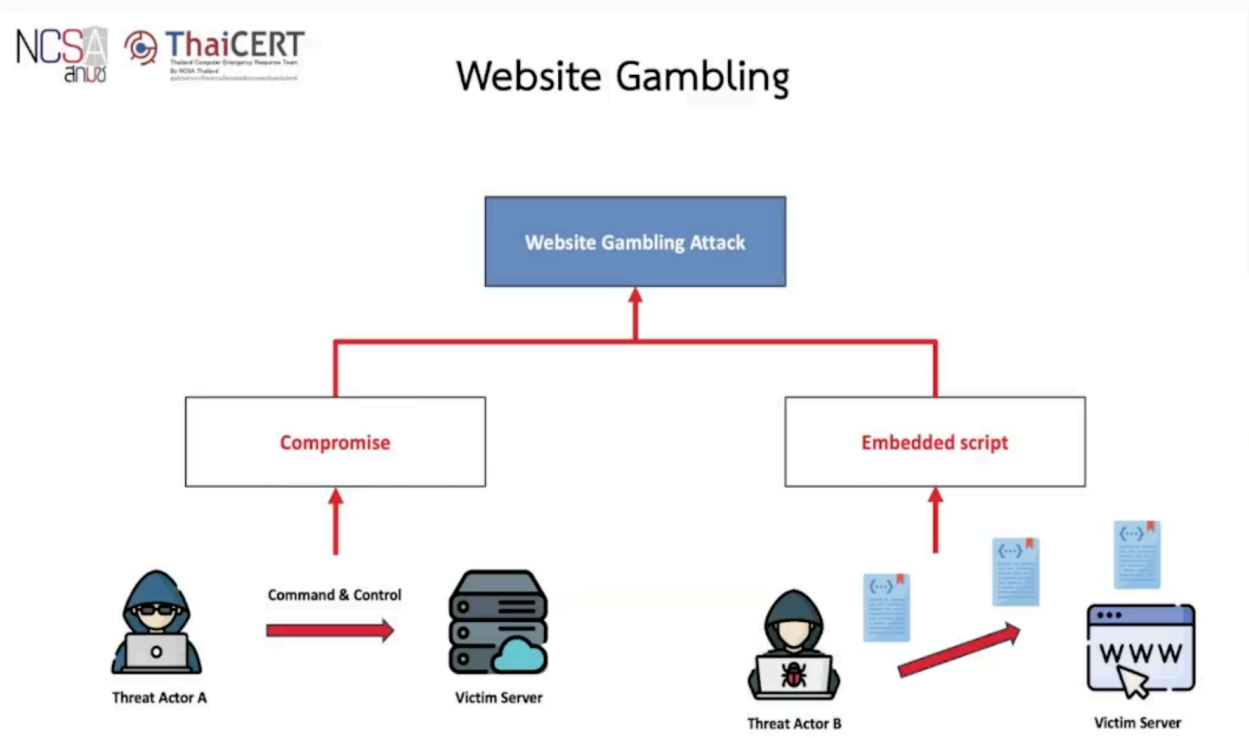
สรุปเนื้อหาการอบรมออนไลน์ หัวข้อป้องกันเว็ปการศึกษาเสริมความปลอดภัย ให้ไร้เว็ปพนัน NCSA สมช

ในการออกแบบเว็ปไซต์จะมี หลัก ๆ อยู่ 2 แบบคือ
- Custom Website หรือเว็ปไซต์ที่เขียนโค๊ดขึ้นมาเอง
- CMS Website เป็นเว็ปสำเร็จรูปที่เป็น open source ยกตัวอย่างเช่น
    +wordpress
    +Joomla
    + Drupal

