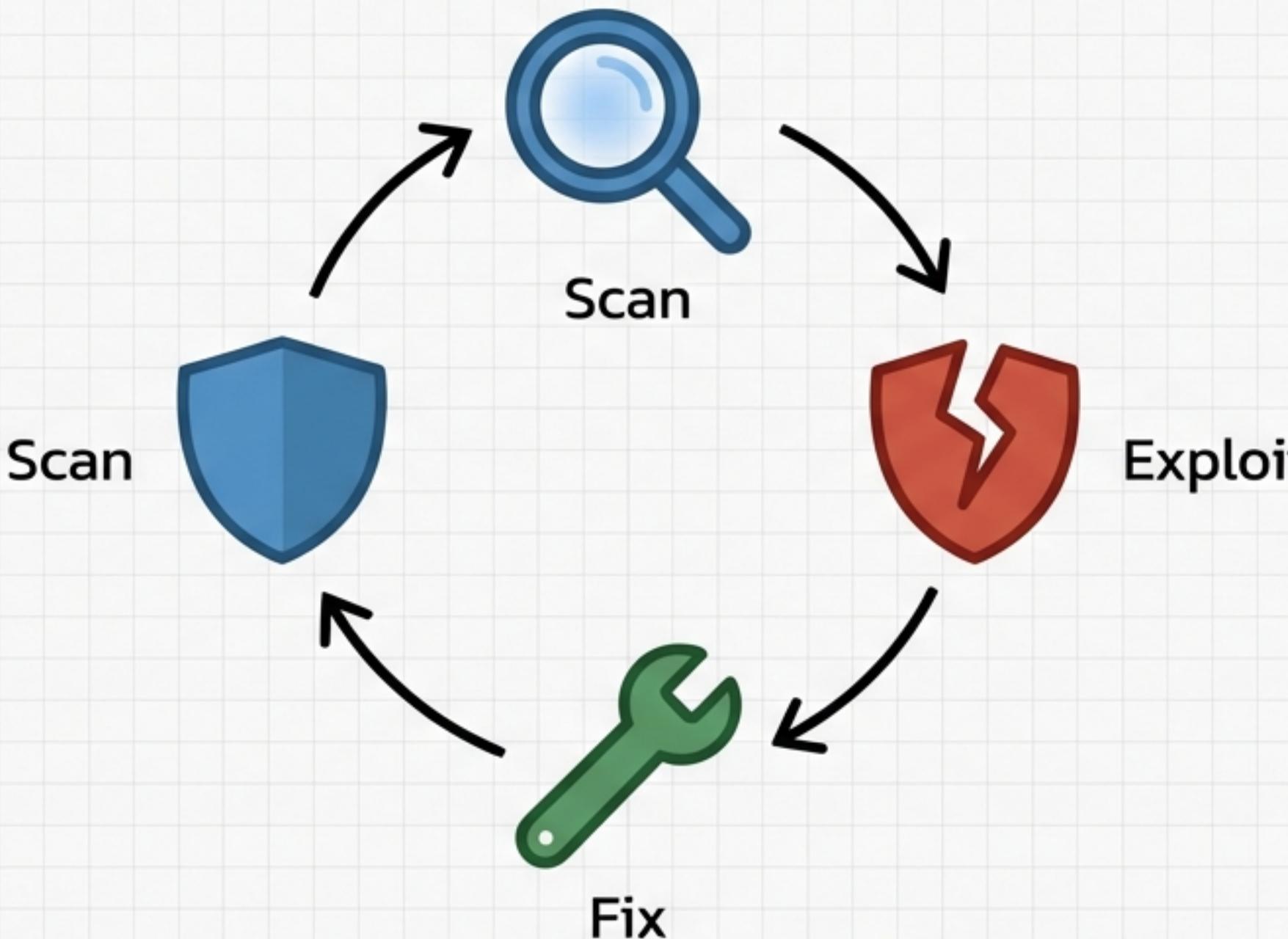


Web Security Hands-on Workshop

เรียนรู้ผ่านการลงมือทำ: Scan • Exploit • Fix • Scan



5 Hours | Localhost Environment

การก่อของเรา



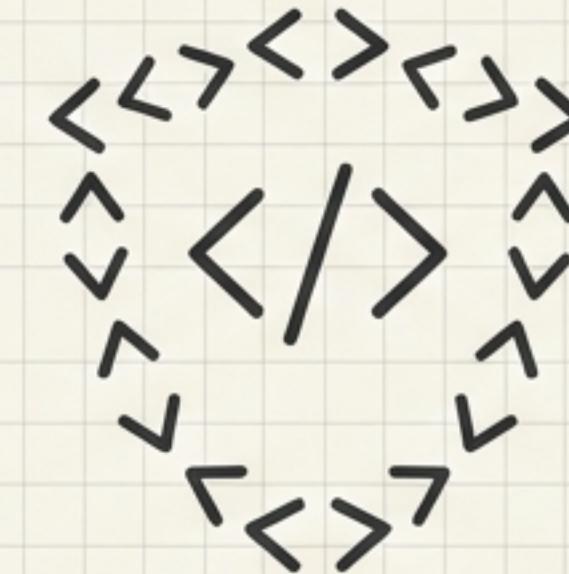
เข้าใจช่องโหว่

เรียนรู้ข้อผิดพลาดที่พบบ่อย
ใน Web Application



เห็นผลกระทบจริง

ทดลองโจมตีเพื่อเห็นความเสียหายด้วยตาตัวเอง



