Incident Analysis Report

Prepared by: Abdelrahman wael

October 28, 2025

Abstract

This document provides a simple, human-friendly incident analysis summary derived from web server logs and standard forensic procedures. Replace all example values with actual data extracted from your environment.

Summary Table

Item	**Example / Explanation**
Attacker IP	`192.168.1.50`
Origin City	Moscow (from geo-IP lookup)
Vulnerable Script	`vulnerable.php`
First SQLi Request URI	`/vulnerable.php?id=1' OR '1'='1`
Request URI Reading Databases	`/vulnerable.php?id=1 UNION SELECT schema_name FROM information_schema.schemata`
Users Table Name	`users`
Hidden Directory Discovered	`/admin_panel/`
Credentials Used	`admin / 123456`
Malicious Script Uploaded	`shell.php`

1 Detailed Human-Friendly Explanation

1. **Attacker IP:** Check your server log (e.g., `/var/log/nginx/access.log`). Look for SQLi or scan patterns. Example:

```
192.168.1.50 - - [27/Oct/2025:10:34:22] "GET /index.php?id=1' OR '1'='1 HTTP/1.1" 200
```

2. **Origin City:** Use a GeoIP tool such as `https://iplocation.io` to determine where the IP originated.

3. **Vulnerable Script:** Identify which script was targeted in the URL path. Example: `/vulnerable.php`.

4. **First SQLi Attempt:** Check logs by timestamp for the first SQL pattern. Example: `/vulnerable.php?id=1' OR '1'='1`.

5. **Database Enumeration Request:** Look for UNION-based SQL commands referencing `information_schema`. Example:

```
/vulnerable.php?id=1 UNION SELECT schema_name FROM information_schema.schemata
```

6. **Users Table:** Attackers often target a table called `users` or `accounts`. Example query:

   ```
   SELECT * FROM users;
   ```

7. **Hidden Directory:** Review directory scanning attempts (e.g., `/admin_panel/`, `/backup/`, `/uploads/`).

8. **Credentials Used:** Analyze POST data in login requests. Example:

   ```
   POST /login.php username=admin&password=123456
   ```

9. **Malicious Script:** Check uploaded files or `/uploads/` folder for suspicious PHP or JSP shells like `shell.php`.

2 Recommended Next Steps

- Block the attacker's IP at both firewall and web server.
- Patch and sanitize the vulnerable script using prepared statements.
- Reset exposed credentials and enforce MFA.
- Delete any uploaded malicious files and restore from clean backups.
- Review directory permissions and disable directory listing.
- Preserve logs for forensic analysis and future legal procedures.

Notes

Replace all placeholders with real log-derived data. Maintain consistent timestamps and provide screenshots of log evidence where possible.