Website Gambling Attack (การถูกฝังเว็ปพนัน) หลัก ๆมี อยู่ 2 กรณี
1. Compromise(ประณีประนอม) อาศัยช่องโหว่ของระบบ เจาะเข้ามาเพื่อควบคุมเครื่องอย่างสมบูรณ์
2. Embedded script ไม่ได้เข้ามาควบคุมเครื่องปลายทางโดยตรง แต่ฝังบางสิ่งบางอย่างเข้าไปจากช่องว่างของระบบที่ให้ฝังคริปต์



Case1 Compromise Google dorking or hacking

เราสามารถใช้ google เพื่อค้นหา ช่องโหว่ ของเว็ปไซต์ของเราได้ โดยระบุ site และ คีย์เวิร์ดที่ต้องการให้ google ค้นหา

เช่น กรณีเว็ปไซต์คณะวิทยาศาสตร์ ให้ระบุคำค้นดังบรรทัดด้านล่างนี้

site:sci.ubu.ac.th “พนัน” “บาคาร่า”

กรณีศึกษา “Drupal” ช่องโหว่ CVE-2018-7600

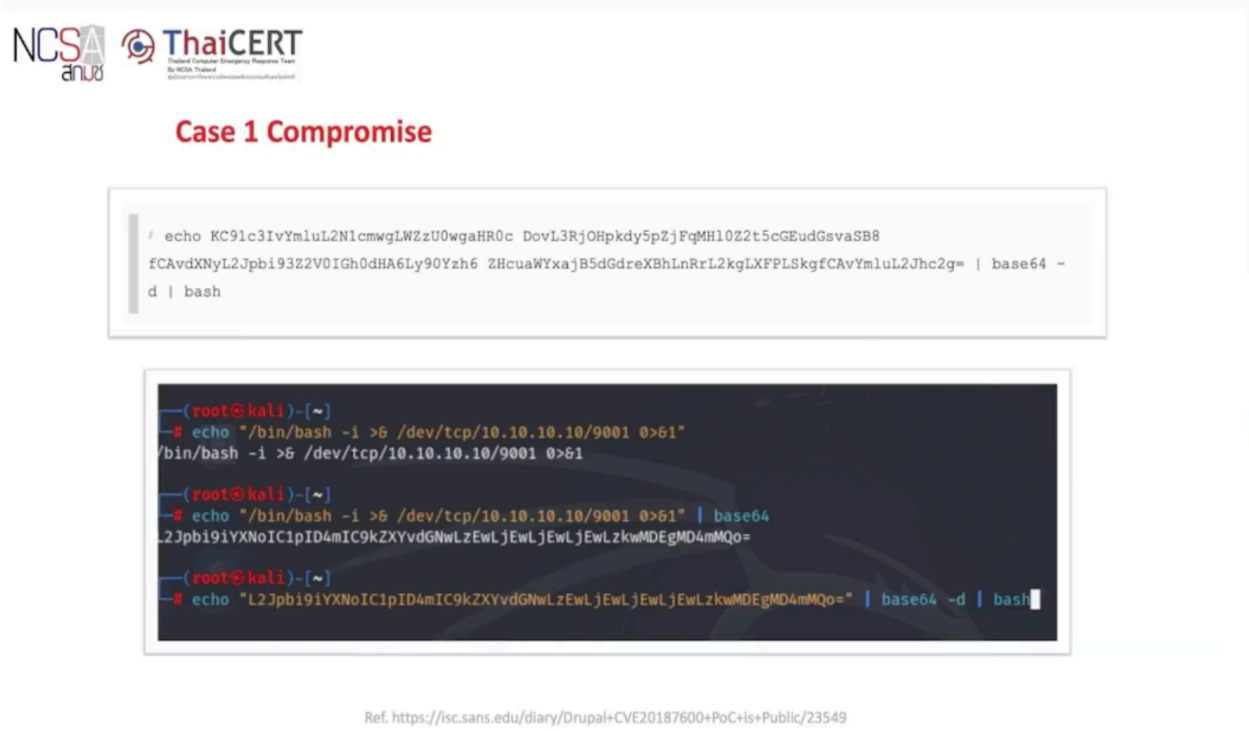
Drupal CVE-2018-7600 PoC
https://www.drupal.org/sa-core-2018-002
CVE IDs:
CVE-2018-7600

Description:
A remote code execution vulnerability exists within multiple subsystems of Drupal 7.x and 8.x. This potentially allows attackers to exploit multiple attack vectors on a Drupal site, which could result in the site being completely compromised.

