Vol. 4, No. 3, Desember 2023, hal. 743-753

Jurnal Computer Science and Information Technology (CoSciTech)

p-ISSN: 2723-567X
e-ISSN: 2723-5661

http://ejurnal.umri.ac.id/index.php/coscitech/index

Implementasi Penilaian Keamanan *Website Komprehensif* Pada Situs SMA Negeri 1 Surade

Salman Alhidamkara*[1], Ivana Lucia Kharisma[2],Kamdan[3]
Email: *[1]salman.alhidamkara_ti20@nusaputra.ac.id, [2]ivana.lucia@nusaputra.ac.id,[3]kamdan@nusaputra.ac.id

[1]Program Studi Teknik Informatika, Fakultas Teknik, Komputer dan Desain, Universitas Nusa Putra
[2]Program Studi Teknik Informatika, Fakultas Teknik, Komputer dan Desain, Universitas Nusa Putra
[3]Program Studi Teknik Informatika, Fakultas Teknik, Komputer dan Desain, Universitas Nusa Putra

Diterima: 11 Desember 2023 | Direvisi: - | Disetujui: 1 Januari 2024

Abstrak

Di era perkembangan media informasi yang pesat, pemanfaatan *website* sebagai sumber informasi telah mencapai signifikansi yang meluas di berbagai sektor. Namun, seiring peningkatan penggunaan *website*, perhatian terhadap aspek keamanan menjadi sangat penting. Urgensinya untuk menganalisis kerentanannya terbukti dalam upaya mendeteksi potensi ancaman di masa depan dan memastikan kelangsungan kerahasiaan, konsistensi, akurasi, *validitas* data, dan ketersediaan informasi dikenal sebagai Prinsip CIA (*Confidentiality, Integrity, Availability*), menjadi landasan utama dalam menjaga keamanan informasi. Situs Sekolah Menengah Atas Negeri 1 Surade menyajikan informasi umum dan data sensitif yang terkait dengan data sekolah dan siswa. Temuan dari penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam meningkatkan kesadaran tentang keamanan situs *web*, serta memberikan panduan berharga bagi lembaga pendidikan dan pemerintah. Tujuannya adalah untuk memperkuat keamanan situs *web* mereka, dengan harapan mengurangi potensi ancaman *siber* di Indonesia. Penelitian ini diharapkan dapat menjadi dasar untuk langkah-langkah *preventif* yang lebih efektif dan kebijakan keamanan informasi yang lebih ketat.

Kata kunci : *keamanan,website, CIA ,sman 1 surade*

Implementation of a Comprehensive Website Security Assessment on the Surade 1 Public High School Site

Abstract

In the era of rapid development in media information, the utilization of websites as sources of information has gained widespread significance across various sectors. However, with the increasing use of websites, attention to security aspects has become crucial. The urgency to analyze its vulnerabilities has proven essential in efforts to detect potential threats in the future and ensure the continuity of confidentiality, consistency, accuracy, data validity, and information availability—known as the CIA Principles (Confidentiality, Integrity, Availability)—forming the fundamental basis for safeguarding information security.

The website of Public High School 1 Surade presents both general and sensitive data related to school and student information. The findings of this research are expected to make a significant contribution in raising awareness about website security, offering valuable guidance to educational institutions and the government. The aim is to strengthen the security of their websites, with the hope of reducing potential cyber threats in Indonesia. This research is anticipated to serve as a foundation for more effective preventive measures and stricter information security policies.

Keywords : *security, website, CIA, sman 1 surade*

1. PENDAHULUAN

Dalam era perkembangan media informasi yang cepat, pemanfaatan *website* sebagai sumber informasi telah menjadi sangat penting dan tersebar luas di berbagai sektor [1]. Seiring dengan meningkatnya penggunaan *website*, keamanan menjadi perhatian yang signifikan, sehingga analisis celah keamanan diperlukan untuk mendeteksi potensi ancaman di masa depan. Hal ini bertujuan untuk memastikan kerahasiaan, konsistensi, akurasi, validitas data, dan ketersediaan informasi, yang merupakan prinsip utama dalam menjaga keamanan informasi, dikenal sebagai Prinsip CIA Triad [2]. Penggunaan teknologi informasi akan memberikan hasil yang efektif dengan menerapkan tata kelola yang baik [3]. Di berbagai bidang teknologi dan informasi yang berkembang pesat, baik pada *software* maupun *hardware*, termasuk dalam bidang aplikasi *web* yang membantu dunia kerja, terjadi penyampaian dan penerimaan informasi, serta kemudahan dan kecepatan pengiriman dari berbagai tempat [4].

Pengujian penetrasi adalah proses krusial untuk mengevaluasi keamanan sistem komputer dengan mensimulasikan serangan dari sumber yang tidak sah, seperti peretasan dan *jailbreaking* [2]. Keamanan sistem komputer melibatkan aspek teknis, manajerial, legalitas, dan politis [2]. Aplikasi berbasis *website* sering kali menjadi target peretasan dan pembobolan sistem karena pertumbuhan penggunaannya yang pesat [5].

Salah satu lembaga yang memiliki *website* adalah Sekolah Menengah Atas Negeri 1 Surade di Indonesia. Website ini digunakan untuk menyampaikan informasi kepada orang tua siswa, murid, dan staf sekolah. Keamanan *website* menjadi sangat penting karena menyimpan informasi sensitif seperti data pribadi siswa, jadwal pelajaran, catatan akademik, dan data penting lainnya. Ketidakamanan *website* dapat menimbulkan risiko pencurian data, penyebaran informasi palsu, atau gangguan operasional sekolah.

