ETDA
สพธอ
www.etda.or.th

งานสัมมนา

“ยกระดับความมั่นคงปลอดภัยของเว็บไซต์ให้ได้มาตรฐาน”

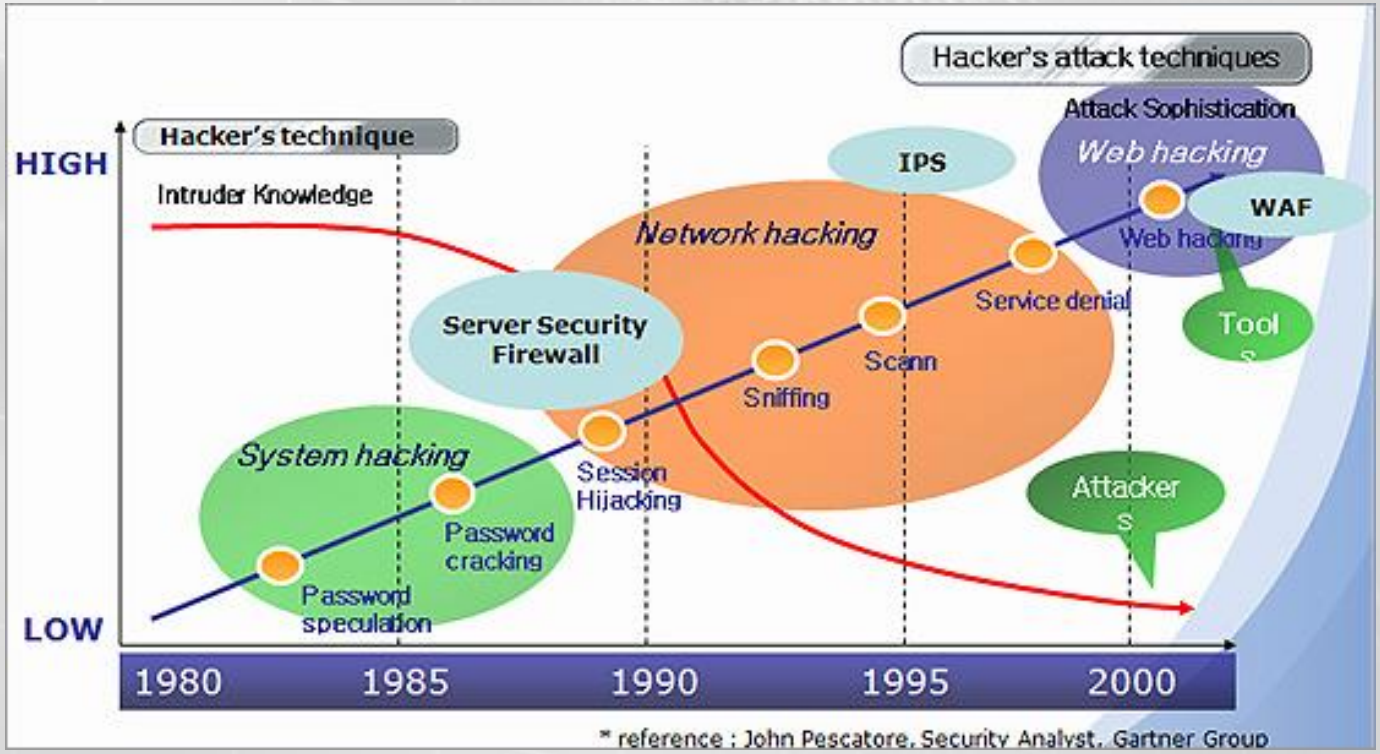
โดย นายพรพรหม ประภากิตติกุล

Security+ ,ISO27001 Lead auditor ,GPEN ,GCIH

วิทยากรจากทีม ThaiCERT สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)

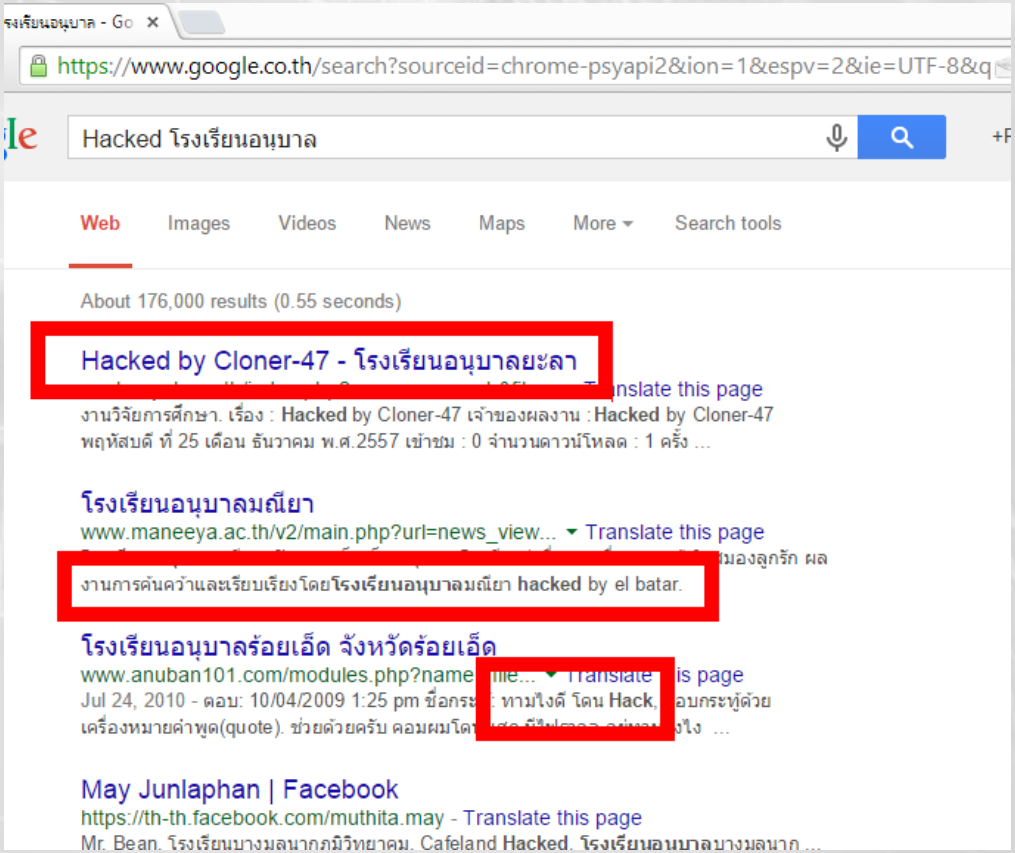
แนวโน้มการโจมตีระบบคอมพิวเตอร์จากอดีตถึงปัจจุบัน

สถานการณ์การโจมตีของโลกที่เปลี่ยนไป จากเดิมเจาะผ่านบริการ มาเจาะผ่านเครือข่าย และสุดท้ายมาจบลงที่เจาะเว็บไซต์

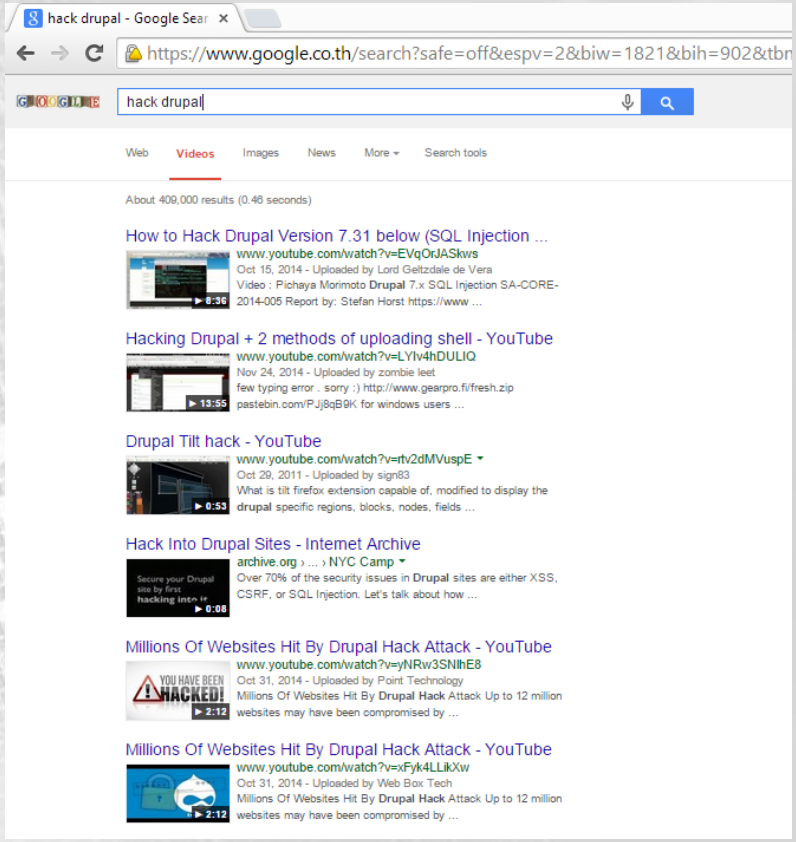


เปิดเว็บโรงเรียนอนุบาลก็ยังโดนแฮก

เปิดเว็บโรงเรียนอนุบาลก็ยังโดนแฮก



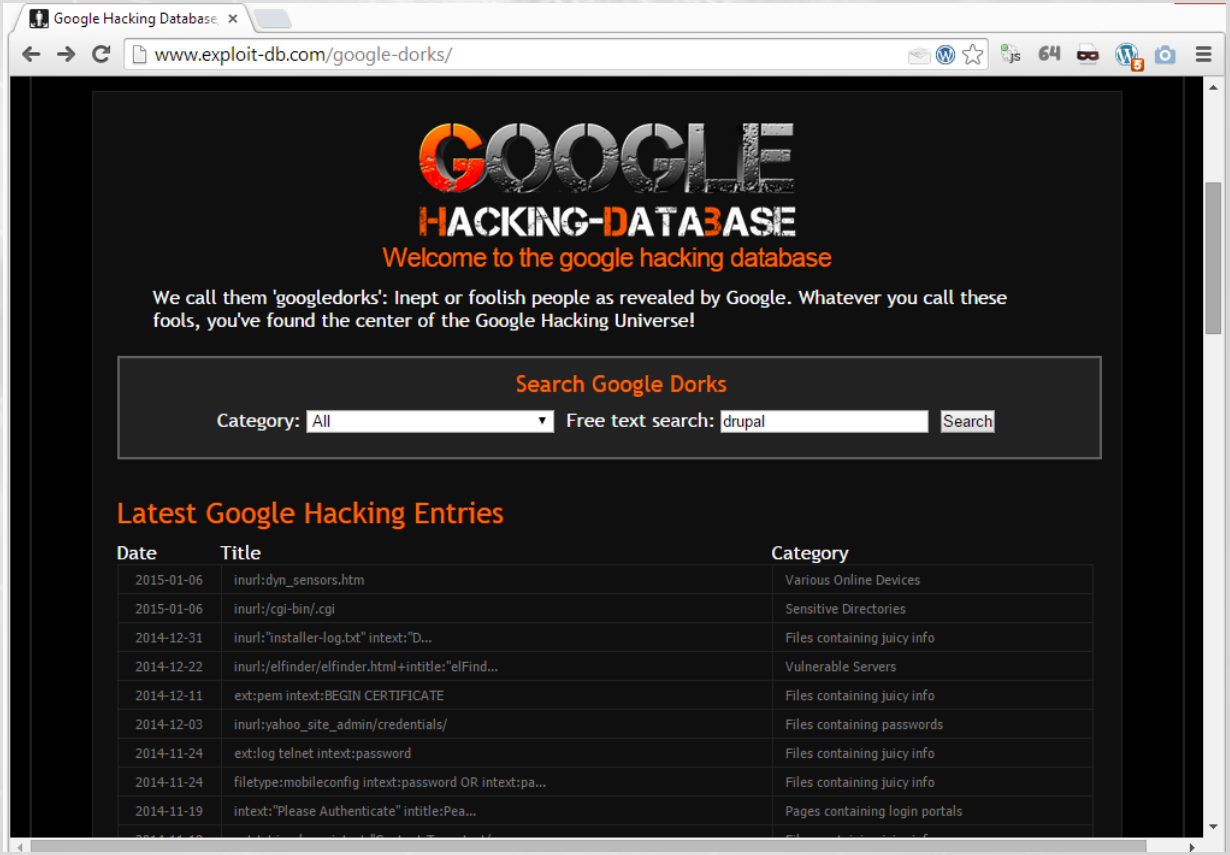
แฮกเกอร์เป็นกันง่าย มีบทเรียนออนไลน์และไซต์ทดสอบ



