INOMATEC: Jurnal Inovasi dan Kajian Multidisipliner Kontemporer
Vol. 01, No. 03, Tahun 2026, Hal. 367-373, ISSN: 3123-5573 (Online)

Keamanan Server Aplikasi terhadap *Remote Code Execution* (RCE)

Rakhamdi Rahman[1], Lionil Situmeang[2]
[1,2,3] Institut Teknologi Bacharuddin Jusuf Habibie Parepare
rakhmadi.rahman@ith.ac.id[1], lionilsitumeang@gmail.com[2]

Article Info
Article history:
Received December 29, 2025
Revised December 31, 2025
Accepted January 09, 2026

Keywords:
Remote Code Execution, Keamanan Server, Cybersecurity, Penetration Testing, Vulnerability Assessment

ABSTRACT
Remote Code Execution (RCE) is one of the most critical security threats to application servers because it allows attackers to remotely execute malicious commands and potentially take full control of the system. This study aims to evaluate the security level of application servers against RCE threats through vulnerability assessment, controlled penetration testing, and security configuration analysis. The methodology includes vulnerability scanning, limited penetration testing, and security log analysis. The results indicate that the main weaknesses originate from server misconfiguration, poor input validation, and inconsistent patching practices. Recommendations proposed include strengthening patch management, implementing server hardening, deploying Web Application Firewall (WAF), and establishing continuous monitoring mechanisms. These measures are expected to significantly enhance server resilience against RCE-based cyberattacks.

Article Info
Article history:
Received December 29, 2025
Revised December 31, 2025
Accepted January 09, 2026

Kata Kunci:
Remote Code Execution, Application Server Security, Cybersecurity, Penetration Testing, Vulnerability Assessment

ABSTRAK
Remote Code Execution (RCE) merupakan salah satu ancaman keamanan paling kritis pada server aplikasi karena memungkinkan penyerang menjalankan perintah berbahaya dari jarak jauh dan mengambil alih sistem. Penelitian ini bertujuan mengevaluasi tingkat keamanan server aplikasi terhadap ancaman RCE dengan melakukan analisis kerentanan, simulasi serangan terkontrol, serta pengujian konfigurasi keamanan. Metodologi yang digunakan meliputi vulnerability scanning, penetration testing terbatas, serta analisis log keamanan. Hasil penelitian menunjukkan bahwa kelemahan utama berasal dari kesalahan konfigurasi server, input validation yang buruk, serta patching yang tidak konsisten. Rekomendasi diberikan berupa penguatan sistem patching, hardening server, penerapan WAF, dan mekanisme monitoring berkelanjutan.

Corresponding Author:
Rakhmadi Rahman
Sistem Informasi Institut Teknologi Bacharuddin Jusuf Habibie Parepare
E-mail: rakhmadi.rahman@ith.ac.id

Pendahuluan

Server aplikasi merupakan komponen utama dalam operasional sistem informasi modern. Namun, tingginya ketergantungan pada layanan berbasis web meningkatkan risiko terhadap serangan siber, salah satunya adalah *Remote Code Execution* (RCE). RCE

memungkinkan penyerang menjalankan kode tanpa otorisasi pada server target, sehingga berdampak pada hilangnya kontrol sistem, pencurian data, kerusakan layanan, hingga kerugian finansial.

Masalah keamanan ini sering muncul akibat kesalahan konfigurasi, celah pada *framework*, kurangnya validasi input, serta keterlambatan pembaruan keamanan. Oleh karena itu, evaluasi keamanan server terhadap ancaman RCE menjadi penting untuk memastikan sistem tetap aman dan andal.

