การจัดการความมั่นคงปลอดภัยทางข้อมูล

Information Security Management

Chapter 6 : เทคนิคการเข้าโจมตีระบบ

เนื้อหา

- OWASP
- SQL Injection
- Cross-Site Scripting
- Remote Code Execution
- Session Hijacking

OWASP

OWASP เป็นองค์กรไม่แสวงหาผลกำไรที่ให้ความรู้เพื่อเน้นให้ระบบคอมพิวเตอร์ มีความปลอดภัยมากยิ่งขึ้น ในหลายแง่มุมไม่ว่าจะเป็นการทดสอบแฮก การเขียนโค้ดให้ปลอดภัย และการกำหนดนโยบายหรือมาตรฐานด้านความปลอดภัยให้แอปพลิเคชัน

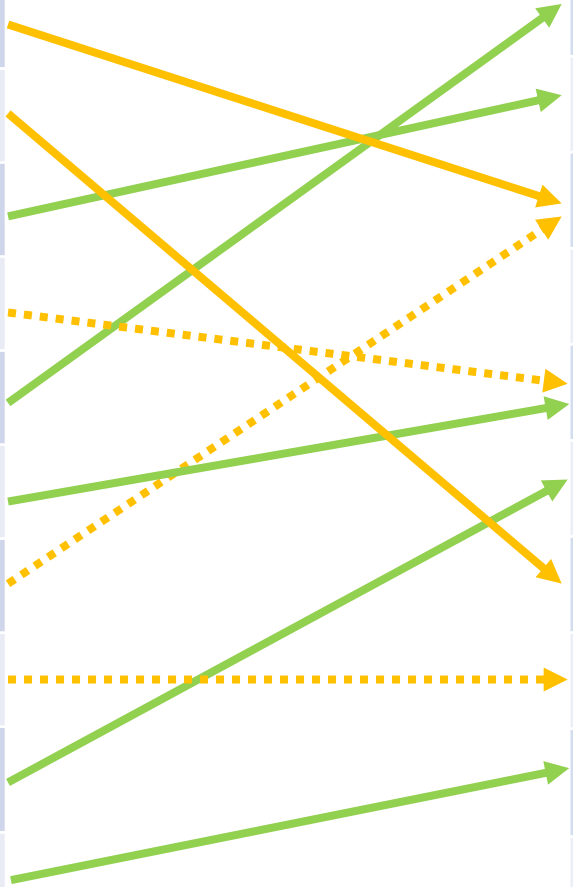
Web Application (2017)
A1:Injection
A2:Broken Authentication
A3:Sensitive Data Exposure
A4:XML External Entities (XXE)
A5:Broken Access Control
A6:Security Misconfiguration
A7:Cross-Site Scripting (XSS)
A8:Insecure Deserialization
A9:Using Components with Known Vulnerabilities
A10:Insufficient Logging & Monitoring

API (2019)
API1:Broken Object Level Authorization
API2:Broken User Authentication
API3:Excessive Data Exposure
API4:Lack of Resources & Rate Limiting
API5:Broken Function Level Authorization
API6:Mass Assignment
API7:Security Misconfiguration
API8:Injection
API9:Improper Assets Management
API10:Insufficient Logging & Monitoring

Mobile Application (2016)
M1:Improper Platform Usage
M2:Insecure Data Storage
M3:Insecure Communication
M4:Insecure Authentication
M5:Insufficient Cryptography
M6:Insecure Authorization
M7:Client Code Quality
M8:Code Tampering
M9:Reverse Engineering
M10:Extraneous Functionality