เด็กที่ไหนก็เป็นแฮกเกอร์ได้ แค่เปิดวีดีโอแล้วทำตามกับเว็บไซต์เป้าหมาย

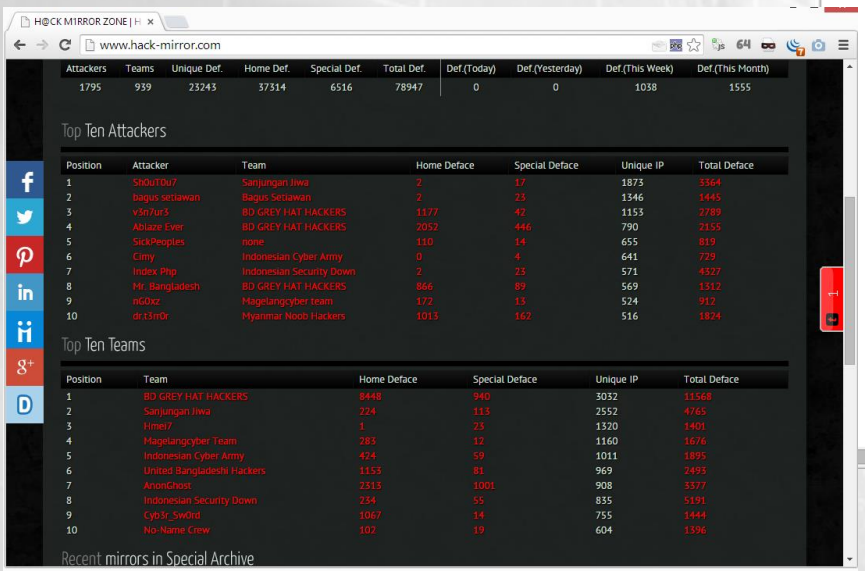
หาเป้าหมายด้วย GHDB (Google Hacking Database)

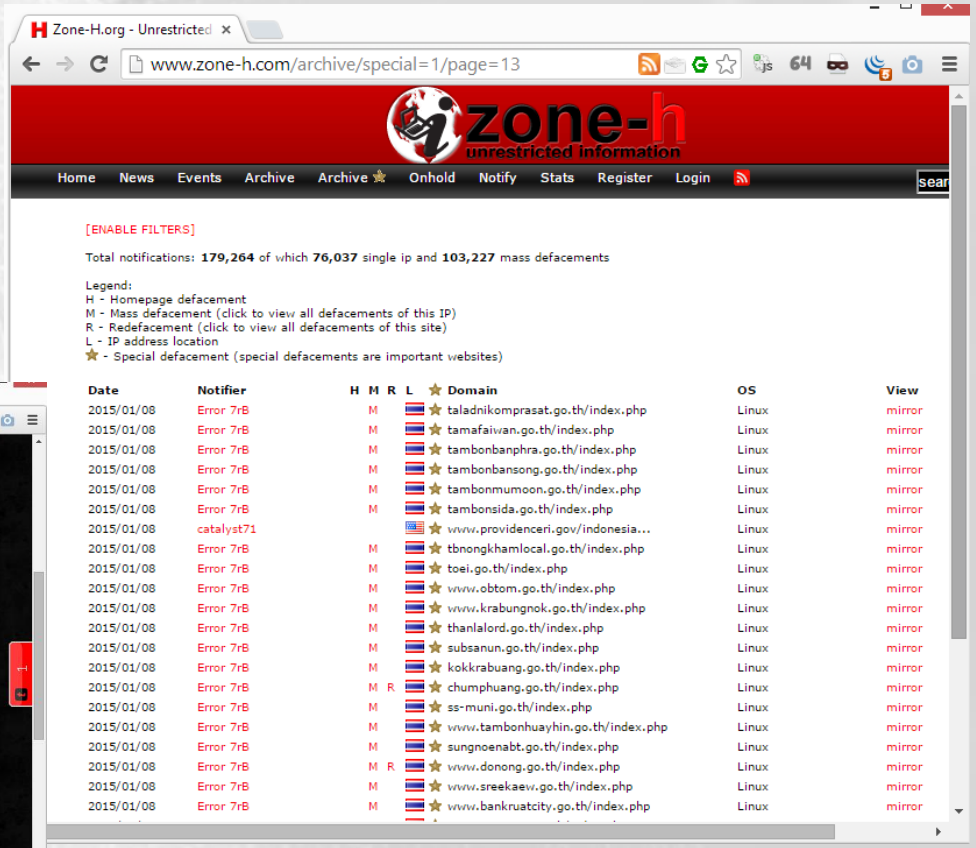
สวรรค์ของ Script kiddies



Hall of frame ของแฮกเกอร์ทั่วโลก

- http://dark-h.org
- http://www.zone-h.org
- http://zone-hc.com/
- http://www.hack-mirror.com/
- .....





ปี 2557 ไทยเซิร์ตรวบรวมข้อมูล Web defacement หน่วยงานภายในประเทศไทยถึง 2500 รายการ โดย 70% เป็นหน่วยงานภาครัฐของไทย

H A C K any website with only one click !!
Try it this amazing new hack software !!

https://www.youtube.com/watch?v=cIs1oRgWEE8

What Does a Malicious Hacker Do?

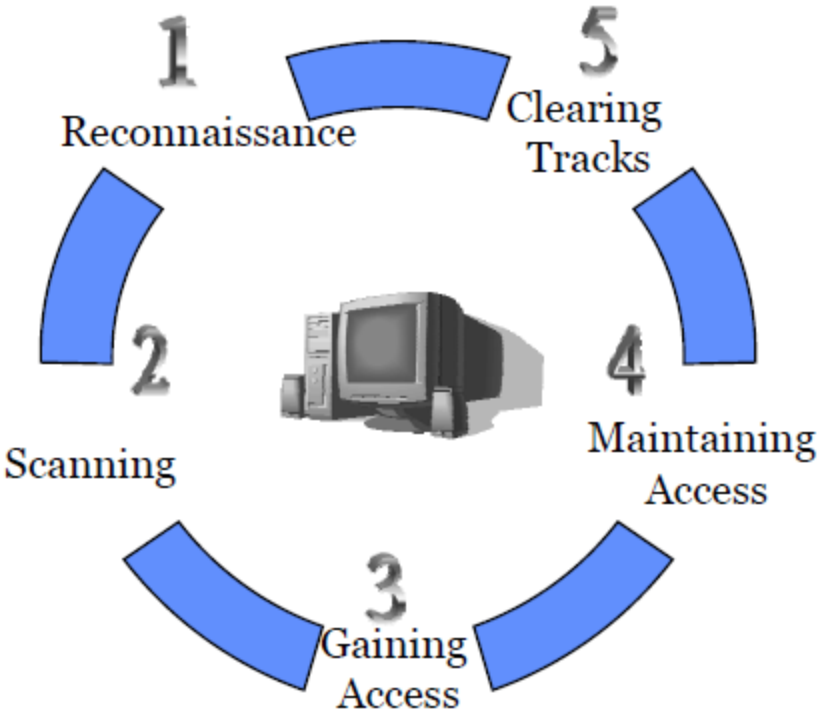
Reconnaissance
- Active/passive

Scanning

Gaining access
- Operating system level/application level
- Network level
- Denial of service

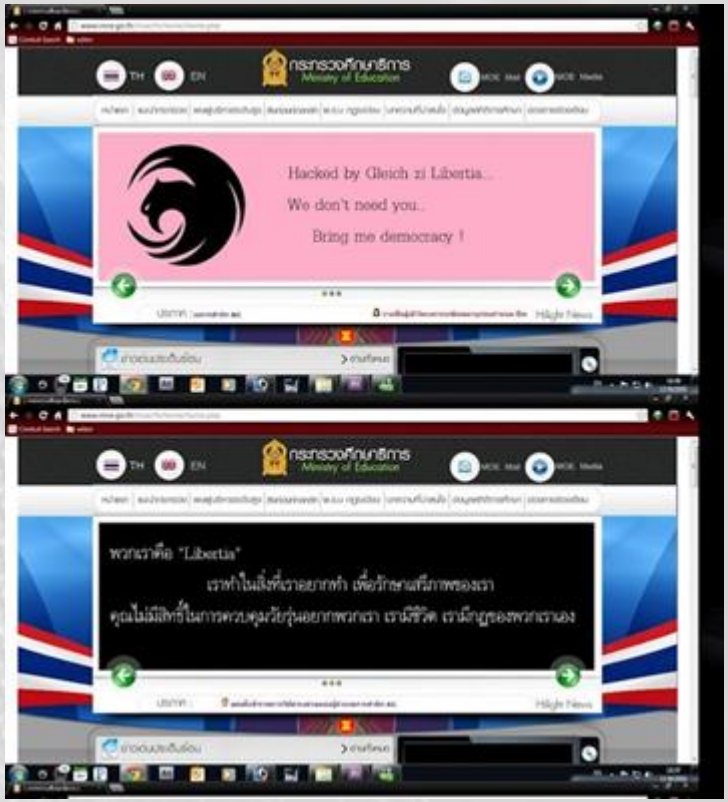
Maintaining access
- Uploading/altering/ downloading programs or data

Clearing tracks

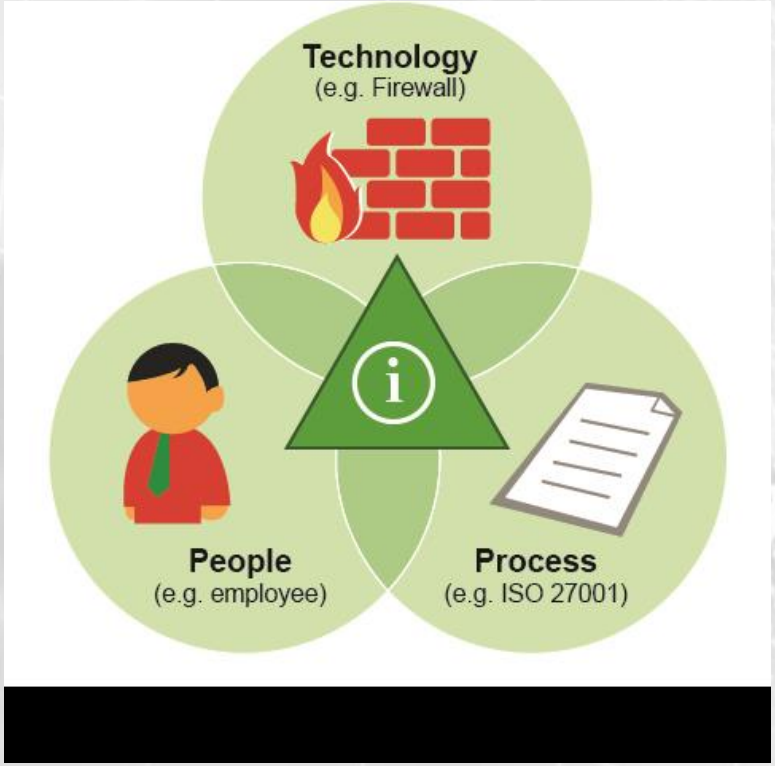


วัตถุประสงค์ของการโจมตีเว็บไซต์มีหลากหลาย

- ใช้เป็นตัวแทนในการโจมตีระบบอื่นๆต่อไป
  - Botnet
- เหตุผลทางการสังคม/การเมือง
  - Discredit , Squeeze (Sony haked)
  - DoS / DDoS
- ทำเพื่อสนุก อยากลองวิชา
- เผยแพร่มัลแวร์
  - 1,735 เคสในปี 2557
- หลอกลวงเอาเงิน / หาผลประโยชน์
  - Phishing (1,010 เคสในปี 2557)
  - SEO
- อยากให้มีคนรู้จัก
  - Hall of frame