Tinjauan Pustaka

a. Remote Code Execution (RCE)

Remote Code Execution (RCE) merupakan salah satu jenis serangan siber paling berbahaya karena memungkinkan penyerang mengeksekusi perintah berbahaya pada sistem dari jarak jauh tanpa otorisasi. Serangan ini biasanya memanfaatkan kelemahan pada aplikasi web, server, *library*, atau *framework* yang digunakan. Menurut OWASP, RCE termasuk dalam kategori risiko kritis karena dapat menyebabkan kehilangan kontrol penuh atas sistem, pencurian data, hingga penghentian layanan. RCE sering terjadi akibat validasi input yang buruk, kesalahan konfigurasi, serta penggunaan komponen perangkat lunak yang rentan.

b. Mekanisme Terjadinya RCE

Berdasarkan berbagai studi keamanan, RCE umumnya muncul melalui beberapa mekanisme berikut:

1) **Command Injection:** Terjadi ketika aplikasi mengizinkan input pengguna diteruskan langsung ke perintah sistem tanpa proses sanitasi.
2) **Insecure Deserialization:** Kerentanan yang memungkinkan penyerang menyisipkan *payload* berbahaya melalui proses serialisasi data.
3) **Server Misconfiguration:** Konfigurasi server yang tidak aman seperti port terbuka, *permission* yang longgar, serta fitur eksekusi berbahaya yang tidak dinonaktifkan.

c. Dampak Serangan RCE

Literatur menunjukkan bahwa dampak RCE sangat signifikan terhadap keamanan sistem. Dampaknya meliputi:

1) Pengambilalihan server (*full system compromise*)
2) Kebocoran dan manipulasi data
3) Penyebaran *malware* dan *ransomware*
4) Kerusakan sistem dan gangguan operasional
5) Kerugian finansial dan reputasi organisasi

Banyak insiden besar dunia siber terjadi karena celah RCE yang tidak terdeteksi atau tidak ditangani secara tepat waktu.

d. Praktik Evaluasi Keamanan Server

Beberapa penelitian menyarankan perlunya evaluasi keamanan yang sistematis, di antaranya:

1) Vulnerability Assessment: Untuk mendeteksi potensi celah keamanan.

2) Penetration Testing: Untuk menguji apakah celah benar-benar dapat dieksploitasi.
3) Security Configuration Review: Untuk memastikan konfigurasi server sesuai standar keamanan.
4) Log Analysis dan Monitoring: Untuk mendeteksi aktivitas mencurigakan.

Penelitian terdahulu menunjukkan bahwa kombinasi pendekatan manual dan otomatis memberikan hasil evaluasi keamanan yang lebih komprehensif.

e. Teknik Mitigasi dan Pencegahan

Berdasarkan berbagai sumber literatur keamanan siber, beberapa langkah mitigasi yang direkomendasikan meliputi:

1) Patch Management: Pembaruan sistem operasi, *framework*, dan *library* secara berkala.
2) Server Hardening: Menghapus layanan tidak perlu, memperketat *permission*, dan menutup port yang tidak digunakan.
3) Input Validation dan Secure Coding: Melakukan sanitasi input dan menerapkan prinsip keamanan dalam pengembangan aplikasi.
4) Web Application Firewall (WAF): Untuk memblokir serangan otomatis dan *payload* berbahaya.
5) Monitoring Keamanan Berkelanjutan: Melalui IDS/IPS, analisis log, dan sistem deteksi ancaman.

f. Ringkasan Literatur

Dari literatur yang ada, dapat disimpulkan bahwa:

1) RCE merupakan ancaman kritis bagi server aplikasi.
2) Penyebab utama berasal dari kesalahan konfigurasi, validasi input yang buruk, dan *patching* yang tidak konsisten.
3) Evaluasi keamanan harus dilakukan secara terstruktur menggunakan kombinasi *tools* dan analisis manual.
4) Diperlukan pendekatan pencegahan yang berkelanjutan, bukan hanya reaktif.

Metode Penelitian

Penelitian ini menggunakan pendekatan evaluasi keamanan secara eksperimental dan analitis untuk mengidentifikasi, menguji, serta mengevaluasi tingkat kerentanan server aplikasi terhadap ancaman *Remote Code Execution* (RCE). Tahapan penelitian dijelaskan sebagai berikut:

a. Desain Penelitian

Penelitian ini bersifat:

1) Deskriptif - Evaluatif: Untuk menggambarkan kondisi keamanan server.
2) Eksperimental: Melalui simulasi serangan terkontrol terhadap server uji.
3) Penelitian dilakukan pada lingkungan pengujian (*test environment*) agar tidak mengganggu sistem produksi.

b. Tahapan Penelitian

Pengumpulan Data

Tahap ini dilakukan untuk memahami karakteristik sistem yang diuji, meliputi:

a) Identifikasi arsitektur server dan aplikasi.
b) Inventarisasi sistem operasi, *framework*, dan *library* yang digunakan.
c) Pengumpulan kebijakan keamanan dan konfigurasi server yang diterapkan.
d) Dokumentasi versi perangkat lunak dan status *patch*.