CVE คือช่องโหว่ ๆ หนึ่งที่ถูกปล่อยออกมา

Keyword:

- Remote execution
- Remote shell /Reverse shell



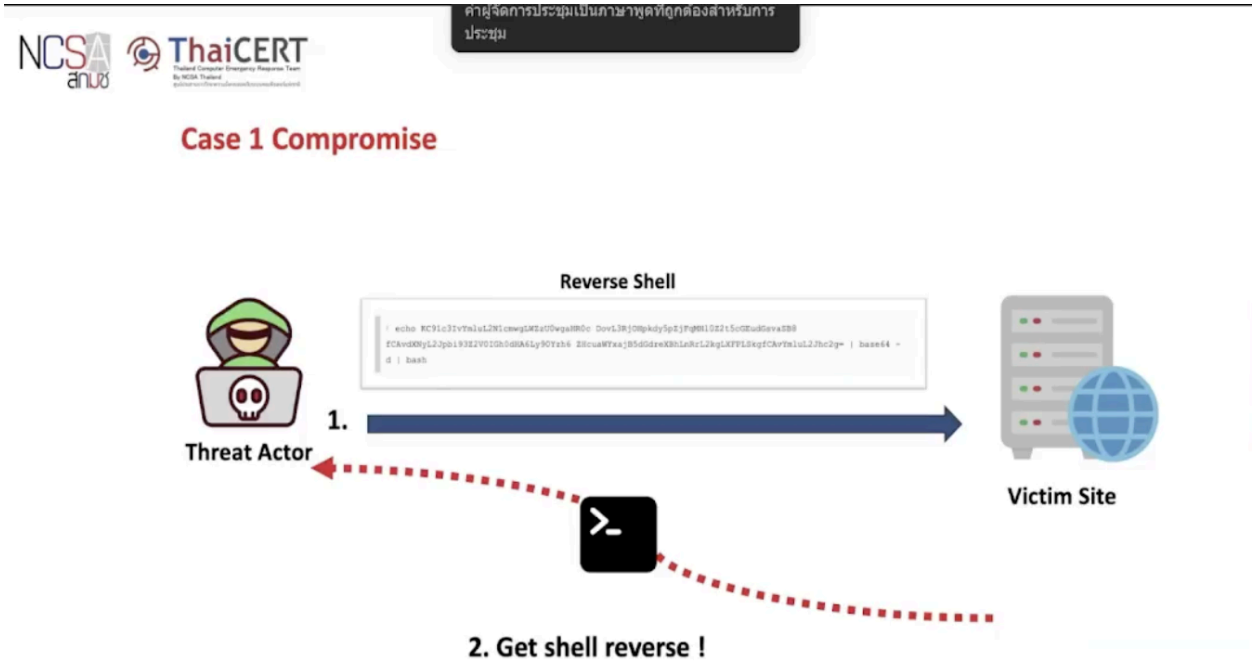
Ref. https://isc.sans.edu/diary/Drupal+CVE20187600+PoC+is+Public/23549

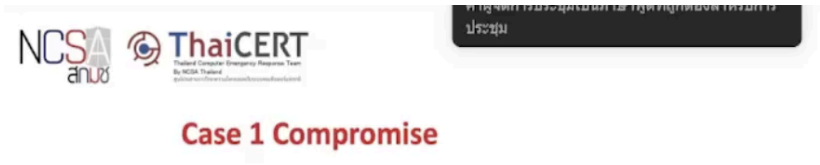
คำสั่งข้างบนเป็น bash64 สามารถแปลงเป็นคำสั่งจริงได้ด้านล่าง