ปัจจัยของการถูกแฮกเว็บไซต์



- ผู้ดูแลเครื่องบริการเว็บ (Web admin) ตั้งรหัสผ่าน 123456 ในหน้าล๊อกอินเข้าสู่ส่วนบริหารจัดการเว็บไซต์
- ไม่มีการควบคุมเรื่องการใช้งาน USB Drive จนทำให้ติดมัลแวร์บนเครื่องให้บริการเว็บไซต์
- ไม่มีอุปกรณ์ด้านการป้องกันการโจมตีเว็บไซต์

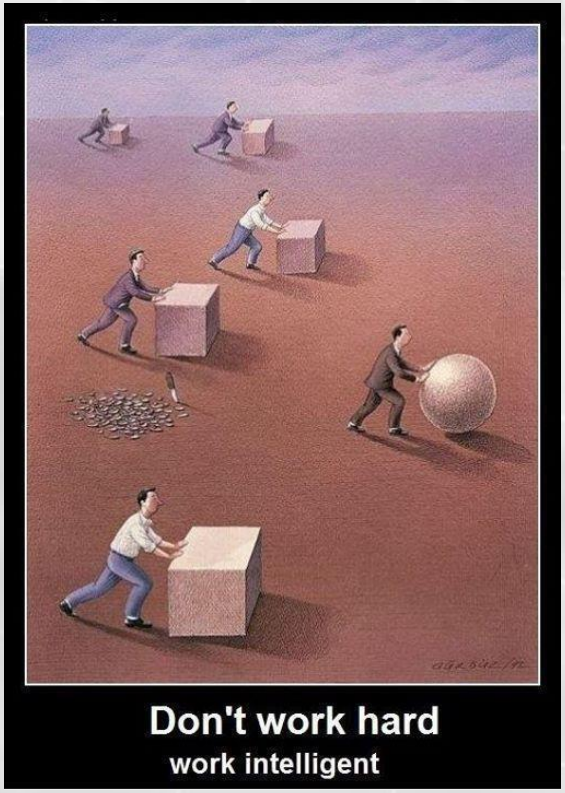
สถานการณ์ปัญหาเกี่ยวกับการโจมตีเว็บไซต์ในประเทศไทย

- เว็บไซต์ส่วนใหญ่ที่ถูกแฮกอยู่ในส่วนของ**ภาครัฐและภาคการศึกษา**
- ผู้พัฒนาเว็บไซต์พัฒนาโดย**คำนึงถึงแต่ฟังก์ชันการใช้งาน ไม่คำนึงถึงเรื่องความมั่นคงปลอดภัย** ส่งผลให้เว็บไซต์ที่พัฒนามีช่องโหว่
- ผู้ดูแลเครื่องบริการเว็บ**ไม่มีความตระหนักในเรื่องความมั่นคงปลอดภัย** ทำให้การแก้ไขปัญหาไม่ได้รับความร่วมมือ
- การขาดบุคคลากร โดยบางแห่งมีแต่เพียง เว็บมาสเตอร์ หรือในบางแห่งไม่มีเจ้าหน้าที่ด้านสารสนเทศเลย ทำให้ **ไม่สามารถตรวจสอบและแก้ไขปัญหาเว็บไซต์ที่ต้นเหตุ**
  - ทำได้เพียงแก้ปัญหาที่ปลายเหตุ ซึ่งปัญหาของการแฮกเข้ามานั้นยังคงอยู่และเกิดเหตุซ้ำอยู่ตลอดเวลา

วิเคราะห์ปัจจัยของการถูกแฮกเว็บไซต์

- ต่อให้มีกระบวนการที่ดีแต่คนไม่ปฏิบัติตามก็ **“ไม่เกิดประโยชน์”**
- ต่อให้มีเทคโนโลยีการป้องกันเว็บไซต์ที่ดีแค่ไหน แต่ถ้าคนใช้งานไม่เป็นก็ **“ไม่เกิดประโยชน์”**
- คนเป็นปัจจัยที่ทำให้เกิดปัญหา แต่ในขณะเดียวกันก็เป็นปัจจัยเดียวที่ถูกใช้ในการแก้ปัญหาอย่างมีประสิทธิภาพได่
- หากทีมมี **“คนที่มีประสิทธิภาพ”** ปัจจัยทุกอย่างจะถูกเชื่อมโยงและแก้ไขให้ดีขึ้น
- ศักยภาพไม่ได้เกิดขึ้นเอง **“ต้องพัฒนา”**

เมื่อ “คนมีประสิทธิภาพ”



- ผู้ดูแลเครื่องบริการเว็บ (Web admin) ตั้งรหัสผ่าน 123456 ในหน้าล๊อกอินเข้าสู่ส่วนบริหารจัดการเว็บไซต์
  - **แต่ถ้ามีกระบวนการและการควบคุมที่ดีย่อมช่วยแก้ไขปัญหาได้**
- ไม่มีการควบคุมเรื่องการใช้งานระบบจนทำให้ติดมัลแวร์บนเครื่องให้บริการเว็บไซต์
  - **แต่ถ้ามีการระบุเรื่องแนวทางการปฏิบัติงานด้านความมั่นคงปลอดภัยไว้ชัด ย่อมช่วยแก้ไขปัญหาได้**
- ไม่มีอุปกรณ์ป้องกันด้านการโจมตีเว็บไซต์
  - **แต่ถ้าคนมีความรู้ บางเรื่องก็ไม่จำเป็นต้องจัดหาอุปกรณ์ใดๆ รวมถึงคนที่มีประสิทธิภาพย่อมหมั่นแสวงหาความรู้เพิ่มเติมอยู่ตลอดเวลา**

แล้วจะป้องกันการถูกแฮกเว็บไซต์ได้ยังไง

แนวทาง	เวลา	เงิน	ประสิทธิภาพ
เรียนรู้วิธีการโจมตี และนำมาลองประยุกต์ทดสอบกับเว็บไซต์	มาก	มาก	ปานกลาง
เรียนรู้จากประสบการณ์ที่เคยพบ แก้ไขในแต่ละจุด	ปานกลาง	น้อย	ปานกลาง
เรียนรู้วิธีการป้องกันจากสิ่งที่มีคนรวบรวมแล้วนำมาประยุกต์ใช้	**ปานกลาง**	**น้อย**	**มาก**
จัดหาอุปกรณ์ด้านการป้องกันการโจมตีเว็บไซต์	น้อย	มาก	ปานกลาง

Website Security Standard ช่วยท่านได้

หมายเหตุ : ตารางดังกล่าวจัดทำขึ้น เพื่อเป็นตัวอย่างในการพิจารณาแนวทางการแก้ไขปัญหา ซึ่งสามารถประยุกต์นำไปประยุกต์ใช้เพื่อการประเมินแนวทางการป้องกันการถูกแฮกเว็บไซต์ได้ต่อไป

Website Security Standard (WSS) คืออะไร

- **ชื่อภาษาไทย** : มาตรฐานการรักษาความมั่นคงปลอดภัยสำหรับเว็บไซต์
- **เวอร์ชันเอกสาร** : 1.0
- **สถานะเอกสาร** : ข้อเสนอแนะมาตรฐานด้านเทคโนโลยีสารสนเทศและการสื่อสารที่จำเป็นต่อธุรกรรมทางอิเล็กทรอนิกส์
- **วันที่ประกาศ** : 30 กันยายน 2557
- **รหัสเอกสาร** : ขมธอ.1 – 2557
- **ประกาศโดย** : สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน) กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร
- **ลิงก์ดาวน์โหลด** : https://standard.etda.or.th/wp/wp-content/uploads/2014/09/Website-Security-Standard_V6E6.2.pdf
- **จุดประสงค์** : เพื่อส่งเสริมให้ผู้ที่เกี่ยวข้องกับการบริหารจัดการและดูแลเว็บไซต์สามารถพัฒนาหรือจัดทำเว็บไซต์ให้มีความมั่นคงปลอดภัย และดำเนินมาตรการในการป้องกัน ตรวจสอบ ลดความเสี่ยง หรือสามารถรับมือกับภัยคุกคามที่มีต่อเว็บไซต์ เพื่อสร้างความเชื่อมั่นในการทำธุรกรรมทางอิเล็กทรอนิกส์

แหล่งข้อมูลอ้างอิงของ WSS

- NIST SP 800-44 Guidelines on Securing Public Web Servers
- OWASP Open Web Application Security Project
- ข้อแนะนำแก้ไขและป้องกันข้อบกพร่องหรือจุดอ่อนของเว็บไซต์ของไทยเซิร์ต (ThaiCERT)
- คู่มือ “How to Secure Your Website” ของ ส านักงานส่งเสริมเทคโนโลยีสารสนเทศประเทศญี่ปุ่น (Information-Technology Promotion Agency (IPA), Japan)

ความต้องการของการจัดทำ WSS

- ให้หน่วยงานมีความรู้และความตระหนักถึงเรื่องการแฮกหรือการโจมตีเว็บไซต์
- ให้หน่วยงานมีความรู้เกี่ยวกับแนวทางการป้องกันการแฮกเว็บไซต์ ครอบคลุมทั้งการดูแลและการพัฒนาเว็บไซต์ให้มีความมั่นคงปลอดภัย
- ให้หน่วยงานมีความรู้ในเรื่องการรับมือสถานการณ์ภัยคุกคามที่อาจเกิดขึ้นต่อเว็บไซต์และสามารถนำไปประยุกต์ใช้กับการทำงานได้
- ให้หน่วยงานมีแนวทางในการประเมินตนเอง (Self-assessment) เกี่ยวกับความมั่นคงปลอดภัยของเว็บไซต์ที่อยู่ในความดูแล

ความต้องการของการจัดทำ WSS (ต่อ)

- ให้หน่วยงานสามารถยืนยันความสอดคล้องกับมาตรฐาน เพื่อให้ในการประกาศการรับรองตนเองกับหน่วยงานภายนอก

- ให้หน่วยงานสามารถขอรับการรับรอง (Certification) มาตรฐานการรักษาความมั่นคงปลอดภัยสำหรับผู้ดูแลและพัฒนาเว็บไซต์จากหน่วยตรวจสอบและรับรอง (Conformity assessment body)

ความครอบคลุมเนื้อหา WSS

(1) การวางแผน (Planning) ได้แก่ การวางแผนด้านความมั่นคงปลอดภัยของเว็บไซต์แนวทางการเลือกผู้รับจดทะเบียนชื่อโดเมน แนวทางการเลือกผู้ให้บริการเว็บโฮสติ้ง และแนวทางในการเลือกใช้ระบบบริหารจัดการเว็บไซต์ (CMS)

(2) การติดตั้งและการตั้งค่าที่เกี่ยวข้องกับเว็บไซต์ (Installation and Configuration) เป็นข้อกำหนดที่มุ่งเน้นให้มีการติดตั้งและการตั้งค่าของ โปรแกรมสำาหรับให้บริการเว็บ ระบบบริหารจัดการเว็บไซต์ ระบบฐานข้อมูลและ Server-side script engine