Web Application

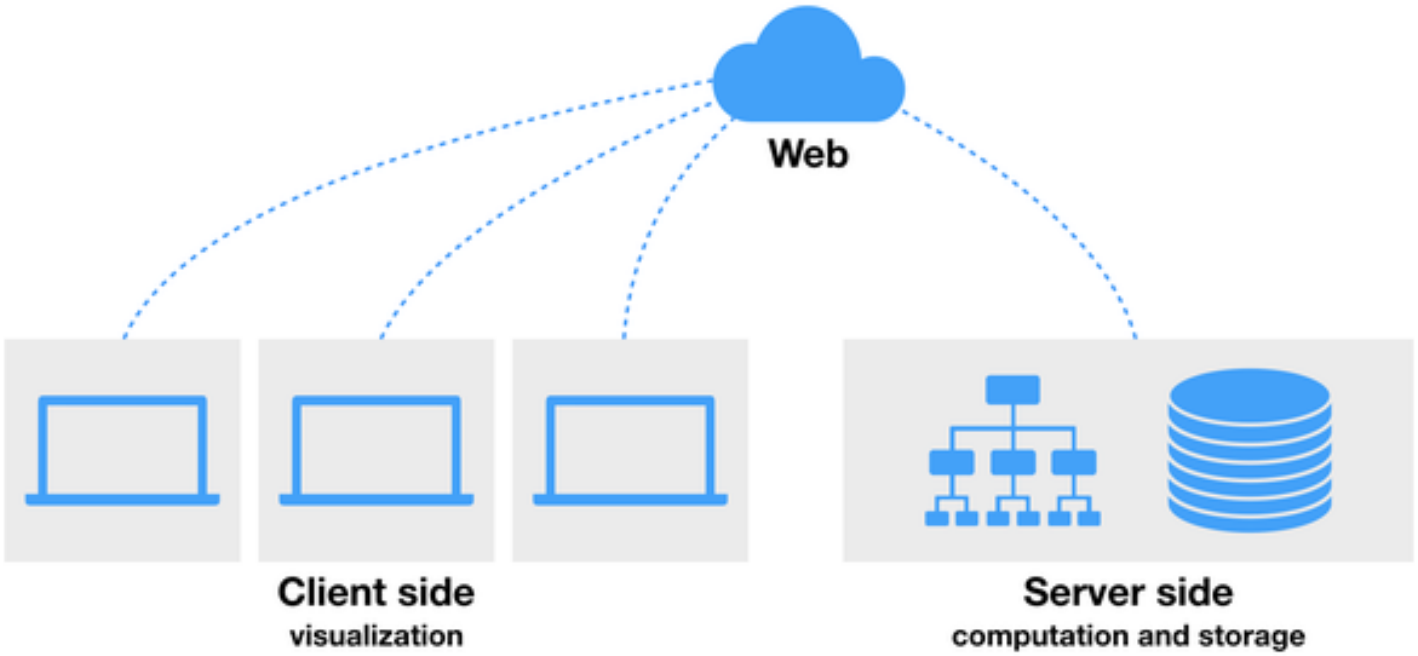
Web Application (2017)
A01:Injection
A02:Broken Authentication
A03:Sensitive Data Exposure
A04:XML External Entities (XXE)
A05:Broken Access Control
A06:Security Misconfiguration
A07:Cross-Site Scripting (XSS)
A08:Insecure Deserialization
A09:Using Components with Known Vulnerabilities
A10:Insufficient Logging & Monitoring



Web Application (2021)
A01:Broken Access Control
A02:Cryptographic Failures
A03:Injection
A04:Insecure Design (New)
A05:Security Misconfiguration
A06:Vulnerable and Outdated Components
A07:Identification and Authentication Failures
A08:Software and Data Integrity Failures (New)
A09:Security Logging and Monitoring Failures
A10:Server-Side Request Forgery (New)

SQL Injection

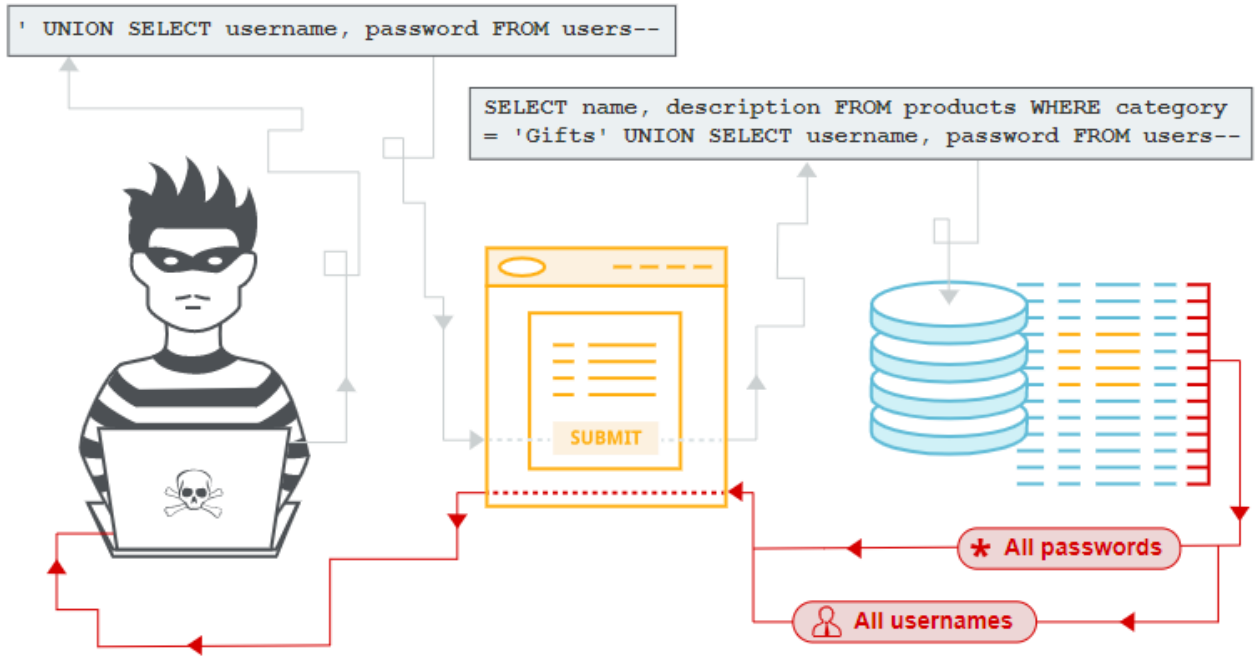
Server-side Script VS Client-side Script



HTML, CSS, JavaScript, VBScript

PHP, Python, JAVA, ASP.net, SQL, API

What SQL Injection Is ?



