CAK3BAB3 IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

PENETRATION TESTING DAN VULNERABILITY TESTING

ANGGOTA KELOMPOK:

1. Zahra Puspita Paramastri (2211102081)
2. Dewi Aminah Chan (2211102089)
3. Hana Setia Putri A.R (2211102155)
4. Gea Amarlinda Sassy M (2211102301)

DOSEN:

Muhamad Azrino Gustalika,S.Kom.,M.Tr.T

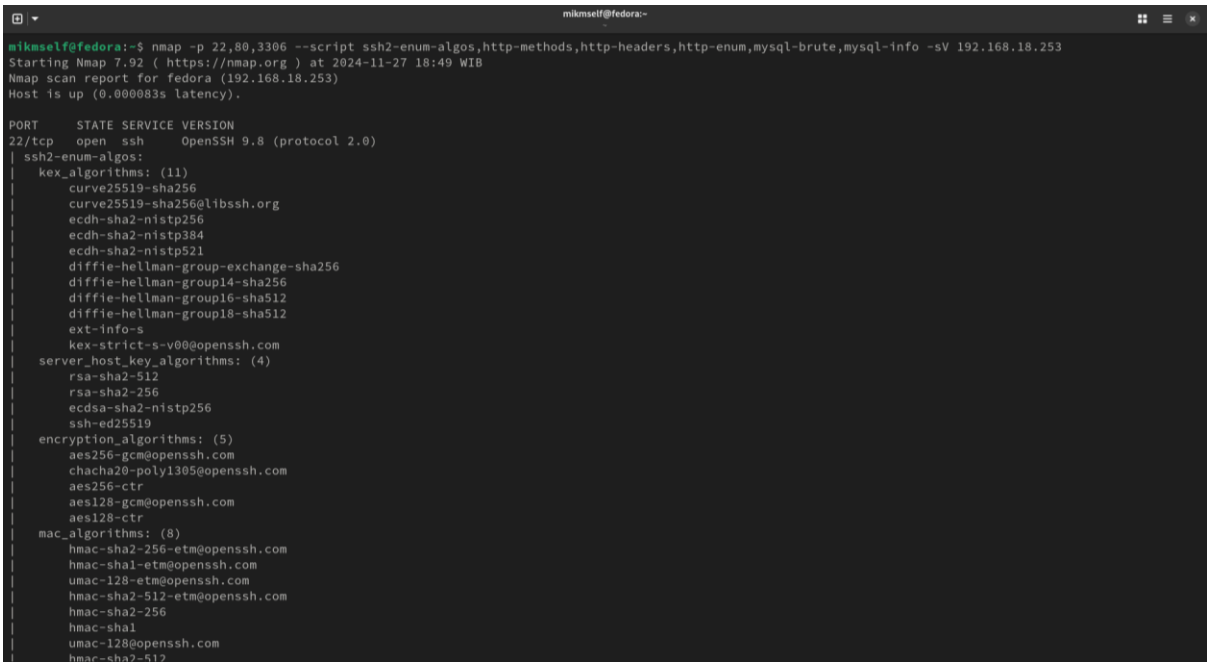
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
PURWOKERTO