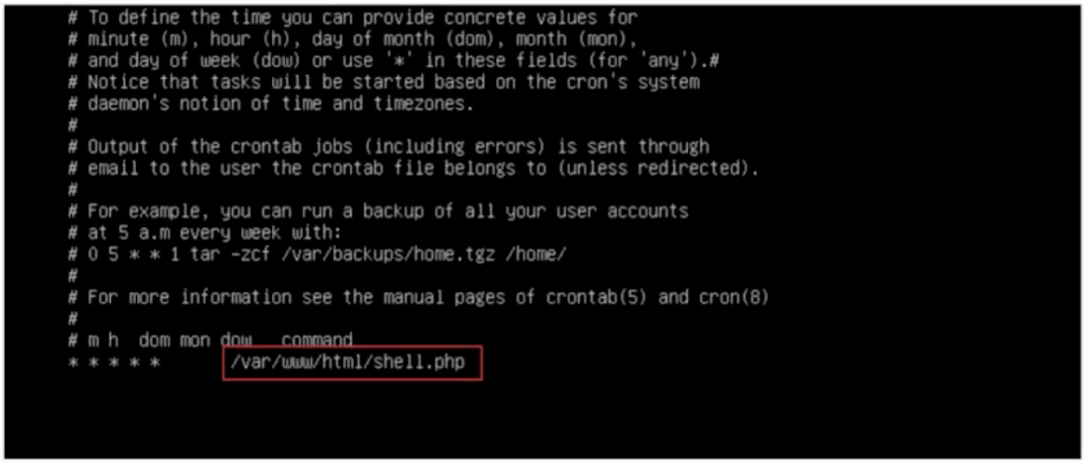
echo base64 command | base64 -d | bash

```
soowoi — -zsh — 77×25

soowoi@iMac-khxng-soowoi ~ % echo Hello World
Hello World
soowoi@iMac-khxng-soowoi ~ % echo Hello World | base64
SGVsbG8gV29ybGQK
soowoi@iMac-khxng-soowoi ~ % echo SGVsbG8gV29ybGQK | base64 -d
Hello World
soowoi@iMac-khxng-soowoi ~ %
```

