RISTEC : Research in Information Systems and Technology
Vol. 6 No. 1 Tahun. 2025

Security Audit for Detection and Mitigation of AMP Injection from Online Gambling Sites on the Wordpress Platform with a Code Analysis and Wordfence Scanning Approach

Muhamad Fathur Fawaz[1*], Yopi Nugraha[2]
[1,2]Sistem Informasi, Fakultas Ilmu Terapan dan Sains, Institut Pendidikan Indonesia,
Jl. Terusan Pahlawan No.32, Sukagalih, Garut, Jawa Barat 44151, Indonesia
e-mail : *fathur38@institutpendidikan.ac.id*

Abstract: This This study identifies and addresses the issue of AMP injection on WordPress sites infected by attacks from online gambling platforms. The attack involves modifying the core file template-loader.php, injecting malicious code that can lead to site redirection and compromise the integrity of the site. Mitigation actions applied, such as removal of infected files, restoration of files to their original version, and scanning with Wordfence, successfully restored the site to a secure state. Additionally, using Google Search Console to remove illegal pages helped reduce the presence of harmful content indexed by search engines. The findings emphasize the importance of regular maintenance of core files and plugins, as well as the use of effective security tools to prevent further exploitation and ensure the ongoing security of WordPress sites. These findings provide practical guidance for WordPress site owners to protect their sites from cyber threats.

Keywords: AMP Injection, WordPress Security, Core File Modification, Online Gambling Attack, Wordfence

Abstrak: Penelitian Penelitian ini mengidentifikasi dan mengatasi masalah injeksi AMP pada situs WordPress yang terinfeksi oleh serangan dari platform perjudian online. Serangan ini melibatkan modifikasi file inti template-loader.php, yang menyisipkan kode berbahaya yang dapat mengarah pada pengalihan situs dan merusak integritas situs. Tindakan mitigasi yang diterapkan, seperti penghapusan file terinfeksi, pengembalian file ke versi asli, dan pemindaian dengan Wordfence, berhasil mengembalikan situs ke kondisi aman. Selain itu, penggunaan Google Search Console untuk menghapus halaman ilegal mengurangi jejak konten berbahaya yang terindeks. Hasil penelitian ini menegaskan pentingnya pemeliharaan rutin pada file inti dan plugin, serta penggunaan alat keamanan yang efektif untuk mencegah eksploitasi lebih lanjut dan menjaga keamanan situs WordPress secara berkelanjutan. Temuan ini memberikan panduan praktis bagi pemilik situs WordPress dalam melindungi situs mereka dari ancaman siber.

Kata kunci: AMP Injection, Keamanan WordPress, Modifikasi File Inti, Serangan Perjudian Online, Wordfence

PENDAHULUAN

WordPress adalah platform manajemen konten (CMS) yang paling banyak digunakan di dunia, dengan lebih dari 40% situs web yang berjalan di atasnya (Ruohonen, 2019) Popularitas yang tinggi ini menjadikan WordPress sebagai target yang menarik bagi pihak yang berniat jahat, terutama bagi mereka yang berusaha untuk menyebarkan situs ilegal, seperti situs perjudian online. (Sead Fadilpašić, 2025) melaporkan bahwa lebih dari 35.000 situs web telah disusupi dalam kampanye peretasan besar-besaran yang mengarahkan pengunjung ke halaman kasino ilegal. Penyerang

menggunakan kode berbahaya untuk mengalihkan lalu lintas situs yang sah ke situs perjudian ilegal, yang sering kali tidak terdeteksi oleh pengguna.

Pada penelitian (Kayleigh Martin, 2025) menginformasikan bahwa penyerang menggunakan plugin WordPress palsu untuk menyuntikkan tautan spam kasino ke footer situs yang sah. Teknik ini menyembunyikan tautan berbahaya dari pemindaian tradisional dan dapat merusak reputasi situs serta menurunkan peringkat SEO.

Informasi diatas diperkuat dengan data bahwa sekitar 95,62% situs web yang dibuat pada tahun 2021 adalah situs web berbasis WordPress (Laksmiati, 2023). Dalam beberapa kasus, situs-situs yang sah dan aman dapat secara tidak sengaja menjadi platform untuk memuat konten ilegal, dengan penyerang menggunakan teknik seperti *AMP (Accelerated Mobile Pages) injection* untuk menempelkan situs perjudian online atau konten berbahaya lainnya ke situs WordPress yang sah ("Experiencia Implementando AMP En WordPress," 2020). Serangan ini dapat merusak reputasi pemilik situs, mengganggu pengalaman pengguna, dan membuka potensi eksploitasi data yang berbahaya.

Salah satu tantangan terbesar yang dihadapi oleh pemilik situs WordPress adalah deteksi dan mitigasi dari ancaman yang berasal dari pihak ketiga yang menginjeksi kode berbahaya ke dalam situs mereka. Pada kasus serangan AMP injection, kode berbahaya dimasukkan ke dalam halaman yang dilihat oleh pengunjung, seringkali tanpa disadari oleh pemilik situs atau pengunjung itu sendiri. Hal ini membuka pintu bagi eksploitasi lebih lanjut, termasuk manipulasi data dan pencurian informasi sensitif pengguna (Buchyk & Kuroiedov, 2024).

Penelitian ini bertujuan untuk mengidentifikasi dan memitigasi AMP injection yang dilakukan oleh situs ilegal (seperti situs perjudian online) yang memanfaatkan situs WordPress yang sah sebagai platform mereka. Fokus penelitian ini adalah pada bagaimana modifikasi kode pada file inti WordPress yang tidak terdeteksi dapat mengarah pada terjadinya serangan AMP injection, serta bagaimana pemilik situs dapat mengatasi masalah ini dengan pendekatan berbasis analisis kode dan alat pemindaian keamanan seperti **Wordfence**. Wordfence adalah plugin keamanan yang banyak digunakan untuk WordPress yang menggabungkan beberapa lapisan pertahanan, seperti deteksi ancaman waktu nyata, pemindaian berbasis tanda tangan dan heuristik, serta firewall aplikasi web terintegrasi. Lapisan pertahanan ini mendeteksi dan mengurangi ancaman siber seperti injeksi SQL dan cross-site scripting (Nair, 2024)

Melalui pendekatan ini, penelitian ini bertujuan untuk memberikan solusi konkret bagi pemilik situs WordPress dalam mengidentifikasi dan menangkal serangan AMP injection yang berpotensi merusak reputasi dan keamanan situs mereka. Selain itu, hasil penelitian ini diharapkan dapat memberikan wawasan tambahan tentang pentingnya audit keamanan yang rutin dan penggunaan

alat pemindaian yang efektif dalam mencegah penyalahgunaan platform WordPress oleh pihak yang tidak bertanggung jawab.