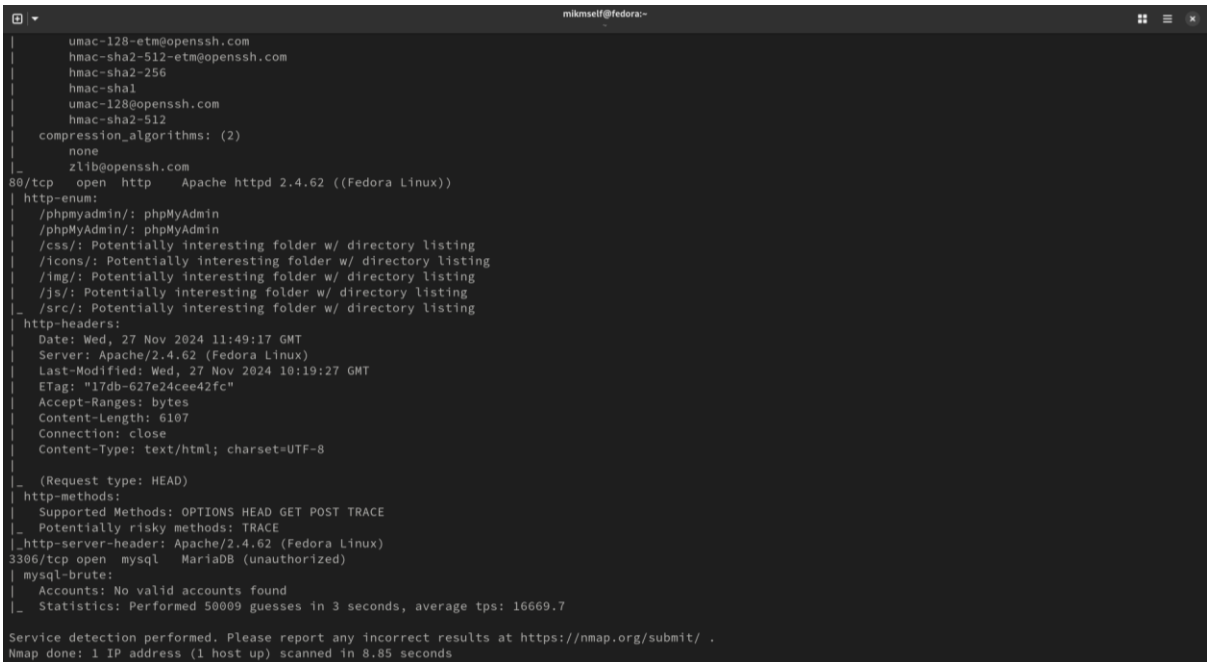
2024

Vulnerability Testing

Menggunakan IP 192.168.18.253 sebagai target. dengan toools Nmap dan Nikto. Kita akan menggunakan nmap terlebih dahulu

---





Perintah Utama:

nmap -p 22,80,3306 --script ssh2-enum-algos,http-methods,http-headers,http-enum,mysql-brute,mysql-info -sV 192.168.18.253

Perintah ini melakukan pemindaian mendalam terhadap port tertentu pada target IP **192.168.18.253** untuk mendeteksi layanan, versinya, dan potensi kerentanannya menggunakan skrip bawaan Nmap.

Detail Parameter:

1. `-p 22,80,3306`
    - **Penjelasan**: Menentukan port spesifik yang akan dipindai:
        - **Port 22**: Digunakan untuk protokol **SSH** (Secure Shell).
        - **Port 80**: Digunakan untuk layanan **HTTP** (web server).
        - **Port 3306**: Digunakan untuk protokol **MySQL/MariaDB** (database server).
    - **Fungsi**: Mempercepat proses pemindaian dengan hanya memeriksa port yang disebutkan, alih-alih memindai semua port secara default.
2. `--script ssh2-enum-algos,http-methods,http-headers,http-enum,mysql-brute,mysql-info`
    - **Penjelasan**: Menjalankan serangkaian skrip Nmap bawaan untuk memeriksa lebih dalam tentang konfigurasi dan potensi kerentanan di layanan yang terdeteksi. Berikut detail tiap skrip:
        - `ssh2-enum-algos`:
            - Mengidentifikasi algoritma enkripsi, autentikasi, dan pertukaran kunci yang didukung oleh server SSH.
            - Penting untuk mendeteksi apakah server menggunakan algoritma yang usang atau lemah.
        - `http-methods`:

