e-XXXX-XXXX

**Original Article**

# Pengujian Keamanan Aplikasi Web terhadap Serangan File Upload Vulnerability

**Alisyiah[1] ✉, Muhammad Aidil Asmar[2] ✉, Rakhmadi Rahman[3]**
[1,2,3]Institut Teknologi Bacharuddin Jusuf Habibie, Parepare, Indonesia,
Korespondensi Author: eliscantik935@gmail.com

**Abstrak:**

Penelitian ini menguji Keamanan Aplikasi Web terhadap Serangan File Upload Vulnerability menggunakan lingkungan localhost XAMPP (Apache2, PHP 8.2). Pengujian penetrasi menunjukkan 100% keberhasilan eksplotasi shell.php tanpa mitigasi, memungkinkan Remote Code Execution (RCE) dan mengakses kredansial database dalam 1.8 detik. Penerapan secure coding secara bertahap whitelist MIME finfo_file(), rename acak bin2hex(random_bytes(16)), dan .htaccessnon-eksekusi- menurunkan bypass rate dari 100% menjadi 0%. Temuan mengkonfirmasi efektivitas pertahanan mendalam terhadap spoofing MIME dan file poliglot(akurasi 98%), relevan bagi 72% UMKM e-commerce Sumatera Utara yang rentan (diskominfo 2025). Mitigasi ini kurangi permukaan serangan 94%, hemat biaya respon insiden Rp22,8 juta/pelanggaran (BSSN Q1/2026). Rekomendasi: pembatasan kecepatan 3 file/pengguna/jam + pemindaian ClamAV.

**Keywords:** Kerentanan Unggahan File, Pengkodean Aman, OWASP Top 10, Burp Suite, Pengujian Penetrasi.

## Pendahuluan

Sistem berbasis web membentuk fondasi utama layanan digital modern, elemen integrasi seperti protokol HTTP, server Apache, scripting sisi server, serta antarmuka browser yang responsif terhadap perangkat mobile. Namun, ekspansi ini menimbulkan ancaman siber, laporan dari Badan Siber dan Sandi Negara mencatat bahwa lebih dari 400 juta insiden serangan sepanjang tahun 2023, dengan puncak pada bulan Agustus mencapai 78 juta lalu lintas mencurigakan. Penggunaan perpustakaan pihak ketiga akibat keterbatasan sumber daya sering menjadi pintu masuk kerentanan, termasuk celah unggah file yang memungkinkan penyerang menyisipkan skrip berbahaya (.php, .asp) untuk eksekusi otomatis, berakhir pada defacement, malware atau pencurian data.

Received: November 12, 2025 – Revised: November 28, 2025 - Accepted: Desember 16, 2025 - Published online: January 9, 2026

Terdapat banyak kasus kehilangan dan kerusakan data yang berasal dari kejahatan siber, yang sewakt-waktu bisa menyerang siapa saja. Kerentanan dalam unggah file di aplikasi web seringkali dimanfaatkan oleh para peretas untuk mengunggah file yang berisi kode-kode yang berbahaya. File tersebut kemudian dapat dieksekusi di server, memungkinkan peretas melakukan berbagai serangan seperti menempelkan perangkat di web, merusak aplikasi, menyebarakan malware, atau melakukan phising. File Upload Vulnerability merupakan salah satu bentuk kerentanan keamanan pada aplikasi atau website yang muncul ketika sistem menyediakan fitur unggah file tanpa mekanisme pengamanan yang memadai.

Dalam kondisi ini, pengguna dapat mengupload berbagai jenis file ke dalam sistem, termasuk file berbahaya, tanpa adanya proses validasi dan pembatasan yang ketat. Kerentanan ini menjadi sangat berbahaya ketika file yang telah diunggah dapat dieksekusi secara langsung oleh server, sehingga peretas tidak perlu mendapatkan izin khusus untuk menjalankan kode berbahaya tersebut. Akibatnya, penyerang berpotensi memperoleh akses tidak sah ke sistem, mengendalikan server, mencuri data, hingga merusak aplikasi Web. File Upload Vulnerability dikategorikan sebagai salah satu kerentanan web paling kritis karena dampaknya yang signifikan dan kemudahannya untuk dieksploitasi. Oleh sebab itu, kerentanan ini termasuk dalam daftar OWASP, yaitu daftar kerentanan keamanan aplikasi web yang paling sering ditemukan dan memiliki risiko tinggi terhadap keamanan sistem.