Dengan temuan ini, diharapkan penelitian ini dapat memberikan kontribusi bagi pengembangan metodologi dan alat yang lebih efektif dalam menangani risiko keamanan pada situs WordPress yang terinfeksi AMP injection. Penelitian ini juga memberikan pedoman praktis bagi pemilik situs untuk melindungi situs mereka dari eksploitasi yang semakin berkembang, serta meningkatkan kesadaran tentang perlunya sistem pertahanan yang lebih kuat untuk platform CMS yang sangat populer ini.

KAJIAN PUSTAKA

Beberapa teori berkorelasi dan mendukung penelitian ini. Teori-teori ini dijelaskan sebagai berikut:

Keamanan Wordpress

WordPress merupakan sistem manajemen konten (CMS) yang sangat populer, dengan 43,4% dari situs web di dunia menggunakannya (W3Techs, 2025). Keberadaannya yang luas dan kompleksitas ekosistemnya, terutama terkait dengan plugin dan tema yang bervariasi, menjadikan WordPress sasaran utama bagi serangan siber. Serangan seperti SQL Injection, Cross-Site Scripting (XSS), dan injeksi lainnya menjadi ancaman serius terhadap keamanan WordPress (Ruohonen, 2019). Meskipun WordPress menawarkan fleksibilitas dan kemudahan penggunaan, kerentanannya yang terkait dengan plugin dan tema pihak ketiga membutuhkan perhatian khusus dalam hal pengelolaan dan pembaruan keamanan.

Tantangan dalam mengamankan WordPress bersifat multidimensional, melibatkan aspek teknis, operasional, dan manajerial. Studi sebelumnya menekankan pentingnya pembaruan berkala dan pemantauan integritas file untuk mencegah serangan (Martínez et al., 2018). Namun, gap yang ada adalah kurangnya fokus pada penerapan metode mitigasi berbasis FMEA (Failure Modes and Effects Analysis) yang dapat memberikan gambaran lebih komprehensif tentang potensi kerentanannya, terutama dalam konteks perubahan yang dilakukan pada file inti WordPress. Oleh karena itu, penelitian ini bertujuan untuk mengisi gap tersebut dengan mengintegrasikan pendekatan berbasis FMEA dalam upaya mitigasi dan pencegahan untuk mengurangi potensi kerentanannya.

Injeksi AMP *(Accelerated Mobile Pages)*

AMP adalah framework sumber terbuka yang dikembangkan oleh Google untuk meningkatkan kecepatan pemuatan halaman web di perangkat seluler. Meskipun tujuannya adalah untuk meningkatkan pengalaman pengguna, AMP juga memiliki potensi disalahgunakan dalam konteks serangan siber, terutama phishing. (Veronika Telychko, 2023) mencatat bahwa AMP digunakan

dalam serangan phishing dengan menyembunyikan URL berbahaya di balik halaman AMP yang sah, memungkinkan peretas untuk menghindari deteksi oleh sistem keamanan.

(Dylan Duncan, 2023) menulis bahwa AMP sering digunakan dalam kampanye phishing, dimana peretas memanfaatkan AMP untuk menyembunyikan URL berbahaya, yang dapat menghindari deteksi mesin pencari dan alat pemindai, sementara tetap mempertahankan tampilan halaman yang tampak sah bagi pengunjung.

Dalam penelitian ini, deteksi eksploitasi dilakukan melalui Google Search Console, yang mengidentifikasi bahwa halaman AMP di situs WordPress yang sah telah dimodifikasi untuk mengarahkan pengunjung ke situs berbahaya (Zainab, 2025). (Patchstack, 2023) mengungkapkan bahwa eksploitasi ini dapat terjadi tanpa menggunakan plugin AMP, di mana peretas dapat memodifikasi bagian halaman AMP untuk menyisipkan link atau skrip berbahaya yang mengarahkan pengguna ke situs yang tidak sah atau mencuri data pribadi pengguna.

Studi lebih lanjut menunjukkan bahwa injeksi AMP berpotensi digunakan untuk mengalihkan pengguna ke situs berbahaya, menyisipkan kode berbahaya pada halaman AMP yang tidak terdeteksi, dan mengekspos data pribadi pengguna (Nair, 2024).

Penelitian ini menyoroti bahwa meskipun tidak menggunakan plugin AMP, modifikasi halaman AMP yang dilakukan tanpa kontrol keamanan yang tepat dapat mengekspos situs terhadap eksploitasi lebih lanjut. Google Search Console digunakan untuk mendeteksi halaman-halaman AMP yang terinfeksi, yang mengarah pada peningkatan kesadaran tentang pentingnya pemantauan dan deteksi dini terhadap kerentanannya ini.

Analisis Kode Sumber Injeksi AMP

Analisis kode sumber merupakan metode utama dalam mendeteksi injeksi AMP pada situs WordPress. Penelitian oleh (Laksmiati, 2023) menunjukkan bahwa analisis kode secara statis dan dinamis memiliki peran yang sangat penting dalam mendeteksi perubahan kode yang mencurigakan. Teknik ini memungkinkan untuk menemukan modifikasi atau penambahan kode yang tidak sah pada file sistem, yang berpotensi menambah kerentanannya terhadap serangan.

Beberapa alat otomatis seperti Wordfence telah digunakan untuk memindai kode sumber dan mendeteksi kerentanannya. Wordfence memeriksa dan memblokir permintaan HTTP berbahaya dengan memanfaatkan firewall tingkat aplikasi (Liu & Gouda, 2009). Meskipun Wordfence efektif dalam menangani kerentanan yang ada, gap penelitian terletak pada kurangnya analisis mendalam mengenai peran Wordfence dalam mendeteksi dan mengatasi injeksi AMP secara spesifik. Walaupun dari tulisan (Chloe Chamberland, 2023) Laporan dari Wordfence menunjukkan bahwa pemindaian berkala menggunakan Wordfence dapat mendeteksi perubahan mencurigakan yang dilakukan pada file inti WordPress dan plugin yang terinstal, yang bisa jadi berisi kode berbahaya

akibat injeksi AMP atau teknik serangan lainnya. Penelitian ini bertujuan untuk mengevaluasi efektivitas Wordfence dalam mendeteksi dan memitigasi risiko keamanan yang berkaitan dengan AMP injection di situs WordPress.