Penelitian menggunakan metode *Footprinting* dan *Vulnerability Scanning* pada *website* kampus telah mengungkapkan beberapa kerentanannya dengan tingkat risiko dari sedang hingga rendah [6]. Pada tahun 2022, penggunaan *Acunetix Vulnerability Scanner* pada *website* ITP menunjukkan beberapa celah keamanan dengan tingkat ancaman pada level 3 atau tinggi, termasuk ancaman seperti *SQL injection*, *cross-site scripting (XSS)*, *file inclusion*, dan serangan *brute force* pada halaman login [7]. Pada tahun 2018, penelitian menggunakan metode *Penetration Testing (Pentest)* pada *website* SMA Negeri 2 Sumbawa Besar juga mengungkapkan beberapa celah keamanan dengan tingkat risiko sedang dan rendah, termasuk 13 sub file *vulnerability* yang dapat dimanfaatkan oleh penyerang [8].

Aktivitas peretasan komputer melibatkan pelanggaran privasi dan keamanan jaringan pada berbagai tingkatan. Dampak dari aktivitas peretasan tersebut dapat bervariasi, mulai dari eksplorasi keamanan hingga kegiatan ilegal yang merusak atau bahkan menghapus file, situs *web*, atau perangkat lunak. Beberapa perusahaan besar juga menyewa tim peretas untuk mengeksplorasi celah keamanan dalam jaringan mereka. Teknik *footprinting* melibatkan pengumpulan data atau informasi yang berhubungan dengan target penelitian, dan untuk tujuan ini, beberapa aplikasi seperti *CMD (Command Prompt)*, *Zenmap*, dan *whois* domain dapat digunakan. *Vulnerability Scanning* adalah proses yang digunakan untuk mengidentifikasi kerentanan dalam jaringan [6]. *Penetration testing*, pada sisi lain, adalah bentuk simulasi yang digunakan untuk menguji dan mengevaluasi tingkat keamanan suatu sistem [9]. *ISSAF (Information System Security Assessment Framework)* adalah suatu teknik evaluasi keamanan yang digunakan untuk mengevaluasi jaringan, sistem, dan aplikasi. [10]

2. METODE PENELITIAN

Penelitian ini mengadopsi metode Evaluasi Keamanan *Website Komprehensif* untuk menyelidiki aspek keamanan pada situs *web* Sekolah Menengah Atas Negeri 1 Surade (smanegeri1surade.sch.id). Pendekatan ini diterapkan dengan tujuan melakukan penilaian menyeluruh terhadap keamanan situs *web*, melibatkan serangkaian pengujian yang dirancang khusus. Penelitian ini bertujuan untuk mengidentifikasi serta mengevaluasi potensi kerentanan dan kelemahan keamanan yang mungkin terdapat pada situs *web* tersebut, dengan fokus pada keamanan menyeluruh dari berbagai aspek yang relevan.

2.1. Evaluasi Kebutuhan

Dalam rangka menjalankan penelitian ini, perlu dipastikan bahwa semua perangkat dan alat yang diperlukan dapat memenuhi persyaratan yang ditetapkan. Spesifikasi perangkat yang esensial untuk pelaksanaan penelitian dapat diidentifikasi melalui Tabel 1 Laptop yang digunakan harus memiliki konfigurasi RAM setidaknya 4 GB, kapasitas SSD minimal 120 GB, menjalankan sistem operasi Windows 10, dan memiliki koneksi internet dengan kecepatan sekitar 30 mbps.Memastikan bahwa perangkat keras dan perangkat lunak yang digunakan sesuai dengan standar keamanan dan dapat beroperasi secara harmonis akan mendukung kesuksesan pelaksanaan penelitian ini. Oleh karena itu, pemilihan perangkat dan alat harus mencakup tidak hanya spesifikasi teknis, tetapi juga faktor-faktor keamanan dan kompatibilitas untuk meminimalkan risiko dan memastikan integritas data penelitian.

Tabel 1.Kebutuhan Hardware

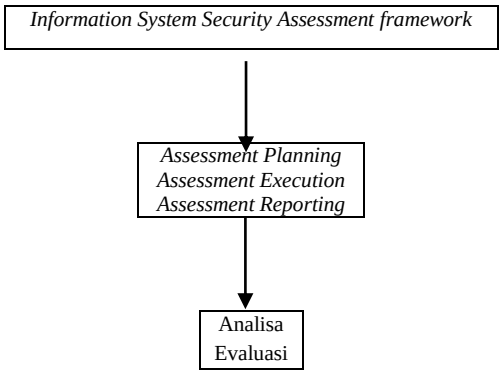
No	Perangkat	Spesifikasi
1	Laptop / PC	-Lenovo ideapad S145 - Ram 20GB - SSD 2 TB
2	Sistem Operasi	Kali Linux
3	*Penetration Tool*	*Metalsploit*
4	*Vulnerabilities Tool*	*OWAS ZAP*
6	*Network*	Indihome Sukabumi 30 mbps

Ketentuan ini dirancang untuk memastikan bahwa penelitian dapat dilakukan secara efisien dan tanpa hambatan teknis yang tidak perlu. Untuk mendapatkan gambaran yang lebih lengkap tentang peralatan yang akan digunakan, informasi tambahan dapat dicari di Tabel 2, yang memberikan daftar rinci alat-alat yang relevan dengan penelitian ini. Hal ini penting agar penelitian dapat berjalan dengan lancar dan memastikan bahwa semua aspek teknis dan peralatan yang diperlukan telah dipertimbangkan dengan cermat. Selain itu, dalam pemilihan perangkat keras, pertimbangan keamanan dan kompatibilitas perangkat juga harus diperhatikan.