ฐานข้อมูลได้รับการออกแบบมาเพื่อส่งเสริมการเข้าถึงและการจัดการข้อมูลได้สะดวก แต่ถ้าใช้งานไม่ถูกวิธีอาจเป็นจุดอ่อนของระบบได้

ซึ่งเทคนิคหรือรูปแบบ การโจมตีของ Hacker โดยอาศัยช่องโหว่ของโปรแกรมทำให้สามารถ แอบใส่ คำสั่ง SQL เข้าไปทาง Input เรียกว่า “Injection”

How SQL Injection Works (ปกติ)

จำลองโดยมีฐานข้อมูลของตารางไวน์ (Wines) ซึ่งมีจากองุ่นหลากหลายชนิด

1. ต้องการแบบฟอร์มที่ให้ผู้ใช้งานค้นหาข้อมูล เช่น ผู้ใช้เลือกที่จะค้นหาไวน์ที่ทำจากองุ่นชนิด “lagrein”
2. ต้องทำการดึงข้อความที่ค้นหามาใส่ตัวแปร เช่น $variety = $_POST['variety'];
3. นำค่าในตัวแปรที่ได้ไปใช้ Query ในคำสั่ง WHERE เช่น

```
$query = "SELECT * FROM wines WHERE variety='$variety'";
```

4. ทำการสั่ง Submit ไปที่ Database Server
5. MySQL ส่งกลับข้อมูลทั้งหมดใน Table : Wines โดยที่ค่า variety มีค่า “lagrein”

How SQL Injection Works (เจาะ)

จำลองโดยมีฐานข้อมูลของตารางไวน์ (Wines) ซึ่งมีจากองุ่นหลากหลายชนิด

1. เปลี่ยนค่าจากชื่อชนิดองุ่นแบบธรรมดา เป็นมีการเชื่อมของวรรคตอนและเครื่องหมายพิเศษ เช่น

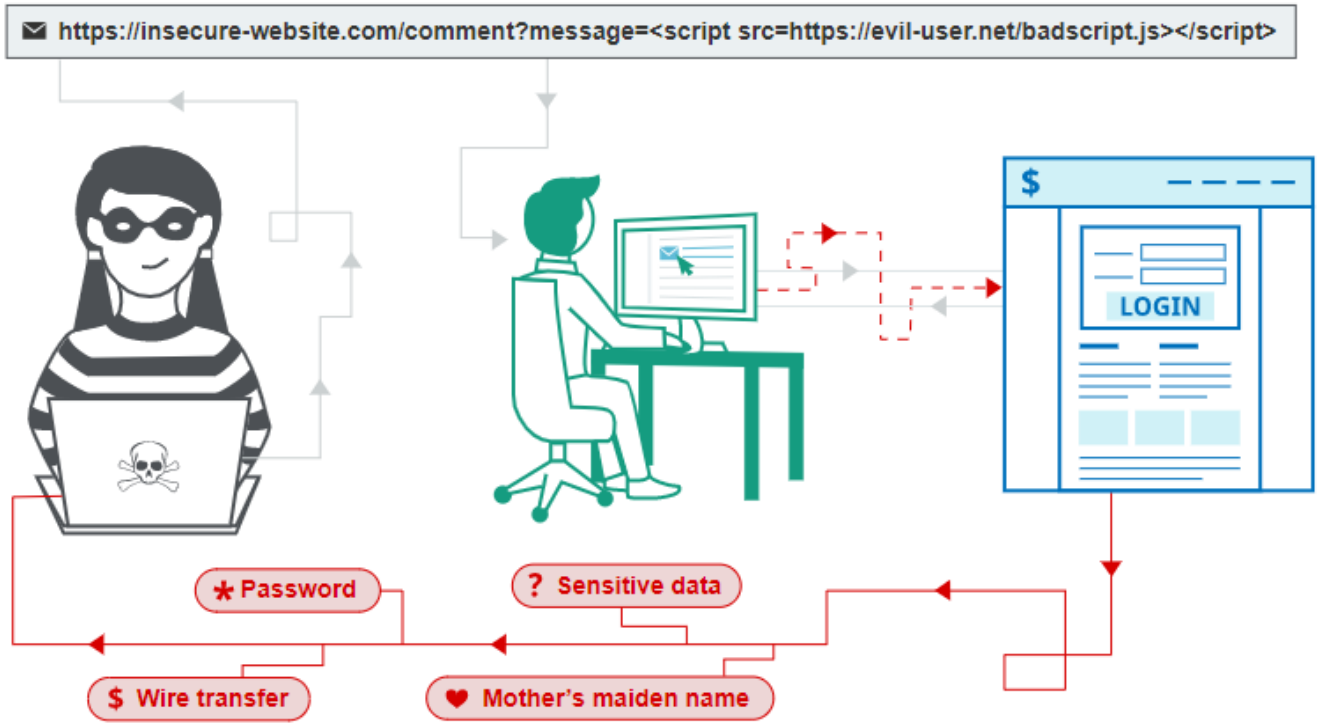
   lagrein' or 1=1;

SELECT * FROM wines WHERE variety = '**lagrein**’

SELECT * FROM wines WHERE variety = '**admin**'

SELECT * FROM wines WHERE variety = '**admin' or 1=1;**'