Vulnerability Assessment

Pada tahap ini dilakukan pemindaian kerentanan menggunakan beberapa teknik dan alat bantu:

a. **Port scanning:** Menggunakan Nmap untuk memeriksa port terbuka dan layanan aktif.
b. **Vulnerability scanning:** Menggunakan OWASP ZAP / Burp Suite / OpenVAS untuk mendeteksi potensi celah keamanan.
c. Identifikasi potensi: *Command Injection*, *Insecure Deserialization*, *Server Misconfiguration*, *Outdated Dependencies*.
d. Hasil pemindaian dicatat dan diprioritaskan berdasarkan tingkat risiko.

Pengujian Serangan (Controlled Penetration Testing)

Pengujian dilakukan secara terbatas dan terkontrol dengan skenario:

a. Simulasi RCE melalui *input injection*.
b. Pengujian fungsi yang berpotensi mengeksekusi perintah sistem.
c. Upaya eksploitasi celah yang ditemukan.
d. Evaluasi apakah: Sistem dapat dieksploitasi, Server memberikan akses *shell*, atau Terjadi *privilege escalation*.

Pengujian mengikuti etika dan standar keamanan, dilakukan hanya pada sistem yang diizinkan.

Analisis Log dan Monitoring

Tahap ini dilakukan untuk menilai kemampuan deteksi dan respon sistem:

a. Analisis log server (*access log*, *error log*, *security log*).
b. Identifikasi indikasi percobaan serangan.
c. Evaluasi apakah sistem memberikan peringatan atau hanya diam (*silent failure*).

Evaluasi dan Rekomendasi

Data hasil pengujian dianalisis dengan cara:

a. Mengklasifikasikan tingkat kerentanan berdasarkan *severity* (tinggi, sedang, rendah).
b. Membandingkan kondisi sebelum dan sesudah rekomendasi.
c. Menyusun langkah mitigasi berupa: *Patch management*, *Server hardening*, Validasi input, Implementasi WAF, dan Monitoring keamanan berkelanjutan.

Kerangka Alur Penelitian

Identifikasi Sistem -> Pengumpulan Data -> Vulnerability Scanning -> Pengujian RCE Terbatas -> Analisis Log -> Evaluasi & Rekomendasi.

Hasil dan Pembahasan

Evaluasi Keamanan

Berdasarkan hasil *vulnerability assessment*, *penetration testing* terbatas, dan analisis konfigurasi server, diperoleh beberapa temuan utama sebagai berikut:

Konfigurasi Server Belum Optimal

1) Terdapat port terbuka yang tidak diperlukan.
2) Pengaturan *permission* file masih longgar.
3) Beberapa fitur eksekusi sistem masih aktif dan berpotensi disalahgunakan.

Validasi Input Lemah

1) Ditemukan parameter input yang memungkinkan percobaan command injection.
2) Mekanisme sanitasi input belum diterapkan secara konsisten pada seluruh *endpoint*.

Patch Management Tidak Konsisten

1) Beberapa komponen server dan library masih menggunakan versi lama.
2) Belum terdapat prosedur pembaruan keamanan yang terstruktur.

Monitoring dan Logging Minim

1) Tidak terdapat mekanisme deteksi intrusi secara real-time.
2) Analisis log belum dilakukan secara rutin.

Hasil Simulasi Serangan RCE

Pengujian dilakukan secara terbatas pada lingkungan uji (*test environment*). Hasilnya menunjukkan bahwa:

1) Percobaan *command injection* berhasil mengeksekusi perintah sistem dengan hak akses terbatas.
2) Server sempat memberikan akses *shell* terbatas sebelum diblokir oleh mekanisme keamanan bawaan.
3) Tidak terjadi *privilege escalation*, namun potensi tetap ada jika tidak dilakukan perbaikan keamanan.
4) Hasil ini membuktikan bahwa sistem masih memiliki peluang untuk dieksploitasi oleh penyerang dengan teknik yang lebih canggih.