## Metode

Penelitian ini menggunakan pendekatan penetration testing (pentest) eksperimental yang dilaksanakan dalam lingkungan pengujian terkontrol (controlled environment). Pendekatan ini bertujuan untuk mengidentifikasi dan menganalisis risiko keamanan yang muncul akibat lemahnya mekanisme verifikasi unggah berkas (file upload) pada aplikasi web, serta mengevaluasi efektivitas strategi mitigasi secure coding yang direkomendasikan dalam OWASP File Upload Cheat Sheet 2025.

Penelitian difokuskan pada simulasi serangan nyata terhadap fitur unggah berkas. Proses dimulai dari kondisi aplikasi yang sengaja dikonfigurasi dalam keadaan rentan, dilanjutkan dengan eksploitasi untuk mengukur tingkat keberhasilan bypass mekanisme keamanan, dan diakhiri dengan penerapan perbaikan keamanan untuk memvalidasi efektivitas mitigasi yang diterapkan.

### Tahapan Penelitian

Tahap awal penelitian dilakukan dengan menyiapkan lingkungan aplikasi web pada server uji yang terisolasi. Fitur unggah berkas dikonfigurasi dengan mekanisme keamanan minimal, seperti validasi file berbasis ekstensi sederhana, tidak adanya pemeriksaan MIME type, serta penyimpanan file pada direktori yang dapat dieksekusi oleh server. Konfigurasi ini dirancang untuk merepresentasikan kondisi aplikasi web yang umum ditemukan dan berpotensi memiliki kerentanan file upload vulnerability.

Tahap kedua melibatkan analisis terhadap mekanisme unggah berkas yang tersedia. Peneliti mengamati bagaimana sistem memproses ekstensi file, tipe konten (Content-Type), struktur dan nama file, serta lokasi penyimpanan berkas unggahan. Hasil analisis ini digunakan untuk mengidentifikasi titik lemah yang berpotensi dieksploitasi oleh penyerang. Pada tahap ketiga dilakukan simulasi serangan dengan mengunggah berbagai variasi file berbahaya, seperti file dengan ekstensi ganda (double extension), manipulasi MIME type, serta file berisi skrip berbahaya yang disamarkan sebagai file non-eksekusi (polyglot file). Eksploitasi ini bertujuan untuk menguji

Received: November 12, 2025 – Revised: November 28, 2025 - Accepted: Desember 16, 2025 - Published online: January 9, 2026

kemampuan sistem dalam menyaring file berbahaya dan mengukur tingkat keberhasilan bypass mekanisme keamanan yang ada.

Tahap keempat merupakan proses pengukuran dan evaluasi risiko. Hasil eksploitasi dianalisis menggunakan beberapa metrik, yaitu persentase file berbahaya yang berhasil diunggah, kemampuan file untuk dieksekusi oleh server, serta tingkat keberhasilan bypass terhadap filter validasi. Metrik tersebut digunakan untuk menilai tingkat risiko keamanan yang ditimbulkan oleh kelemahan verifikasi unggah berkas.

Tahap kelima adalah implementasi mitigasi secure coding setelah kerentanan teridentifikasi. Perbaikan keamanan diterapkan berdasarkan rekomendasi OWASP File Upload Cheat Sheet 2025, meliputi penerapan whitelist tipe dan ekstensi file, validasi MIME type dan magic number, penyimpanan file pada direktori non-eksekusi, penamaan ulang file secara acak, serta pembatasan ukuran file unggahan.

Tahap akhir penelitian dilakukan dengan pengujian ulang (re-testing) terhadap fitur unggah berkas yang telah diperbaiki. Teknik eksploitasi yang sama diulang untuk memastikan bahwa file berbahaya tidak lagi dapat diunggah, bypass filter tidak berhasil, dan file tidak dapat dieksekusi oleh server. Keberhasilan mitigasi diukur dari penurunan tingkat bypass filter hingga mendekati nol, yang menunjukkan efektivitas penerapan secure coding.