เขียนโค้ดให้ปลอดภัย

ฝึกฝนแนวคิด Secure Design และการแก้ไขโค้ดที่ถูกต้อง

เราจะเรียนรู้ผ่านกระบวนการ: Scan → Exploit → Fix → Scan

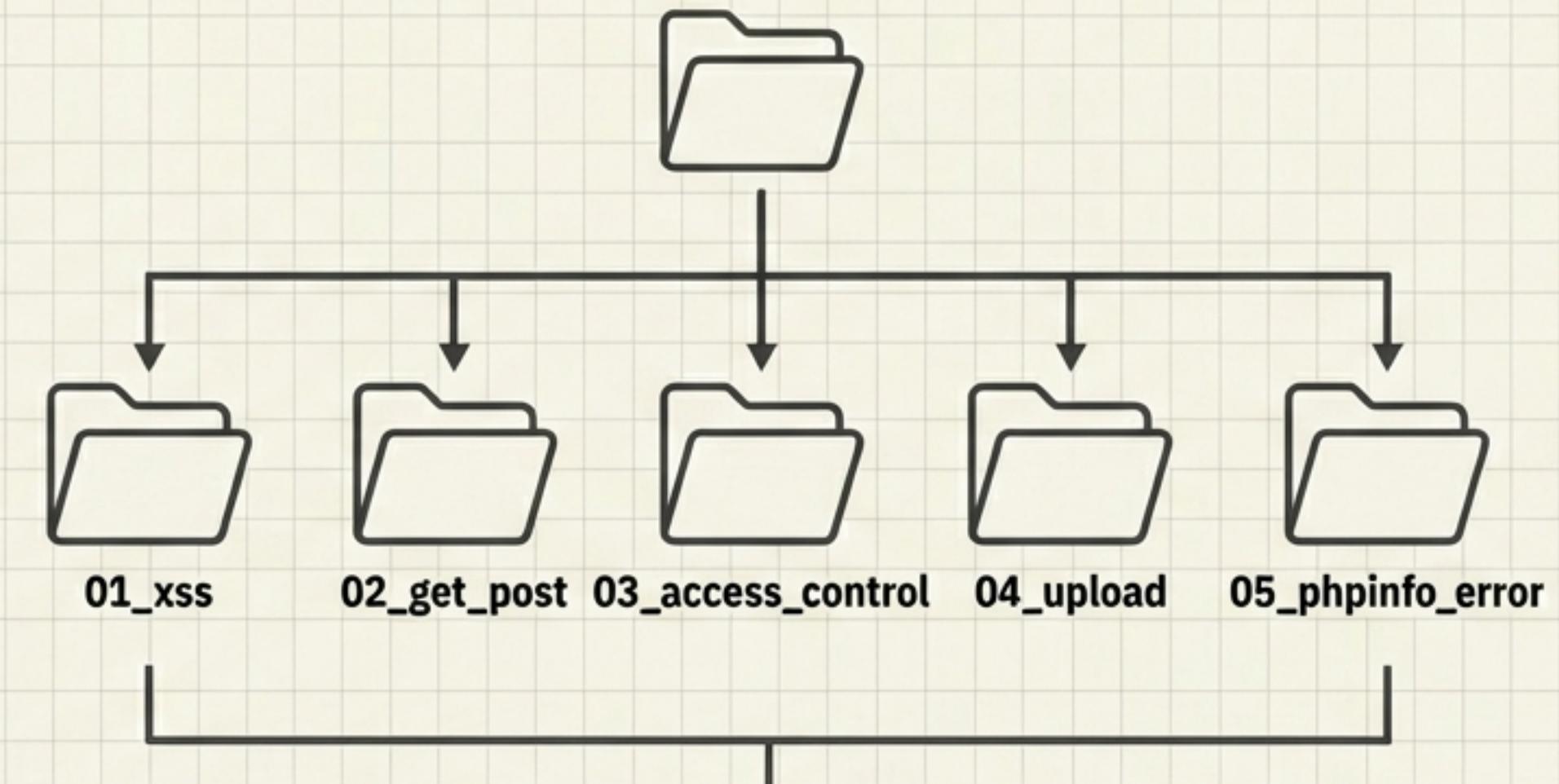
สภาพแวดล้อมการทดสอบ

System Specs

- OS: Windows
- Server: XAMPP (Apache + PHP)
- Tools: OWASP ZAP, Browser

Note: ไม่ใช้ฐานข้อมูล (No Database)
เน้น Logic และ File Handling

