

Term Project

By Saenyakorn Siangsanoh 6232035721

The purpose of this project is to educate you with the fundamentals of security in web applications (secure coding) by fixing the vulnerable application.

Table of Contents

- [How to start DVWA](#)
- [Instructions](#)
- [Vulnerabilities](#)
 - [1. Bruce Force](#)
 - [Exploitability](#)
 - [Weakness Prevalence](#)
 - [Weakness Detecability](#)
 - [Technical Impact](#)
 - [Fixes](#)
 - [2. Command Injection](#)
 - [Exploitability](#)
 - [Weakness Prevalence](#)
 - [Weakness Detecability](#)
 - [Technical Impact](#)
 - [Fixes](#)
 - [3. CSRF](#)
 - [Exploitability](#)
 - [Weakness Prevalence](#)
 - [Weakness Detecability](#)
 - [Technical Impact](#)
 - [Fixes](#)
 - [4. SQL Injection](#)
 - [Exploitability](#)
 - [Weakness Prevalence](#)
 - [Weakness Detecability](#)
 - [Technical Impact](#)
 - [Fixes](#)
 - [5. CSP \(Content Security Policy\) Bypass](#)
 - [Exploitability](#)
 - [Weakness Prevalence](#)
 - [Weakness Detecability](#)
 - [Technical Impact](#)
 - [Fixes](#)
 - [6. XSS \(Cross-site scripting\)](#)
 - [Exploitability](#)

- Weakness Prevalence
- Weakness Detecability
- Technical Impact
- Fixes
- 7. Weak Session IDs
 - Exploitability
 - Weakness Prevalence
 - Weakness Detecability
 - Technical Impact
 - Fixes
- 8. Javascript
 - Exploitability
 - Weakness Prevalence
 - Weakness Detecability
 - Technical Impact
 - Fixes

How to start DVWA

Just use `docker-compose.yml` from this folder. Then run

```
docker compose up -d
```

Instructions

For each vulnerability, suggest/show a fix for it. If it is a threat (cannot be fixed), please suggest a mitigation methodology. Please **highlight/explain** the concept clearly.


Vulnerabilities

1. Bruce Force

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Average (2)	Widespread (3)	Easy (3)	Moderate (2)

URL: <http://localhost:8080/vulnerabilities/brute/>

คือการสุ่มหา password ที่ถูกต้องจาก username ที่เราอาจจะรู้อยู่แล้วด้วยการลองสร้าง password ขึ้นมาแบบสุ่ม ๆ แล้วลอง login ดูว่าเข้าสู่ระบบได้หรือไม่



[Home](#)
[Instructions](#)
[Setup / Reset DB](#)
[Brute Force](#)
[Command Injection](#)
[CSRF](#)
[File Inclusion](#)
[File Upload](#)
[Insecure CAPTCHA](#)
[SQL Injection](#)
[SQL Injection \(Blind\)](#)
[Weak Session IDs](#)
[XSS \(DOM\)](#)
[XSS \(Reflected\)](#)
[XSS \(Stored\)](#)
[CSP Bypass](#)
[JavaScript](#)

[DVWA Security](#)
[PHP Info](#)
[About](#)

[Logout](#)


Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin



More Information

- [https://www.owasp.org/index.php/Testing_for_Brute_Force_\(OWASP-AT-004\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004))
- <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

Username: admin
Security Level: low
PHPIDS: disabled

[View Source](#)[View Help](#)

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

Exploitability

ซึ่งการทำ brute force ไม่ได้ทำได้ยากมากนัก แต่อาจจะต้องใช้ effort เล็กน้อยในการเขียน script เพื่อให้ hacker สามารถสุ่มรหัสและลอง login ได้อย่างต่อเนื่อง

Weakness Prevalence

เป็น vulnerability ที่มีความเป็นไปได้ที่จะเกิดขึ้นได้ทุกที่ โดยเฉพาะเว็บที่มีการ login โดยใช้ username และ password ที่ไม่มี CAPTCHA หรือ 2FA หรือไม่มี attempt limitation

Weakness Detecability

ตรวจสอบได้ง่าย โดยการลอง login ด้วยจำนวนครั้งมาก ๆ ว่ามีปฏิกิริยาอะไรหรือไม่ ถ้าไม่แปลว่าสามารถ brute force ได้

Technical Impact

ค่อนข้างร้ายแรง เพราะว่าหาก hacker สามารถเข้าสู่ระบบด้วย admin ได้ อาจจะทำให้ข้อมูลของระบบเสียหายได้

