unsafe File Upload Leading to Remote Code Execution (RCE)

High **ryuring** published **GHSA-hv78-cwp4-8r7r** 3 days ago

Package

BcDatabasesService.php (baserCMS)

Affected versions

<= 5.2.2

Patched versions

5.2.3

Description

Details

The application's restore function allows users to upload a `.zip` file, which is then automatically extracted. A PHP file inside the archive is included using `require_once` without validating or restricting the filename. An attacker can craft a malicious PHP file within the zip and achieve arbitrary code execution when it is included.

Vector: Malicious ZIP upload + insecure `require_once`

PoC

1. Restore backup

```php
public function restoreDb(array $postData, array $uploaded): bool
{
    set_time_limit(0);

    $postData = array_merge([
        'encoding' => 'UTF-8'
    ], $postData);

    if (BcUtil::isOverPostSize()) {
        throw new BcException(__d('baser_core',
            '送信できるデータ量を超えています。合計で %s 以内のデータを送信してください。',
            ini_get('post_max_size')
        ));
    }

    if (empty($_FILES['backup']['tmp_name'])) {
        if (!empty($uploaded['backup']) && $uploaded['backup']->getError() === 1) {
            $message = __d('baser_core', 'サーバに設定されているサイズ制限を超えています。');
        } else {
            $message = __d('baser_core', 'バックアップファイルが送信されませんでした。');
        }
        throw new BcException($message);
    }

    $tmpPath = TMP . 'schema' . DS;
    if(!is_dir($tmpPath)) {
        (new BcFolder())->create($tmpPath);
    }
    $name = $uploaded['backup']->getClientFileName();
    $uploaded['backup']->moveTo($tmpPath . $name);
    $bcZip = new BcZip();
    if (!$bcZip->extract($tmpPath . $name, $tmpPath)) {                          // extract file shell in zip file
        throw new BcException(__d('baser_core', 'アップロードしたZIPファイルの展開に失敗しました。'));
    }
    unlink($tmpPath . $name);

    $result = true;
    try {
        /* @var \BaserCore\Service\BcDatabaseService $dbService */
        $this->_loadBackup($tmpPath, $postData['encoding']);                     // load file shell
    } catch (\Throwable $e) {
        throw $e;
    }

    $dbService = $this->getService(BcDatabaseServiceInterface::class);
    $dbService->updateSequence();

    $this->resetTmpSchemaFolder();
    BcUtil::clearAllCache();
    return $result;
}
```