Analisis Risiko dan Dampak

Apabila kerentanan tidak segera ditangani, potensi risiko yang dapat terjadi meliputi:

1) Pengambilalihan kendali server (*full system compromise*).
2) Kebocoran dan manipulasi data sensitif.
3) Gangguan operasional layanan.
4) Penyebaran *malware* atau *ransomware*.
5) Kerugian finansial serta penurunan reputasi organisasi.

Dengan demikian, ancaman RCE memiliki tingkat risiko yang tinggi dan memerlukan penanganan serius.

Pembahasan

Hasil penelitian ini sejalan dengan temuan berbagai literatur yang menyatakan bahwa RCE umumnya dipicu oleh:

1) Kesalahan konfigurasi server.
2) Validasi input yang tidak memadai.
3) Pengelolaan *patch* yang lemah.

Selain itu, penelitian ini menegaskan bahwa konfigurasi *default* server belum cukup untuk menjamin keamanan. Diperlukan:

1) Penerapan *patching* yang konsisten.
2) *Hardening* server sesuai standar keamanan.
3) Validasi input yang ketat dan berlapis.
4) Monitoring keamanan berkelanjutan serta penerapan WAF sebagai proteksi tambahan.

Ringkasan Temuan

Berdasarkan hasil evaluasi dapat disimpulkan bahwa:

- Server masih memiliki kerentanan terhadap ancaman RCE.
- Faktor utama berasal dari konfigurasi yang tidak optimal, validasi input yang lemah, dan ketidakteraturan *patching*.
- Implementasi rekomendasi keamanan yang tepat dapat secara signifikan menurunkan risiko eksploitasi.

Kesimpulan

Berdasarkan hasil evaluasi keamanan yang telah dilakukan, dapat disimpulkan bahwa server aplikasi masih memiliki potensi kerentanan terhadap ancaman *Remote Code Execution* (RCE). Kerentanan tersebut terutama disebabkan oleh konfigurasi server yang belum optimal, mekanisme validasi input yang belum diterapkan secara menyeluruh, serta pengelolaan *patch* yang tidak konsisten. Hasil simulasi serangan menunjukkan bahwa perintah sistem masih dapat dieksekusi dengan hak akses terbatas, yang menandakan bahwa peluang eksploitasi tetap terbuka apabila tidak dilakukan perbaikan.

Ancaman RCE memiliki tingkat risiko yang tinggi karena berpotensi menyebabkan pengambilalihan kendali sistem, kebocoran data, gangguan operasional, hingga kerugian finansial dan reputasi. Oleh karena itu, diperlukan langkah mitigasi yang komprehensif melalui penerapan *patch management* yang terstruktur, *hardening server*, validasi input yang ketat, implementasi *Web Application Firewall* (WAF), serta monitoring keamanan yang berkelanjutan. Dengan penerapan langkah tersebut, tingkat keamanan server aplikasi diharapkan dapat meningkat secara signifikan dan lebih mampu menghadapi potensi serangan RCE di masa mendatang.

Daftar Pustaka

OWASP Foundation. (2021). *OWASP Top 10 - Web Application Security Risks*. https://owasp.org/Top10/

OWASP Foundation. (2018). *OWASP Testing Guide v4*. https://owasp.org/www-project-web-security-testing-guide/

MITRE Corporation. (2023). *Common Weakness Enumeration (CWE) Software Weaknesses*. https://cwe.mitre.org/

MITRE Corporation. (2023). *Common Vulnerabilities and Exposures (CVE)*. https://cve.mitre.org/

NIST. (2018). *Guide to Enterprise Patch Management Technologies*. NIST Special Publication 800-40r3.
NIST. (2017). *Framework for Improving Critical Infrastructure Cybersecurity*.
SANS Institute. (2020). *Remote Code Execution: Threats and Mitigation Strategies*.
ENISA. (2020). *Good Practices for Web Application Security*.
Stuttard, D., & Pinto, M. (2011). *The Web Application Hacker's Handbook*. Wiley.
Brooks, C. (2018). *Cybersecurity Essentials*. Jones & Bartlett Learning.