Fixes

- ใช้ CAPTCHA หรือ 2FA เพิ่มเติมในการ login เข้าสู่ระบบ
- จำกัดจำนวนครั้งในการ login เข้าสู่ระบบของ user คนหนึ่งต่อช่วงระยะเวลาหนึ่ง

2. Command Injection

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Average (2)	Uncommon (1)	Average (2)	Severe (3)

URL: <http://localhost:8080/vulnerabilities/exec/#>

คือการส่ง command ผ่าน input ใน website กลับไปให้ server ซึ่ง server จะเอา command นั้นไป execute ตรง ๆ โดยไม่มีการตรวจสอบ

The screenshot shows the DVWA interface at <http://localhost:8080/vulnerabilities/exec/#>. The main heading is "Vulnerability: Command Injection". Below it is a "Ping a device" section with a form "Enter an IP address:" containing the text "8.8.8.8 && ls" and a "Submit" button. The output of the command is displayed in red text: "PING 8.8.8.8 (8.8.8.8): 56 data bytes", followed by four lines of ping results, and then "4 packets transmitted, 4 packets received, 0% packet loss", "round-trip min/avg/max/stddev = 49.079/50.815/51.814/1.047 ms", and finally "help", "index.php", and "source". Below this is a "More Information" section with four links: <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>, <http://www.ss64.com/bash/>, <http://www.ss64.com/nt/>, and https://www.owasp.org/index.php/Command_Injection. The page also features a sidebar with navigation links like Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (highlighted), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security, PHP Info, About, and Logout. At the bottom, it shows "Username: admin", "Security Level: low", "PHPIDS: disabled", and "View Source" / "View Help" buttons. The footer text is "Damn Vulnerable Web Application (DVWA) v1.10 *Development*".

Exploitability

ความยากในการ exploit อยู่ในระดับปานกลาง เนื่องจาก hacker จำเป็นที่จะต้องรู้ linux command เสียก่อน

Weakness Prevalence

เป็น vulnerability ที่มีโอกาสเจอได้น้อย เพราะว่าปัจจุบันมี library ในการ execute command ที่มีการตรวจสอบ input อยู่เยอะมากแล้ว แต่อาจจะเจอได้กับ website ที่มีอายุเยอะ หรือ website ที่ไม่ได้ใช้ library ที่มีการตรวจสอบ input

Weakness Detecability

อาจจะตรวจสอบไม่ได้จาก client-side source code เพราะ command จะส่งไป execute ที่ server แต่ก็สามารถตรวจสอบได้ว่ามี vulnerability หรือไม่ โดยการลองส่ง basic linux command ไปให้ server แล้วดู response

Technical Impact

ร้ายแรงมาก เพราะเสมือนว่า hacker เข้าถึง server ของระบบได้แล้ว ที่เหลือ hacker อยากจะทำอะไรก็สามารถทำได้เลย เช่นแอบเอา SSH Public key ของตัวเองเอาไปใส่ไว้ใน server เพื่อที่ hacker จะได้ login เข้า server ตรง ๆ ได้ หรืออาจจะแอบติดตั้ง malware ไว้ใน server ก็ได้

Fixes

- หยุดการใช้ `exec` / `eval` ตรง ๆ แล้วเปลี่ยนไปใช้ library ที่มีคนทำเรื่องการตรวจสอบ input ให้แล้ว
- validate input ทุกครั้ง ก่อนที่จะนำไปใช้งาน โดยการใช้ whitelist หรือ blacklist ก็ได้ แต่เพื่อความปลอดภัยสูงสุด จะแนะนำให้ใช้ whitelist แทน เช่น Regex

3. CSRF

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Easy (3)	Uncommon (2)	Easy (3)	Moderate (2)

URL: http://localhost:8080/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change#

http://localhost:8080/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change#

ในกรณีของ DVWA จะสังเกตได้ว่า URL หลังจากลองเปลี่ยน password คือ

http://localhost:8080/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change#

ดังนั้นเราสามารถสร้าง URL ที่มี parameter ตามที่เราต้องการแล้วส่งไปให้เพื่อนกด จากนั้นเราก็จะรู้ password ของเพื่อนคนนั้นโดยปริยาย

Exploitability

การ exploit vulnerability นี้ค่อนข้างง่าย เนื่องจากการแก้ URL query parameters เท่านั้น