C:\xampp\htdocs\web-labs\

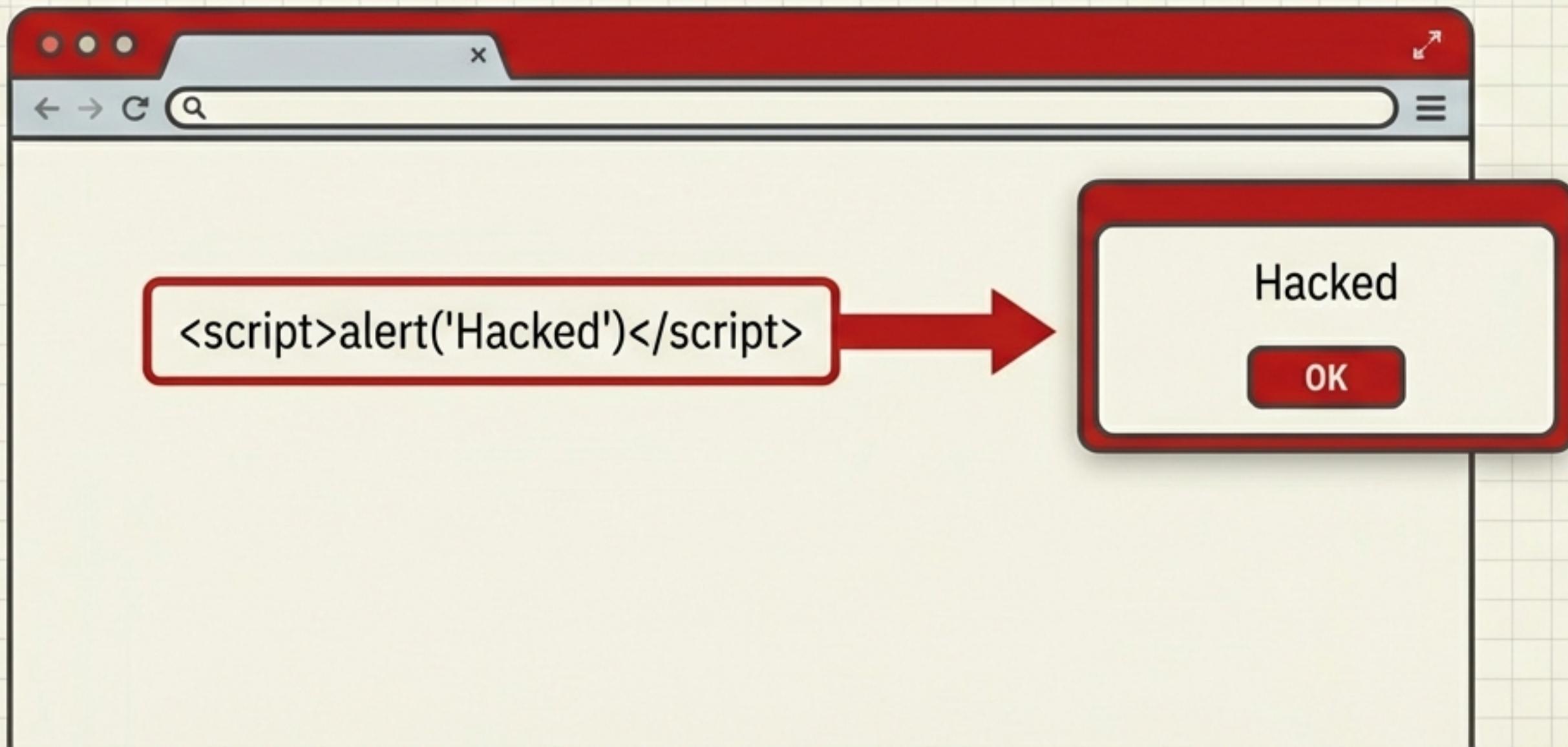


ไฟล์ทดสอบแบ่งเป็น vuln.php (ก่อนแก้) และ secure.php (หลังแก้)

Lab 01: Cross-Site Scripting (XSS)

เมื่อ Input ของผู้ใช้ถูกลายเป็นคำสั่ง

- เกิดจากการนำข้อมูลจากผู้ใช้ไปแสดงผลโดยไม่ตรวจสอบ
- Browser รันโค้ดในสิทธิ์ของเว็บไซต์



Impact

- Cookie Theft
- Redirect
- Content Injection

การป้องกัน XSS: Escape Output

Vulnerable

```
1 // vuln.php  
2 $name = $_GET['name'];  
3 echo $name;
```

User input ถูกแสดงผลตรงๆ

Secure

```
1 // secure.php  
2 $name = $_GET['name'];  
3 echo htmlspecialchars($name);
```

เปลี่ยนอักขระพิเศษเป็น HTML entities

ไม่เชื่อ Input จากผู้ใช้ และต้อง Escape ข้อมูลก่อนแสดงผลเสมอ

Lab 02: Parameter Trust (GET/POST)

