**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN MENGENAI PROBLEM DIRECTORY TRAVERSAL DENGAN PENYELESAIAN MENGGUNAKAN SOLUSI VALIDASI INPUT**

*Laporan ini disusun untuk memenuhi tugas 1 mata kuliah Pengembangan Web (Teori)*



Disusun oleh kelompok B4:

Asri Husnul Rosadi 221524035

Faris Abulkhoir 221524040

Mahardika Pratama 221524044

Muhamad Fahri Yuwan 221524047

Najib Alimudin Fajri 221524053

Septyana Agustina 221524058

Sarah 221524059

Dosen Pengampu:

Joe Lian Min, M.Eng.

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2024**

# **DAFTAR ISI**

[**DAFTAR ISI** i](#_Toc175586464)

[**A.** **IDENTIFIKASI PROBLEM** 1](#_Toc175586465)

[**B.** **DESKRIPSI PROBLEM** 1](#_Toc175586466)

[**C.** **METODOLOGI EKSPERIMEN** 1](#_Toc175586467)

[**D.** **PELAKSANAAN EKSPERIMEN** 2](#_Toc175586468)

[**E.** **ANALISIS HASIL EKSPERIMEN** 3](#_Toc175586469)

# **IDENTIFIKASI PROBLEM**

Serangan directory traversal adalah kerentanan keamanan yang memungkinkan penyerang mengakses file dan direktori di luar jalur direktori yang diizinkan aplikasi web. Jika aplikasi tidak memvalidasi input dengan benar, penyerang dapat menggunakan karakter traversal direktori seperti ../ untuk mengakses file sensitif di luar direktori web yang seharusnya tidak bisa diakses.

# **DESKRIPSI PROBLEM**

Aplikasi web yang dirancang untuk mengakses file berdasarkan parameter URL perlu melindungi diri dari serangan directory traversal. Jika file sensitif, seperti file konfigurasi, dapat diakses melalui manipulasi parameter URL, maka aplikasi tersebut dapat mengalami kebocoran data sensitif dan potensi risiko keamanan.

# **METODOLOGI EKSPERIMEN**

Variabel Terikat: Keamanan aplikasi (apakah file sensitif dapat diakses atau tidak).

Variabel Bebas: Validasi input pada parameter URL (no validation, with validation).

Alat Pengukuran:

* Pengujian manual dengan mengakses URL tertentu untuk mengukur apakah file sensitif seperti danger.txt dapat diakses.
* Pengamatan langsung terhadap output di browser dan log server.

Skenario Pengujian:

1. Tanpa Validasi Input:

* Tujuan: Menentukan apakah file sensitif dapat diakses tanpa adanya validasi pada parameter URL.
* Langkah-langkah:

1. Akses file yang sah dalam direktori files.
2. Coba akses file danger.txt di luar direktori files dengan menggunakan karakter traversal direktori (../).

* Pengukuran: Apakah konten danger.txt ditampilkan di browser.

1. Dengan Validasi Input:

* Tujuan: Menguji apakah validasi input mencegah akses ke file sensitif di luar direktori files.
* Langkah-langkah:

1. Implementasikan validasi input untuk memfilter karakter traversal direktori.
2. Ulangi akses ke file yang sah dan coba akses config.php dengan cara yang sama seperti pada skenario pertama.

* Pengukuran: Apakah validasi berhasil mencegah akses dan mengeluarkan pesan error yang sesuai.

Alat Pengukuran:

* Browser: Untuk mengakses URL dan melihat hasil respons.
* Log Server: Untuk memantau dan mencatat setiap akses yang dilakukan selama eksperimen.

# **PELAKSANAAN EKSPERIMEN**

Langkah 1: Siapkan Struktur File

Struktur file yang digunakan:

/root-directory

│

├── app.php # File PHP utama

├── danger.txt # File konfigurasi sensitif

└── /files # Direktori file yang dapat diakses

├── file1.txt # File teks yang bisa diakses

├── file2.txt # File teks yang bisa diakses

Langkah 2: Implementasikan Kode Awal

Kode awal app.php:

<?php

$file = $\_GET['file'];

$baseDir = \_\_DIR\_\_ . '/files/';

$filePath = $baseDir . $file;

if (file\_exists($filePath)) {

echo file\_get\_contents($filePath);

} else {

echo "File tidak ditemukan.";

}

?>

Langkah 3: Uji Akses File

* Uji File Valid: Akses file di dalam direktori files seperti file1.txt untuk memastikan file dapat dibaca:

http://localhost:8000/yourapp.php?file=file1.txt

* Uji Serangan Directory Traversal: Coba akses file config.php dengan traversal direktori:

http://localhost:8000/yourapp.php?file=../danger.txt

Langkah 4: Tambahkan Validasi

Jika file danger.txt dapat diakses, tambahkan validasi input untuk mencegah serangan directory traversal:

<?php

$file = $\_GET['file'];

$baseDir = \_\_DIR\_\_ . '/files/';

if (strpos($file, '../') !== false || strpos($file, '..\\') !== false) {

die("Invalid file name.");

}

$filePath = $baseDir . basename($file);

if (file\_exists($filePath)) {

echo file\_get\_contents($filePath);

} else {

echo "File tidak ditemukan.";

}

?>

# **ANALISIS HASIL EKSPERIMEN**

**Hasil:**

* Tanpa Validasi Input: File danger.txt dapat diakses jika tidak ada validasi input pada kode aplikasi. Ini menunjukkan bahwa aplikasi tidak membatasi akses ke file yang berada di luar direktori yang diizinkan, membuka kemungkinan untuk serangan directory traversal.
* Dengan Validasi Input: Setelah menerapkan validasi input, file danger.txt tidak dapat diakses. Validasi ini berhasil mencegah akses ke file yang berada di luar direktori yang diizinkan.es File Valid: File yang berada di dalam direktori files dapat diakses dengan benar, menampilkan konten file seperti file1.txt.

**Pros:**

* Perlindungan Terhadap Directory Traversal: Validasi input berhasil mencegah akses ke file di luar direktori yang diizinkan, mengurangi risiko serangan directory traversal.
* Kontrol Akses yang Lebih Baik: Validasi memastikan bahwa hanya file yang berada dalam direktori yang diizinkan (files) yang dapat diakses, meningkatkan keamanan aplikasi.
* Keamanan Berlapis: Menambahkan lapisan perlindungan di tingkat aplikasi, memberikan keamanan tambahan di luar pengaturan server.

**Cons:**

* Peningkatan Kompleksitas Kode: Menambahkan validasi input meningkatkan kompleksitas kode aplikasi, yang dapat mempengaruhi pemeliharaan dan meningkatkan kemungkinan kesalahan implementasi.
* Overhead Kinerja: Validasi input memerlukan pemrosesan tambahan, meskipun dampaknya biasanya kecil, ini tetap menjadi pertimbangan untuk aplikasi dengan beban tinggi.
* Pengujian dan Pemeliharaan Ekstra: Validasi input memerlukan pengujian tambahan untuk memastikan semua potensi celah ditangani dan pemeliharaan berkelanjutan untuk menangani perubahan atau pola serangan baru.

**Kesimpulan:**

* Tanpa Validasi Input: Aplikasi tidak dapat membatasi akses ke file yang berada di luar direktori yang diizinkan, menunjukkan kerentanan terhadap serangan directory traversal dan potensi kebocoran data.
* Dengan Validasi Input: Penerapan validasi input berhasil mencegah akses tidak sah ke file di luar direktori files, menunjukkan bahwa validasi input efektif dalam meningkatkan keamanan aplikasi.