        - Memeriksa metode HTTP yang didukung oleh server (GET, POST, TRACE, dll).
        - Tujuan utamanya adalah menemukan metode berisiko seperti TRACE yang dapat digunakan untuk serangan.
    - `http-headers`:
        - Mengambil dan menganalisis header HTTP dari server.
        - Informasi seperti versi server, konfigurasi cache, atau header khusus dapat menunjukkan potensi kerentanan.
    - `http-enum`:
        - Mencoba menemukan direktori, file, atau aplikasi web yang bisa diakses pada server HTTP.
        - Berguna untuk mendeteksi struktur direktori yang terbuka atau file sensitif.
    - `mysql-brute`:
        - Melakukan brute force attack pada server MySQL untuk mencoba kredensial default atau lemah.
        - Digunakan untuk mengidentifikasi akun database yang rentan.
    - `mysql-info`:
        - Mengumpulkan informasi tentang server MySQL, seperti versi, mode autentikasi, dan fitur yang diaktifkan.
3. `-sV`
    - **Penjelasan**: Mengaktifkan deteksi versi layanan (Service Version Detection).
    - **Fungsi**: Memeriksa layanan yang berjalan di port tertentu dan menentukan versinya. Informasi ini berguna untuk mengidentifikasi kerentanan spesifik yang mungkin terkait dengan versi tersebut.
4. `192.168.18.253`
    - **Penjelasan**: Alamat IP target yang akan dipindai.

- **Fungsi**: Mengarahkan Nmap untuk memfokuskan pemindaian pada host ini.

Fungsi Utama Perintah Ini

- **Tujuan**: Menganalisis keamanan tiga layanan utama yang berjalan pada server (SSH, HTTP, dan MySQL).
- **Manfaat**: Mendapatkan informasi mendetail tentang konfigurasi layanan, algoritma enkripsi, versi layanan, dan potensi kerentanan yang bisa dimanfaatkan.
- **Penggunaan**: Sangat cocok untuk tahap awal penilaian keamanan (vulnerability assessment) pada jaringan atau server.

Penetration Testing

Dokumentasi Penemuan

1. Hasil Pemindaian

Port yang Teridentifikasi Terbuka:

1. **Port 22 (SSH)**
    - **Layanan:** OpenSSH 9.8
    - **Protokol yang Didukung:** SSH 2.0
    - **Algoritma Pertukaran Kunci (Key Exchange Algorithms):**
        - curve25519-sha256
        - ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521
        - diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512
        - ext-info-s
    - **Algoritma Enkripsi:**
        - aes256-gcm@openssh.com, aes128-gcm@openssh.com
        - chacha20-poly1305@openssh.com
        - aes256-ctr, aes128-ctr
    - **Fingerprint Host Key:**
        - RSA-SHA2-512, RSA-SHA2-256
        - ECDSA-SHA2-NISTP256
        - SSH-ED25519
    - **Algoritma MAC (Message Authentication Code):**
        - hmac-sha2-256, hmac-sha2-512
        - umac-128-etm@openssh.com
2. **Port 80 (HTTP)**
    - **Layanan:** Apache HTTPD 2.4.62
    - **Direktori Terbuka:**

        - `/phpmyadmin/`
        - `/css/`, `/icons/`, `/img/`, `/js/`, `/src/`
    - **Header Server HTTP:**
        - **Server:** Apache/2.4.62 (Fedora Linux)
        - **Last-Modified:** Wed, 27 Nov 2024
        - **ETag:** "17db-627e24cee42fc"
    - **Metode HTTP yang Didukung:**
        - OPTIONS, HEAD, GET, POST, TRACE (TRACE terindikasi berisiko).
3. **Port 3306 (MySQL/MariaDB)**
    - **Layanan:** MariaDB
    - **Status:** Unauthorized Access, tidak ada akun valid yang ditemukan.
    - **Tes Brute Force:**
        - Hasil: Tidak ada akun dengan kredensial default berhasil ditemukan.