Tabel 2. Alat yang dimanfaatkan

No	Perangkat	Spesifikasi
1	*Phase 1 Assessment Planning*	*Nslookup cmd*
2	*Phase 2 Assessment Execution*	*Metalsploit*
3	*Phase 3 Assessment Reporting*	*Manual*

2.2 Skenario Pengujian



Gambar 1. Alur Pengujian

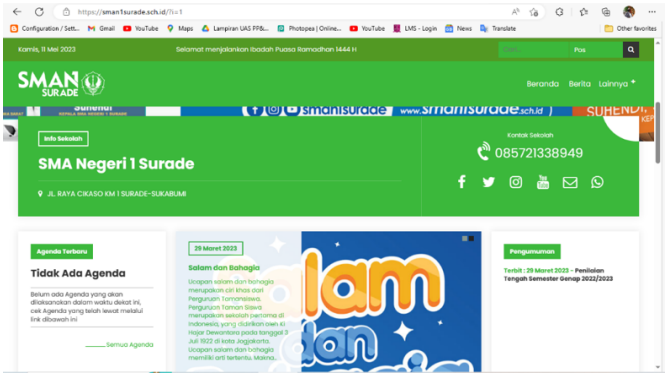
3 Phase *Information System Security Assessment framework* adalah sebagai berikut :

a. Phase 1 *Information Gathering*

Pada fase ini, peneliti menggunakan sumber daya internet untuk mengumpulkan sebanyak mungkin informasi terkait dengan suatu situs *web*. Pendekatan yang digunakan mencakup metode teknis seperti *nslookup*, serta metode non-teknis melalui mesin pencari, daftar alamat email, dan sumber daya lainnya. Kegiatan *Information Gathering* ini tidak melibatkan interaksi langsung dengan sistem target; sebaliknya, peneliti bergantung pada sumber informasi publik seperti internet dan dokumen yang dapat diakses dari entitas-entitas yang menyediakan informasi publik, termasuk perpustakaan dan sumber daya sejenis.

Dalam langkah-langkah yang dilakukan pada fase ini, peneliti melakukan pencarian informasi terkait dengan suatu situs web dari berbagai sumber, termasuk *website* sekolah, forum, dan mesin pencari. Tujuannya adalah untuk mengidentifikasi *host* yang digunakan oleh situs *web* tersebut, serta upaya untuk memahami informasi tentang sistem operasi yang digunakan, aplikasi *web server*, dan bahasa pemrograman yang diterapkan.

Langkah selanjutnya adalah perakitan pemetaan jaringan untuk mengidentifikasi *port* yang terbuka dan layanan yang aktif pada setiap *host*. Sebagai contoh, untuk mendapatkan informasi tentang situs *web* Sekolah Menengah Negeri 1 Surade, data dapat dikumpulkan melalui kunjungan ke situs *web* resmi sekolah di *https://smanegeri1surade.sch.id/*. Tangkapan layar dari tampilan situs *web* tersebut dapat memberikan gambaran lebih lanjut tentang struktur dan konten yang ada.



Gambar 2. Tabel *website* SMA Negeri 1 Surade

Dari situs *web* tersebut, peneliti berhasil menghimpun beragam informasi, termasuk namun tidak terbatas pada profil sekolah, struktur organisasi, jadwal kegiatan, galeri foto, dan berita terbaru. Selain itu, rincian kontak sekolah, seperti alamat, nomor telepon, dan alamat email, juga berhasil diidentifikasi.

Pada fase ini, peneliti menjalankan proses pengumpulan informasi terkait dengan situs *web* Sekolah Menengah Atas Negeri 1 Surade dengan mengakses berbagai sumber yang tersedia di internet. Seluruh data yang berhasil diperoleh kemudian dikelompokkan dan disusun secara teratur dalam sebuah tabel.

Tabel 3. Informasi *website*

Informasi Website	Detail
URL *website*	http://sman1surade.sch.id/
Tanggal pembuatan *website*	Tidak diketahui
Bahasa pemrograman yang digunakan	PHP, JavaScript, HTML
Sistem operasi server	Linux
Server web	*Apache*
Versi CMS yang digunakan	Joomla! 3.9.26
Daftar halaman *website*	Beranda, Profil, Berita, Guru, Siswa, Fasilitas, Kontak Kami
Daftar *plugin* yang digunakan	*Akeeba Backup Core, JCE Editor, JCH Optimize, Joomla! Update Notification, RSFirewall!, Widgetkit*

b. *Network Mapping*

Setelah berhasil menghimpun informasi pada tahap *Information Gathering*, langkah selanjutnya adalah menjalankan proses *network mapping*. Tujuan dari *network mapping* adalah untuk mengungkap struktur dan konfigurasi jaringan yang terhubung dengan target. Namun, perlu diingat bahwa kegiatan *network mapping* memiliki potensi merugikan keamanan sistem target, sehingga harus dijalankan dengan penuh kehati-hatian dan tanggung jawab. Peneliti harus memastikan bahwa tindakan *network mapping* yang diimplementasikan tidak menyebabkan kerusakan atau gangguan terhadap kinerja sistem target.

Peneliti menggunakan alat seperti *Nmap* untuk melakukan pemindaian *port* di dalam jaringan yang terkoneksi dengan target. Hasil dari pemindaian ini memungkinkan peneliti untuk mengidentifikasi jenis layanan yang sedang berjalan di setiap *port* dan membentuk peta jaringan (*network map*) yang mengilustrasikan bagaimana semua komponen jaringan target terkoneksi satu sama lain. Sebelum menjalankan network mapping, peneliti juga harus memastikan untuk memperoleh izin resmi dari pihak berwenang.

2. Phase 2 (penilaian) melibatkan dua tahap, yaitu pemindaian kerentanan dan pengujian penetrasi. Berikut ini adalah langkah-langkah yang dilakukan pada Phase 2 (penilaian):

a. Melakukan pemindaian kerentanan pada *website* menggunakan alat seperti *Nmap*. Tujuan dari pemindaian ini adalah untuk mengidentifikasi potensi kerentanan keamanan yang mungkin ada di dalam *website*.

b. Melaksanakan pengujian penetrasi dan *enumerate* lebih lanjut pada *website* untuk menguji kekuatan sistem keamanan *website*. Selain itu, tujuan dari tahap ini adalah untuk mencari informasi seperti *username* dan *password* yang dapat digunakan dalam evaluasi keamanan *website*.