Secara keseluruhan, meskipun penelitian sebelumnya telah banyak membahas kerentanannya WordPress dan serangan berbasis AMP, gap dalam penelitian ini terletak pada kurangnya fokus pada integrasi teknik mitigasi berbasis FMEA dan pengaruhnya terhadap pengurangan kerentanan. Selain itu, pemanfaatan alat keamanan seperti Wordfence juga perlu dikaji lebih dalam dalam konteks serangan AMP. Penelitian ini mengisi celah tersebut dengan menawarkan pendekatan yang lebih holistik dalam mengidentifikasi dan mengatasi kerentanan yang ada, serta menyediakan rekomendasi yang lebih komprehensif untuk melindungi situs WordPress dari eksploitasi yang semakin kompleks.

METODE PENELITIAN

Studi ini menggunakan pendekatan kualitatif deskriptif dengan desain studi kasus yang berfokus pada masalah injeksi AMP pada situs WordPress yang terkait dengan perjudian online. Penelitian ini bertujuan untuk memberikan solusi praktis terhadap masalah tersebut melalui eksperimen mitigasi pada situs yang telah teridentifikasi mengalami serangan serupa. Penelitian ini merupakan pengalaman dari penulis dalam menangani situs yang terkena serangan AMP dan bertujuan untuk membagikan temuan serta solusi yang telah diterapkan.

1. Pemilihan Situs Sampel

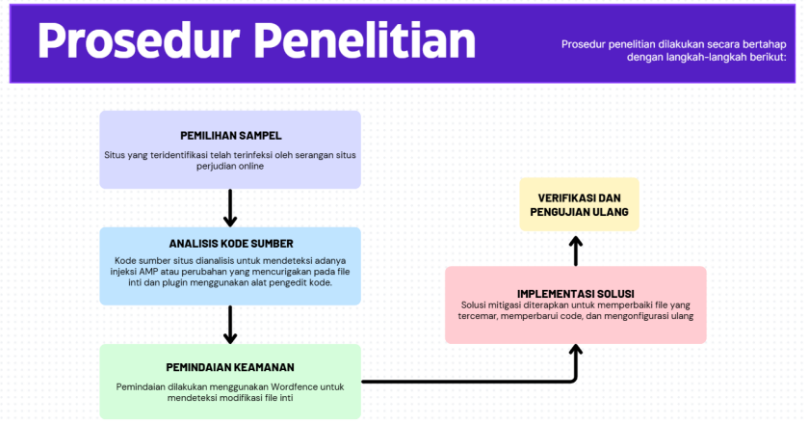
Situs yang digunakan dalam penelitian ini adalah satu situs WordPress yang sebelumnya telah teridentifikasi terinfeksi oleh serangan yang berkaitan dengan situs perjudian online. Situs ini dipilih sebagai objek studi kasus berdasarkan pengalaman penulis yang telah menangani dan memperbaiki masalah yang terjadi pada situs tersebut.

Kriteria Pemilihan Situs:

- **Riwayat serangan**: Situs ini telah terkena serangan yang dimanfaatkan oleh platform perjudian ilegal, menggunakan teknik injeksi AMP.
- **Trafik tinggi**: Situs ini memiliki volume trafik yang signifikan, yang menunjukkan relevansi dan dampak yang dapat ditimbulkan dari serangan tersebut pada pengunjung.
- **Kerentanannya terhadap AMP**: Situs ini teridentifikasi rentan terhadap eksploitasi injeksi AMP dan memerlukan perbaikan untuk meningkatkan keamanannya.

Penelitian ini berfokus pada satu situs tersebut untuk menggali lebih dalam permasalahan yang terjadi serta menerapkan solusi yang telah terbukti efektif dalam menangani masalah serupa. Hasil temuan dan solusi yang diterapkan diharapkan dapat memberikan panduan bagi pemilik situs WordPress lainnya yang menghadapi masalah serupa.

2. Prosedur Penelitian



Gambar 1. Prosedur Penelitian

Diagram Alur Prosedur Penelitian (lihat Gambar 1) menggambarkan langkah-langkah yang diambil dalam penelitian ini. Di bawah ini adalah prosedur yang dijelaskan lebih rinci:

A. Identifikasi dan Pemindaian Masalah:

a) **Pemindaian Keamanan**: Pemindaian dilakukan menggunakan Wordfence untuk mendeteksi adanya modifikasi yang mencurigakan pada file inti atau plugin, serta mendeteksi masalah terkait injeksi AMP. Pemindaian ini bertujuan untuk mencari tanda-tanda potensi eksploitasi lebih lanjut.

b) **Analisis Kode Sumber**: Kode sumber situs diuji secara manual untuk memeriksa perubahan atau modifikasi yang tidak sah pada file inti dan plugin, dengan fokus pada kode yang mungkin terkait dengan injeksi AMP.

B. Implementasi Solusi Mitigasi

Berdasarkan hasil analisis yang dilakukan, solusi mitigasi diterapkan, yang meliputi:

a) **Pembaruan file inti dan plugin**: File yang rentan diperbarui untuk menutup celah keamanan.

b) **Pengaturan ulang pengalihan AMP**: Proses ini memastikan bahwa pengalihan AMP berjalan sesuai prosedur yang aman, meminimalkan risiko penyalahgunaan.

c) **Pemantauan situs**: Penggunaan alat pemindai keamanan seperti Wordfence untuk memantau situs secara berkelanjutan, guna mendeteksi ancaman baru yang mungkin muncul.

C. **Pengujian dan Verifikasi Solusi Mitigasi:**

Setelah penerapan solusi mitigasi, langkah-langkah berikutnya adalah verifikasi dan pengecekan ulang:

a) **Pemindaian ulang menggunakan Wordfence**: Pemindaian dilakukan untuk memeriksa apakah situs terhindar dari potensi eksploitasi lebih lanjut dan apakah solusi mitigasi telah efektif.

b) **Verifikasi integritas situs**: Pengecekan dilakukan untuk memastikan bahwa situs tidak lagi terpengaruh oleh injeksi AMP dan langkah-langkah mitigasi telah berhasil.