ผู้ใช้สามารถแก้ไขข้อมูลได้เสมอ

- นักพัฒนามักเชื่อค่าที่ส่งมาจาก Hidden Field หรือ URL



ระบบทำงานผิดพลาด เพราะเชื่อค่าจาก Client



การป้องกัน Logic Flaw

ตัดสิน Logic ที่ Server เท่านั้น

Validate / Whitelist

ตรวจสอบค่าที่รับมาว่าถูกต้องตามรูปแบบหรือไม่



No Client Trust

ห้ามใช้ Input จาก Client เป็นตัวตัดสินสิทธิ์หรือราคา



Server Authority

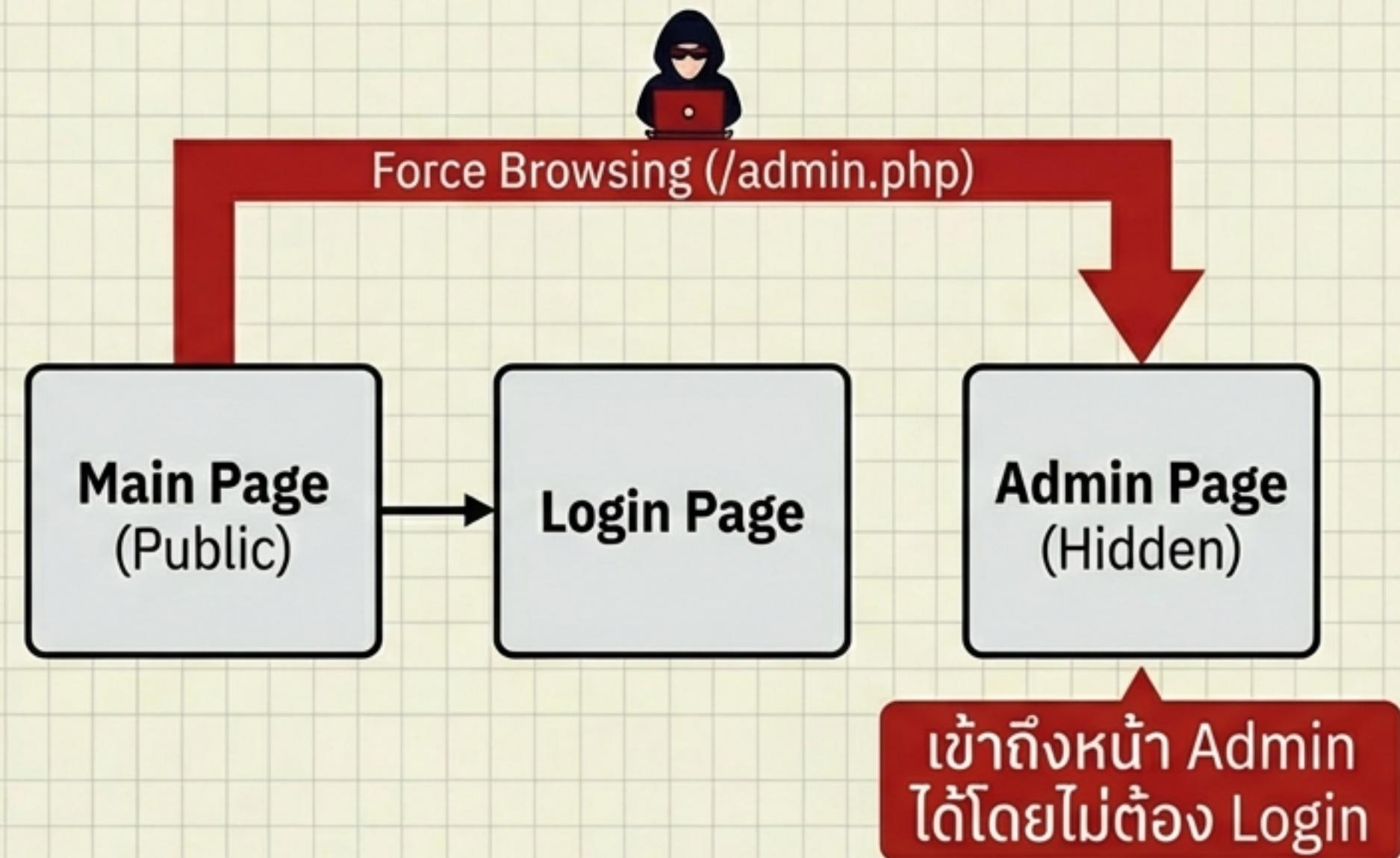
ค้นหาราคาจากฐานข้อมูลหรือ Array ผ่าน Server โดยใช้ ID แทน



Lab 03: Broken Access Control

รู้ URL ไม่ได้แปลว่ามีสิทธิ์

- URL-based access control:
คิดว่าถ้าไม่ทำปุ่ม Link
ผู้ใช้ก็เข้าไม่ได้



ระบบทำงานผิดพลาดเพราะเชื่อค่าจาก Client



การสร้างระบบ Access Control ที่แข็งแกร่ง



Authentication

= ยืนยันตัวตน (Who are you?)



Authorization

= ตรวจสอบสิทธิ์ (What can you do?)

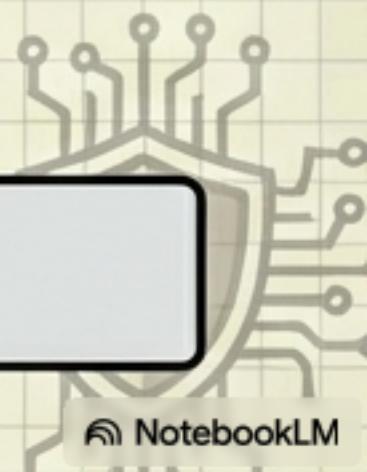
Code Logic

```
session_start();
if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'admin') {
    die('Access Denied');
}
// Show Admin Content
```



ตรวจสอบสิทธิ์ทุกหน้า
(Check permissions on every page)