3. Phase 3 (*Report*) melibatkan pembuatan laporan hasil penilaian. Langkah-langkah yang terlibat dalam phase 3 adalah:

a. *Report*

Setelah mengidentifikasi ancaman atau potensi serangan pada *website*, langkah berikutnya adalah menyusun laporan dan memberikan rekomendasi terkait tindakan pengamanan yang perlu diambil untuk meningkatkan keamanan *website* berdasarkan hasil analisis tersebut. [8]

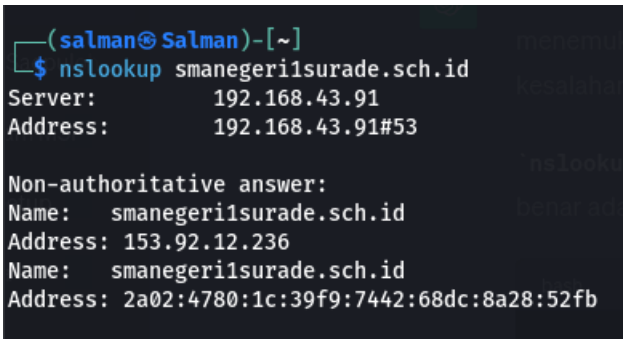
3. HASIL DAN PEMBAHASAN

3.1 Hasil 3 *Phase Information System Security Assessment* :

Pada Phase pertama terdapat *Information Gathering* dan *Network Mapping*, dan di Phase kedua terdapat *Assessment*, dengan Phase terakhir yaitu *report*. Berikut adalah hasil-hasil utama dari setiap Phase ini:

1. Phase 1: *Information Gathering* dan *Network Mapping*

- Hasil *Information Gathering* :



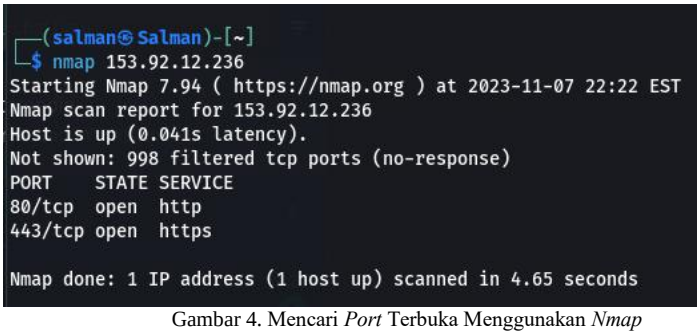
Gambar 3. Mencari Alamat IP Mengunakan *nslookup* di *cmd*

Pada gambar 3 menunjukkan bahwa domain "*smanegeri1surade.sch.id*" dapat diakses melalui dua alamat IP, satu untuk protokol *IPv4* (alamat IP versi 4) dan satu untuk protokol *IPv6* (alamat IP versi 6). Penyedia layanan internet dan *server web* dapat menggunakan dua jenis protokol ini untuk mengakses situs *web* tersebut, tergantung pada dukungan yang mereka miliki. Jadi, hasil dari perintah nslookup ini adalah mengidentifikasi alamat IP yang terkait dengan nama domain yang dicari, serta menunjukkan bahwa hasilnya bukan otoritatif (dari sumber utama yang memiliki otoritas DNS atas domain tersebut). Melalui penggunaan *nslookup*, Alamat IP dari *website* SMA Negeri 1 Surade berhasil diidentifikasi. Dua alamat IP terkait dengan domain "smanegeri1surade.sch.id" ditemukan, yaitu alamat IP versi 4 (IPv4) dan alamat IP versi 6 (IPv6).

- Hasil *Network Mapping* :

Informasi jaringan diperoleh melalui pemindaian menggunakan alat seperti *Nmap*. Pemindaian ini memungkinkan identifikasi jenis layanan yang berjalan di setiap port dan menyusun peta jaringan (*network map*) yang memvisualisasikan koneksi antar komponen jaringan target.

Setelah kita mendapatkan IP dari *website* tersebut langkah selanjutnya adalah dengan melihat *Port* yang terbuka menggunakan *Nmap*.



Gambar 4. Mencari *Port* Terbuka Menggunakan *Nmap*

Hasil yang tampilkan pada gambar 4 adalah output dari perintah *`nmap`*, yang digunakan untuk melakukan pemindaian port dan mencari tahu layanan yang berjalan pada alamat IP yang dituju, dalam hal ini, alamat IP *153.92.12.236*. Berikut adalah rincian dari hasil tersebut :

Starting Nmap 7.94 (https://nmap.org) pada 2023-11-07 22:22 EST

• Pada awal pemindaian, tercatat versi Nmap yang digunakan, bersama dengan tanggal dan waktu dimulainya pemindaian. *Nmap*, sebagai alat pemindaian jaringan *open-source*, berfungsi untuk mengidentifikasi *port* yang terbuka dan layanan yang beroperasi pada alamat IP yang dituju.

Nmap scan report for 153.92.12.236

• Pada laporan pemindaian untuk alamat IP 153.92.12.236, mengindikasikan bahwa *Nmap* sedang mencari informasi terkait dengan alamat IP tersebut.

Host is up (0.041s latency)

• Pada fase ini, terlihat bahwa *host* dengan alamat IP 153.92.12.236 sedang aktif (*host is up*) dan memiliki waktu *latency* sekitar 0.041 detik, menunjukkan respons yang cepat terhadap permintaan.

Not shown : 998 filtered tcp ports (no-response)

• Pada keterangan mengenai *port-port* yang tidak terlihat dalam hasil pemindaian. Informasi ini mengindikasikan bahwa ada 998 *port* TCP yang difilter (*filtered*), yang berarti *Nmap* tidak menerima respons apapun dari *port-port* tersebut. Oleh karena itu, tidak dapat ditentukan apakah *port-port* tersebut terbuka atau tertutup.