3. Proses Manual yang Dilakukan Selama Analisis Kode

Pada tahap analisis kode manual, fokus utama diberikan pada **file-template-loader.php**, yang merupakan target umum untuk modifikasi berbahaya. File ini dibandingkan dengan versi asli dari WordPress 6.7.2 untuk mendeteksi perbedaan yang mencurigakan, seperti penyisipan fungsi tambahan is_bot() yang digunakan untuk mengontrol pengunjung bot atau penyisipan kode berbahaya.

4. Konfigurasi dan Penggunaan Wordfence

Wordfence digunakan untuk memindai dan mengidentifikasi kerentanan situs, serta melacak modifikasi pada file inti dan plugin WordPress. Pengaturan Wordfence dilakukan dengan mengaktifkan **firewall tingkat aplikasi** dan **pemindai malware**, yang memungkinkan identifikasi otomatis terhadap potensi ancaman. Proses konfigurasi mencakup pengaturan pemindaian otomatis dan manual untuk memastikan bahwa file yang digunakan adalah versi yang tidak dimodifikasi.

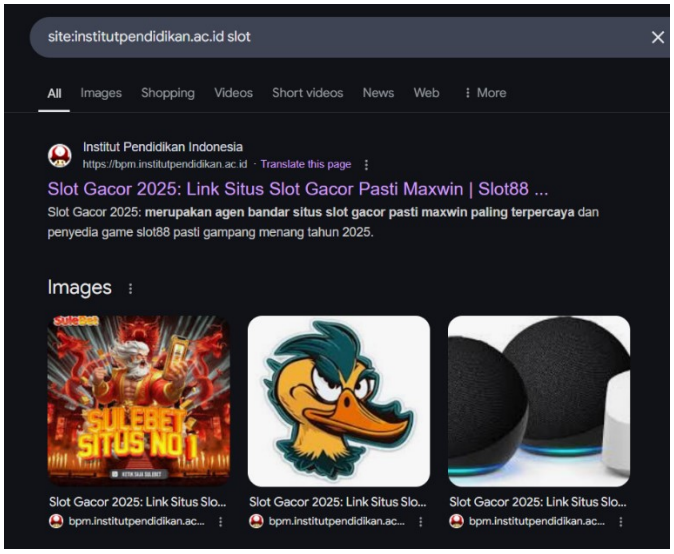
HASIL DAN PEMBAHASAN

Teknik Audit Keamanan:

Dalam penelitian ini, beberapa teknik audit keamanan digunakan untuk mengidentifikasi dan menganalisis situs WordPress yang terinfeksi injeksi AMP, khususnya yang terkait dengan situs perjudian online. Langkah-langkah yang diambil meliputi:

1. Information Gathering Menggunakan Google

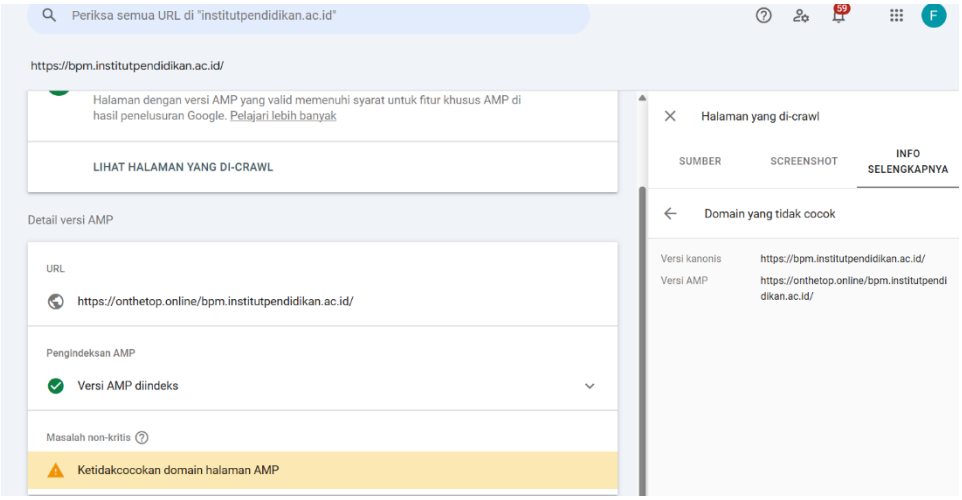
Menggunakan sintaks pencarian “site:” pada Google untuk mencari situs yang berpotensi terinfeksi dengan kata kunci seperti "slot". Hasil pencarian menunjukkan beberapa situs yang terindikasi sudah diserang dengan memanipulasi pengalihan ke halaman perjudian online.



Gambar 2. Pencarian Informasi Indikasi Serangan

2. Pemeriksaan URL dengan Google Search Console

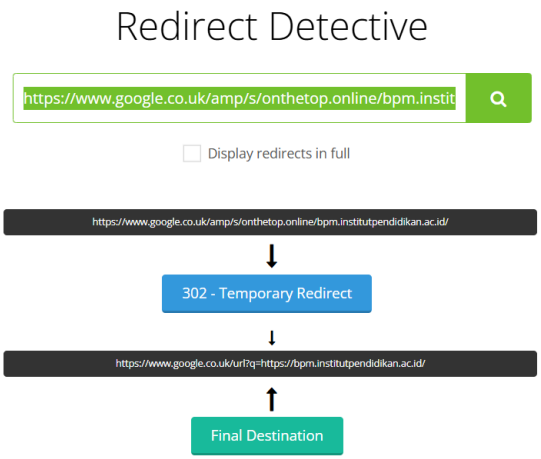
Melaksanakan inspeksi URL melalui Google Search Console untuk mengidentifikasi halaman AMP yang tidak sesuai. Teknik ini berfungsi untuk mengidentifikasi halaman-halaman yang mungkin sudah dimodifikasi atau mengarah ke situs berbahaya yang tidak sah.