Weakness Prevalence

ไม่ค่อยพบในปัจจุบัน เพราะการใช้ OAuth มีความนิยมมากขึ้น เนื่องจากง่ายและไม่ต้อง implement ระบบ login / forgot password ของตัวเอง

Weakness Detecability

ตรวจสอบได้ง่าย เนื่องจากดูจาก URL ที่ถูกส่งมาก็พอจะทราบได้ว่าต้องเป็นการโจมตีบางอย่าง

Technical Impact

ค่อนข้างร้ายแรง หากคนที่โดยการโจมตีนี้เป็น admin ดังนั้น hacker ก็จะสามารถใช้งานระบบได้อย่างเต็มที่

Fixes

- หยุดการรับ input จาก query parameters
- ให้ผู้ใช้กรอก current password เพื่อเป็นการ confirm ว่าต้องการเปลี่ยน password จริง ๆ
- ใช้ CSRF Token ที่จะตรวจสอบว่า request ที่กำลังจะเกิดขึ้นเป็น request เกิดที่ต่อกันมาจริง ๆ หรือไม่

4. SQL Injection

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Moderate (2)	widespread (3)	Average (2)	Server (3)

URL: <http://localhost:8080/vulnerabilities/sqli/?id=%27+OR+1%3D1%3B+---+&Submit=Submit#>

เป็นการที่ hacker สามารถใส่ SQL command ผ่าน input ของ web application เพื่อให้เกิดผลบางอย่างเช่น เพิ่มข้อมูลที่ไม่ควรเพิ่ม, ดูข้อมูลที่ไม่ควรดู, หรือลบข้อมูลที่ไม่ควรลบ

Exploitability

การ exploit vulnerability นี้อยู่ในระดับปานกลาง เนื่องจาก hacker ตอนจินตนาการด้วยว่า developer implement input นี้อย่างไร และ hacker จะต้องมีความรู้ SQL พื้นฐานประมาณหนึ่ง

Weakness Prevalence

ค่อนข้างพบได้แพร่หลายในเว็บที่มีอายุ เนื่องจาก server program ส่วนใหญ่จะ execute SQL command โดยตรง โดยไม่มีการ validate input ที่ผู้ใช้กรอกเข้ามา

Weakness Detecability

ค่อนข้างตรวจสอบได้ยาก เนื่องจาก code ส่วนที่มีปัญหามักจะอยู่ฝั่ง server ซึ่ง hacker จะต้องลอง exploit ดูเท่านั้น

Technical Impact

ค่อนข้างร้ายแรง เนื่องจากว่าหากเราไม่ได้ทำ Database backup ไว้ เราก็อาจจะโดน hacker drop database ทั้งจนทำให้ข้อมูลหายทั้งหมดได้