PORT STATE SERVICE

80/tcp open http

443/tcp open https

• Pada catatan ini adalah informasi tentang dua *port* yang berhasil diidentifikasi: *Port* 80 (*http*) dalam keadaan terbuka (*open*), menunjukkan bahwa layanan *web http* berjalan pada *port* ini. *Port* 443 (*https*) juga dalam keadaan terbuka, menunjukkan bahwa layanan *web https* berjalan pada port ini. *Port* 443 sering digunakan untuk koneksi web yang aman menggunakan *enkripsi SSL/TLS.Nmap done : 1*

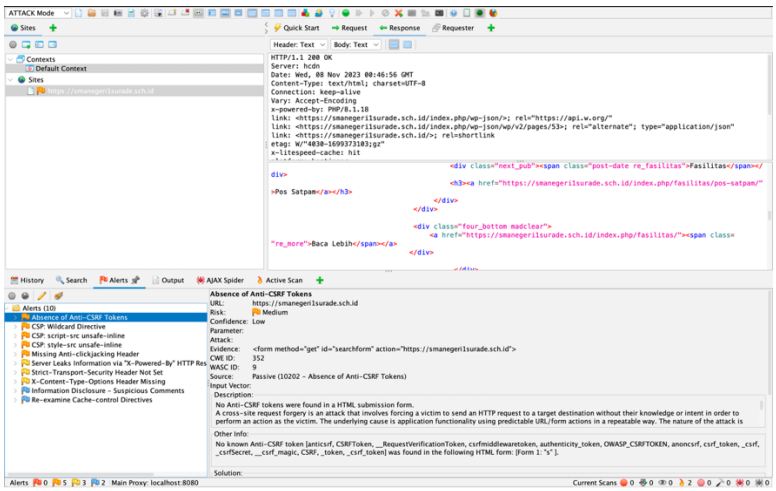
Nmap done: 1 IP address (1 host up) scanned in 4.65 seconds

• Pada akhir laporan, tercatat bahwa pemindaian telah selesai. *Nmap* berhasil mendeteksi 1 alamat IP yang aktif (*1 host up*) dan menyelesaikan pemindaian dalam waktu 4.65 detik. Dengan demikian, hasil pemindaian *Nmap* menunjukkan bahwa *host* dengan alamat IP 153.92.12.236 memiliki dua port terbuka, yaitu port 80 (*http*) dan port 443 (*https*), menandakan bahwa *host* tersebut menjalankan layanan web.

2. *Phase 2 : Assessment*

Hasil Pemindaian Kerentanan:

Proses pemindaian kerentanan menggunakan *Nmap* dilaksanakan untuk mengidentifikasi potensi kerentanan keamanan pada *website*. Hasil pemindaian memberikan wawasan tentang kerentanan yang mungkin ada di dalam *website*. Dalam langkah ini, peneliti menjalankan proses pemindaian kerentanan pada situs *web* menggunakan alat *OWASP ZAP*. Tujuannya adalah untuk mengidentifikasi potensi kerentanan keamanan yang mungkin ada dalam situs web tersebut.



Gambar 5. Pemindaian kerentanan menggunakan OWAS ZAP

Dari Gambar 5 diatas diuraikan penjelasan mengenai sepuluh kerentanan yang telah disebutkan beserta langkah-langkah pemecahannya:

1. *Absence of Anti-CSRF Tokens*

Kerentanan ini adalah kelemahan di mana aplikasi web tidak menggunakan *token anti-CSRF* (*Cross-Site Request Forgery*) untuk melindungi permintaan yang sensitif dari serangan *CSRF*. *CSRF* terjadi ketika serangan memaksa pengguna yang terautentikasi melakukan tindakan tanpa persetujuan mereka.

2. *CSP : Wildcard Directive*

Kerentanan ini adalah kerentanan terkait Kebijakan Keamanan Konten (*Content Security Policy/CSP*) di mana direktif *wildcard* (*) digunakan, memungkinkan semua sumber skrip internal dimuat. Penggunaan wildcard dapat meningkatkan risiko eksekusi skrip yang tidak aman.

3. *CSP: style-src unsafe-inline*

Kerentanan ini merujuk pada kerentanan di *CSP* di mana gaya internal (*inline*) yang tidak aman diizinkan. Hal ini meningkatkan risiko eksekusi gaya yang tidak aman dan berpotensi menjadi sasaran serangan *Cross-Site Scripting* (*XSS*).

4. *Missing Anti-clickjacking Header*

Kerentanan ini muncul ketika aplikasi *web* tidak mengatur *header Anti-clickjacking*, *seperti X-Frame-Options*, untuk melindungi halaman *web* dari serangan *clickjacking*. *Clickjacking* terjadi ketika penyerang menipu pengguna untuk melakukan tindakan tanpa sepengetahuan mereka.

5. *Server Leaks Information via "X-Powered-By" http Response Header Field(s)*

Kerentanan ini terjadi ketika *server* mengungkapkan informasi sensitif, seperti teknologi *server* yang digunakan, melalui *header respons http X-Powered-By*. Informasi ini dapat digunakan oleh penyerang untuk merancang serangan yang lebih terfokus.

6. *Strict-Transport-Security Header Not Set*

Kerentanan ini muncul ketika aplikasi *web* tidak mengatur *header Strict-Transport-Security*, yang memastikan komunikasi hanya melalui protokol *https* yang aman. Tanpa *heade*r ini, ada risiko pemotongan *SSL/TLS* dan potensi serangan *man-in-the-middle.*

7. *X-Content-Type-Options Header Missing*

Kerentanan ini terkait dengan ketidak setelan *header X-Content-Type-Options*, yang mengontrol perilaku penentuan jenis konten oleh *browser*. Tanpanya, risiko penentuan jenis konten yang tidak aman meningkat, memungkinkan serangan terhadap kelemahan jenis konten. Kerentanan ini terjadi ketika aplikasi *web* tidak mengatur *header X-Content-Type-Options* yang mengontrol perilaku penentuan jenis konten oleh *browser*.