https://www.facebook.com/NCSA.Thailand/videos/1264121957933355







มีการสร้าง cronjob สำหรับรันคำสั่งเป้าหมาย ตามวัน/เวลาที่กำหนด

ข้อแนะนำ

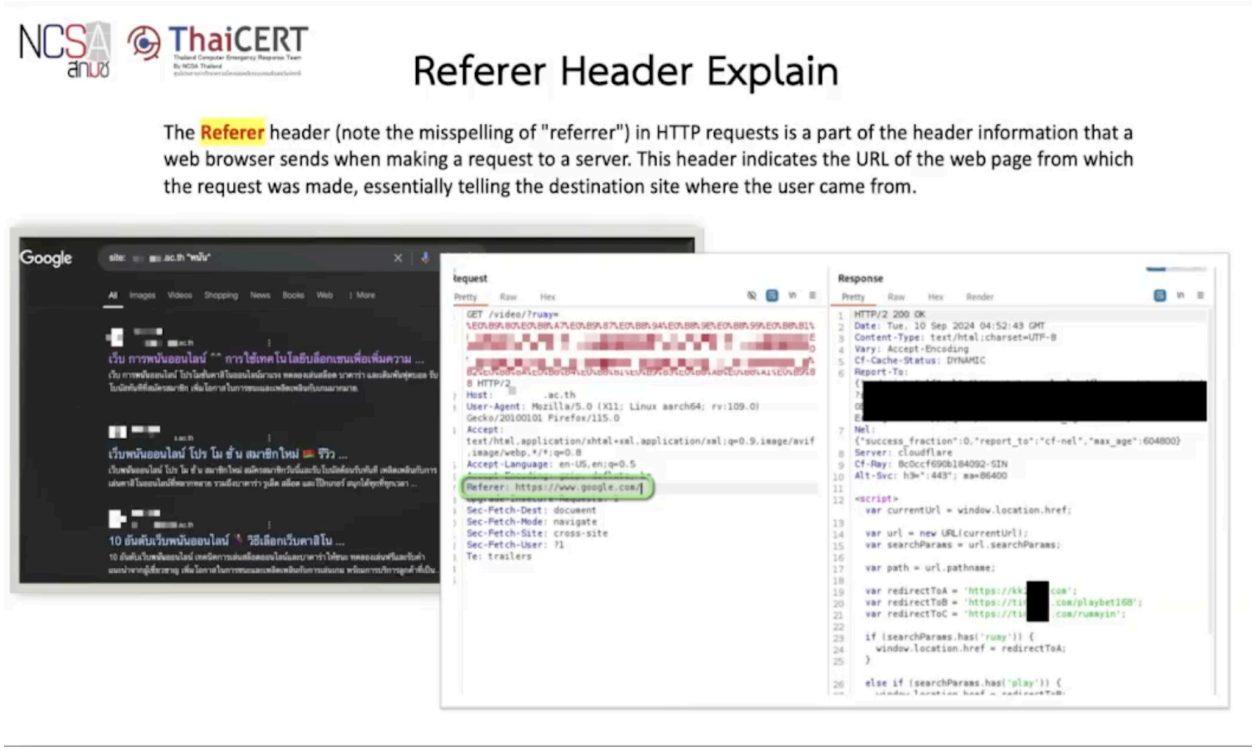
หลังจากปิดช่องโหว่แล้ว ให้ตรวจสอบหลังบ้าน และจำกัดความเสียหายไว้ด้วย (มีไฟล์แปลกๆ อยู่หลังบ้านอยู่อีกไหม)

สรุป

1.อัปเดตซอฟต์แวร์ให้เป็นปัจจุบัน
2. ตรวจสอบหลังบ้านว่าไม่มีอะไรเขียน หรือตั้งเวลาการทำงานไว้

วิเคราะห์พฤติกรรมของเว็ปไซต์การพนัน

หากเราเอาลิงค์ที่แปะไว้ในเว็ปมาเข้าตรงๆ จะไม่สามารถเข้าได้ แต่ถ้าคลิ๊กลิงค์จากกูเกิ้ลเข้าได้เพราะมีการเช็ค Referer header



Case 2 Embeded

เว็ปไซต์ถามตอบ (Webboard)

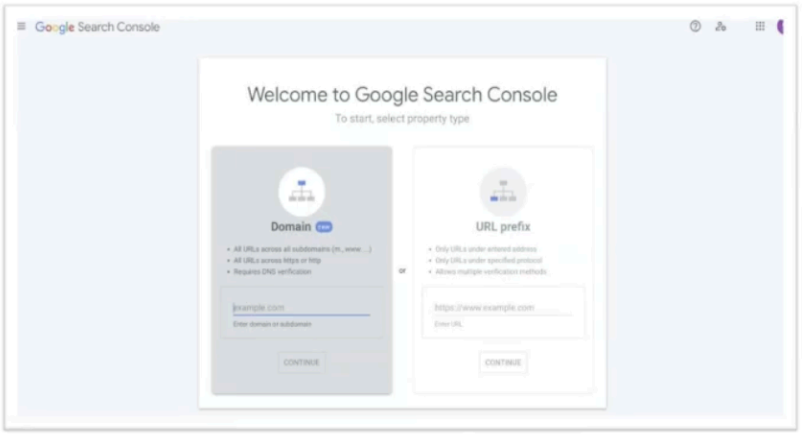
- เว็ปพนันมักจะมาโพสต์ลง ในช่องความคิดเห็นทำให้มีข้อมูลเว็ปพนันฝังอยู่ในเว็ป เมื่อค้นผ่าน google

Google Search Console

แจ้งลบผลการค้นหาจากเว็ปไซต์ที่เราดูแล เมื่อเราจัดการปิดช่องโหว่เสร็จและไม่ต้องการให้ผลการค้นหาปรากฏในผลการค้นหาจาก google

Google Search Console

To delete a specific search path or URL from Google Search Console.