Gambar 3. Inspeksi URL pencarian halaman AMP yang tidak cocok

3. Pengecekan *Redirect* Menggunakan *Redirect Detective*

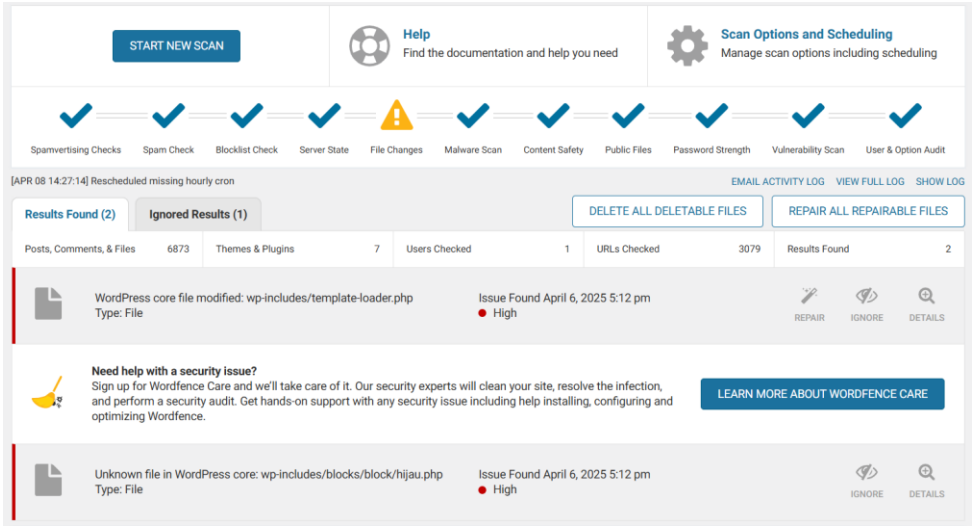
Menggunakan ***Redirect Detective*** untuk memeriksa link redirect dari situs yang terkait dengan perjudian daring. Alat ini membantu mengidentifikasi apakah ada pengalihan tidak sah atau penyalahgunaan pengalihan AMP yang mengarah ke situs berbahaya.



Gambar 4. Pengecekan Link Redirect

4. Pemindaian Keamanan dengan Wordfence

Pemindaian dilakukan pada situs yang teridentifikasi terindikasi diserang menggunakan **Wordfence**. Alat ini membantu mendeteksi adanya modifikasi file inti atau plugin yang terinfeksi, serta potensi kerentanannya yang dapat dimanfaatkan untuk eksploitasi lebih lanjut.



Gambar 4. Hasil Scanning Wordfence

Berdasarkan hasil pemindaian menggunakan Wordfence, dua masalah utama terdeteksi yang menunjukkan adanya kerentanannya pada situs WordPress. Pertama, ditemukan modifikasi pada file inti WordPress yaitu pada file wp-includes/template-loader.php. Modifikasi ini dapat mengindikasikan bahwa file inti telah diubah secara tidak sah, yang sering digunakan oleh penyerang untuk menyisipkan kode berbahaya atau memodifikasi fungsionalitas situs.

Hasil Analisis Kode

Dalam tahap analisis kode, perbedaan antara file template-loader.php yang telah dimodifikasi (terinfeksi) dan versi asli dari WordPress 6.7.2 sangat mencolok dan menunjukkan adanya eksploitasi yang signifikan. Berikut adalah perbandingan dan analisis hasil temuan:

Perbedaan Utama dalam Kode:

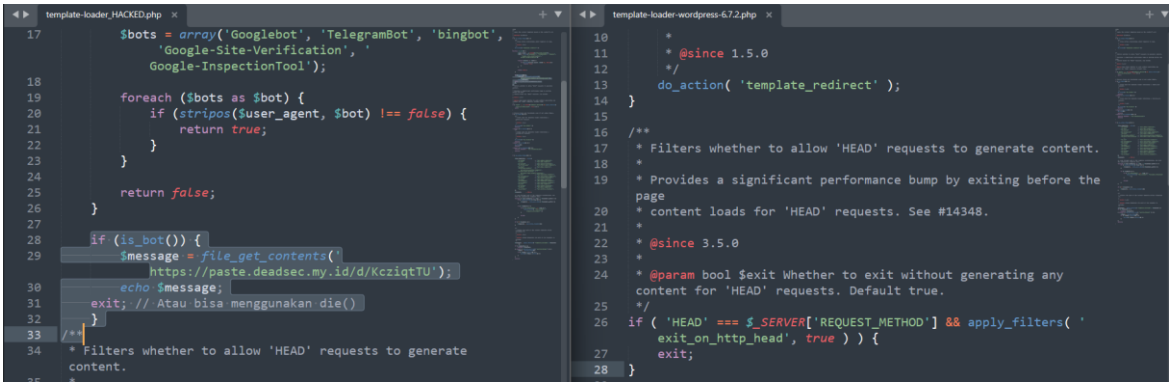
1. **Penambahan Fungsi is_bot() (Hacked Version):**
    a. Dalam versi yang dimodifikasi, fungsi **is_bot()** ditambahkan untuk memeriksa apakah pengunjung adalah bot berdasarkan user-agent. Jika pengunjung terdeteksi sebagai bot, situs akan menampilkan pesan yang diambil dari **URL eksternal** (https://paste.deadsec.my.id/d/KcziqtTU - tidak untuk diakses).
    b. **Analisis**: Fungsi ini mengindikasikan bahwa penyerang mencoba mengalihkan atau mengontrol respons situs hanya untuk pengunjung tertentu (bot). Tujuannya mungkin untuk menyembunyikan konten asli situs dari mesin pencari atau mengalihkan bot ke halaman phishing atau berbahaya.
2. **Penyisipan Kode file_get_contents() untuk Mengambil Pesan Berbahaya (Hacked Version):**
    a. Pada versi yang dimodifikasi, jika **is_bot()** mengembalikan nilai true, situs akan menampilkan pesan yang diambil dari URL eksternal menggunakan fungsi file_get_contents(). URL tersebut adalah **URL yang mencurigakan**, yang jelas merupakan bagian dari eksploitasi untuk mendistribusikan informasi yang berpotensi berbahaya.
    b. **Analisis**: Penggunaan file_get_contents() untuk mengambil data dari sumber eksternal sangat berisiko. Ini bisa digunakan untuk mendownload skrip atau data yang merusak untuk dijalankan pada situs, yang jelas merupakan indikasi serangan.
3. **Modifikasi pada Struktur Kondisional (if Statements) dan Fungsi yang Tidak Ada di Versi Asli:**
    a. Pada **versi yang dimodifikasi**, sejumlah pengecekan dan pengalihan kondisi yang tidak ada dalam versi asli ditambahkan, seperti pengalihan pada kondisi is_bot(), serta pemeriksaan yang berhubungan dengan cara konten halaman dimuat atau dihentikan.
    b. **Analisis**: Modifikasi semacam ini dapat mempengaruhi cara situs berfungsi, misalnya dengan memblokir pengunjung yang terdeteksi sebagai bot atau dengan mengubah cara template WordPress dimuat, yang dapat merusak fungsionalitas normal situs.

