Rapport d'Analyse des Logs Apache2

Date du rapport : 23 février 2026
Période analysée : 17 février 2025 (10h14 – 10h33)
Serveur : Apache/2.4.57 (Debian)
Fichiers analysés :

- `Apache_bidon_access.txt` (logs d'accès)
- `Apache_bidon_error.txt` (logs d'erreurs)

Sommaire des Incidents Critiques

Type dIncident	Gravité	IP Source	Description	Preuve
Injection SQL	**Critique**	203.0.113.45	Tentative d'exfiltration de données via `UNION SELECT` sur `/search.php`.	Logs access + ModSecurity (ID 942100).
Traversal de répertoire	**Élevée**	192.0.2.55	Accès non autorisé à `/etc/passwd` via encodage URL (`%2e%2e`).	Erreur 400 + ModSecurity (ID 930120).
Fuite d'information	**Moyenne**	198.51.100.23	Tentative d'accès au fichier `.env` (contenant des crédentials).	Log access (404).
Erreur PHP critique	**Élevée**	203.0.113.88	Fonction `connect_database()` non définie dans `/api.php` (ligne 12).	PHP Fatal Error.
Surcharge du serveur	**Critique**	-	`MaxRequestWorkers` atteint → Déni de service potentiel.	`AH00484` (10:33:02).
Configuration Apache				

Élevée	-	Échec de résolution DNS pour `debian-web-01` → Arrêt du service.	`AH00059` (10:31:12).	
Variable non définie	**Faible**	185.199.108.12	Clé `user_id` manquante dans `/login.php` (ligne 42).	PHP Warning.

Analyse Détaillée

1. Attaques et Comportements Suspects

A. Injection SQL (203.0.113.45)

- **Requête malveillante :**

```
GET /search.php?q=1 UNION SELECT username,password FROM users--
```

- **Détails :**
  - Utilisation de **sqlmap** (outil automatisé d'injection SQL).
  - Détecté par **ModSecurity** (règle `942100`).
  - Réponse HTTP **403** (blocage réussi), mais la tentative révèle une vulnérabilité potentielle.
- **Risque :** Exfiltration de données sensibles (mots de passe en clair ?).
- **Visualisation :**

```
graph LR
  A[Attaquant] ⟶|SQLi| B[/search.php]
  B ⟶|403| C[ModSecurity]
  C ⟶|Bloqué| A
```

B. Traversal de Répertoire (192.0.2.55)

- **Requête malveillante :**

```
GET /../../etc/passwd HTTP/1.1
```

- Encodage URL (`%2e%2e` = `..`).
- **Détails :**
  - Détecté par ModSecurity (règle `930120`).
  - Réponse HTTP **400** (requête invalide).
- **Risque :** Accès aux fichiers système si le serveur est mal configuré.
- **Recommandation :** Vérifier les permissions et désactiver `FollowSymLinks`.

C. Tentative d'Accès à `.env` (198.51.100.23)

- **Requête :**

```
GET /.env HTTP/1.1
```

- **Détails :**
  - Fichier `.env` souvent utilisé pour stocker des **crédentials** (DB, API keys).
  - Réponse **404** (fichier non trouvé ou protégé).
- **Risque :** Fuites de données si le fichier est accessible.
- **Action :** Vérifier que `.env` est **hors de la racine web** (`/var/www/`).

---

2. Erreurs Techniques

A. Erreur PHP Critique (203.0.113.88)

- **Erreur :**

```
PHP Fatal error: Uncaught Error: Call to undefined function
connect_database() in /var/www/html/api.php:12
```

- **Cause probable :**
  - Fonction `connect_database()` non déclarée ou **extension PHP manquante** (ex: `pdo_mysql`).
  - Fichier de configuration non inclus.
- **Impact :** Indisponibilité de `/api.php` (réponse HTTP **500**).
- **Solution :**
  - Vérifier les `include` dans `api.php`.
  - Installer les extensions PHP manquantes :

```
sudo apt install php-mysql
```

B. Variable PHP Non Définie (185.199.108.12)

- **Avertissement :**

```
PHP Warning: Undefined array key "user_id" in /var/www/html/lo-
gin.php on line 42
```

- **Cause :** Accès à `$_POST['user_id']` ou `$_GET['user_id']` sans vérification.
- **Risque :** Comportement imprévisible ou faille de sécurité (ex: **injection de variables**).
- **Solution :** Toujours valider les entrées :

```
$user_id = $_POST['user_id'] ?? null;
if (!$user_id) { /* Gérer l'erreur */ }
```

C. Surcharge du Serveur (MaxRequestWorkers)

- **Erreur :**

```
AH00484: server reached MaxRequestWorkers setting
```

- **Cause :**
  - Trop de requêtes simultanées (possible **DDoS** ou trafic légitime élevé).
  - Valeur par défaut de `MaxRequestWorkers` trop basse pour la charge.
- **Solution :**
  - Augmenter `MaxRequestWorkers` dans `/etc/apache2/mods-enabled/mpm_event.conf` :

    ```
    MaxRequestWorkers 250  # Valeur à ajuster selon la RAM
    ```

  - Surveiller le trafic avec **fail2ban** ou **mod_evasive**.