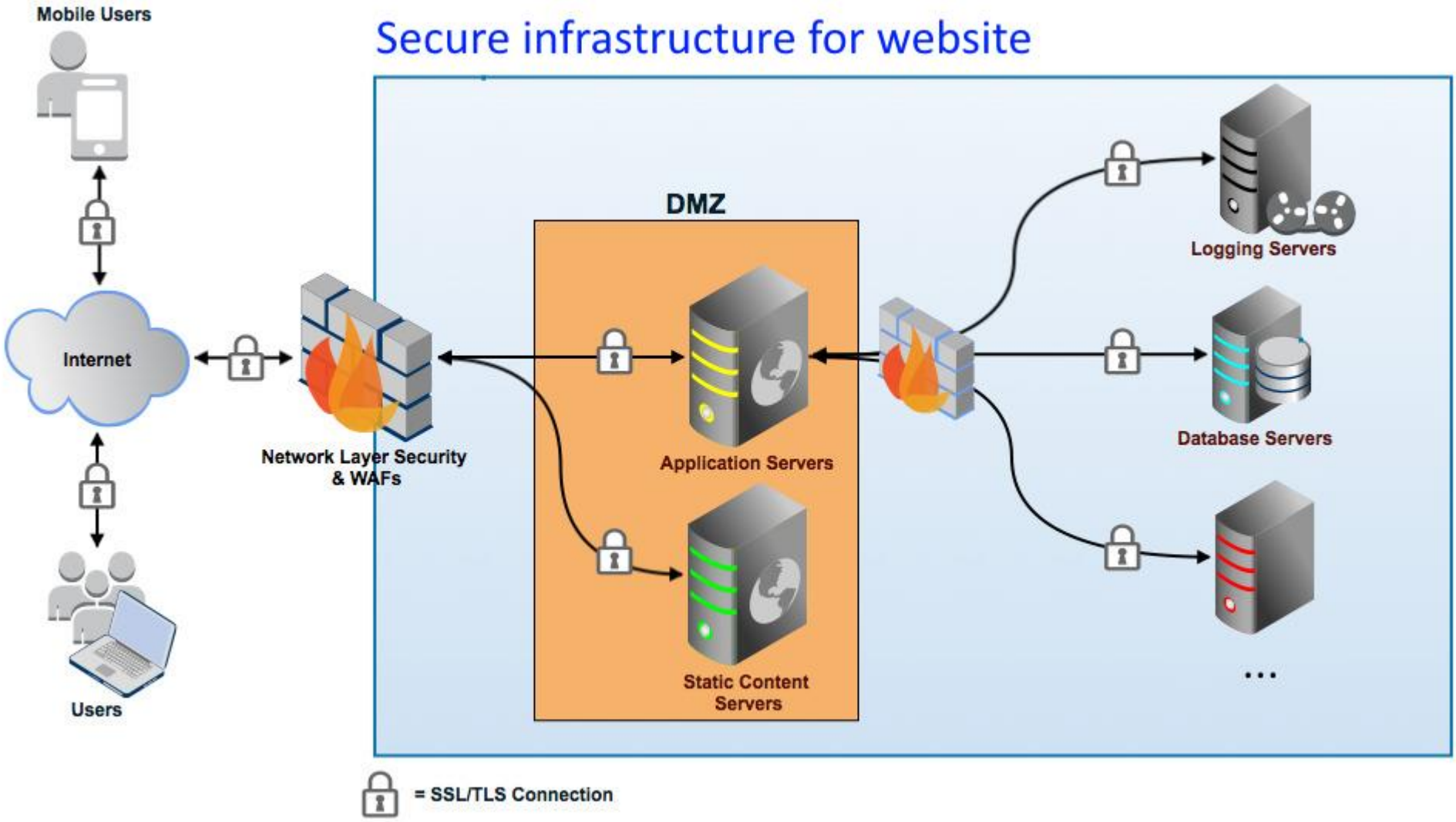
(3) การพัฒนาโปรแกรมประยุกต์บนเว็บอย่างมั่นคงปลอดภัย เน้นการป้องกันการโจมตีด้วยเทคนิคต่างๆ ที่พบบ่อยจากไทยเซิร์ต (ThaiCERT) แนวทางการป้องกันจากเอกสารของ IPA และ OWASP

(4) การรับมือเหตุภัยคุกคาม (Incident Handling) เน้นให้ผู้ดูแลเครื่องบริการเว็บสามารถรับมือกับเหตุภัยคุกคามด้านความมั่นคงปลอดภัยที่เกิดขึ้นกับเว็บไซต์ได้แก่ กรณีเว็บไซต์ถูกบุกรุกและควบคุม (Intrusions) กรณีการถูกโจมตีในลักษณะ (Denial of services: DoS) และ กรณีโดเมนถูก ขโมย (Domain Hijack) เป็นต้น

ความคลอบคลุมเนื้อหา WSS (ต่อ)

- ครอบคลุมกลุ่มคนสำคัญที่ต้องเข้าร่วมการป้องกัน
  - ผู้ดูแลเครื่องบริการเว็บ (เครื่องแม่ข่ายเว็บ โฮสติ้งส์ เว็บมาสเตอร์)
    - มีผลต่อการตั้งค่าของสภาพแวดล้อมที่เกี่ยวข้องกับเว็บไซต์
  - ผู้พัฒนาเว็บไซต์ (ผู้พัฒนาและจัดทำเว็บไซต์)
    - มีผลต่อเว็บไซต์และฐานข้อมูล
- ครอบคลุมอุปกรณ์และปัจจัยต่างๆที่ต้องป้องกัน
  - เครือข่าย
  - บริการบนเครื่องแม่ข่าย
  - เว็บไซต์
  - ฐานข้อมูล

Secure infrastructure for website



การแฮกเว็บไซต์ที่พบบ่อย (1)

1. แฮกผ่านช่องโหว่และปัจจัยบริการที่เปิดให้บริการบนเครื่องให้บริการเว็บ
   - ตัวอย่างเช่น
     - FTP
     - Remote desktop
     - SSH
     - Database
     - Web server
     - Server-side script Engine

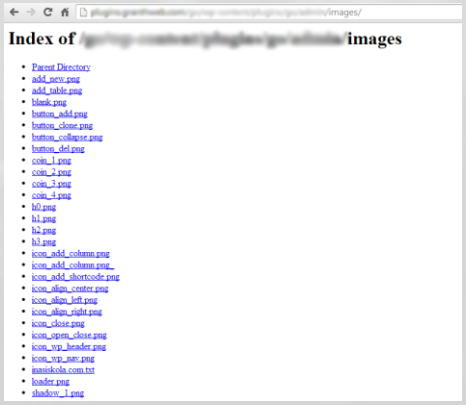
การแฮกเว็บไซต์ที่พบบ่อย (1) (ต่อ)

1. แฮกผ่านช่องโหว่และปัจจัยบริการที่เปิดให้บริการบนเครื่องให้บริการเว็บ
   - สาเหตุ เช่น
     - ไม่อัปเดตซอฟต์แวร์ที่ใช้สำหรับให้บริการ
     - ตั้งค่าบริการที่ไม่ปลอดภัย
       - เช่น กำหนดรหัสผ่านเป็น 123456
     - เปิดการใช้งานซอฟต์แวร์หรือบริการที่ไม่จำเป็น และไม่มีการดูแล
     - ใช้งานการตั้งค่าเริ่มต้นของบริการที่ติดตั้งมาทันที
       - เช่น Default password ของบริการ การเปิดให้มี Anonymous login

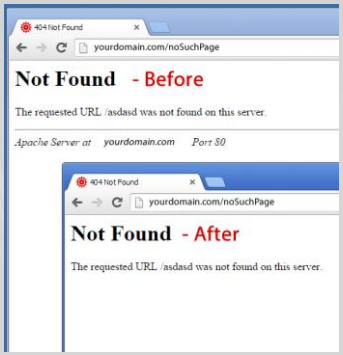
การตั้งค่าเครื่องบริการเว็บอย่างมั่นคงปลอดภัย

1. การตั้งค่าโปรแกรมสำหรับให้บริการเว็บ (Web server software)
2. การตั้งค่าระบบบริหารจัดการเว็บไซต์ (CMS)
3. การตั้งค่าฐานข้อมูล (Database system)
4. การตั้งค่า Server-side Script Engine
5. การกำหนดและรักษารหัสผ่าน

การตั้งค่าโปรแกรมสำหรับให้บริการเว็บ (Web server)

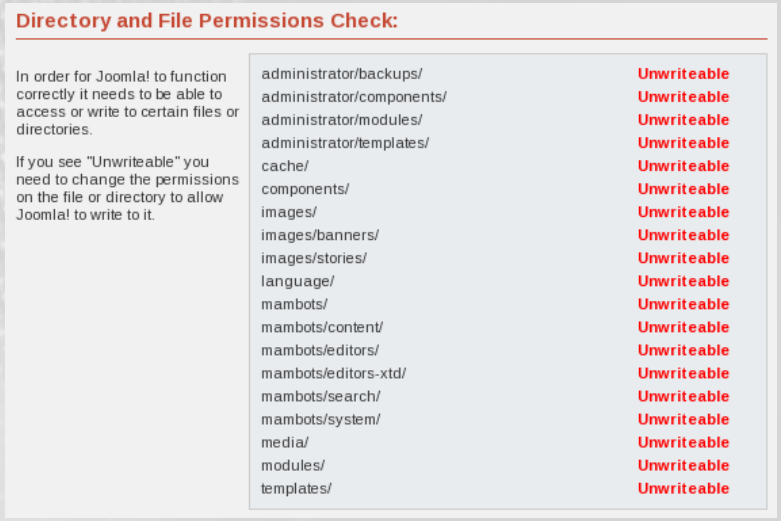


1. อัปเดตโปรแกรมอย่างสม่ำเสมอ
2. ปิดการแสดงข้อความแสดงข้อผิดพลาด (Error Message)
3. กำหนดสิทธิในการเข้าถึงไฟล์ที่เกี่ยวข้องทั้งหมดให้เหมาะสมกับการใช้งาน เช่น ไม่เปิดโหมด Directory listing
4. ลบไฟล์ตัวอย่างโปรแกรม ตัวอย่างไฟล์ข้อมูล บัญชีผู้ใช้ที่ไม่ได้ใช้งาน เช่น บัญชีซึ่งมีการใช้งานระหว่างกระบวนการติดตั้งเครื่องบริการเว็บทั้งหมด
5. ในกรณีที่เว็บไซต์มีความสำคัญและต้องการจำกัดการใช้งาน ให้จำกัดหมายเลขไอพีปลายทางที่อนุญาตให้เชื่อมต่อ (Whitelist IP)
6. ปิดบริการต่างๆ ที่ไม่จำเป็นบนเครื่องบริการเว็บ เช่น Phpmyadmin เป็นต้น รวมถึงโปรแกรมบริการประเภท Remote Access เช่น Remote Desktop, VNC, SSH, Telnet



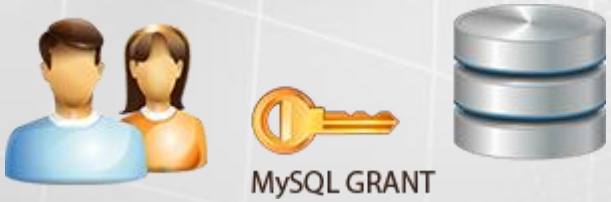
การตั้งค่าระบบบริหารจัดการเว็บไซต์ (CMS)