Berikut adalah penjelasan hasil pemindaian menggunakan *OWASP ZAP* untuk setiap kerentanan yang terdeteksi, beserta solusi pemecahannya :

1. *Absence of Anti-CSRF Tokens*

Aplikasi web yang tidak menggunakan token *anti-CSRF* untuk melindungi permintaan yang sensitif dari serangan *CSRF* dapat memicu potensi serangan *CSRF* yang mengakibatkan aksi tidak sah atas nama pengguna yang terautentikasi jika tidak diperbaiki dengan menyertakan token *anti-CSRF* dalam setiap permintaan yang mengubah keadaan dan dampaknya adalah kerentanan terhadap serangan yang merugikan. Berikut adalah implementasi solusinya ;

```
<!-- Membuat token anti-CSRF pada halaman -->

<form action="/submit" method="post">
    <input type="hidden" name="csrf_token" value="{{ generate_csrf_token() }}">
    <!-- ... elemen-elemen formulir lainnya ... -->
    <button type="submit">Submit</button>
</form>

<!-- Memeriksa token pada sisi server ( menggunakan Python dan Flask) -->

def submit_form():
    if request.method == 'POST':
        csrf_token = request.form.get('csrf_token')
        if csrf_token != session.get('csrf_token'):
            abort(403)  # Token CSRF tidak valid
        # Proses formulir
```

2. *CSP : Wildcard Directive*

Kebijakan Keamanan Konten (*CSP*) yang menggunakan direktif wildcard (*) meningkatkan risiko eksekusi skrip yang tidak aman. Solusinya adalah menghapus *wildcard* dan menggantinya dengan daftar sumber yang diizinkan secara eksplisit, sehingga mengurangi dampak risiko tersebut. Berikut adalah implementasi solusinya ;

```
<!-- Menerapkan kebijakan CSP dengan daftar sumber yang diizinkan -->
<meta http-equiv="Content-Security-Policy" content="default-src 'self' https://trusted-source.com; script-src 'self' https://trusted-scripts.com">
```

3. *CSP : style-src unsafe-inline*

Kebijakan Keamanan Konten (*CSP*) yang mengizinkan gaya internal (*inline*) yang tidak aman dapat meningkatkan risiko eksekusi gaya yang tidak aman dan potensi serangan *XSS*. Solusinya adalah menghilangkan gaya internal yang tidak aman dan menggunakan gaya eksternal yang aman, sehingga mengurangi risiko tersebut. Berikut adalah implementasi solusinya ;

```
<!-- Menghilangkan gaya internal yang tidak aman -->
<meta http-equiv="Content-Security-Policy" content="style-src 'self' https://trusted-styles.com">
```

4. *Missing Anti-clickjacking Header*

Aplikasi web yang tidak mengatur *header Anti-clickjacking* dapat meningkatkan risiko serangan *clickjacking* dan solusinya adalah mengatur *header X-Frame-Options* dengan nilai *"DENY"* atau *"SAMEORIGIN"* untuk melindungi halaman *web*, mengurangi dampak risiko tersebut. Berikut adalah implementasi solusinya ;

```
<!-- Mengatur header X-Frame-Options -->
<meta http-equiv="X-Frame-Options" content="DENY">
```

5. *Server Leaks Information via "X-Powered-By" http Response Header Field(s)*

Server yang mengungkapkan informasi sensitif melalui header respons *http X-Powered-By* dapat memungkinkan pengumpulan informasi oleh penyerang dan solusinya adalah menghapus atau mengganti nilai *header X-Powered-By* dengan informasi yang lebih umum untuk mengurangi dampak potensial. Berikut adalah implementasi solusinya ;

```
# Menghapus atau mengganti nilai header X-Powered-By di server Nginx
server_tokens off;
```

6. *Strict-Transport-Security Header Not Set*

Aplikasi *web* yang tidak mengatur *header Strict-Transport-Security* meningkatkan risiko pemotongan *SSL/TLS* dan potensialnya serangan *man-in-the-middle*; solusinya adalah mengatur *header Strict-Transport-Security* untuk memastikan komunikasi hanya melalui protokol *https* yang aman, mengurangi dampak risiko tersebut. Berikut adalah implementasi solusinya ;

```
# Mengatur header Strict-Transport-Security di server Nginx
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

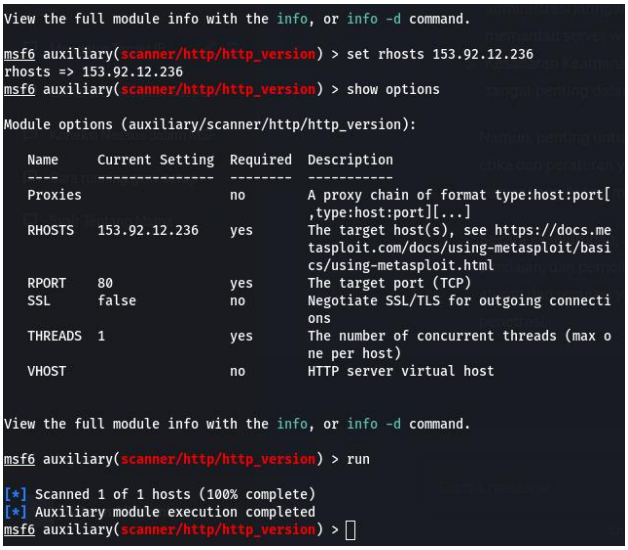
7. *X-Content-Type-Options Header Missing*

Aplikasi web yang tidak mengatur header *X-Content-Type-Options* meningkatkan risiko penentuan jenis konten yang tidak aman; solusinya adalah mengatur header *X-Content-Type-Options* dengan nilai *"nosniff"* untuk mengontrol perilaku penentuan jenis konten oleh browser, mengurangi dampak risiko tersebut.