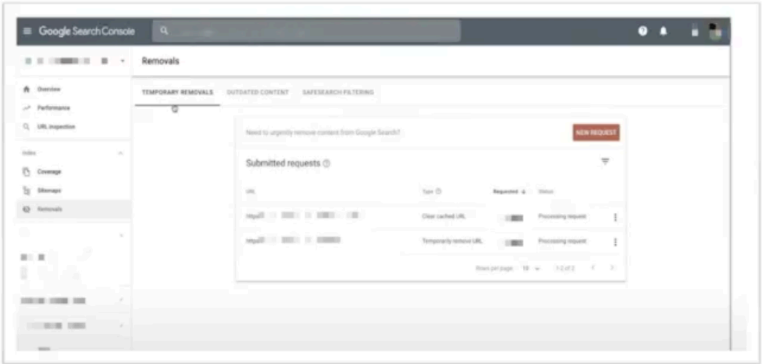


- **URL:** https://search.google.com/search-console/welcome?utm_source=about-page

เลือกเครื่องมือ Removal Tools จากนั้นกรอก site หรือเพจที่เราต้องการลบออกจากผลการค้นหา

Google Search Console

To delete a specific search path or URL from Google Search Console, follow these steps:
- **Use the "Removals" Tool**: This tool allows you to temporarily or permanently remove URLs from Google's search results. You can find it under the "Index" section in Google Search Console. Go to the URL removal section and submit the search path (e.g., /your-page-path) for removal. This will block the URL from appearing in Google search results.



การแก้ไขเว็ปพนัน เราต้องรู้จัก Attack Surface หรือพื้นที่ที่ hacker สามารถเจาะเข้าไปในระบบ เพื่อที่เราจะสามารถปกปิดช่องโหว่เหล่านั้นได้

WordPress:

1. อัปเดตซอฟต์แวร์, ธีม, และปลั๊กอิน
2. ใช้รหัสผ่านที่แข็งแกร่งและเปิดใช้งาน Two-Factor Authentication (2FA)
3. ติดตั้งปลั๊กอินรักษาความปลอดภัย เช่น Wordfence Security

Joomla:

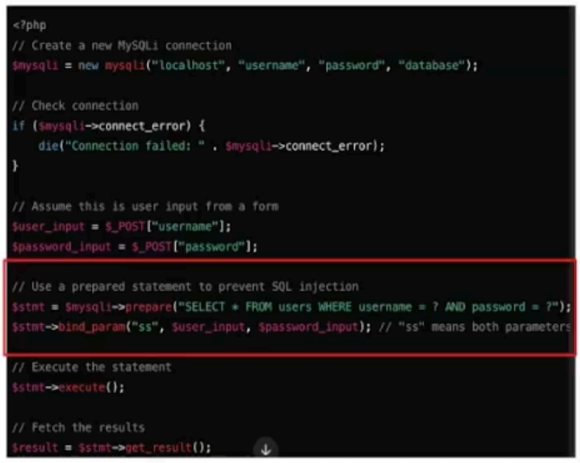
1. อัปเดต Joomla และส่วนขยายอย่างสม่ำเสมอ
2. ใช้รหัสผ่านที่แข็งแกร่งและเปิดใช้งาน 2FA
3. ติดตั้งส่วนขยายรักษาความปลอดภัย เช่น RSFirewall