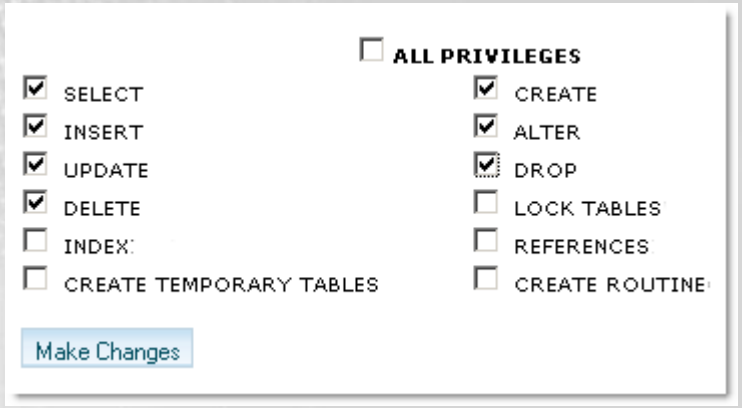
1. กำหนดสิทธิการใช้งานไฟล์ต่างๆให้เหมาะสม
2. ตรวจสอบว่ามีไฟล์หรือโปรแกรมเสริม รวมถึงบัญชีการใช้งานที่ไม่จำเป็น หรือไม่ได้ใช้งานปรากฏอยู่หรือไม่ถ้ามีให้ลบทิ้งเพื่อลดโอกาสที่อาจถูกโจมตี
3. อัปเดตเวอร์ชั่นของ CMS อยู่เสมอ โดยดาวน์โหลดไฟล์จากเว็บไซต์หลักของผู้ให้บริการระบบบริหารจัดการเว็บไซต์เท่านั้น
4. ตั้งค่ารหัสผ่านของบัญชีใช้งานให้เป็นรหัสผ่านที่มีความมั่นคงปลอดภัย
5. เปลี่ยน table prefix ของฐานข้อมูลที่มาในระหว่างการติดตั้งระบบบริหารจัดการเว็บไซต์ เนื่องจากอาจถูกใช้เป็นช่องทางให้ผู้ประสงค์ร้ายสามารถทราบถึงโครงสร้างและตารางในฐานข้อมูลได้



การตั้งค่าฐานข้อมูล (Database system)

1. จำกัดการใช้งานจากเครื่องที่มีสิทธิการเข้าถึงฐานข้อมูล (Whitelist IP)
2. ลบบัญชีผู้ใช้ที่ไม่ได้มีการใช้งานออกจากระบบฐานข้อมูล หรือเปลี่ยนรหัสผ่านของบัญชีผู้ใช้ดังกล่าวให้เป็นรหัสผ่านที่มีความมั่นคงปลอดภัย
3. ตั้งค่าฐานข้อมูล โดยต้องไม่อนุญาตให้ใช้งานรหัสผ่านที่มีค่าว่าง (Null password)
4. แยกสิทธิการใช้งานโดยสร้างบัญชีผู้ใช้งานแยกกันในแต่ละแอปพลิเคชันที่เชื่อมต่อเข้ามา และกำหนดสิทธิโดยยึดหลัก Least Priviledge
5. อัปเดตเวอร์ชั่นของโปรแกรมระบบฐานข้อมูล
6. รหัสผ่านที่เก็บในฐานข้อมูล ต้องมีการเข้ารหัสเสมอ
7. อัปเดตเวอร์ชั่นของโปรแกรมระบบฐานข้อมูล





การตั้งค่า Server-side Script Engine

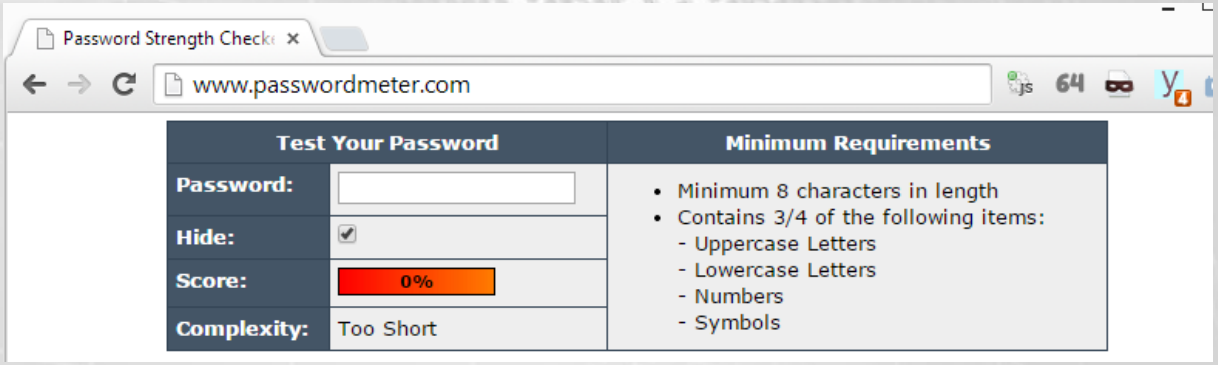
1. อัปเดตโปรแกรมอย่างสม่ำเสมอ
2. กำหนดค่าติดตั้งไม่ให้ Server-side Script Engine แสดงข้อมูลเวอร์ชั่นที่ใช้งานเนื่องจากอาจเป็นช่องทางให้ผู้ประสงค์ร้ายล่วงรู้เวอร์ชั่นและค้นหาช่องโหว่ต่อไป
3. กำหนดค่าติดตั้ง Server-side Script Engine ให้ตรงกับการทำงานของระบบ เพื่อลดความจากการถูกโจมตี

Example : Secure configuration for PHP

```
expose_php = off
file_uploads = off
allow_url_fopen = off
allow_url_include = off
disable_functions = shell_exec,system,passthru,exec,curl_exec,proc_open,parse_ini_file
open_basedir = /var/www
magic_quotes_gpc = On
register_globals off
```

การกำหนดและรักษารหัสผ่าน

1. ตั้งค่ารหัสผ่านให้มีความมั่นคงปลอดภัย (Strong password) โดยรหัสผ่านควรประกอบด้วยตัวอักษรทั้งตัวเล็กและตัวใหญ่ผสมกัน มีตัวเลขและสัญลักษณ์พิเศษอย่างน้อย 1 หลัก และต้องมีความยาวทั้งหมดไม่น้อยกว่า 8 หลัก
2. กำหนดให้มีการเปลี่ยนรหัสผ่านอย่างสม่ำเสมอจะช่วยลดโอกาสจากการถูกคาดเดารหัสผ่าน
3. ไม่เก็บรหัสผ่านที่ไม่มีการเข้ารหัสลับบนเครื่องบริการเว็บ หากจำเป็นต้องมีการเก็บรหัสผ่านควรอยู่ในรูปที่มีการเข้ารหัส เช่น เก็บเป็นค่าแฮช (Hash value) ควรใช้ขั้นตอนวิธี(Algorithm) โดยใช้อัลกอริทึมที่ได้รับความเชื่อถือ เช่น SHA-224 SHA-256 SHA-384 SHA-512 เป็นต้น



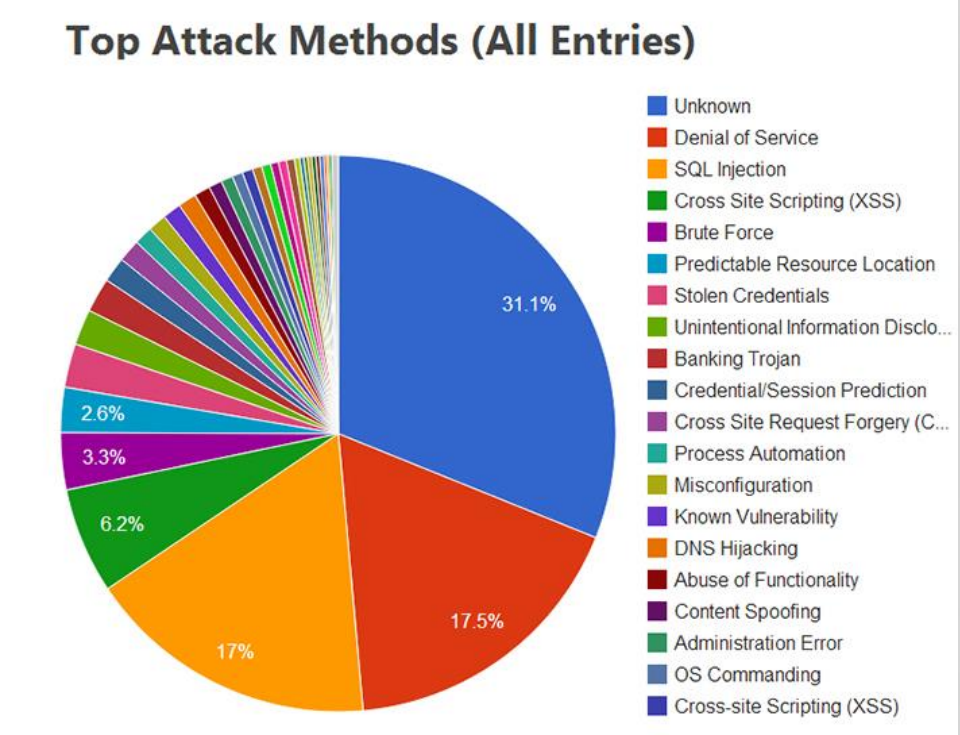
http://www.passwordmeter.com/

การแฮกเว็บไซต์ที่พบบ่อย (2)

1. แฮกผ่านช่องโหว่และปัจจัยบริการที่เปิดให้บริการบนเครื่องให้บริการเว็บ
2. แฮกผ่านช่องโหว่ของเว็บไซต์และโปรแกรม
    - เทคนิคที่พบบ่อย เช่น
        - Malicious injection (SQL Injection ,Command Injection ,....)
        - Session Hijacking
        - Cross-site Scripting
        - CSRF
        - Sensitive Data Explosure

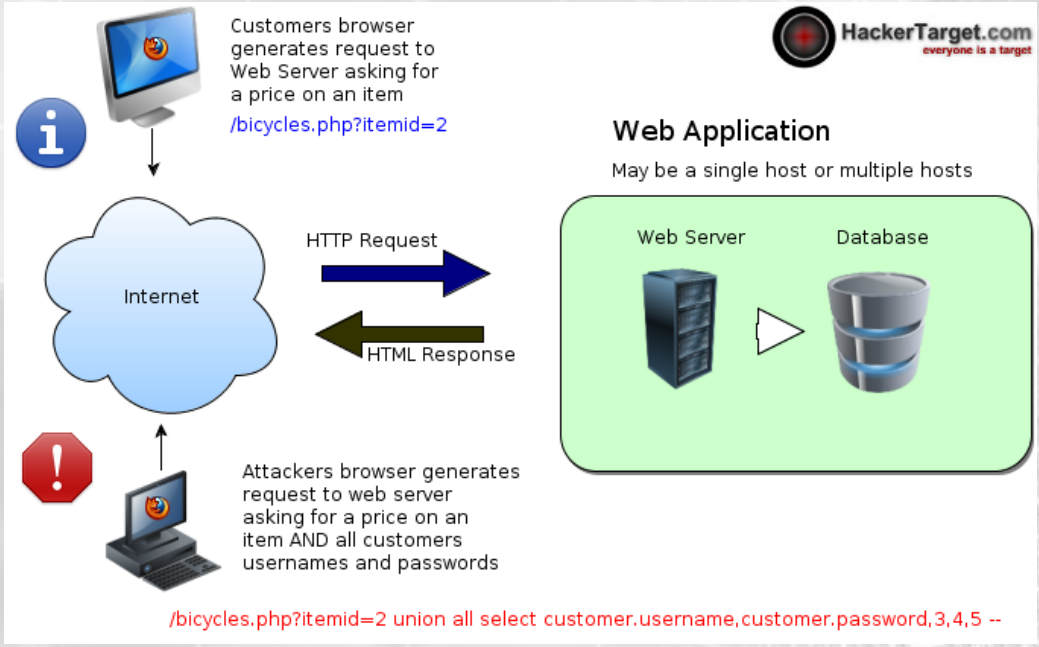
สถิติการแฮกเว็บไซต์ด้วยเทคนิคต่างๆจาก Trustwave

ในปี 2556 พบรูปแบบการโจมตีเว็บไซต์ด้วยเทคนิค DoS (17.5%) SQL Injection (17%) และ Cross-site scripting (6.2%) ถูกแจ้งใน 3 อันดับแรก ตามลำดับ จากรายงานของ Trustwave

