis dari celah keamanan website target sebagai pegangan bagi pengelola website untuk kedepannya [4], [6]. Pada pencatatan hasil uji scanning Badstore, disertai dengan dokumentasi berupa *screenshoot* pada halaman pengujian.

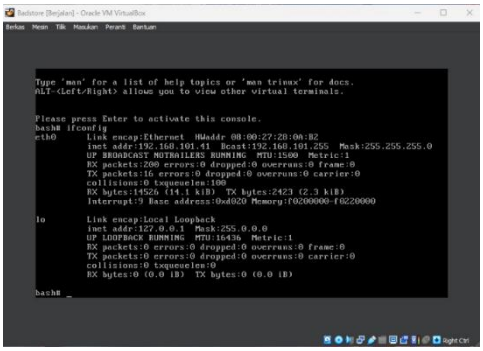
3. Hasil dan Pembahasan

3.1 Ruang Lingkup (Scope)

Penggunaan Oracle VM Virtual Box dengan menginstall dan membuat virtual machine "Kali Linux" dan "BadStore" menggunakan ISO https://old.kali.org/kali-images/kali-2022.3/kali-linux-2022.3-installer-amd64.iso . Dilanjutkan dengan menginstall *tools* Nmap, SQLMap dan OpenVAS sebagai uji *scanner*. Kemudian BadStore dicoba dengan menggunakan perintah *ifconfig* setelah sistem berjalan, maka diperoleh alamat IP 192.168.101.41 . Alamat IP ini di *copy*, lalu dipanggil melalui *browser*. Apabila halaman badstore sudah terlihat, *SQL Injection* diuji dengan memasukkan perintah *1=1* pada bagian "Quick Item Search" maka didapatkan alamat IP baru untuk diuji pada Kali Linux.

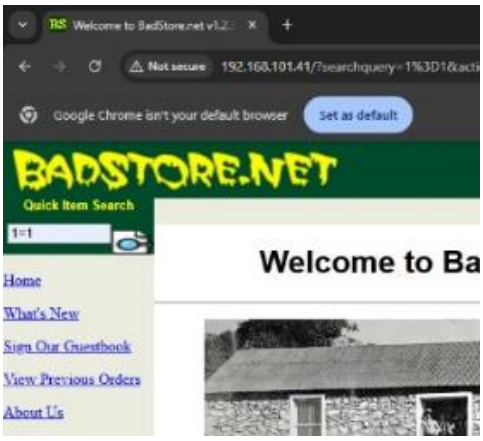
1. Uji *ifconfig* Badstore pada Kali Linux, kemudian mencoba perintah *1=1* pada halaman Badstore, ditunjukkan pada Gambar 2 dan Gambar 3 sebagai berikut:

   Pada Gambar 2, didapatkan hasil scan uji Badstore pada Kali Linux, untuk mendapatkan alamat IP Badstore.



Gambar 2. Mencari IP Badstore

Gambar 3, menunjukkan halaman badstore pada Kali Linux, untuk mendapatkan alamat IP Badstore. berhasil dibuka, dan mencoba untuk uji *1=1



Gambar 3. Menguji 1=1 pada halaman Badstore

3.2 Footprinting (Nmap)

Pada tahapan ini, menggunakan tools Nmap yang sudah diinstal untuk melakukan scanning port agar dapat diakses untuk diuji vulnerability testing. Kemudian menguji SQLMap untuk mencari kerentanan SQL Injection, dengan memaksukkan script sqlmap -u http://192.168.101.41/images/Badstore.jpg --dbs. Setelah berhasil, maka diperoleh beberapa kerentanan, dengan terdapat 'CRITICAL' dan 'WARNING' . Dipastikan terdapatnya SQL Injection, dilanjutkan menguji sistem CVSS untuk memperoleh hasil risiko level kerentanan pada CCS Injection, dengan menggunakan hasil informasi dari scanning Nmap. Dan hasil dari scanning, didapatkan risk level 'High' dengan state 'VULNERABLE'. Proses dari footprinting, ditunjukkan pada Gambar 4, Gambar 5 dan Gambar 6, sebagai berikut:

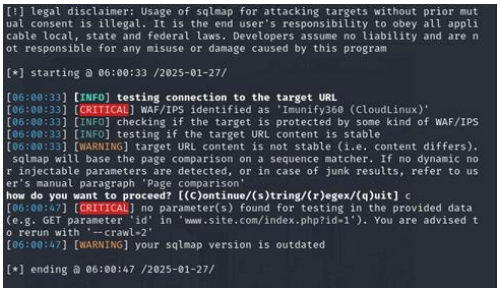
1. Uji scanning Nmap
   Melakukan *scanning port* untuk uji vulnerability testing, dengan menginput alamat IP Badstore.

```
└─$ nmap 192.168.101.41
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-12 04:49 PST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.20 seconds
```