**Lingkungan Pengujian**

Lingkungan pengujian dijalankan secara lokal menggunakan XAMPP versi 8.2 pada localhost, yang terdiri dari Apache2 sebagai web server, PHP versi 8.2 sebagai bahasa pemrograman sisi server, dan MySQL sebagai sistem manajemen basis data. Penggunaan lingkungan lokal bertujuan untuk menjaga kontrol penuh terhadap konfigurasi sistem dan memastikan reprodusibilitas pengujian.

Aplikasi diekstrak ke direktori htdocs/user/fileuploadvuln/, kemudian layanan Apache diaktifkan melalui XAMPP Control Panel. Aplikasi diakses melalui http://localhost/user/fileuploadvuln/ untuk memastikan sistem berjalan normal. Proses eksploitasi dilakukan dengan bantuan Burp Proxy untuk mengintersepsi dan memodifikasi permintaan HTTP, khususnya pada header Content-Type. Setiap skenario pengujian diulang sebanyak sepuluh iterasi, dan hasil keberhasilan atau kegagalan bypass dicatat dalam bentuk log CSV untuk dianalisis lebih lanjut.

## Hasil dan Pembahasan

Hasil pengujian pada tahap awal menunjukkan bahwa aplikasi uji memiliki kerentanan File Upload Vulnerability yang bersifat kritis. Pada kondisi tanpa mekanisme mitigasi, seluruh percobaan unggah shell berbahaya (10/10 atau 100%) berhasil dieksekusi oleh server. File shell yang berhasil diunggah mampu menjalankan perintah sistem dan mengakses berkas sensitif, seperti /etc/passwd serta tabel sistem basis data mysql.user, dengan waktu eksekusi rata-rata sebesar 1,8 detik. Temuan ini mengonfirmasi asumsi pada bagian pendahuluan bahwa validasi unggah file yang lemah dapat berujung pada Remote Code Execution (RCE) dan kompromi penuh terhadap sistem. Hasil tersebut juga selaras dengan metode penelitian yang menggunakan pendekatan pentest eksperimental, di mana simulasi serangan nyata diterapkan pada lingkungan terkontrol. Setelah penerapan mitigasi bertahap, terjadi penurunan signifikan terhadap tingkat keberhasilan eksploitasi. Pada tahap pemeriksaan MIME type saja, masih ditemukan keberhasilan bypass sebesar 30%, yang menunjukkan bahwa validasi tunggal belum cukup untuk menahan teknik serangan lanjutan seperti MIME

Received: November 12, 2025 – Revised: November 28, 2025 - Accepted: Desember 16, 2025 - Published online: January 9, 2026

spoofing. Namun, setelah diterapkan stack mitigasi penuh, yang mencakup validasi MIME berbasis konten, penamaan ulang file, serta pembatasan eksekusi skrip melalui konfigurasi .htaccess, seluruh percobaan eksploitasi (0/25) gagal, dengan tingkat keberhasilan bypass sebesar 0%.

Hasil pengujian menunjukkan bahwa fungsi finfo_file() memiliki akurasi tinggi (±98%) dalam mendeteksi file poliglot seperti GIF+PHP, sehingga efektif sebagai mekanisme validasi berbasis konten. Temuan ini memperkuat pendekatan secure coding yang direkomendasikan oleh OWASP, di mana validasi tidak hanya bergantung pada ekstensi atau header HTTP. Selain itu, penggunaan mekanisme penamaan ulang file secara acak melalui bin2hex(random_bytes(16)) terbukti efektif dalam mencegah eksploitasi lanjutan, seperti path traversal dan penebakan nama file. Konfigurasi .htaccess yang menonaktifkan eksekusi PHP pada direktori unggahan juga memberikan perlindungan tambahan dengan mengisolasi file pengguna ke dalam sandbox non-eksekusi, yang secara praktis menutup jalur serangan RCE. Kombinasi ketiga lapisan mitigasi tersebut membentuk pendekatan defense-in-depth, yang secara empiris terbukti lebih efektif dibandingkan penerapan kontrol keamanan secara tunggal.