Fixes

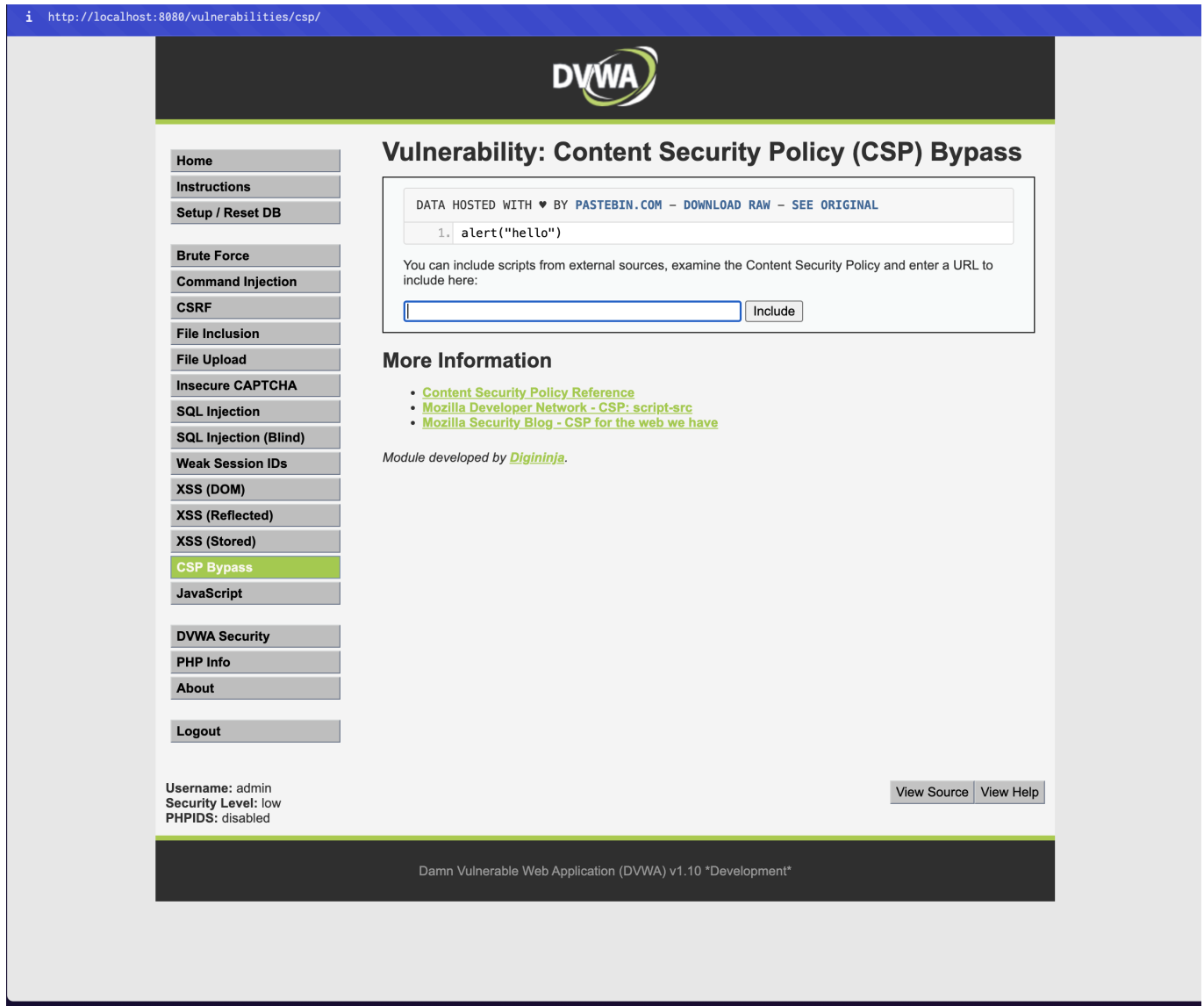
- ใช้ ORM (Object-relational mapping) แทนการเขียน SQL command ด้วยตัวเอง เนื่องจากเป็น library ที่มีการ develop อย่างต่อเนื่องโดยคนทั่วโลก
- ทำการ validate input อย่างละเอียดก่อนใช้งานเสมอ

5. CSP (Content Security Policy) Bypass

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Moderate (1)	Uncommon (1)	Average (2)	Minor (1)

URL: <http://localhost:8080/vulnerabilities/csp/>

เป็นการที่ developer เปิดช่องให้ hacker สามารถโหลด script จากที่อื่นมาใส่ใน website ได้



Exploitability

ค่อนข้างยากประมารหนึ่ง เนื่องจาก hacker จะต้องใช้ effort สำหรับการเตรียม script เพื่อ load เข้า website เป้าหมาย

Weakness Prevalence

ไม่ค่อยพบ เพราะว่า website ส่วนใหญ่ไม่มีความจำเป็นต้อง load script จากภายนอกที่ user เป็นคนกรอกเข้ามา

Weakness Detecability

ตรวจพอค่อนข้างปานกลาง hacker ต้องใช้เวลาในการหาจุดที่ developer เปิดช่องให้ load script จากข้างนอกได้

Technical Impact

ไม่ค่อยร้ายแรง เนื่องจาก script นั้นจะอยู่แค่ใน browser ของ hacker เท่านั้น ยกเว้นแต่ developer เก็บ external script กลับไปที่ server ด้วย