4. **File Eksternal dan Pengalihan Berbahaya (Hacked Version):**
    a. Kode dimodifikasi juga memanfaatkan file eksternal dari **paste.deadsec.my.id**, yang mengindikasikan bahwa penyerang mencoba menyembunyikan atau menyamarkan aktivitas mereka dengan memuat informasi atau kode dari situs eksternal yang dikendalikan oleh penyerang.
    b. **Analisis**: Penggunaan file eksternal semacam ini adalah tanda pasti dari **serangan injeksi** atau **pengalihan berbahaya** yang dapat digunakan untuk mengeksploitasi situs lebih lanjut atau mencuri data pengguna.

Perbandingan dengan Versi Asli:

Versi asli dari **template-loader.php** di WordPress **6.7.2** tidak memiliki fungsi is_bot() atau pemanggilan ke URL eksternal. Fungsi yang ada lebih terbatas pada penanganan **request untuk favicon, robots.txt, dan feed**, serta pemrosesan template berdasarkan kondisi standar yang sudah ada pada WordPress.

- **Analisis**: Versi asli tidak menyertakan mekanisme untuk memanipulasi bot atau memuat data eksternal, yang menunjukkan bahwa perubahan yang dilakukan pada versi yang dimodifikasi bertujuan untuk mengeksploitasi situs atau mengarahkan bot ke halaman berbahaya. Modifikasi ini merupakan indikasi kuat dari **serangan injeksi atau penyisipan kode berbahaya** yang dapat mengubah perilaku normal situs.



Gambar 5. Proses Analisa Code

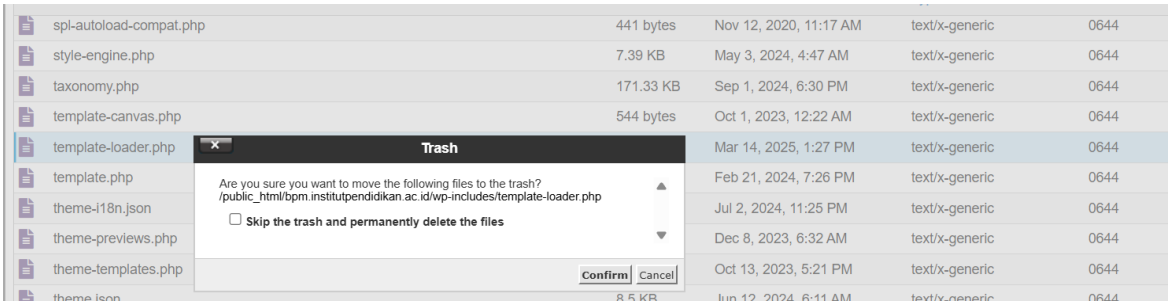
Dari analisis di atas, dapat disimpulkan bahwa **file template-loader.php yang telah dimodifikasi** menunjukkan adanya **serangan injeksi** atau **perubahan berbahaya** yang dapat memanipulasi cara situs berfungsi, mengalihkan pengunjung, atau bahkan memanipulasi hasil pencarian mesin pencari (SEO). Modifikasi ini berpotensi membahayakan integritas situs WordPress, dan sangat disarankan untuk segera memperbaiki file tersebut dengan menggantinya dengan versi asli untuk mengembalikan fungsionalitas yang normal dan menghilangkan potensi eksploitasi lebih lanjut.

Implementasi Solusi Mitigasi

Dalam mengatasi serangan yang terjadi pada situs WordPress, beberapa langkah mitigasi yang efektif diterapkan untuk memperbaiki dan mengembalikan integritas situs. Langkah-langkah mitigasi yang dilakukan meliputi:

1. **Penghapusan File yang Telah Diretas (*Hacked*)**

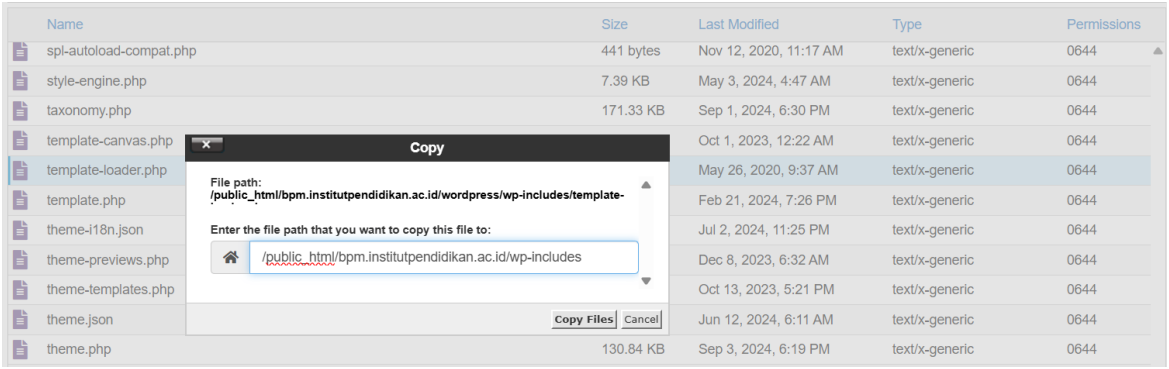
   Langkah pertama adalah **menghapus file template-loader.php** yang telah dimodifikasi oleh penyerang. File ini dimodifikasi untuk menyisipkan kode berbahaya, dan menghapusnya adalah langkah pertama yang krusial untuk memutuskan jalur akses yang telah disalahgunakan oleh peretas.



Gambar 6. Proses Penghapusan

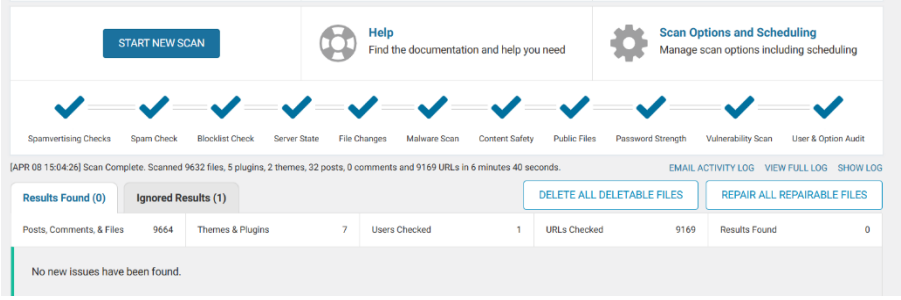
2. **Pengunduhan Ulang dan Penggantian dengan File Asli**

   Setelah file yang terinfeksi dihapus, langkah berikutnya adalah **mengunduh ulang file versi asli dari WordPress versi terbaru**. Versi terbaru ini diunduh dari sumber resmi **WordPress.org** untuk memastikan bahwa file yang digunakan bebas dari modifikasi atau kerentanannya.