```
# Mengatur header X-Content-Type-Options di server Nginx
add_header X-Content-Type-Options "nosniff" always;
```

- Hasil Pengujian Penetrasi:

Pengujian penetrasi dan enumerasi lebih lanjut dijalankan untuk menguji kekuatan sistem keamanan *website*. Pengujian ini juga mencakup pencarian informasi seperti *username* dan *password* untuk evaluasi keamanan *website*.



Gambar 13. Hasil Penetrasi dengan *Metalsploit*

Hasil pemindaian menunjukkan bahwa saya telah memindai 1 dari 1 *host* (100% selesai) dan eksekusi modul *auxiliary* selesai.

Tujuan dari tindakan ini adalah untuk melakukan deteksi versi protokol *http* pada target yang ditentukan. Dengan informasi ini, Anda dapat memahami versi protokol yang digunakan oleh *server web* target, yang dapat membantu dalam memahami potensi kerentanan atau konfigurasi keamanan.

Sebagai penilaian, Anda telah berhasil menjalankan modul *Metasploit* sesuai dengan tujuannya, yaitu untuk melakukan deteksi versi protokol *http* pada target tertentu. Namun, sebagai praktik terbaik, pastikan Anda selalu menjalankan tindakan semacam ini hanya pada sistem yang Anda miliki izin atau yang Anda administrasikan. Etika dan legalitas harus selalu diperhatikan saat menggunakan alat-alat seperti *Metasploit*.

3. Phase 3 : *Report*

- Hasil Laporan:

Setelah mengidentifikasi ancaman dan potensi serangan pada *website*, laporan hasil penilaian disusun. Laporan ini mencakup rekomendasi tindakan pengamanan yang perlu diambil untuk meningkatkan keamanan *website* berdasarkan hasil analisis dari

setiap Phase sebelumnya. Laporan ini menjadi dasar untuk langkah-langkah selanjutnya dalam upaya meningkatkan keamanan sistem atau *website* yang dituju.

1. *Report Information Gathering*

Langkah 1: Menggunakan *nslookup*

Dalam langkah ini, tujuan utama adalah untuk menemukan alamat IP dari domain "*smanegeri1surade.sch.id*" menggunakan perintah *nslookup* di *cmd*. Hasil dari langkah ini adalah:

Domain "*smanegeri1surade.sch.id*" dapat diakses melalui dua alamat IP, yaitu alamat IP versi 4 (IPv4) dan alamat IP versi 6 (IPv6). Hal ini menunjukkan bahwa penyedia layanan internet dan server web dapat menggunakan dua jenis protokol ini untuk mengakses situs *web* tersebut, tergantung pada dukungan yang mereka miliki.

Hasil dari perintah *nslookup* ini adalah *non-otoritatif*, yang berarti hasilnya bukan dari sumber utama yang memiliki otoritas DNS atas domain tersebut.

Langkah 2 : *Network Mapping* dengan *Nmap*

Setelah mendapatkan alamat IP dari *website* tersebut, langkah selanjutnya adalah mencari *port* yang terbuka menggunakan *Nmap*. Hasil dari pemindaian ini adalah sebagai berikut:

- Pemindaian dimulai dengan menggunakan *Nmap* versi 7.94 pada tanggal dan waktu tertentu.
- Laporan pemindaian menunjukkan hasil dari pemindaian terhadap alamat IP 153.92.12.236. Pemindaian ini bertujuan untuk mengidentifikasi *port* yang terbuka dan layanan yang berjalan di alamat IP tersebut.
- Hasil pemindaian menunjukkan bahwa *host* dengan alamat IP 153.92.12.236 aktif dengan waktu *latency* sekitar 0.041 detik, menunjukkan respons yang cepat terhadap permintaan.
- Ada 998 *port* TCP yang tidak ditampilkan dalam hasil pemindaian karena mereka difilter dan tidak mendapatkan respons, sehingga tidak dapat ditentukan apakah *port-port* tersebut terbuka atau tertutup.
- Ditemukan dua *port* terbuka : *Port* 80 (*http*) dalam keadaan terbuka, menunjukkan bahwa layanan *web http* berjalan pada port ini. *Port* 443 (*https*) juga dalam keadaan terbuka, menunjukkan bahwa layanan *web https* berjalan pada port ini. Port 443 sering digunakan untuk koneksi web yang aman yang menggunakan enkripsi SSL/TLS.
- Laporan pemindaian mencatat bahwa pemindaian selesai dalam waktu 4.65 detik.

Assessment (Penilaian)

Langkah 1: Pemindaian Kerentanan dengan *OWASP ZAP*

Pada tahap *Assessment*, dilakukan proses pemindaian kerentanan pada *website* menggunakan *OWASP ZAP*. Hasil pemindaian menunjukkan identifikasi 10 kerentanan potensial beserta solusinya:

- *Absence of Anti-CSRF Tokens* : Aplikasi web tidak menggunakan token *anti-CSRF*. Solusinya adalah menyertakan *token anti-CSRF* dalam setiap permintaan yang sensitif.
- *CSP : Wildcard Directive* : Kebijakan Keamanan Konten (*CSP*) menggunakan direktif wildcard (*), yang memungkinkan semua sumber eksternal dimuat. Solusinya adalah mengganti wildcard dengan daftar sumber yang diizinkan secara eksplisit.
- *CSP : script-src unsafe-inline* : Kebijakan Keamanan Konten (*CSP*) mengizinkan penggunaan skrip internal (*inline*) yang tidak aman. Solusinya adalah menghilangkan penggunaan skrip internal yang tidak aman dan menggunakan skrip eksternal yang aman.
- *CSP : style-src unsafe-inline* : Kebijakan Keamanan Konten (*CSP*) mengizinkan penggunaan gaya internal (*inline*) yang tidak aman. Solusinya adalah menghilangkan penggunaan gaya internal yang tidak aman dan menggunakan gaya eksternal yang aman.
- *Missing Anti-clickjacking Header* : Aplikasi *web* tidak mengatur *header Anti-clickjacking*. Solusinya adalah mengatur *header X-Frame-Options* dengan nilai *"DENY"* atau *"SAMEORIGIN"*.