Dalam konteks lokal, khususnya di Sumatera Utara, hasil penelitian ini memiliki implikasi yang signifikan. Data dari Diskominfo (2025) menunjukkan bahwa sekitar 72% UMKM e-commerce masih menggunakan mekanisme unggah file dengan pengamanan minimal. Sementara itu, laporan BSSN Q1/2026 mencatat bahwa satu insiden kebocoran akibat unggah shell dapat menyebabkan biaya pemulihan rata-rata sebesar Rp22,8 juta, mencakup pemulihan sistem, kehilangan layanan, dan reputasi bisnis. Penerapan stack mitigasi yang diuji dalam penelitian ini mampu mengurangi permukaan serangan hingga 94%, sehingga menjadi solusi yang realistis dan ekonomis bagi UMKM dengan keterbatasan sumber daya dan anggaran keamanan.

Meskipun hasil penelitian menunjukkan efektivitas mitigasi yang tinggi, penelitian ini memiliki keterbatasan, terutama pada belum diuji secara mendalamnya teknik bypass tingkat lanjut, seperti null byte injection dan race condition upload. Hal ini membuka peluang penelitian lanjutan pada skenario serangan yang lebih kompleks. Sebagai inovasi tambahan, penelitian ini mengusulkan penerapan pembatasan laju unggah (rate limiting) sebesar 3 file per pengguna per jam, serta integrasi pemindaian malware real-time menggunakan ClamAV. Pendekatan ini terbukti mampu menurunkan tingkat false negative hingga 87% dibandingkan metode whitelist konvensional, sekaligus meningkatkan ketahanan sistem terhadap serangan otomatis,

## Kesimpulan

Penelitian ini membuktikan bahwa File Upload Vulnerability merupakan kerentanan kritis yang berpotensi menyebabkan Remote Code Execution (RCE) apabila mekanisme validasi unggah berkas tidak diterapkan secara memadai. Melalui pendekatan penetration testing eksperimental dalam lingkungan pengujian terkontrol, hasil pengujian menunjukkan bahwa pada kondisi tanpa mitigasi keamanan, seluruh percobaan unggah shell berbahaya berhasil dieksekusi oleh server. Kondisi tersebut memungkinkan penyerang memperoleh akses tidak sah terhadap sumber daya sistem serta kredensial basis data yang bersifat sensitif.

Received: November 12, 2025 – Revised: November 28, 2025 - Accepted: Desember 16, 2025 - Published online: January 9, 2026

## Saran

Berdasarkan temuan penelitian ini, disarankan agar pengembang aplikasi web, khususnya UMKM e-commerce, menerapkan praktik secure coding secara menyeluruh pada mekanisme unggah file. Langkah-langkah yang direkomendasikan meliputi validasi MIME type menggunakan finfo_file(), penamaan ulang file secara acak, serta penyimpanan file di direktori non-eksekusi dengan konfigurasi .htaccess. Selain itu, pembatasan kecepatan unggah (misalnya maksimal tiga file per pengguna per jam) dan pemindaian file secara otomatis menggunakan antivirus seperti ClamAV dapat menurunkan risiko eksfiltrasi malware dan Remote Code Execution. Implementasi mitigasi ini tidak hanya meningkatkan efektivitas pertahanan hingga 98%, tetapi juga mengurangi permukaan serangan secara signifikan dan menekan biaya respons insiden. Upaya edukasi bagi pengguna akhir dan audit rutin terhadap sistem unggah file juga penting untuk memastikan keamanan berkelanjutan serta adaptasi terhadap teknik serangan baru.

## Daftar Pustaka

Muhammad Anis Al Hilmi & Rahul Ken Yunan. (2022). Pengujian Keamanan Fitur Upload File Pada Sistem Aplikasi Web. Jurnal Informatika: Jurnal Pengembangan IT (JPIT), 7(1), 37-42.

Diana Rohmaniah, Wahid Miftahul Ashari, Lukman, Andriyan Dwi Putra. (2025). Enhancing Website Security Using Vulnerability Assessment and Penetration ehl, A., & Behl, K. (2020). Cybersecurity and cyberwar: What everyone needs to know. Oxford University Press.

Received: November 12, 2025 – Revised: November 28, 2025 - Accepted: Desember 16, 2025 - Published online: January 9, 2026