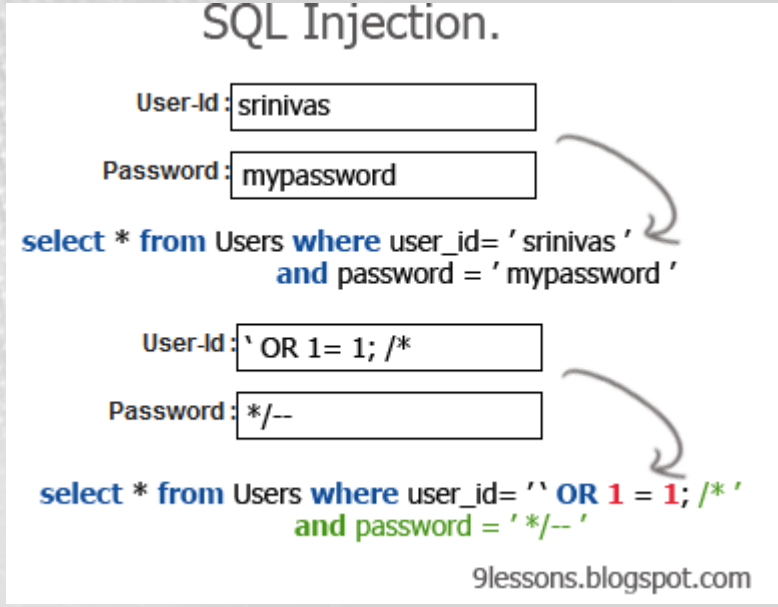


Malicious injection

เทคนิคการโจมตีด้วยเทคนิค SQL Injection



โจมตีโดยการส่งค่าซึ่งเป็นคำสั่ง SQL อันตรายผ่าน Input ของเว็บไซต์ เพื่อไปประมวลผลโดยตรงยังฐานข้อมูล



ผลของการโจมตีทำให้สามารถ Bypass การตรวจสอบเช่น การล๊อกอิน การเข้าถึงและปรับปรุงข้อมูลภายในฐานข้อมูล รวมถึงการส่งคำสั่งไปประมวลผลยังระบบปฏิบัติการได้ เช่น การสั่งเปิด Firewall บนระบบปฏิบัติการ

เทคนิคการโจมตีด้วยเทคนิค Command Injection

โจมตีโดยการส่งค่าซึ่งเป็นคำสั่ง เพื่อไปประมวลผลโดยตรงยังฐานระบบปฏิบัติการ เช่น การสั่งดูไฟล์บนระบบ หรือสั่งโจมตีระบบอื่นๆด้วยการ Ping เป็นต้น

ผลของการโจมตีทำให้ผู้ไม่ประสงค์ดีสามารถส่งคำสั่งไปประมวลผลยังระบบปฏิบัติการได้โดยตรง เช่น สั่งให้ Attack ระบบเครือข่ายอื่นๆด้วยคำสั่ง Ping เป็นต้น

การป้องกันการโจมตีด้วยเทคนิค Malicious injection

1. มีการทำ **Prepared Statement และ/หรือ Stored Procedure** ซึ่งเป็นวิธีการที่จะแยกคำสั่งในการประมวลผลและค่าที่จะนำไปประมวลผลบนฐานข้อมูลออกจากกัน

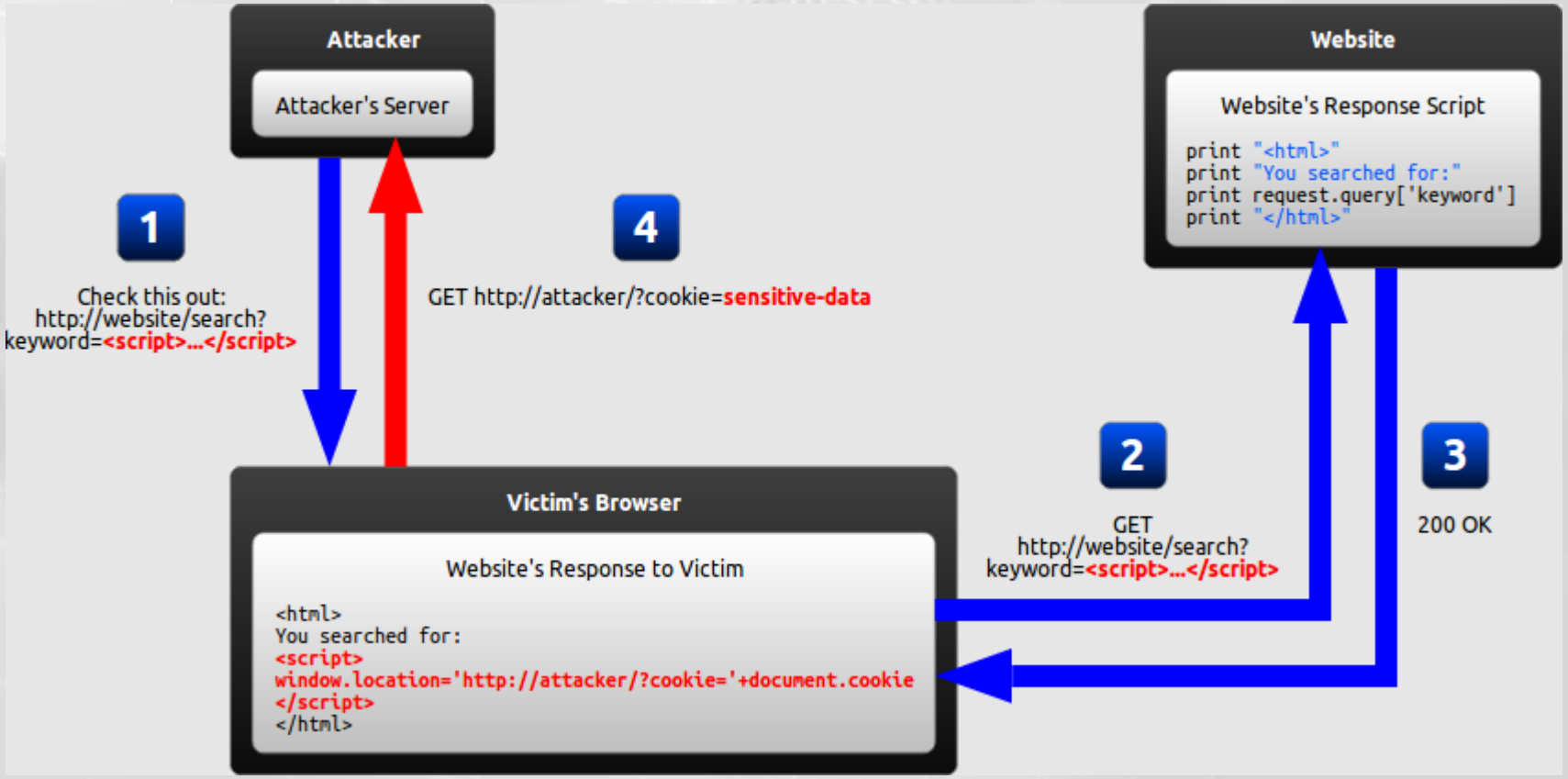
2. มีการทำ **Input validation** เป็นวิธีการที่ใช้ในการตรวจสอบข้อมูลที่ได้รับก่อนส่งมาประมวลผล ด้วยวิธีการทำ Whitelist และ Blacklist

3. มีการทำ **Encoding** หรือทำ **Sanitization** ก่อนนำค่ามาประมวลผล เพื่อป้องกันการโจมตีด้วยเทคนิคต่าง ๆ ข้อมูลที่ผ่านกระบวนการดังกล่าวจะถูกแปลงรูปแบบของข้อมูลที่ส่ง มาจากฝั่งผู้ใช้บริการให้อยู่ในรูปแบบที่ระบบนำไปประมวลผลได้โดยไม่อันตราย เช่น เปลี่ยนจาก ' OR 1=1 --' เป็น \' OR 1=1 --\'