---

2. Analisis Risiko

1. **Port 22 (SSH)**
    - **Kerentanan:** Protokol SSH dapat dieksploitasi melalui brute force attack jika menggunakan kata sandi lemah. Algoritma yang tidak aman dapat digunakan untuk serangan downgrade.
    - **Dampak Potensial:** Akses ke server melalui layanan shell aman ini bisa memberikan kontrol penuh kepada penyerang.
    - Tingkat Risiko: Medium-High
2. **Port 80 (HTTP)**
    - **Kerentanan:**
        - Directory listing aktif pada beberapa direktori penting, termasuk direktori phpMyAdmin yang sering menjadi target eksploitasi.
        - TRACE diizinkan, yang dapat digunakan untuk serangan XST (Cross-Site Tracing).

- **Dampak Potensial:** Potensi kebocoran data sensitif seperti konfigurasi file atau bahkan eksploitasi via phpMyAdmin.
- **Tingkat Risiko:** High

3. **Port 3306 (MariaDB)**
    - **Kerentanan:** Akses terbuka ke port database dapat dimanfaatkan oleh penyerang untuk mencoba autentikasi brute force atau mengidentifikasi celah pada database itu sendiri.
    - **Dampak Potensial:** Eksposur data sensitif atau kerentanan lainnya yang berpotensi merusak integritas sistem.
    - **Tingkat Risiko:** High

---

3. Rekomendasi Mitigasi

1. **Port 22 (SSH):**
    - **A**ktifkan autentikasi berbasis kunci publik dan nonaktifkan login berbasis kata sandi.
    - Batasi akses SSH menggunakan firewall untuk IP tertentu.
    - Terapkan mekanisme perlindungan brute force seperti Fail2Ban.
2. **Port 80 (HTTP):**
    - Nonaktifkan directory listing dengan mengubah konfigurasi Apache:
      Options -Indexes
    - Nonaktifkan metode TRACE pada server HTTP untuk mencegah serangan XST.
    - Amankan direktori phpMyAdmin dengan autentikasi tambahan atau firewall.
3. **Port 3306 (MariaDB):**
    - Batasi akses port 3306 hanya untuk IP tertentu yang dipercaya.
    - Gunakan kredensial yang kuat untuk akun database.
    - Aktifkan logging dan analisis secara berkala untuk mendeteksi aktivitas mencurigakan.