Policy: ปฏิเสธการเข้าถึงเป็นค่าเริ่มต้น



Lab 04: File Upload Vulnerabilities

ไฟล์อันตรายในคราบรูปภาพ

- ระบบตรวจสอบเพียงแค่นามสกุลไฟล์หรือ Content-Type ที่ Client ส่งมา

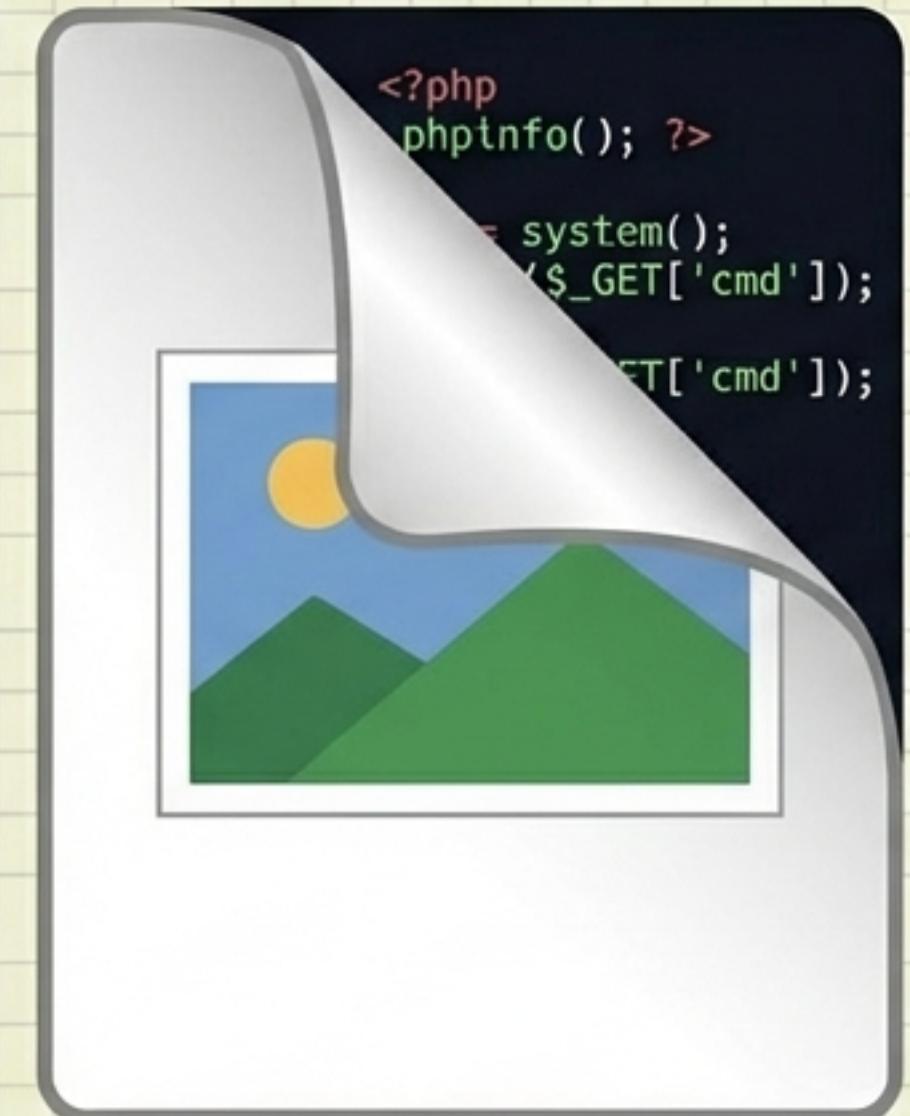


image.jpg.php

Remote Code Execution (RCE):
ผู้โจมตีสามารถสั่งยืดเครื่อง Server ได้



การจัดการไฟล์อัปโหลดอย่างปลอดภัย



Check Real MIME Type

ตรวจสอบเนื้อหาไฟล์จริง ไม่ใช่แค่ Header



Rename File

ตั้งชื่อไฟล์ใหม่เสมอ (Randomize) เพื่อป้องกันการเรียกใช้ไฟล์



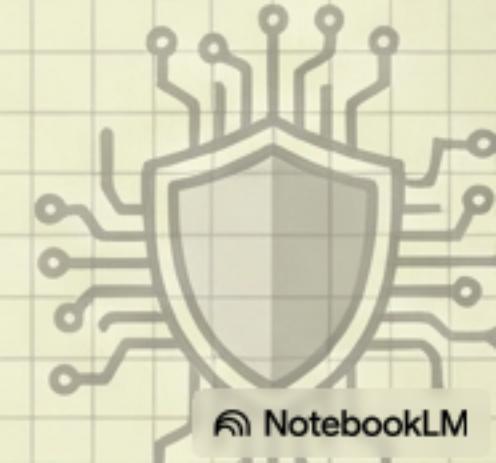
Store Outside Web Root

เก็บไฟล์ไว้นอกโฟลเดอร์ที่เข้าถึงได้ผ่านเว็บ



No Execution

ตั้งค่า Server ไม่ให้รัน Script ในโฟลเดอร์ Upload



Lab 05: Information Disclosure

ข้อมูล Debug คือลายแทงของผู้โจมตี

The Risk

Version 7.4		PHP Info
System		Linux 7.4 112.5-35/eme l-08-25.0:t/k 2.030:a8-4.fill, m86_64
Build Date		Alt 17, 2019 02:00:55
Server API		PHP Canulcare
Virtual Directory Support		disabled
Configuration File (php.ini) Path		/var/www/html/cip/php.ini
Configuration File (php.ini) Scan		/var/oin

Stack Trace
Fatal error: Uncaught Error: Call to undefined function in /var/www/html/app/controllers/UserController.php:35
Stack trace:
#0 /var/www/html/app/core/Router.php(62): UserController->index()