Cross-Site Scripting



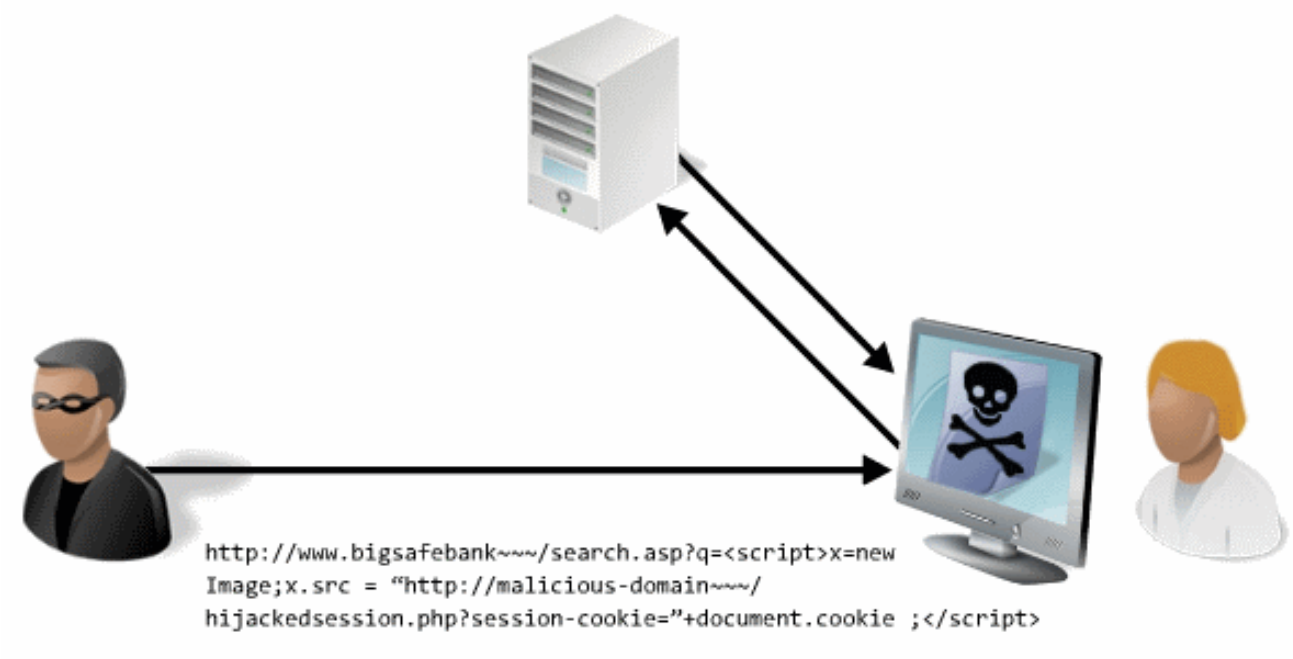
เทคนิคการฝังโค้ดเข้าไปกับหน้าเวปเพจที่มีช่องโหว่ เมื่อผู้ใช้โหลดหน้าเวปเพจไป ค่าที่สำคัญบางอย่างจะถูกเปิดเผย เช่น ค่าของ Cookie, Username, Password เรียกเทคนิค “Cross-Site Scripting (XSS)”

ประเภทของการโจมตี XSS attacks

- **Reflected XSS** : ช่องโหว่ที่สามารถแสดง HTML/JavaScript ออกมาทันที่เมื่อมีการ Input เข้าไป
- **Stored (Persistent) XSS** : มีการ Stored ลง Database แล้วต้องดึงขึ้นมาโชว์
- **DOM-based XSS** : ช่องโหว่นั้นมีอยู่ในโค้ดฝั่งไคลเอ็นต์แทนที่จะเป็นโค้ดฝั่งเซิร์ฟเวอร์ หรือเอาเซอเวอร์ไปเขียน code โจมตีไคแอนท์

