OS Command Injection Leading to Remote Code Execution (RCE)

Critical **ryuring** published **GHSA-qxmc-6f24-g86g** 2 weeks ago

Package

php **baserproject/basercms** (Composer)

Affected versions	Patched versions
<= 5.2.2	5.2.3

Description

Summary (English)

baserCMS contains an OS command injection vulnerability in the core update functionality.
An authenticated administrator can execute arbitrary OS commands on the server due to improper handling of user-controlled input that is directly passed to `exec()` without sufficient validation or escaping.

概要

baserCMS のコアアップデート機能において、管理画面から送信されるパラメータの一部が、適切な検証やエスケープを行わないまま `exec()` 関数に渡されている箇所が存在します。
この問題により、**認証済みの CMS 管理者がサーバー上で任意の OS コマンドを実行できる（Remote Code Execution, RCE）** 状態となります。

本脆弱性は、画面操作や CSRF の欠如といった UI レベルの問題ではなく、**サーバー側で受け取った入力値を OS コマンドとして直接実行している設計**に起因します。
そのため、UI 上でボタンが非表示になっている場合や、CakePHP の CSRF / FormProtection（SecurityComponent）により正規の POST リクエストのみが受理される場合であっても、**管理者セッションにおいて正規トークンを含むリクエストが処理されれば攻撃は成立します**。

脆弱性情報

項目	内容
CWE	CWE-78: Improper Neutralization of Special Elements used in an OS Command
影響	Remote Code Execution (RCE)
深刻度	Critical
攻撃条件	管理者権限が必要
再現性	再現可能（複数回確認）
検証環境	baserCMS 5.2.2（Docker / 開発環境）

影響を受ける箇所

- **Controller**
  - `PluginsController::get_core_update()`
- **Service**
  - `PluginsService::getCoreUpdate()`
- **Affected Endpoint**
  - `/baser/admin/baser-core/plugins/get_core_update`

技術的詳細

脆弱なコードフロー

```
PluginsController::get_core_update()
  ↓ POST データから php パラメータを取得
PluginsService::getCoreUpdate($targetVersion, $php, $force)
  ↓ $php を検証・エスケープせずにコマンド文字列へ連結
exec($command)
```

該当コード（抜粋）

PluginsController.php

```
$service->getCoreUpdate(
    $request->getData('targetVersion') ?? '',
    $request->getData('php') ?? 'php',
    $request->getData('force'),
);
```

PluginsService.php

```
$command = $php . ' ' . ROOT . DS . 'bin' . DS . 'cake.php composer ' .
           $targetVersion . ' --php ' . $php . ' --dir ' . TMP . 'update';

exec($command, $out, $code);
```

`$php` パラメータはユーザー入力であり、以下の対策はいずれも行われていません。

- 許可リストによる制限
- 正規表現による検証
- `escapeshellarg()` 等によるエスケープ

攻撃シナリオ

1. 攻撃者が CMS 管理者としてログイン
2. 管理画面のコアアップデート機能に対して POST リクエストを送信
3. `php` パラメータに OS コマンドを含む文字列を指定
4. サーバー側で `exec()` が実行され、任意の OS コマンドが実行される

攻撃入力例（概念）

```
php=php;id>/tmp/rce_test;#
```

検証結果（PoC）

実行結果

```
$ docker exec bc-php cat /tmp/rce_test
uid=1000(www-data) gid=1000(www-data) groups=1000(www-data)
```

上記より、`www-data` 権限で OS コマンドが実行可能であることを確認しました。

補足

- 管理画面上の正規フロー（ブラウザ）から再現可能
- CSRF / FormProtection トークンを含む正規リクエストでも成立
- 再現失敗時（400 / 403）の切り分けも確認済み
- curl 等による HTTP リクエストの再送でも再現可能であることを確認済み（正規のトークンを含む同一リクエストの再送）

影響

本脆弱性を悪用された場合、以下が可能となります。

- サーバー情報の取得
- 任意ファイルの読み取り・書き込み
- アプリケーション設定情報（DB 認証情報等）の取得
- アプリケーション権限境界を越えた OS レベル操作

管理者権限が必要ではあるものの、**アプリケーション層から OS 層へ影響が及ぶ設計上の問題**であり、影響は大きいと考えます。

修正案

推奨対応

- ユーザー入力から PHP 実行パスを受け取らない
- 実行する PHP はサーバー側で固定し、`PHP_BINARY` 定数を使用する

```
$php = escapeshellarg(PHP_BINARY);
```

補足修正案

- コマンドライン引数（バージョン番号、ディレクトリ等）についても `escapeshellarg()` によるエスケープを行う
- 可能であれば、シェル解釈を介さない実行方法（配列形式、Process クラス等）の利用を検討する

代替案（非推奨）

- PHP 実行パスに対する許可リスト検証
- 正規表現検証と `escapeshellarg()` の併用

ただし、**攻撃面（Attack Surface）を削減する観点から、ユーザー入力自体を排除する設計を推奨**します。

補足事項

- 本件は UI 表示制御（ボタンの表示・非表示）とは独立して成立します
- エンドポイントが存在する限り、正規トークンを含むリクエストが処理されれば攻撃可能です
- 設計上の入力取り扱いに起因する問題であり、CSRF や UI 制御のみでは防止できません

結論

baserCMS のコアアップデート機能において、ユーザー入力が検証されないまま `exec()` に渡される設計上の問題により、**管理者権限を前提としたリモートコード実行（RCE）** が成立します。
本脆弱性は入力検証および設計の見直しにより修正可能であり、早期の対応が望まれます。

Severity

Critical 9.1 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	High
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	High

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H

CVE ID

CVE-2026-21861

Weaknesses

▶ CWE-78

Credits

kaminuma Reporter