2. Load file shell (insecure `require_once`)

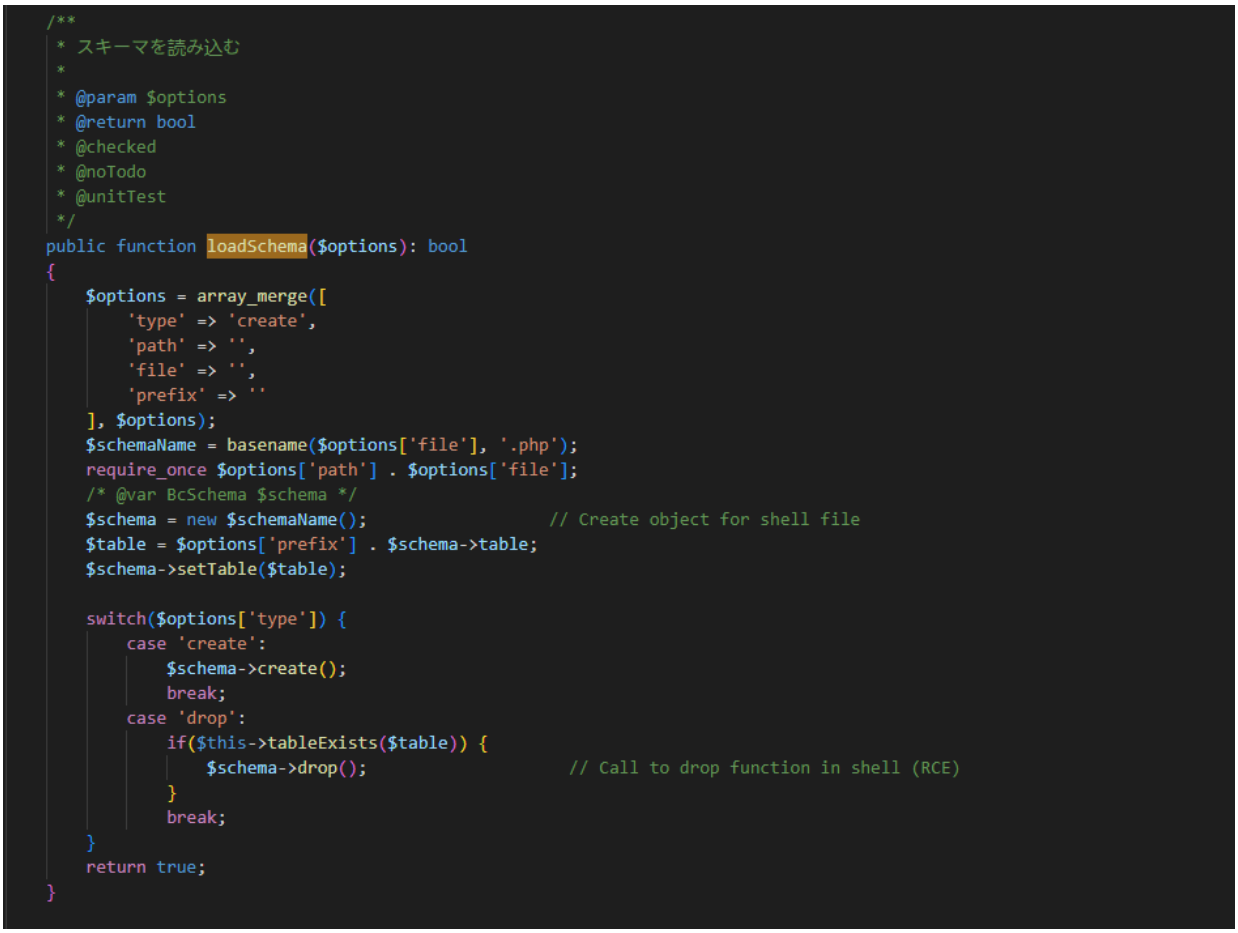
```php
protected function _loadBackup($path, $encoding)
{
    $folder = new BcFolder($path);
    $files = $folder->getFiles();
    if (!is_array($files)) return;

    /* @var BcDatabaseService $dbService */
    $dbService = $this->getService(BcDatabaseServiceInterface::class);

    $prefix = BcUtil::getCurrentDbConfig()['prefix'];

    $db = BcUtil::getCurrentDb();
    $db->begin();
    // テーブルを削除する
    foreach($files as $file) {
        if (!preg_match("/\.php$/", $file)) continue;
        try {
            $dbService->loadSchema([
                'type' => 'drop',
                'path' => $path,
                'file' => $file,
                'prefix' => $prefix
            ]);
        } catch (Throwable $e) {
            $db->rollback();
            throw $e;
        }
    }

    // テーブルを読み込む
    foreach($files as $file) {
        if (!preg_match("/\.php$/", $file)) continue;
        try {
            if (!$dbService->loadSchema([
                'type' => 'create',
                'path' => $path,
                'file' => $file,
                'prefix' => $prefix
            ])) {
                continue;
            }
        } catch (Throwable $e) {
            $db->rollback();
            throw $e;
        }
    }
```



```php
/**
 * スキーマを読み込む
 *
 * @param $options
 * @return bool
 * @checked
 * @noTodo
 * @unitTest
 */
public function loadSchema($options): bool
{
    $options = array_merge([
        'type' => 'create',
        'path' => '',
        'file' => '',
        'prefix' => ''
    ], $options);
    $schemaName = basename($options['file'], '.php');
    require_once $options['path'] . $options['file'];
    /* @var BcSchema $schema */
    $schema = new $schemaName();                    // Create object for shell file
    $table = $options['prefix'] . $schema->table;
    $schema->setTable($table);

    switch($options['type']) {
        case 'create':
            $schema->create();
            break;
        case 'drop':
            if($this->tableExists($table)) {
                $schema->drop();                    // Call to drop function in shell (RCE)
            }
            break;
    }
    return true;
}
```

Impact

Remote Code Execution (RCE)

Severity

High 8.7 / 10

CVSS v3 base metrics

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

Learn more about base metrics

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

CVE ID

CVE-2025-32957

Weaknesses

No CWEs

Credits

MinhhhCuonggg	Finder
Vatvo69	Finder