- *Server Leaks Information via "X-Powered-By" http Response Header Field(s)* : *Server* mengungkapkan informasi sensitif melalui *header respons http X-Powered-By*. Solusinya adalah menghilangkan *header X-Powered-By* atau mengganti nilainya dengan informasi yang lebih umum.
- *Strict-Transport-Security Header Not Set* : Aplikasi *web* tidak mengatur *header Strict-Transport-Security*. Solusinya adalah mengatur *header Strict-Transport-Security*.

- *X-Content-Type-Options Header Missing* : Aplikasi *web* tidak mengatur *header X-Content-Type-Options*. Solusinya adalah mengatur *header X-Content-Type-Options* dengan nilai "*nosniff*".

Langkah 2 : *Penetration Testing* (Pengujian Penetrasi)

- Dilakukan pengujian penetrasi menggunakan Metasploit dengan tujuan melakukan deteksi versi protokol *http* pada target yang ditentukan. Hasil pemindaian menunjukkan bahwa pengujian telah selesai dengan berhasil.
- Pengujian penetrasi menggunakan *Metasploit* berhasil dilakukan untuk deteksi versi protokol *http* pada target.
- Laporan ini menyediakan dokumentasi penting untuk langkah-langkah yang telah dilakukan dalam *fase Information Gathering* dan *Assessment*, serta memberikan dasar untuk tindakan selanjutnya dalam upaya mengamankan sistem atau *website* yang dituju.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan dari ketiga *fase Information System Security Assessment*, langkah-langkah yang dilakukan telah memberikan wawasan mendalam tentang keamanan *website* SMA Negeri 1 Surade.

- Pada Phase 1 : *Information Gathering* dan *Network Mapping*, informasi kritis seperti alamat IP telah berhasil diidentifikasi menggunakan perintah `*nslookup*`, dan *network mapping* dengan *Nmap* telah mengungkap struktur dan konfigurasi jaringan target. Penjelasan rinci tentang alamat IP yang terkait dengan domain "*smanegeri1surade.sch.id*" dan hasil pemindaian *Nmap* memberikan gambaran yang jelas tentang layanan yang berjalan di setiap *port*.
- Dalam Phase 2 : *Assessment*, proses pemindaian kerentanan menggunakan *OWASP ZAP* membuka beberapa kerentanan potensial pada *website*. Pemindaian ini melibatkan identifikasi dan pemecahan sepuluh kerentanan yang melibatkan aspek keamanan seperti *CSRF, CSP*, dan kebijakan keamanan lainnya. Pengujian penetrasi dengan *Metasploit* juga menciptakan pemahaman mendalam tentang kekuatan sistem keamanan *website*.
- Terakhir, Phase 3 : *Report* menghasilkan laporan *komprehensif* yang mencakup temuan dari setiap fase sebelumnya. Laporan ini tidak hanya mengidentifikasi ancaman dan potensi serangan, tetapi juga memberikan rekomendasi tindakan pengamanan untuk meningkatkan keamanan *website*.

Secara keseluruhan, 3 *Phase Information System Security Assessment* memberikan gambaran menyeluruh tentang keadaan keamanan *website* SMA Negeri 1 Surade. Langkah-langkah ini, dari pengumpulan informasi hingga penetrasi dan pembuatan laporan, memberikan dasar yang kokoh untuk mengambil tindakan yang diperlukan guna meningkatkan keamanan dan melindungi *website* dari potensi ancaman di masa depan.

DAFTAR PUSTAKA

[1] A. M. Tania *et al.*, "Copyright@2018. P2M STMIK BINA INSANI Keamanan Website Menggunakan Vulnerability Assessment," *INFORMATICS FOR EDUCATORS AND PROFESSIONALS*, vol. 2, no. 2, pp. 171–180, 2018.

[2] A. Rochman, R. Rohian Salam, dan Sandi Agus Maulana Sekolah Tinggi Manajemen Ilmu Komputer, and S. Likmi, "DI RUMAH SAKIT XYZ," *ANALISIS KEAMANAN WEBSITE DENGAN INFORMATION SYSTEM SECURITY ASSESSMENT FRAMEWORK (ISSAF) DAN OPEN WEB APPLICATION SECURITY PROJECT*, vol. 2, no. 4, 2021.

[3] E. Handoyo, "Analisis Tingkat Keamanan Informasi: Studi Komparasi Framework Cobit 5 Subdomain Manage Security Services (DSS05) dan NIST Sp 800 – 55," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 1, no. 2, pp. 76–83, Oct. 2020, doi: 10.37859/coscitech.v1i2.2199.

[4] R. Hafsari, R. Rahmadani Saputra, and M. Afin Wirdyansah, "Perancangan Absensi Berbasis Web Dengan Metode Waterfall (Studi Kasus: PT. GlobalRiau Data Solusi)," vol. 4, no. 1, pp. 306–312, 2023, doi: 10.37859/coscitech.v4i1.5400.

[5] A. Zirwan, "Pengujian dan Analisis Kemanan Website Menggunakan Acunetix Vulnerability Scanner," *Jurnal Informasi dan Teknologi*, pp. 70–75, Mar. 2022, doi: 10.37034/jidt.v4i1.190.

[6] M. Fatkhurozzi, "Seminar Nasional Informatika Bela Negara (SANTIKA) Analisa Keamanan Website Menggunakan Metode Footprinting dan Vulnerability Scanning pada Website Kampus".

[7] A. Zirwan, "Pengujian dan Analisis Kemanan Website Menggunakan Acunetix Vulnerability Scanner," *Jurnal Informasi dan Teknologi*, pp. 70–75, Mar. 2022, doi: 10.37034/jidt.v4i1.190.

[8] Y. Mulyanto, M. Taufan Asri Zaen, and S. Sihab, "Analisis Keamanan Website SMA Negeri 2 Sumbawa Besar Menggunakan Metode Penetration Testing (Pentest)," *Journal of Information System Research*, vol. 4, no. 1, pp. 202–209, 2022, doi: 10.47065/josh.v4i1.2335.

[9] Y. Thurfah Afifa Rosaliah and B. Hananto, *Pengujian Celah Keamanan Website Menggunakan Teknik Penetration Testing dan Metode OWASP TOP 10 pada Website SIM xxx*. 2021.

[10] B. Tasya Kumala Dewi and M. Andri Setiawan, "Kajian Literatur: Metode dan Tools Pengujian Celah Keamanan Aplikasi Berbasis Web."