D. Échec de Résolution DNS (debian-web-01)

- **Erreur :**

```
AH00059: ap_run_pre_config failed, exiting
```

- **Cause :**
  -

Le nom d'hôte `debian-web-01` n'est pas résolu (fichier `/etc/hosts` ou DNS mal configuré).

- **Solution :**
  - Vérifier `/etc/hosts` :

```
127.0.1.1 debian-web-01
```

  - Redémarrer Apache :

```
sudo systemctl restart apache2
```

---

3. Statistiques et Visualisations

A. Répartition des Codes HTTP (Access Logs)

Code	Count	Signification
200	2	Succès
400	1	Requête invalide (traversal)
403	1	Accès interdit (SQLi)
404	1	Fichier non trouvé (.env)
500	1	Erreur serveur (PHP)
503	1	Service indisponible (surcharge)

B. Timeline des Événements Critiques

```
timeline
    title Chronologie des Incidents (17/02/2025)
    section Attaques
        10:18 : Injection SQL (203.0.113.45)
        10:19 : Tentative .env (198.51.100.23)
        10:22 : Traversal (192.0.2.55)
    section Erreurs
        10:16 : Variable non définie (login.php)
        10:25 : Erreur PHP critique (api.php)
        10:31 : Échec DNS (debian-web-01)
        10:33 : Surcharge (MaxRequestWorkers)
```

C. Origines Géographiques (IPs)

IP	Pays (Estimation)	Type d'Activité
203.0.113.45	APNIC (Asie)	Injection SQL
192.0.2.55	Réservé (Docs)	Traversal
198.51.100.23	ARIN (Amérique)	Accès .env
185.199.108.12	RIPE (Europe)	Variable non définie

Recommandations

1. Sécurité

Action	Priorité	Détails
Mettre à jour ModSecurity	Haute	Ajouter des règles personnalisées pour bloquer les attaques SQL/Traversal.
Déplacer `.env`	Critique	Placer hors de `/var/www/` et restreindre les permissions (`chmod 600`).
Désactiver `FollowSymLinks`	Moyenne	Dans `/etc/apache2/apache2.conf` ou `.htaccess`.
Activer fail2ban	Haute	Bloquer les IPs après 3 tentatives d'injection.
Chiffrer les mots de passe	Critique	Utiliser `password_hash()` en PHP si les mots de passe sont en clair.

2. Performances

Action	Priorité	Détails
Augmenter `MaxRequestWorkers`	Haute	Ajuster en fonction de la RAM (`free -h`).
Optimiser PHP	Moyenne	Installer `opcache` et vérifier les extensions manquantes.
Surveiller les logs	Basse	Utiliser `logrotate` et un outil comme **Graylog** ou **ELK**.

3. Configuration

Action	Priorité	Détails
Corriger `/etc/hosts`		

Haute		Ajouter `127.0.1.1 debian-web-01`.
Vérifier les includes PHP	Moyenne	S'assurer que `api.php` charge les dépendances nécessaires.
Tester les sauvegardes	Basse	Vérifier que les backups de `/var/www/` et `/etc/apache2/` fonctionnent.

Conclusion et Tendances

Points Clés

1. **Vulnérabilités Critiques :**
   - Le serveur est ciblé par des **attaques automatisées** (SQLi, traversal).
   - Risque élevé de **fuite de données** (`.env`, `/etc/passwd`).
2. **Problèmes de Configuration :**
   - Erreurs PHP liées à des **fonctions non définies** ou des **variables non vérifiées**.
   - **Surcharge** due à un `MaxRequestWorkers` trop bas.
3. **Manques de Surveillance :**
   - Aucune détection proactive des attaques (ex: **IDS** comme Snort).

Tendances Observées

- **Augmentation des attaques automatisées** (sqlmap, scans de fichiers sensibles).
- **Erreurs PHP récurrentes** liées à un manque de validation des entrées.
- **Problèmes de scalabilité** (limites Apache atteintes).

Prochaines Étapes

1. **Corriger les failles critiques** (SQLi, traversal, `.env`) **sous 24h**.
2. **Automatiser la surveillance** avec :
   - **fail2ban** pour bloquer les IPs malveillantes.
   - **Prometheus + Grafana** pour surveiller la charge.
3. **Auditer le code PHP** pour éliminer les erreurs de variables non définies.
4. **Planifier un test de charge** pour ajuster `MaxRequestWorkers`.

Rédigé par : [Votre Nom]
Contact : admin@domaine.com
Annexes : Fichiers logs complets disponibles sur demande.