Gambar 7. Proses Copy File Template-loader.php Versi asli

3. **Pemindaian Ulang Menggunakan Wordfence**

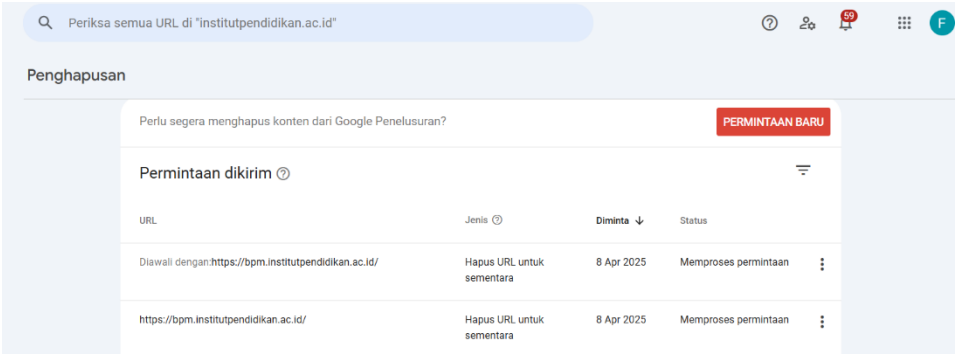


Gambar 8. Proses Scanning Wordfence

Setelah penggantian file selesai, dilakukan **pemindaian ulang menggunakan Wordfence** untuk memastikan bahwa situs tidak lagi terinfeksi dan bebas dari modifikasi berbahaya. Pemindaian ini untuk memastikan bahwa semua file telah dipulihkan dengan benar, dan tidak ada file yang dimodifikasi atau disisipkan kembali oleh penyerang. Verifikasi bahwa mitigasi berhasil dilakukan, serta untuk memastikan bahwa situs sekarang aman dari potensi serangan lanjutan.

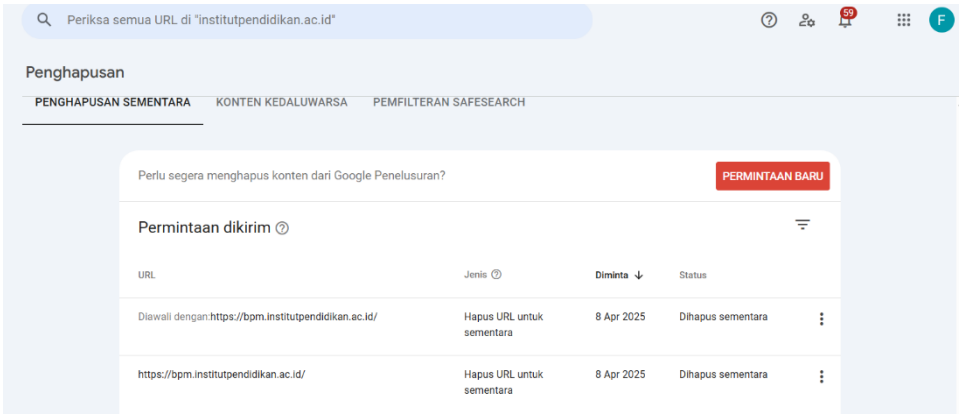
4. **Penghapusan Halaman Ilegal di Google Search**

Setelah Langkah terakhir melibatkan pengajuan permintaan untuk menghapus indeks Google dari halaman-halaman ilegal yang masih muncul dalam hasil pencarian, meskipun telah dihapus dari situs. Langkah yang diambil adalah memanfaatkan Google Search Console untuk mengajukan permohonan penghapusan halaman yang teridentifikasi sebagai pengarah ke konten berbahaya atau yang telah terinfeksi.



Gambar 9. Proses Permintaan Penghapusan Halaman Ilegal Indexing

Halaman yang masih terdeteksi dalam hasil pencarian Google, meskipun telah dihapus dari situs, harus dihapus sepenuhnya dari indeks Google untuk menghindari dampak negatif lebih lanjut pada reputasi situs selain itu menjamin bahwa tidak ada jejak dari halaman yang terinfeksi atau konten berbahaya yang tersisa di mesin pencari, dengan demikian mengurangi potensi kerusakan reputasi situs dan mencegah pengunjung tertipu untuk mengakses halaman yang berbahaya.



Gambar 10. Permintaan Penghapusan Berhasil dilakukan

KESIMPULAN DAN SARAN

Studi ini berhasil mengidentifikasi dan menangani masalah injeksi AMP pada situs WordPress yang telah terinfeksi oleh serangan yang dikendalikan oleh platform perjudian online. Berdasarkan hasil penelitian, dapat disimpulkan bahwa:

1. **Serangan Injeksi AMP**: Para peretas menyalahgunakan situs dengan mengubah file penting WordPress, seperti template-loader.php, dan menyisipkan kode berbahaya. Serangan ini menyebabkan pengalihan yang merugikan dan mengancam integritas situs, dengan potensi pencurian data dan perubahan perilaku situs yang signifikan.
2. **Solusi Mitigasi yang Efektif**: Tindakan mitigasi yang dilakukan, seperti penghapusan file terinfeksi, pengunduhan versi asli file, serta pemindaian ulang dengan Wordfence, terbukti efektif dalam mengembalikan situs ke kondisi aman. Selain itu, penghapusan halaman ilegal melalui Google Search Console membantu menghilangkan jejak konten berbahaya yang telah terindeks oleh mesin pencari, mengurangi potensi kerusakan lebih lanjut.
3. **Pentingnya Pemeliharaan Rutin**: Proses pemulihan situs menegaskan pentingnya pemeliharaan rutin pada file-file inti WordPress dan plugin yang digunakan. Pembaruan sistem secara berkala dan penggunaan alat keamanan yang kuat, seperti Wordfence, sangat diperlukan untuk mencegah eksploitasi yang lebih lanjut dan untuk memastikan bahwa situs tetap aman.

Implikasi Lebih Luas: Penelitian ini tidak hanya memberikan wawasan praktis dalam menangani serangan injeksi AMP di situs WordPress, tetapi juga memperlihatkan tantangan yang dihadapi oleh pengelola situs WordPress dalam melindungi situs dari berbagai jenis serangan siber. Hal ini mempertegas pentingnya pendekatan keamanan yang proaktif, serta penerapan strategi pemeliharaan dan mitigasi yang terorganisir dengan baik.