---

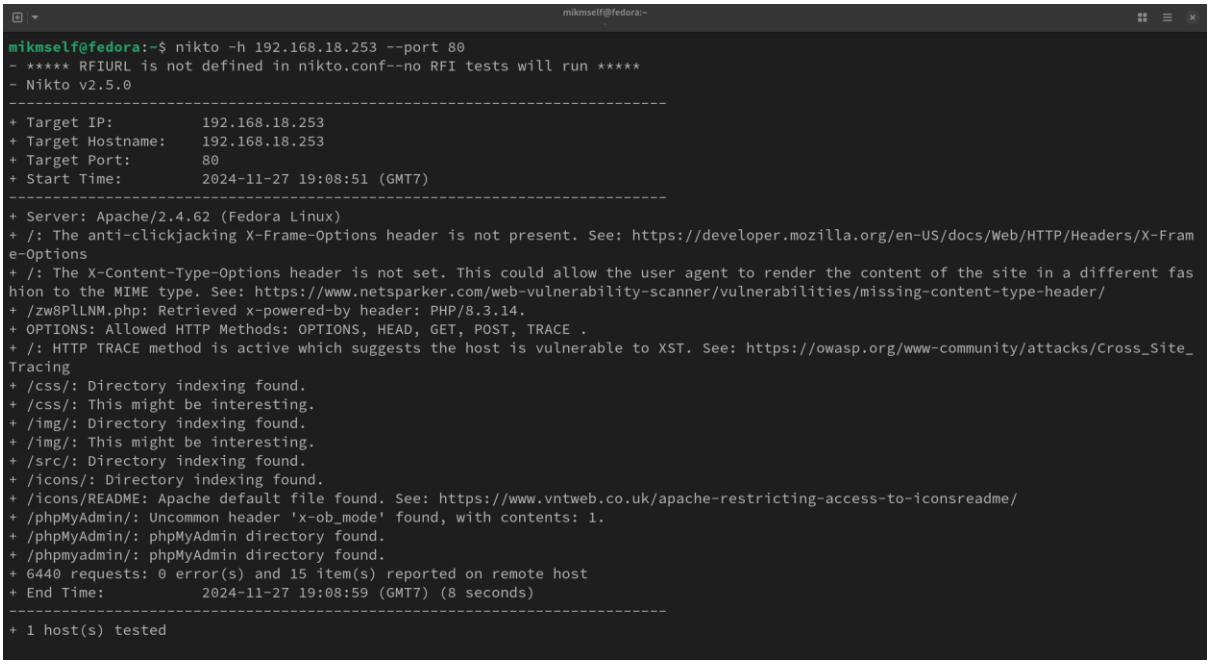
4. Langkah Tindak Lanjut

- Lakukan audit ulang terhadap sistem setelah menerapkan mitigasi.
- Gunakan IDS/IPS untuk memantau aktivitas mencurigakan secara real-time.
- Pertimbangkan melakukan hardening tambahan pada konfigurasi server untuk mengurangi risiko eksploitasi di masa mendatang.

Kesimpulan

Hasil pemindaian mengungkapkan sejumlah kerentanan signifikan pada konfigurasi layanan yang sedang berjalan. Dengan fokus pada mitigasi yang cepat dan efektif, risiko terhadap integritas dan keamanan sistem dapat diminimalisir secara signifikan.

Selanjutnya kita akan mencoba menggunakan nikto



Analisis Temuan Keamanan Berdasarkan Pemindaian Nikto

1. Informasi Umum

- **IP Target**: 192.168.18.253
- **Hostname Target**: 192.168.18.253
- **Port yang Dipindai**: 80 (HTTP)
- **Server Teridentifikasi**: Apache/2.4.62 (Fedora Linux)
- **Waktu Pemindaian**: 27 November 2024, 19:08:51 (GMT+7)
- **Durasi Pemindaian**: 8 detik

---

2. Temuan Keamanan yang Terdeteksi

Nikto melaporkan beberapa kelemahan kritis dan non-kritis yang terdeteksi pada server target. Berikut adalah analisis mendalam:

1. **Header HTTP Tidak Lengkap**:

- **X-Frame-Options Tidak Ditemukan**:
  - Risiko: Membuka potensi serangan **Clickjacking**, di mana elemen berbahaya dapat ditempatkan di atas situs web target.
  - Referensi: MDN Web Docs - X-Frame-Options.
- **X-Content-Type-Options Tidak Dikonfigurasi**:
  - Risiko: Server dapat membiarkan browser merender konten yang tidak sesuai dengan tipe MIME yang diatur, memungkinkan serangan **MIME Sniffing**.
  - Referensi: Netsparker - Missing Content-Type Header.

2. **Metode HTTP yang Rentan**:
   - **Metode TRACE Aktif**:
     - Risiko: Rentan terhadap serangan **Cross-Site Tracing (XST)**, yang dapat memungkinkan eksploitasi data cookie pengguna melalui skrip berbahaya.
     - Referensi: OWASP - Cross Site Tracing.
3. **Directory Indexing Aktif**:
   - Direktori seperti `/css/`, `/img/`, `/src/`, dan `/icons/` ditemukan memiliki **directory listing** yang aktif.
     - Risiko: Informasi terkait struktur direktori dapat dimanfaatkan oleh penyerang untuk mendapatkan akses file sensitif atau menavigasi direktori tanpa otorisasi.
4. **File dan Direktori Sensitif**:
   - **phpMyAdmin Directory Ditemukan**:
     - Risiko: Direktori **phpMyAdmin** sering kali menjadi target serangan brute force atau eksploitasi jika tidak dilindungi dengan baik.
     - Header tambahan: `'x-ob_mode'` ditemukan dengan nilai `1`, menunjukkan kemungkinan pengaturan tertentu yang dapat dieksploitasi.