Fixes

- ปิดช่องให้ user สามารถ load external script ได้

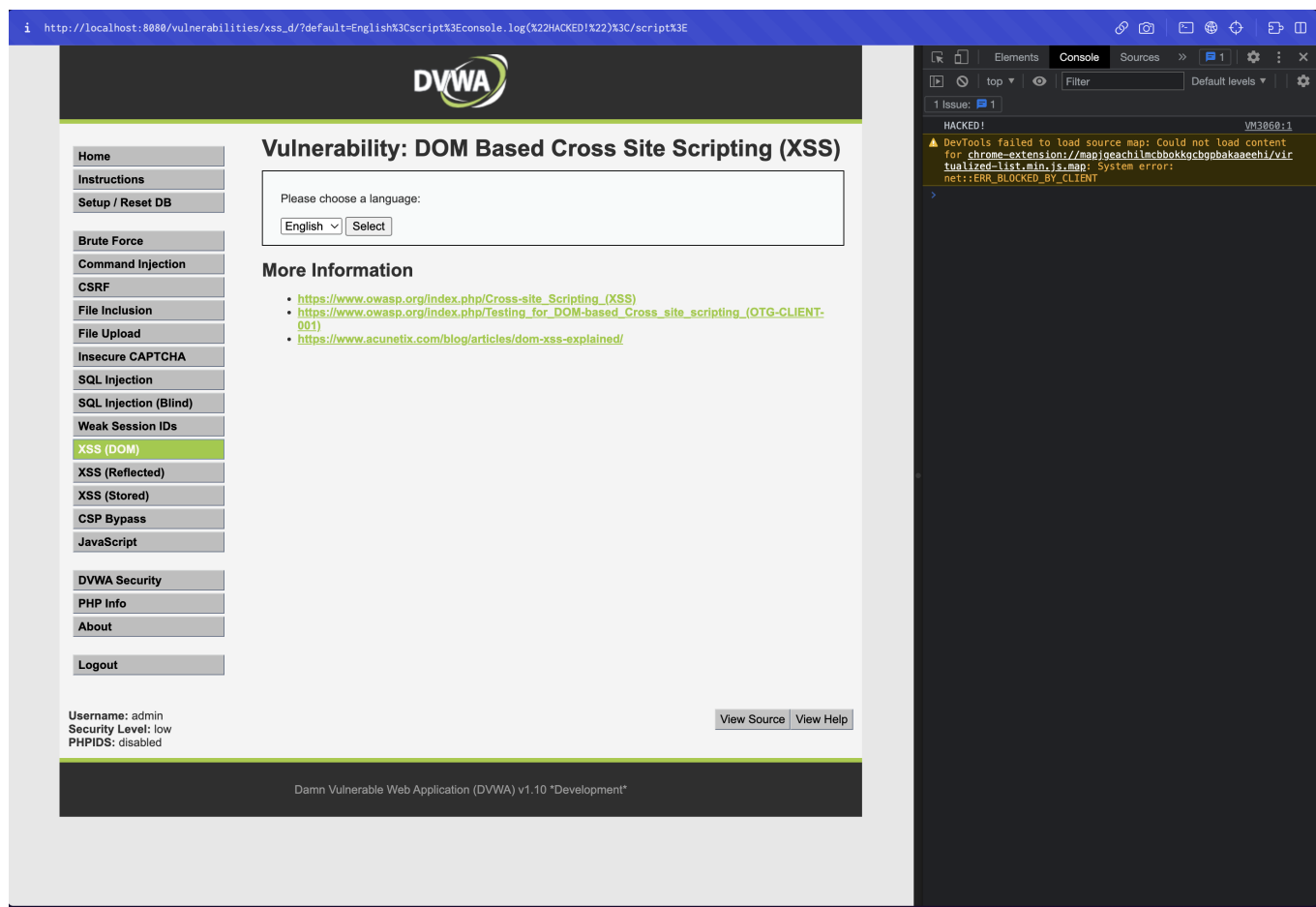
6. XSS (Cross-site scripting)

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Moderate (2)	Widespread (3)	Average (2)	Moderate (2)

URL: [http://localhost:8080/vulnerabilities/xss_d/?default=English%3Cscript%3Ealert\(%27hacked%27\)%3C/script%3E](http://localhost:8080/vulnerabilities/xss_d/?default=English%3Cscript%3Ealert(%27hacked%27)%3C/script%3E)

[default=English%3Cscript%3Ealert\(%27hacked%27\)%3C/script%3E](http://localhost:8080/vulnerabilities/xss_d/?default=English%3Cscript%3Ealert(%27hacked%27)%3C/script%3E)

เป็นการที่ input ของ hacker มี javascript แอบไว้อยู่ ซึ่งจะทำให้ browser execute script นั้นออกมา โดย script นั้นอาจจะทำอะไรบางอย่างเพื่อให้ hacker ได้ผลประโยชน์ เช่น แอบดึงข้อมูลบางอย่างจาก user แล้วส่งข้อมูลกลับไปให้ hacker



Exploitability

การ exploit vulnerability นี้อยู่ในระดับปานกลาง เนื่องจาก hacker จะต้องใช้จินตนาการว่า จะใส่ script แบบไหนดี แล้วจะให้ script นั้นทำอะไร นอกจากนี้ hacker ยังจะต้องรู้ด้วยว่า script ของตัวเองจะทำงานอย่างไรบน browser ที่เป้าหมายใช้