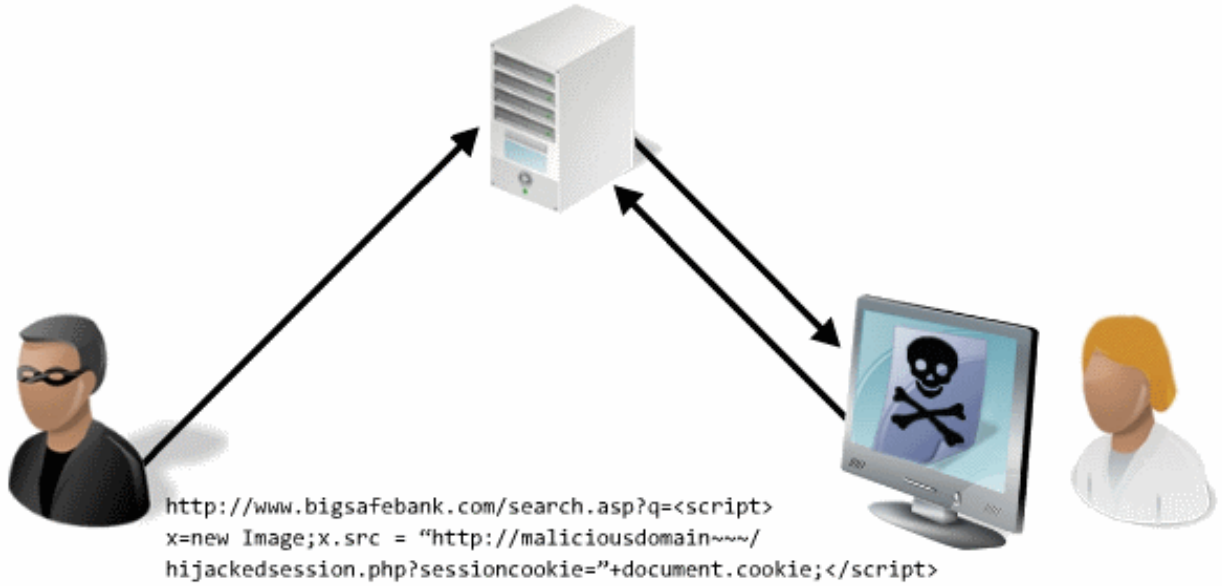
ประเภทของการโจมตี XSS attacks

Reflected XSS คือ อนุญาตให้ผู้โจมตีสามารถแทรกสคริปต์ลงในเนื้อหาของเว็บไซต์หรือแอพพลิเคชัน วิธีนี้ช่วยให้ผู้โจมตีสามารถขโมยข้อมูลส่วนตัว เช่น cookies, account information

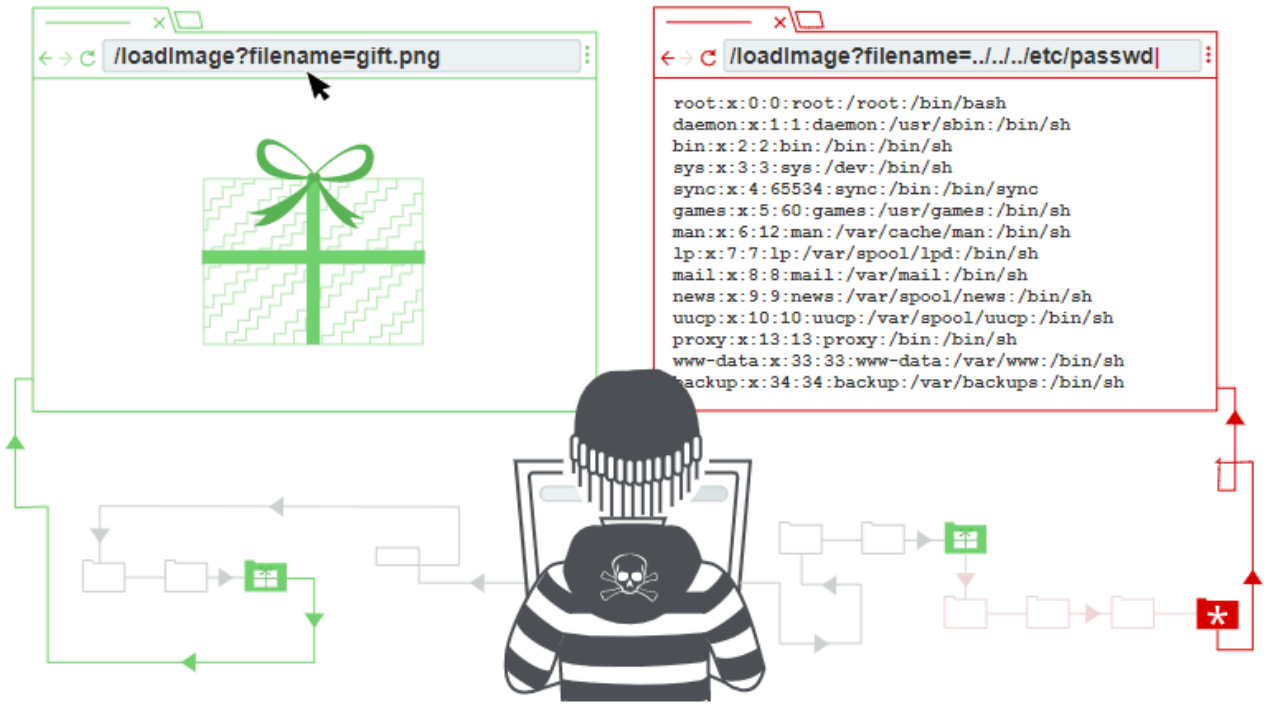


ประเภทของการโจมตี XSS attacks

Stored (Persistent) XSS คือ อนุญาตให้ผู้โจมตีสามารถแทรกสคริปต์ลงในฐานข้อมูลของเว็บไซต์หรือแอพพลิเคชัน ทำให้เว็บไซต์ส่งโค้ดอันตรายแก่ผู้เยี่ยมชมรายอื่น