Keterbatasan Penelitian

Studi ini memiliki keterbatasan yang perlu diperhatikan, antara lain:

- **Sampel yang Terbatas**: Penelitian ini hanya menguji satu situs yang terinfeksi oleh serangan AMP, yang mungkin tidak mewakili keragaman kasus yang lebih luas pada situs WordPress lainnya yang menghadapi jenis serangan serupa.
- **Penerapan Solusi Mitigasi**: Meski mitigasi yang diterapkan efektif pada situs yang diuji, tidak dapat dipastikan bahwa solusi ini akan bekerja secara optimal pada situs dengan konfigurasi yang berbeda atau serangan yang lebih canggih.
- **Terbatasnya Penggunaan Alat**: Fokus utama pada penggunaan Wordfence untuk pemindaian dan mitigasi mungkin membatasi cakupan penelitian terhadap alat-alat keamanan lainnya yang mungkin juga efektif dalam mendeteksi dan mengatasi serangan AMP.

Saran

Berdasarkan temuan dan keterbatasan yang ada, beberapa saran untuk penelitian lanjutan meliputi:

1. **Pengujian pada Situs yang Lebih Beragam**: Penelitian lanjutan sebaiknya mencakup pengujian pada berbagai situs WordPress yang terinfeksi AMP, dengan memperhatikan berbagai konfigurasi situs dan jenis serangan yang lebih bervariasi, guna memperoleh gambaran yang lebih komprehensif.
2. **Pengembangan Metode Mitigasi yang Lebih Lanjut**: Mengingat bahwa teknik injeksi AMP bisa semakin kompleks, penelitian lebih lanjut perlu mengembangkan dan menguji metode mitigasi yang lebih canggih, termasuk penggunaan alat keamanan yang lebih beragam dan pendekatan berbasis AI untuk mendeteksi perubahan yang tidak sah pada kode sumber situs.
3. **Evaluasi Keamanan dengan Berbagai Alat**: Untuk mendapatkan pemahaman yang lebih luas tentang efektivitas alat keamanan dalam menangani injeksi AMP, penelitian lanjutan dapat membandingkan kinerja **Wordfence** dengan alat keamanan lainnya dalam mendeteksi dan mengatasi kerentanannya di situs WordPress.

DAFTAR PUSTAKA

Buchyk, S., & Kuroiedov, A. (2024). Javascript-Based Web Exploit Analysis Model. *Information Systems and Technologies Security*, *2 (8)*, 17–25. https://doi.org/10.17721/ists.2024.8.17-25

Chloe Chamberland. (2023, October 19). *Wordfence Intelligence Weekly WordPress Vulnerability Report*. Https://Www.Wordfence.Com/Blog/2023/10/Wordfence-Intelligence-Weekly-Wordpress-Vulnerability-Report-October-9-2023-to-October-15-2023/.

Dylan Duncan. (2023, August). *Google AMP – The Newest of Evasive Phishing Tactic*. Https://Cofense.Com/Blog/Google-Amp-the-Newest-of-Evasive-Phishing-Tactic/.

Experiencia Implementando AMP en WordPress. (2020). *Mosaic*, *183*. https://doi.org/10.7238/m.n183.2025

Kayleigh Martin. (2025, February 26). *Fake WordPress Plugin Impacts SEO by Injecting Casino Spam*. Https://Blog.Sucuri.Net/2025/02/Fake-Wordpress-Plugin-Impacts-Seo-by-Injecting-Casino-Spam.Html?Utm_source=chatgpt.Com.

Laksmiati, D. (2023). Vulnerability Assessment With Network-Based Scanner Method for Improving Website Security. *Journal of Computer Networks Architecture and High Performance Computing*, *5*(1), 38–45. https://doi.org/10.47709/cnahpc.v5i1.1991

Liu, A. X., & Gouda, M. G. (2009). Firewall Policy Queries. *Ieee Transactions on Parallel and Distributed Systems*, *20*(6), 766–777. https://doi.org/10.1109/tpds.2008.263

Martínez, J. M., Aledo-Hernández, A.-J., Guillén, A., Sanchez-Iborra, R., & Cano, M. (2018). A Comparative Study of Web Content Management Systems. *Information*, *9*(2), 27. https://doi.org/10.3390/info9020027

Nair, S. (2024). Securing Against Advanced Cyber Threats: A Comprehensive Guide to Phishing, XSS, and SQL Injection Defense. *Journal of Computer Science and Technology Studies*, *6*(1), 76–93. https://doi.org/10.32996/jcsts.2024.6.1.9

Patchstack. (2023, December 27). *WordPress Schema & Structured Data for WP & AMP Plugin <= 1.23 is vulnerable to Cross Site Scripting (XSS)*. Https://Patchstack.Com/Database/Wordpress/Plugin/Schema-and-Structured-Data-for-Wp/Vulnerability/Wordpress-Schema-Structured-Data-for-Wp-Amp-Plugin-1-23-Cross-Site-Scripting-Xss-Vulnerability.

Ruohonen, J. (2019). *A Demand-Side Viewpoint to Software Vulnerabilities in WordPress Plugins*. 222–228. https://doi.org/10.1145/3319008.3319029

Sead Fadilpašić. (2025, February 21). *Major website hijacking scam sees over 35,000 sites attacked, redirected to gambling sites, so be on your guard*. Https://Www.Techradar.Com/pro/Security/Major-Website-Hijacking-Scam-Sees-over-35-000-Sites-Attacked-Redirected-to-Gambling-Sites-so-Be-on-Your-Guard?Utm_source=chatgpt.Com.

Sucuri Malware Research Team. (2023, December 28). *WordPress Vulnerability & Patch Roundup December 2023*. Https://Blog.Sucuri.Net/2023/12/Wordpress-Vulnerability-Patch-Roundup-December-2023.Html.

Veronika Telychko. (2023, August 4). *Google AMP Exploited in Phishing Attacks Targeting Enterprise Users*. Https://Socprime.Com/Blog/Google-Amp-Exploited-in-Phishing-Attacks-Targeting-Enterprise-Users/.

Zainab. (2025, July 12). *Casino Redirect Hack in WordPress: Full Cleanup & Prevention Guide (2025)*. Https://Easifytechnologies.Com/Wordpress-Casino-Redirect-Hack-Removal/?Utm_source=chatgpt.Com.