การสร้างเว็บไซต์ที่ปลอดภัย

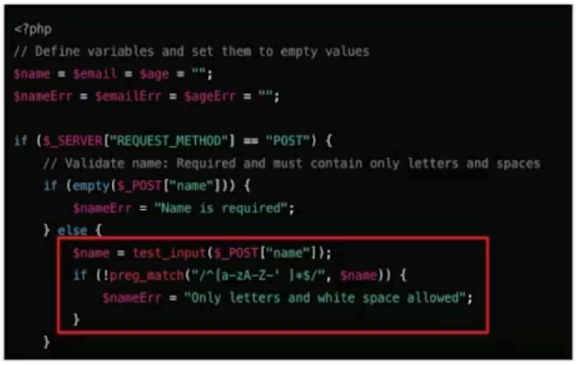
1. Input validation & Output Encoding

คีย์เวิร์ดหลัก คือ Input Validation เราจะไม่เชื่อข้อมูล input ใดๆ จาก user ต้องตรวจสอบทุก ๆ input และลบข้อมูลที่จะเป็นอันตรายก่อนจะนำไปประมวลผลที่ฝั่ง server

- ป้องกันการโจมตีเช่น SQL injection
- ใช้ parameterized queries เพื่อป้องกันการโจมตี เช่น SQL injection
    - ตัวอย่าง php ใช้ prepare statement
- Escape special character in HTML to prevent XSS(Cross-Site Scripting)
- ตัวอย่าง php ใช้ script เช็คว่า เป็นตัวอักษร ตัวเลข ไม่ได้มีอักขระพิเศษอื่นๆ
- สำหรับ form inputs ควรเช็ค data type, size , format และ range



```php
<?php
// Create a new MySQLi connection
$mysqli = new mysqli("localhost", "username", "password", "database");

// Check connection
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}

// Assume this is user input from a form
$user_input = $_POST["username"];
$password_input = $_POST["password"];

// Use a prepared statement to prevent SQL injection
$stmt = $mysqli->prepare("SELECT * FROM users WHERE username = ? AND password = ?");
$stmt->bind_param("ss", $user_input, $password_input); // "ss" means both parameters

// Execute the statement
$stmt->execute();

// Fetch the results
$result = $stmt->get_result();
```



```php
<?php
// Define variables and set them to empty values
$name = $email = $age = "";
$nameErr = $emailErr = $ageErr = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Validate name: Required and must contain only letters and spaces
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        if (!preg_match("/^[a-zA-Z-' ]*$/", $name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }
```

2. Authentication & Session Management
    - การกำหนดนโยบายรหัสผ่านที่แข็งแกร่ง (เช่น ความยาวขั้นต่ำ ความซับซ้อน วันหมดอายุ)
    - แฮชรหัสผ่านด้วยอัลกอริทึมที่ปลอดภัย เช่น bcrypt, Argon2 หรือ PBKDF2
    - ใช้การพิสูจน์ตัวตนหลายปัจจัย (MFA)
    - ใช้คุกกี้เซสชันที่ปลอดภัย (เช่น HttpOnly, Secure, SameSite)
    - หมุนเวียนและทำให้โทเค็นเซสชันไม่ถูกต้องอย่างปลอดภัยเมื่อออกจากระบบ