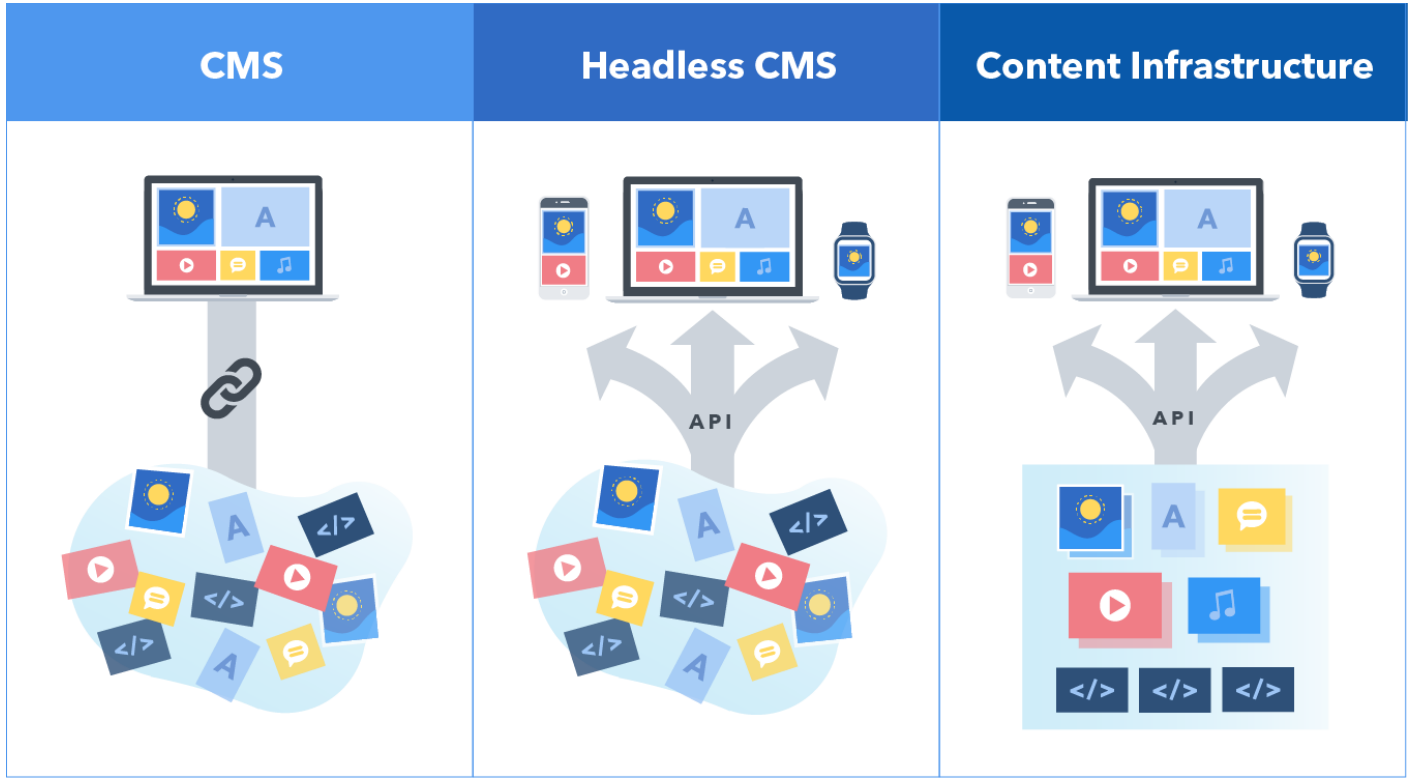
Remote Code Execution



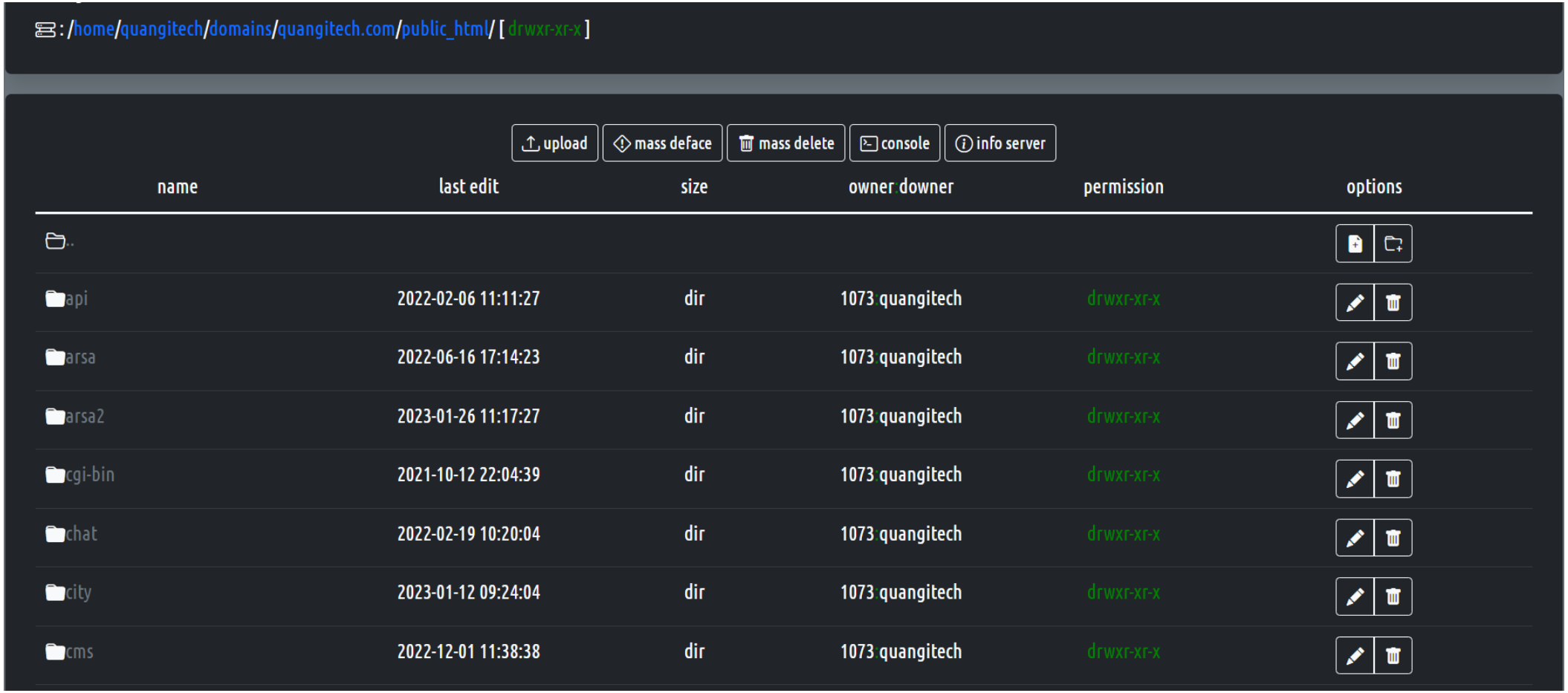
เป็นการโจมตีด้วยการใช้ช่องโหว่ของการใช้งานตรรกะในโปรแกรมเพื่อเรียกใช้งานคำสั่ง หรือ Script ที่อยู่บน Server ซึ่งการเรียกใช้คำสั่งเกิดขึ้นในเบราว์เซอร์ไคลเอ็นต์ แต่จะเป็นการบังคับจากระยะไกลใช้เครื่อง Server เพื่อโจมตีไปยังเครื่องอื่น หรือเข้าใช้งานเครื่อง Server