Gambar 4. Hasil Scan Nmap

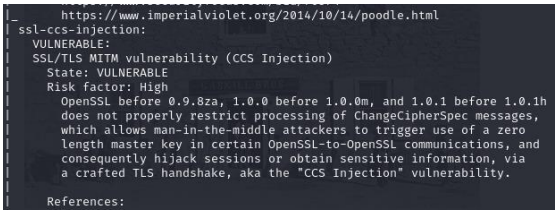
2. Hasil pengujian SQLMap

Pengujian yang dimulai dengan script "sqlmap -u http://192.168.101.41/images/Badstore.jpg --dbs, menghasilkan 'CRITICAL' dan 'WARNING' yang menandakan adanya kerentanan.



Gambar 5. Hasil Scan SQL Injection

3. Hasil pengujian CCS Injection
   Diperoleh hasil risk factor 'High' pada pengujian CCS Injection, menggunakan sistem CVSS (*Common Vulnerability Scoring System*).

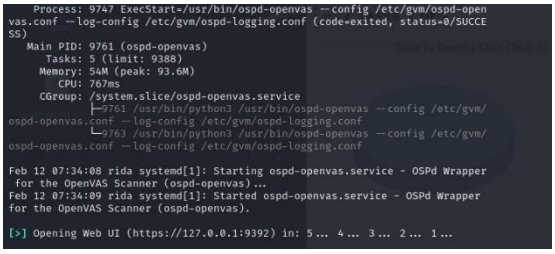


Gambar 6. Hasil Scan CCS Injection

3.2 Vulnerability Scanning (OpenVAS)

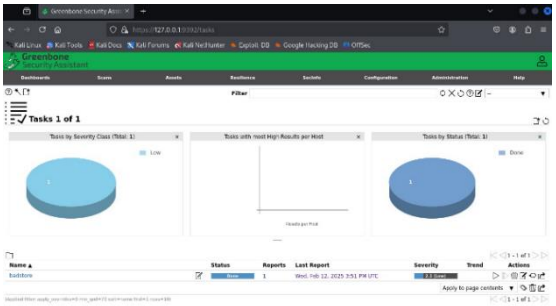
Selain menguji kerentanan SQL Injection menggunakan Nmap, penggunaan *tools* OpenVAS juga digunakan untuk mencari celah kerentanan lainnya dengan *OpenVas gvm-setup gvm-start* https://127.0.0.1:9392. Sebelum melakukan scanning, diperlukannya untuk mengeset target IP yang akan discan. Setelah berhasil, nantinya akan muncul link untuk akses menuju halaman openVAS, kemudian dilanjutkan dengan memasukan alamat IP Badstore, maka uji scanning dijalankan dan mendapatkan hasil. Proses ini ditunjukkan pada Gambar 7, Gambar 8 dan Gambar 9 sebagai berikut:

1. Uji set target IP
   Proses yang dilakukan setelah install OpenVAS, gvm-setup dan gvm-start pada Kali Linux, kemudian mengeset target IP, yang nantinya ditujukan pada opening web OpenVAS https://127.0.0.1:9392.



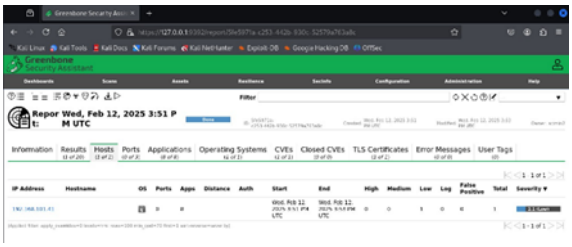
Gambar 7. Set IP target

2. Proses dan hasil scan Badstore pada halaman OpenVas
   Pada Gambar 8, ditunjukkan proses jalannya scanning pada Badstore.



Gambar 8. Proses Scan

Pada Gambar 9, terlihat bahwa hasil scanning Badstore pada OpenVAS diperoleh risk factor 'Low'



Gambar 9. Hasil uji scan Badstore

3.3 Analysist

Berdasarkan hasil uji *scanning*, dapat dicatat analisis terhadap celah keamanan yang terdeteksi, berdasarkan tabel di bawah ini:

Tabel 1. Hasil Analisis

Vulnerability Scanning	Alert	Risk Assessment
Nmap	SQL Injection	High
	CCS Injection	High
OpenVAS	ICMP Timestamp Reply Information Disclosure	Low

3.4 Result

Tahap terakhir dalam pencatatan *vulnerability assessment*. Berfokus pada hasil pengujian yang terdeteksi. Berikut tabel result berdasarkan hasil uji scanning:

Tabel 2. Result

Alert	Risk Assessment	Keterangan	Solusi
SQL Injection	High	Terdapat 2 Critical yang dapat menyebabkan eksekusi perintah arbitrer. Dan 2 Warning yang menandakan adanya potensi untuk perbaikan.	Memastikan semua data sensitif tidak tersedia dengan cara yang tidak diauntetifikasi. Rutin melakukan audit keamanan dan pengujian penetrasi. Serta menerapkan Web Application Firewall (WAF) untuk memantau, menyaring dan memblokir lalu lintas HTTP/HTTPS yang mencurigakan.