    - **Default File Apache (`icons/README`) Teridentifikasi**:
        - Risiko: Menunjukkan server belum di-hardening dengan baik, berpotensi mengungkap informasi konfigurasi sensitif.
        - Referensi: Restricting Access to icons/README.
5. **Pengungkapan Header Server**:
    - Header **X-Powered-By** ditemukan dengan nilai `PHP/8.3.14`. Informasi ini dapat membantu penyerang mengidentifikasi kerentanan spesifik pada versi PHP tersebut.

---

3. Penilaian Risiko

- **Risiko Tinggi**:
    1. **Clickjacking (X-Frame-Options)**: Rentan terhadap eksploitasi visual yang dapat menipu pengguna.
    2. **XST (HTTP TRACE)**: Eksposur TRACE membuka potensi serangan terhadap data autentikasi seperti cookie.
- **Risiko Sedang**:

    1. **Directory Indexing**: Memberikan informasi tentang struktur file, membuka peluang eksploitasi file yang salah dikonfigurasi.
    2. **phpMyAdmin Directory**: Eksposur ke direktori admin database dapat menjadi vektor serangan signifikan.
- **Risiko Rendah**:

  Pengungkapan Versi PHP dan Apache: Informasi ini dapat digunakan dalam tahap reconnaissance oleh penyerang.

---

4. Rekomendasi Mitigasi

1. **Header Keamanan HTTP**:
    - Konfigurasikan header `X-Frame-Options` dengan nilai **SAMEORIGIN** atau **DENY**.
    - Tambahkan header `X-Content-Type-Options: nosniff` untuk mencegah MIME sniffing.
2. **Nonaktifkan Metode TRACE**:
    - Perbarui konfigurasi Apache untuk menonaktifkan metode HTTP TRACE:

      ```
      TraceEnable off
      ```
3. **Nonaktifkan Directory Indexing**:
    - Modifikasi konfigurasi Apache dengan menambahkan:

      ```
      Options -Indexes
      ```
4. **Amankan Direktori phpMyAdmin**:
    - Pindahkan direktori phpMyAdmin ke lokasi non-standar.
    - Terapkan autentikasi tambahan pada direktori tersebut.
5. **Sembunyikan Informasi Versi Server**:
    - Ubah konfigurasi Apache dengan menambahkan:

      ```
      ServerTokens Prod
      ServerSignature Off
      ```
6. **Pembaruan Perangkat Lunak**:
    - Pastikan server Apache dan PHP diperbarui ke versi terbaru untuk menghindari eksploitasi kerentanan yang diketahui.

---

5. Kesimpulan

Hasil pemindaian menunjukkan kelemahan mendasar dalam konfigurasi keamanan server, termasuk header HTTP yang tidak lengkap, directory indexing yang aktif, dan metode TRACE yang diaktifkan. Implementasi langkah mitigasi yang disarankan dapat secara signifikan meningkatkan keamanan server dan mengurangi risiko serangan terhadap target.

Rate Limiting & Denial of Service (DoS)

Celah: Aplikasi ini tidak memiliki mekanisme untuk membatasi jumlah request yang diterima oleh endpoint `/api/send-message`.

Contoh Eksploitasi:

Penyerang dapat menggunakan alat seperti **`curl`** atau **`burpsuite`** untuk mengirimkan ribuan request dalam waktu singkat:

```
for i in {1..10000}; do curl -X POST -d
'{"name":"test","email":"test@test.com","message":"spam"}' -H "Content-Type:
application/json" http://localhost:3000/api/send-message; done
```

Ini dapat:

- Membebani server dan membuatnya tidak responsif.
- Mengisi database dengan data spam.