Remote Code Execution

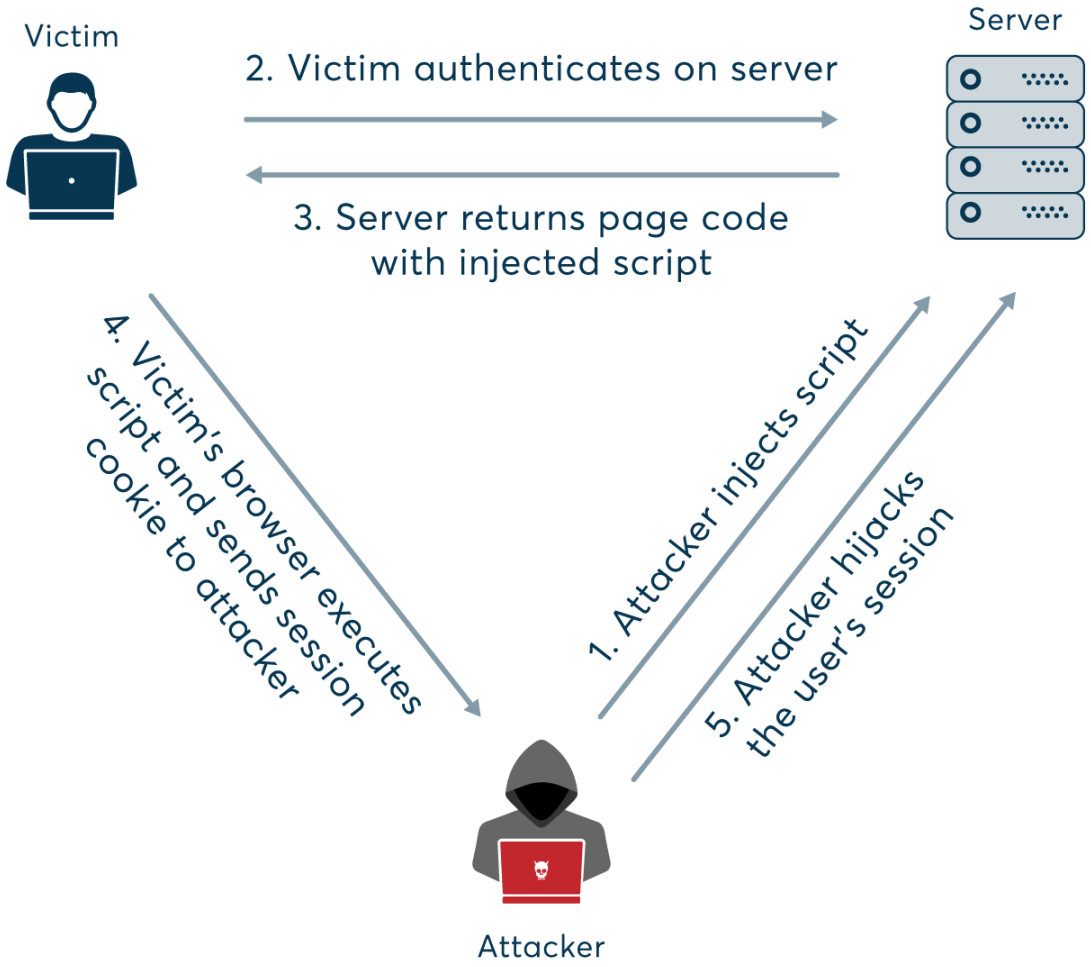
ช่องโหว่นี้ส่วนมากจะพบในโปรแกรมที่ใช้งานในรูปแบบ Template (เว็บที่มีการดึง html เอาไว้ด้านใน)