Weakness Prevalence

ค่อนข้างพบได้แพร่หลายในเว็บที่มีอายุ เนื่องจาก developer มักจะลืม escape ตัวอักษรพิเศษที่มีความหมายใน HTML อยู่ เช่น <, >, &, " และ ' ซึ่งเป็นการเปิดช่องให้ hacker โจมตีด้วย XSS ได้

Weakness Detecability

ตรวจพบได้ยาก เพราะต้องตรวจสอบ source code ดูเท่านั้น หรือไม่ก็ต้องลอง exploit ดู

Technical Impact

ความร้ายแรงค่อนข้างน้อย เพราะไม่ได้เป็นการโจมตีระบบโดยตรง แต่เป็นการเล่นงาน end user มากกว่า

Fixes

- ให้ทำการ escape string ทุกครั้งเมื่อรับ input มา
- ใช้ Framework / Library ที่มีความทันสมัย และมีการแก้ไข security issue อยู่เสมอ

7. Weak Session IDs

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Easy (3)	Uncommon (3)	Easy (3)	Moderate (2)

URL: http://localhost:8080/vulnerabilities/weak_id/

เป็นการที่ session id เดาได้ง่ายเกินไป ซึ่ง hacker อาจจะนำ session id นั้นไปปลอมตัว สวมรอยเป็นคนอื่น เพื่อใช้งานระบบ

Exploitability

ค่อนข้างง่าย เนื่องจากแค่หากเราเข้าไปดู cookie แล้วพบว่า session id สามารถเดาได้ ก็สามารถลอง set session id เป็นอันอื่นดูได้เลย

Weakness Prevalence

ค่อนข้างพบได้ยาก เนื่องจาก developer มักจะใช้ session id ที่ random และยาวมาก แต่ถ้าเป็นการใช้ session id ที่สั้น หรือเป็นตัวเลข ก็อาจจะพบได้ง่าย

Weakness Detecability

ตรวจพบได้ง่าย เพียงแต่ดู cookie จาก browser ก็จะได้รู้ได้เลย

Technical Impact

ความร้ายแรงอยู่ในระดับปานกลาง เพราะไม่ได้เป็นการโจมตีระบบโดยตรง แต่เป็นการเล่นงาน end user มากกว่า

Fixes


- ใช้ session id ที่ random และยาวมาก จนไม่สามารถเดาได้

8. Javascript

Exploitability	Weakness Prevalence	Weakness Detecability	Technical Impact
Difficult (1)	Common (3)	Difficult (3)	Minor (1)

URL: <http://localhost:8080/vulnerabilities/javascript/>

เป็นการใช้ javascript function ที่ developer เป็นคนเขียนทิ้งไว้เพื่อ bypass บางอย่าง หรือใช้ javascript ในการ execute โค้ดที่ไม่ได้รับอนุญาต



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

Vulnerability: JavaScript Attacks

Submit the word "success" to win.

Well done!

Phrase

More Information

- <https://www.w3schools.com/js/>
- <https://www.youtube.com/watch?v=cs7EQdWO5o0&index=17&list=WL>
- <https://ponyfoo.com/articles/es6-proxies-in-depth>

Module developed by [Digininja](#).

Username: admin
Security Level: low
PHPIDS: disabled

View Source

View Help

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

Exploitability

ค่อนข้างยาก เนื่องจาก hacker จะต้องไปดูและทำความเข้าใจ source code ของ website ก่อน

Weakness Prevalence

พบได้ทั่วไป เนื่องจากไม่มีวิธีป้องกัน hacker จากการใช้ function ที่ developer เขียนไว้บน website

Weakness Detecability

ค่อนข้างยาก เนื่องจาก hacker จะต้องไปดูและทำความเข้าใจ source code ของ website ก่อน

Technical Impact

มีผลน้อย เนื่องจากด้วยวิธีนี้ hacker จะโจมตีได้แบบจำกัด เว้นเสียแต่ developer จะเขียน code ไม่ได้

Fixes

- ไม่มีวิธีป้องกันได้ 100% เนื่องจากผู้ใช้ทุกคนสามารถอ่าน source code ของ website ได้
- พยายามใช้ parser ที่มีความปลอดภัยสูง และทำให้ complied code อ่านได้ยากขึ้น ซึ่งจะทำให้ hacker ทำความเข้าใจ code ได้ยากขึ้นด้วย