Cross-site scripting

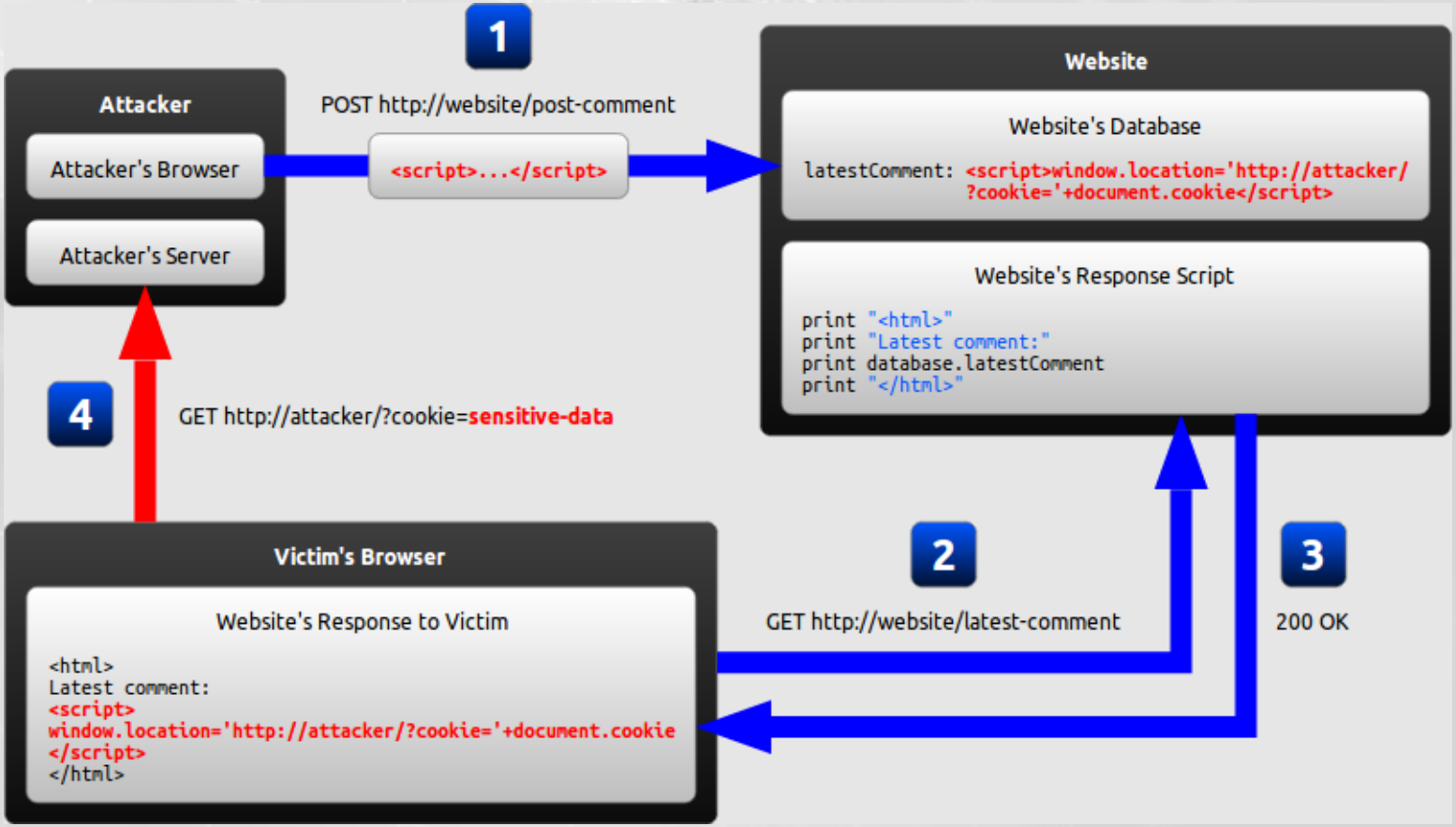
เทคนิคการโจมตีด้วยเทคนิค Reflected XSS

โจมตีโดยการส่ง Script อันตรายไปประมวลผลต่อครั้ง



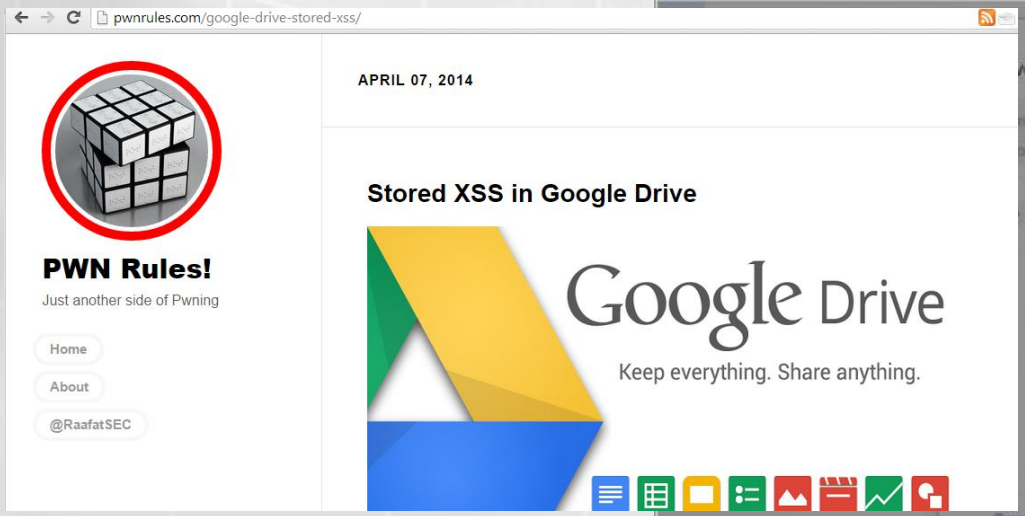
เทคนิคการโจมตีด้วยเทคนิค Persistent XSS

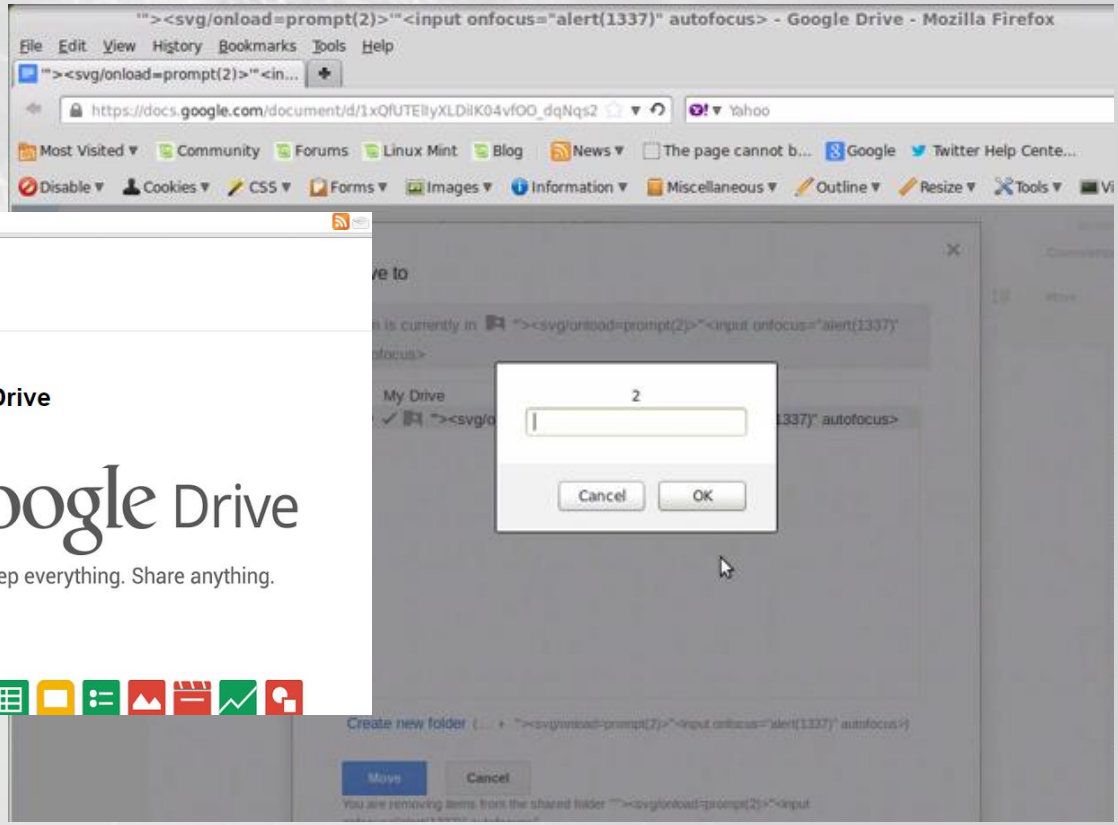
โจมตีโดยการส่ง Script อันตรายไปประมวลผลและเก็บลงฐานข้อมูล



ตัวอย่างการโจมตีด้วย Cross-site scripting ในอดีต

เว็บไซต์อย่าง Google ก็ยังมีช่องโหว่ XSS (APRIL 07, 2014)





การป้องกันการโจมตีด้วยเทคนิค Cross-site scripting

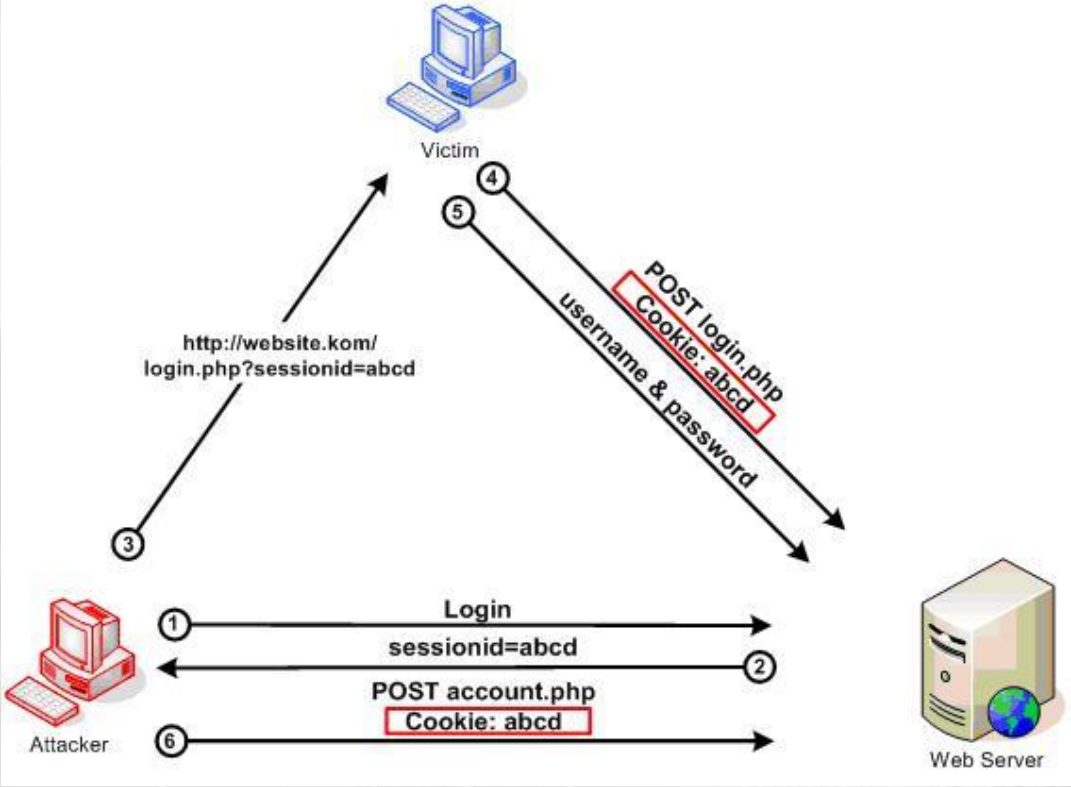
1. มีการทำ **Input validation** เป็นวิธีการที่ใช้ในการตรวจสอบข้อมูลที่ได้รับก่อนส่งมาประมวลผล ด้วยวิธีการทำ Whitelist และ Blacklist รวมถึงทำ Encoding ก่อนนำค่ามาประมวลผล ควรแปลงพวก "Non-alphanumeric data" ให้กลายเป็น HTML character เสียก่อน เช่น เครื่องหมายน้อยกว่า "<" ควรถูกแปลงเป็น "& l t ;" เป็นต้น

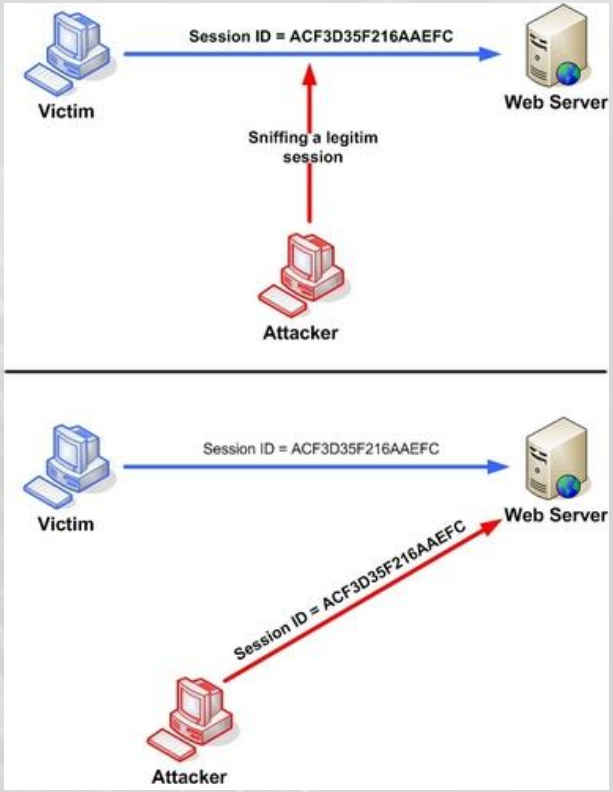
2. มีการทำ **Output validation** ในลักษณะ Sanitization เพื่อป้องกันการแสดงผลข้อมูลที่ไม่พึงประสงค์ยังฝั่งผู้ใช้บริการ ตัวอย่างเช่นการใช้งานฟังก์ชัน ตัวอย่างเช่น การใช้งานฟังก์ชั่น htmlentities ในภาษา PHP

3. มีการใช้งาน **HTTPOnly Cookie flag** เป็นรูปแบบการกำหนดค่าเพิ่มเติม (Flag) สำหรับป้องกันไม่ให้ฝั่งผู้ใช้บริการสามารถการเข้าถึงค่า Cookie ของระบบได้ หากระบบมีช่องโหว่ XSS แฮกเกอร์อาจส่งคำสั่งใช้เกิดการขโมยค่าเซสชัน แต่หากมีการกำหนดค่า HTTPOnly จะสามารถป้องกันการกระทำดังกล่าวได้

Session Hijacking

เทคนิคการโจมตีด้วยเทคนิค Session Hijack





โจมตีโดยการประยุกต์ค่าเซสชันที่แฮกเกอร์ได้รับ ให้กับผู้ใช้งาน และเมื่อผู้ใช้งานล๊อกอินระบบ ก็จะทำให้แฮกเกอร์สามารถสวมรอยเป็นผู้ใช้งานท่านนั้นได้ทันที (Session fixation)

โจมตีโดยการดักรับข้อมูลทางเครือข่ายในลักษณะที่เป็น HTTP (Clear-text) ส่งเมื่อได้ค่าเซสชันแล้วก็นำมากำหนดลงที่บราวเซอร์ของแฮกเกอร์เพื่อสวมรอยเข้าใช้งานแทน