Lanjutan Tabel 2. Result

CCS Injection	High	Ditandakan dengan hasil CCS Injection Vulnerable dan risk factor 'High'	Memberikan batas akses dan hak penggunaan dengan menerapkan CSP (Content Security Policy)
ICMP Timestamp Reply	Low	Terdapat vulnerability ICMP Timestamp Reply Information Disclosure dengan risk factor 'Low' 1	Rutin memperbaharui software dan framework versi terbaru, menutup port yang tidak diperlukan dan menerapkan x-frame options, untuk mencegah serangan clickjacking.

4. Kesimpulan

Berdasarkan hasil pembahasan diatas, dapat disimpulkan bahwa dalam pengujian keamanan pada website Badstore menggunakan Kali Linux yang memanfaatkan tools Nmap dan OpenVAS, diperoleh jenis *vulnerability* SQL Injection dan CCS Injection bernilai *High* dan ICMP *Timestamp Reply* bernilai *Low*. Pada hasil uji *scanning* tersebut, dapat diberikan solusi dan rekomendasi untuk meningatkan performa kemanan website seperti memebatasi hak akses, memperhatikan penyimpanan data sensitif dalam bentuk dienkripsi, melakukan audit dan pengujian penetrasi secara rutin serta mengaktifkan monitoring dan logging yang memadai dalam aplikasi server dengan menerapkan Web Application Firewall (WAF) untuk memantau, mendeteksi dan memblokir aktivitas yang mencurigakan.

Daftar Pustaka

[1] J. Pendidikan and D. Konseling, "Analisis Keamanan Website Universitas Singaperbangsa Karawang Menggunakan Metode Vulnerability Assessment."

[2] Y. Mulyanto and E. Haryanti, "SUMBAWA MENGGUNAKAN METODE VULNERABILITY ASESEMENT," *JINTEKS*, vol. 3, no. 3, 2021, doi: 10.51401.

[3] S. A. Hafsari and H. Harjono, "Analisis Keamanan Website Pendaftaran Mahasiswa Baru Dengan Munggunakan Metode Vulnerability Assessment," *TIN: Terapan Informatika Nusantara*, vol. 4, no. 11, pp. 698–708, Apr. 2024, doi: 10.47065/tin.v4i11.5060.

[4] E. Irawadi Alwi and F. Umar, "Analisis Keamanan Website Menggunakan Teknik Footprinting dan Vulnerability Scanning," 2020.

[5] R. Hermawan, "STRING (Satuan Tulisan Riset dan Inovasi Teknologi) TEKNIK UJI PENETRASI WEB SERVER MENGGUNAKAN SQL INJECTION DENGAN SQLMAP DI KALILINUX."

[6] E. Irawadi Alwi and L. Budi Ilmawan, "Analisis Keamanan Sistem Informasi Akademik (SIAKAD) Universitas XYZ Menggunakan Metode Vulnerability Assessment," 2021.

[7] W. Ardiyasa and A. T. Ndok, "Penetration Testing Keamanan Sistem Informasi Berbasis Web dengan Metode OSSTMM." [Online]. Available: https://ti.stikom-bali.ac.id/.

[8] H. Jurnal and R. Lana Rahardian, "JURNAL INFORMATIKA DAN TEKNOLOGI KOMPUTER ANALISIS KEAMANAN WEB NEW KUTA GOLF MENGGUNAKAN METODE VULNERABILITY ASSESSMENTS DAN PERHITUNGAN SECURITY METRIKS," vol. 2, no. 3, pp. 256–265, 2022.

[9] I. Riadi, A. Yudhana, and P. Korspondensi, "ANALISIS KEAMANAN WEBSITE OPEN JOURNAL SYSTEM MENGGUNAKAN METODE VULNERABILITY ASSESSMENT," vol. 7, no. 4, 2020, doi: 10.25126/jtiik.202071928.

[10] T. Astriani, A. Budiyono, and A. Widjajarto, "Analisa Kerentanan Pada Vulnerable Docker Menggunakan Scanner Openvas Dan Docker Scan Dengan Acuan Standar NIST 800-115," vol. 8, no. 4, 2021, [Online]. Available: http://jurnal.mdp.ac.id

[11] P. Harahap and I. Zufria, "Analisis Keamanan Pada Website UPM SAINTEK UIN-SU Medan Menggunakan Metode Vulnerability Assesment," *FEBRUARI*, vol. 2, no. 1, pp. 10–20, 2024, doi: 10.55537/cosmic.