Solusi:

- Terapkan **rate limiting** menggunakan middleware seperti `express-rate-limit`.
- Validasi pengiriman berdasarkan IP atau autentikasi token.

CSRF (Cross-Site Request Forgery)

Celah: Endpoint tidak memeriksa apakah request berasal dari sumber yang sah.

Contoh Eksploitasi:

Jika ada sesi yang aktif, seorang penyerang dapat membuat halaman palsu dengan kode berikut:

```
<form action="http://localhost:3000/api/send-message" method="POST">
```

```
 <input type="hidden" name="name" value="malicious">
 <input type="hidden" name="email" value="hacker@evil.com">
 <input type="hidden" name="phone" value="123456789">
 <input type="hidden" name="message" value="CSRF Exploit">
 <input type="submit" value="Submit">
</form>
```

Ketika korban membuka halaman tersebut, form akan terkirim ke server dengan kredensial aktif.

Solusi:

- Gunakan token CSRF untuk memastikan validitas pengiriman form.
- Validasi header referer/origin untuk memeriksa asal request.

Insecure Direct Object Reference (IDOR) & Missing Authentication and Authorization

Celah: Jika endpoint API tidak melakukan otorisasi dengan benar, penyerang dapat mengakses data atau melakukan operasi yang seharusnya tidak diizinkan.

Contoh Eksploitasi:

Jika endpoint `/api/send-message` tidak memvalidasi pengguna yang mengaksesnya, penyerang dapat mengirim request dengan data palsu tanpa autentikasi:

```
curl -X POST http://localhost:3000/api/send-message -d
'{"name":"Attacker","email":"attacker@evil.com","phone":"12345","message":"Hacked!"}' -H
"Content-Type: application/json"
```

Solusi:

- Tambahkan autentikasi untuk semua endpoint, seperti JWT atau OAuth.

- Periksa izin pengguna sebelum memproses data.

Log Injection & XSS

Celah: Jika input dari pengguna disimpan langsung tanpa sanitasi, ini dapat memungkinkan penyerang menyisipkan kode berbahaya atau memanipulasi.

Contoh Eksploitasi:

Penyerang dapat mengirimkan input seperti berikut:

{

"name":"<script>alert('LogInjection')</script>",

"email":"attacker@evil.com",

"phone":"12345",

"message":"Injected"

}

Jika log aplikasi mencatat input tanpa sanitasi, penyerang dapat memanfaatkan ini untuk:

- **Membaca log sensitif & menyisipkan skrip berbahaya.**

Solusi:

- **Encode semua input sebelum disimpan ke log.**

Bagian Penetration Testing dalam dokumen terutama meliputi:

1. **Uji Kerentanan (menggunakan Nmap)**

- **Pemindaian mendetail pada IP 192.168.18.253**
- **Memindai port 22 (SSH), 80 (HTTP), dan 3306 (MySQL)**
- **Menganalisis versi layanan, konfigurasi, dan potensi kerentanan**

2. **Pemindaian Nikto**

- **Analisis keamanan web server pada port 80**
- **Mengidentifikasi kerentanan header HTTP**
- **Mendeteksi direktori yang terbuka dan risiko keamanan potensial**

3. **Analisis Kerentanan Keamanan Tambahan**

- **Risiko pembatasan laju & Denial of Service (DoS)**
- **Kerentanan CSRF (Cross-Site Request Forgery)**
- **Referensi Objek Langsung Tidak Aman (IDOR)**
- **Autentikasi dan Otorisasi yang Hilang**
- **Potensi eksploitasi Log Injection & XSS**

Setiap bagian mengikuti metodologi penetration testing standar:

- **Rekognisi (pemindaian)**
- **Identifikasi kerentanan**
- **Penilaian risiko**
- **Rekomendasi mitigasi**

Dokumen menyediakan pendekatan komprehensif untuk mengidentifikasi dan mengatasi kelemahan keamanan potensial pada sistem target.