การป้องกันการโจมตีด้วยเทคนิค Session hijacking

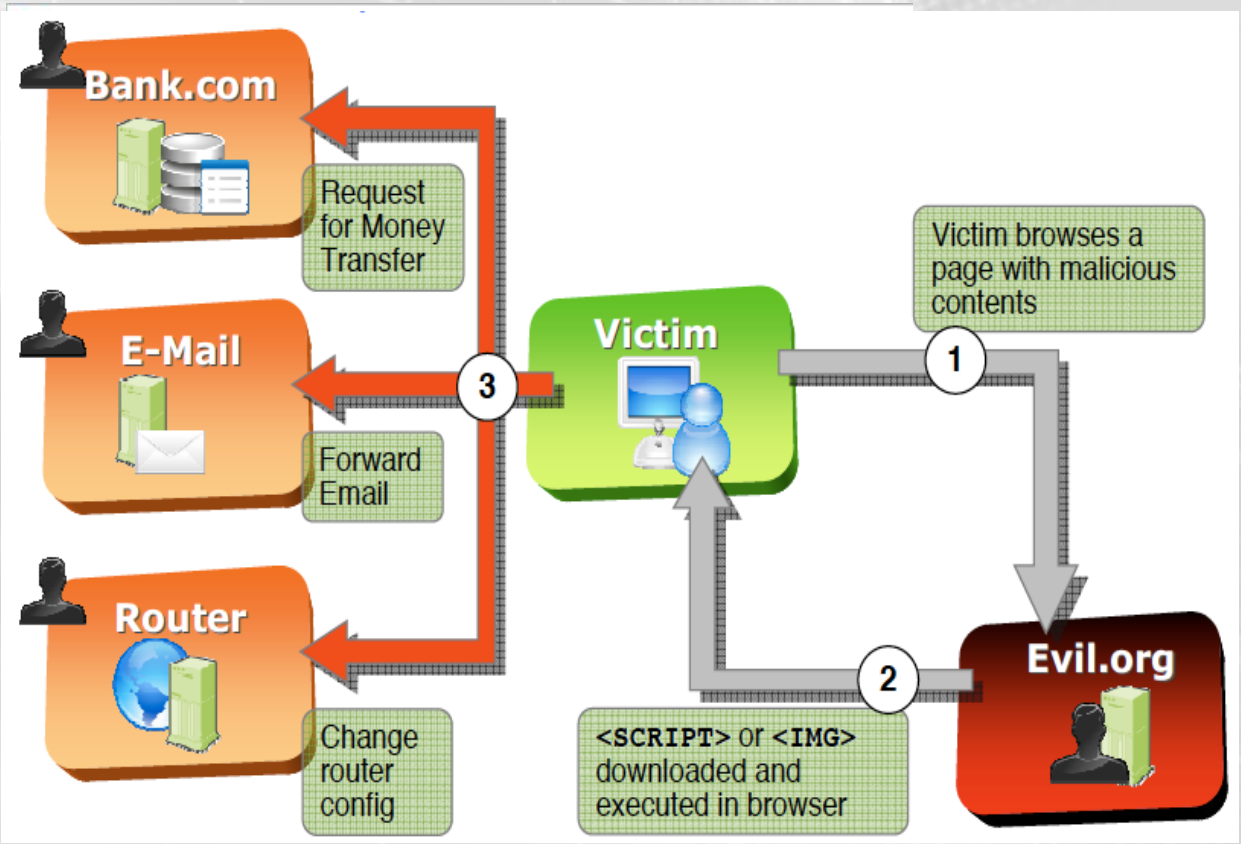
1. มีการใช้ Session ID ที่เป็นค่าสุ่ม (**Random session ID**) คาดเดาไม่ได้ และเป็นค่าที่ไม่มีการนำกลับมาใช้ซ้ำ เพื่อป้องกันการคาดเดา

2. การส่งค่า Session ID ต้องรับส่งในช่องทางการสื่อสารที่มีการเข้ารหัสลับ (**Encrypted connection**) เช่น การส่งข้อมูลผ่านโพรโทคอล HTTPS เพื่อป้องกันการลักลอบดักรับข้อมูล

3. มีการเช็คค่า Session ID ร่วมกับปัจจัยอื่นๆ เช่น **IP-Address User-Agent HTTP-Referer** เพื่อป้องกันการสวมรอยค่าเซสชันจากแฮกเกอร์

CSRF (Cross-site script forgery)

เทคนิคการโจมตีด้วยเทคนิค CSRF



โจมตีในลักษณะคล้ายกับ XSS คือ ทำให้ผู้ใช้งานประมวลผล Script อันตราย เช่น สั่งให้ส่งอีเมล หรือแม้แต่สั่งให้โอนเงิน ซึ่งปัญหาเกิดขึ้นเนื่องจากเว็บไซต์หรือระบบปลายทางไม่มีการตรวจสอบเซสชันที่ดี เช่น ไม่มีการใช้งาน Captcha

ผลของการโจมตีทำให้ผู้ใช้งาน สามารถถูกหลอกให้ประมวลผลคำสั่งอันตรายใดๆ บนเว็บไซต์ที่มีช่องโหว่ดังกล่าว

การป้องกันการโจมตีด้วยเทคนิค CSRF

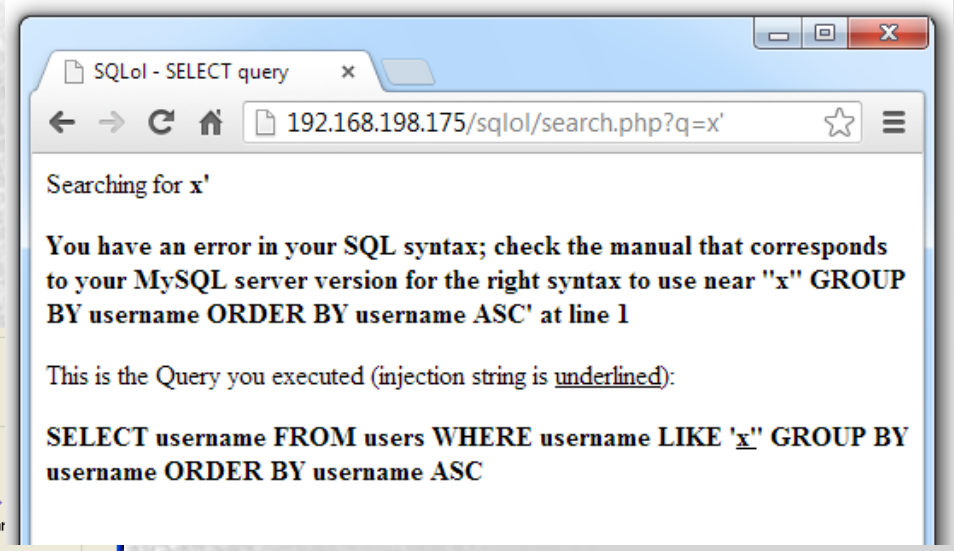
1. มีการใช้งาน **Unique Token และ/หรือตรวจสอบ Referrer ร่วมกับการส่งข้อมูล** หรือคำสั่งผ่านแบบฟอร์ม เพื่อให้แน่ใจว่าข้อมูลในแบบฟอร์มที่จะส่งมาประมวลผลในแต่ละครั้งนั้นเป็นข้อมูลที่เกิดมาจากการที่ผู้ใช้บริการจริง ไม่ใช่โปรแกรมอัตโนมัติหรือสคริปท์ที่ใช้ในการโจมตีแต่อย่างใด

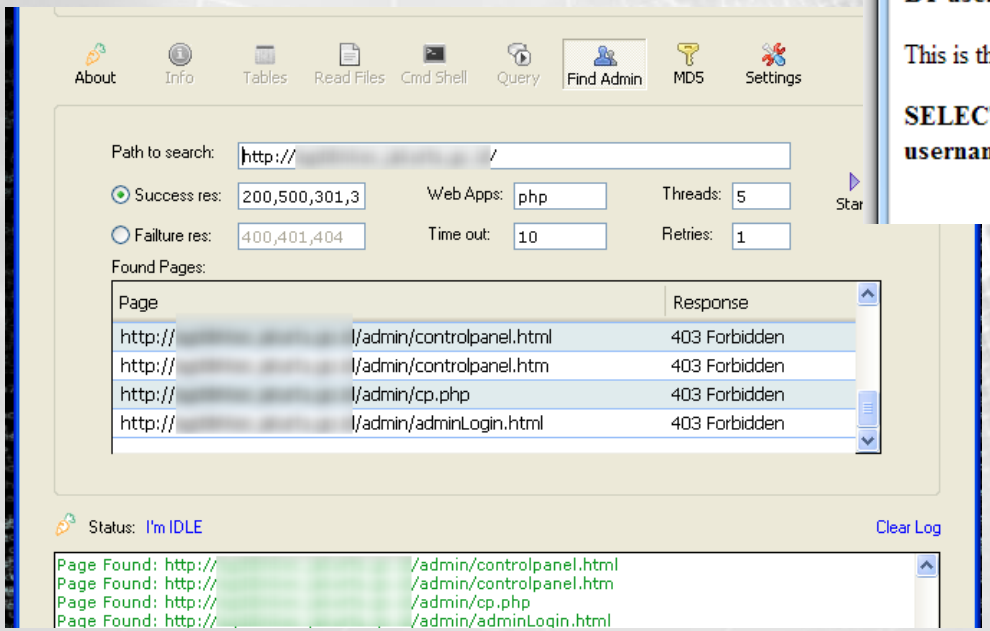
2. มีการ**ใช้งาน Captcha** เพื่อเป็นการยืนยันการใช้งานจากผู้ใช้งานจริง ในการใช้งานฟังก์ชั่นที่สำคัญ เช่น เปลี่ยนจากสถานะเลือกซื้อสินค้า เป็น จ่ายเงินชำระค่าสินค้า ระบบควรจะให้ผู้ใช้บริการ ยืนยันตัวตนอีกครั้ง เช่น ให้กรอกรหัสผ่านใหม่ พร้อมกับใช้ Captcha เป็นต้น

Sensitive Data Exposure

เทคนิคการโจมตีด้วยเทคนิค Sensitive Data Exposure

โจมตีโดยการส่งค่าทดสอบไปยังเว็บไซต์และดูผลลัพธ์จากการประมวลผล เช่น ข้อมูลแสดงข้อความ Error หรือสถานะของ HTTP Header เพื่อนำมาใช้ในการโจมตีเว็บไซต์ต่อไป





ผลของการโจมตีทำให้แฮกเกอร์ได้ข้อมูลสำคัญ เพื่อมาใช้ในการโจมตีเว็บไซต์ได้ต่อไป เช่น ทำให้ทราบว่าเว็บไซต์มีการเชื่อมต่อกับฐานข้อมูล MySQL และเว็บไซต์ดังกล่าวมีช่องโหว่ SQL Injection เป็นต้น

การป้องกันการโจมตีด้วยเทคนิค Sensitive data exposure

1. มีการออกแบบและควบคุมข้อความแจ้งเตือนหรือข้อความแสดงข้อผิดพลาด (**Notification or Error Message**) ไม่ให้แสดงข้อมูลที่เป็นประโยชน์ต่อผู้ประสงค์ร้าย เช่น ข้อมูลเวอร์ชันของโปรแกรมบริการต่างๆ

2. **ไม่ใช้งาน Autocomplete** ในแบบฟอร์มสำคัญ เช่น แบบฟอร์มสำหรับการลงทะเบียน การใช้งานระบบที่มีรหัสผ่าน หรือ แบบฟอร์มที่เกี่ยวข้องกับการชำระเงิน เป็นต้น

3. **ไม่ใช้ชื่อ URL ที่คาดเดาได้ง่าย**ซึ่งใช้ในการเข้าถึงหน้าเว็บสำหรับผู้ดูแลเครื่องบริการเว็บ เช่น admin.php หรือ login.php เป็นต้น

it's DEMO time!

Q & A time


ETDA
สพธอ
www.etda.or.th