Remote Code Execution

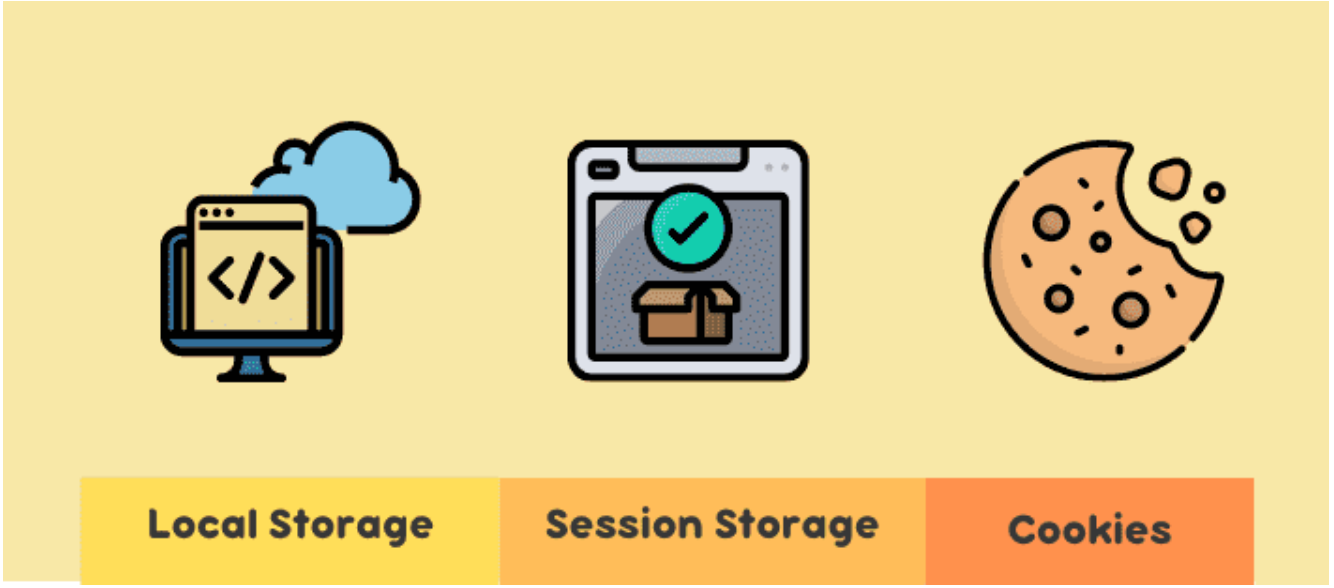


Session Hijacking

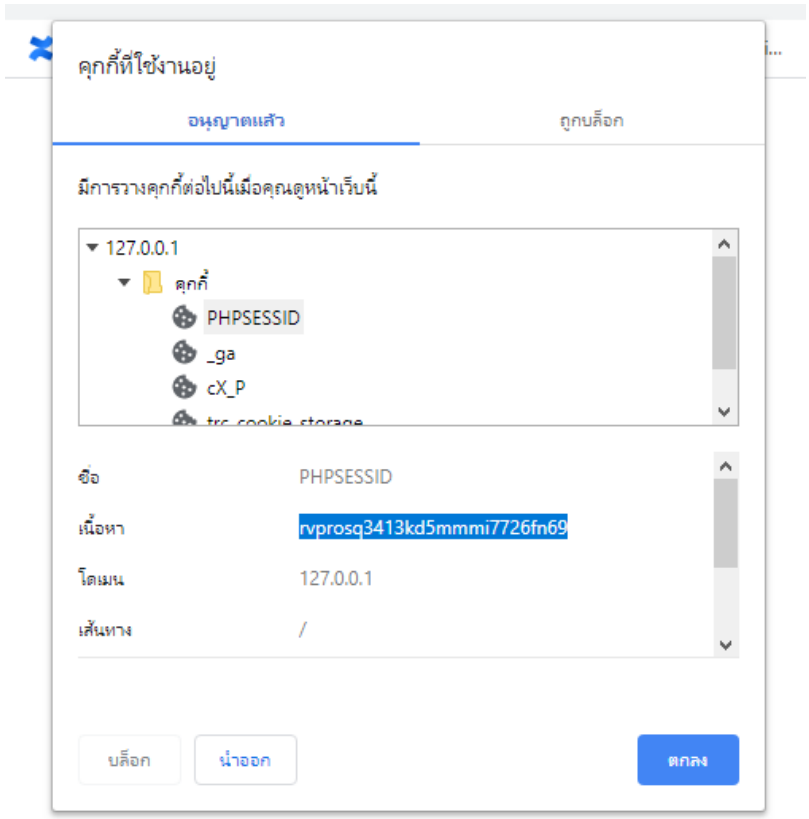


เป็นการโจมตีต่อเนื่องจากการโจมตีช่องโหว่ Cross-Site Scripting (XSS) โดยที่ผู้ไม่หวังดีจะทำการปลอมแปลง Session ของเจ้าของแล้วทำการสวมรอยเป็นผู้ใช้งานนั้นๆ แทน โดยการปลอมแปลง Session ใหม่ เรียกว่าการโจมตี “ขโมยตัวตน (Session Hijacking)”

Local Storage – Session - Cookies



Local Storage – Session - Cookies



เปลี่ยนหมายเลข Session id