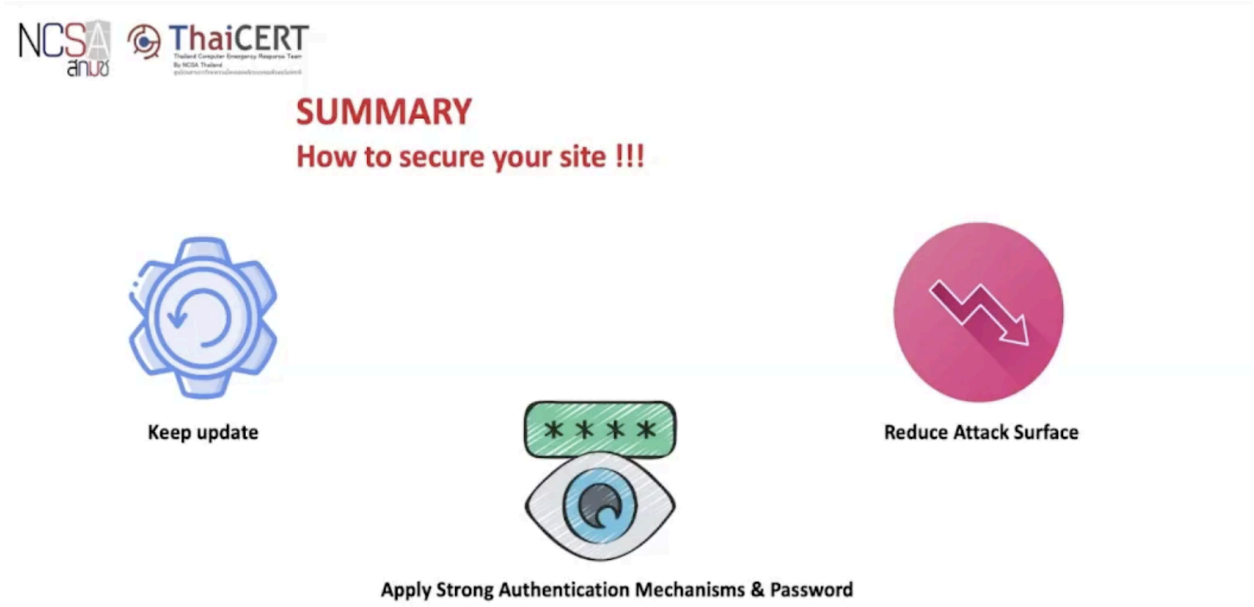
3. Access Control & Authorization
    - การควบคุมการเข้าถึงตามบทบาท (RBAC): กำหนดสิทธิ์ตามบทบาท (เช่น ผู้ดูแลระบบ ผู้ใช้งาน)
    - ตรวจสอบสิทธิ์ของผู้ใช้ในทุกการกระทำที่ละเอียดอ่อน
    - ใช้หลักการ สิทธิ์ขั้นต่ำ เพื่อลดการเข้าถึงข้อมูลที่ละเอียดอ่อน

4. Security Headers

   Implement HTTP security headers เช่น
    - Content-Security-Policy(CSP) to prevent XSS
    - X-Frame-Options to prevent clickjacking
    - Strict-Transport-Security (HSTS) to enforce HTTPS
    - X-Content-Type-Options to prevent MIME-type sniffing

5. Secure file uploads
    - จำกัดประเภทไฟล์และขนาดไฟล์ที่อัพโหลด
    - จัดเก็บไฟล์ที่อัพโหลดอย่างปลอดภัย: ใช้ตำแหน่งจัดเก็บแยกต่างหากที่มีการเข้าถึงแบบจำกัด
    - สแกนไฟล์เพื่อหามัลแวร์ก่อนทำการบันทึก

6. Regular Security Audits & Update
    - อัปเดตซอฟต์แวร์ให้ทันสมัยอยู่เสมอ: ซอฟต์แวร์ ไลบรารี และส่วนที่ต้องพึ่งพาของเว็บไซต์
    - ดำเนินการตรวจสอบความปลอดภัยและการทดสอบการเจาะระบบเป็นประจำ
    - ตรวจสอบบันทึกของเว็บไซต์ของคุณเพื่อดูว่ามีกิจกรรมที่ผิดปกติหรือไม่ (เช่น ความพยายามเข้าสู่ระบบที่ล้มเหลว)



ลิงค์วีดีโอการอบรมเนื้อหาเต็ม
https://www.facebook.com/share/v/oghRfFxiHpK4eLga/