#1 /var/www/html/public/Index.php(20): Router->dispatch()
#2 {main}
thrown in
/var/www/html/app/controllers/UserController.php on line 35

- `phpinfo()`: เปิดเผย Version ของ PHP, Server path, Module
- Error Messages: แสดงโครงสร้างไฟล์และเทคโนโลยีที่ใช้

The Fix

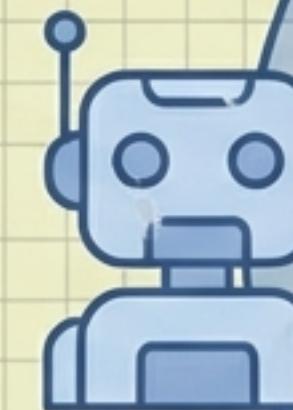
ปิดการแสดง Error
ใน Production

ลบไฟล์ Debug/Test
ออกจากระบบจริง

เครื่องมือช่วยตรวจสอบ: OWASP ZAP

Automated Scanner สำหรับค้นหา Technical Vulnerabilities

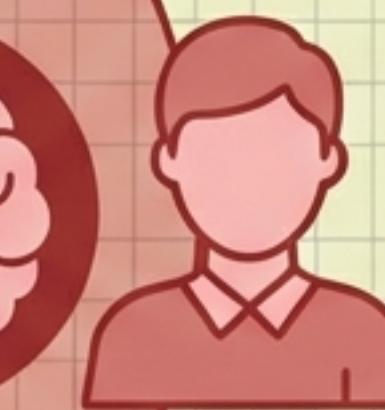
สิ่งที่ทำได้
(Can Do)



- เก็บ Request/Response
 - ตรวจ SQL Injection
 - ตรวจ XSS, Headers

เครื่องมือช่วยค้นหา
แต่การตัดสินใจ
ต้องใช้มนุษย์

สิ่งที่ทำไม่ได้
(Cannot Do)



- ตรวจ Logic Flaw
- การໂກງສ່ວນຄອດ
- Business Logic

Key Security Mindset

1. Client ໂກທິດໄສເນວ



(ອຍ่าເຫັນຂ້ອມູນຈາກ Browser ໂດຍໄມ່ຕຽບສອບ)

2. URL ໄມໃຊ່ສັກົນ



(ການຊ່ວຍ Link ໄມໃຊ່ການປ້ອງກັນ)

3. ໄຟຣີໂຄລວງໄດ້



(ອຍ่าເຫັນຊື່ໄຟຣີ ອີເຕີໂນ ສັງກືຕາເຫັນ)

4. Error ຄົວຂ້ອມູນ



(ອຍໍາມອບຂ້ອມູນຮະບບໃຫ້ຜູ້ໄມ່ຫວັງດີ)

Security ເປັນກະບວນການຕ່ອນເນື່ອງ ໄມໃຊ່ແຄ່ຂັ້ນຕອນເດືອວ

ขอบเขตและจริยธรรม

(Ethics & Boundaries)



Defensive Security:
เนื้อหาเนี้เพื่อการศึกษาและ
การป้องกันระบบ



Environment:
ทดลองบน localhost และ
ระบบของตนเองเท่านั้น



Warning: การโจมตี
ระบบจริงของผู้อื่น
มีความผิดตามกฎหมาย