**LAB 3**

**Fuzzing - Reverse Engineering - Cryptography**

|  |
| --- |
| Họ tên và MSSV: Phạm Thị Kim Nguyên, CL21V7X002  Nhóm học phần: ET21V7X1 |

* *Các sinh viên bị phát hiện sao chép bài của nhau sẽ nhận 0đ cho TẤT CẢ các bài thực hành của môn này.*
* *Bài nộp phải ở dạng PDF, hình minh họa phải rõ ràng chi tiết.* Hình minh hoạ chỉ cần chụp ở nội dung thực hiện, không chụp toàn màn hình.

Lưu ý: sử dụng tài khoản là mã số sinh viên (đã tạo ở Câu 1.4 - Lab01) để thực hiện tất cả các câu trong bài thực hành. Tất cả các câu trong bài thực hành này được thực hiện trên Kali Linux 2022.2

**Câu 1: Thực hiện kỹ thuật Reverse Engineering với công cụ IDA Free**

1.1. Cài đặt [IDA Free](https://hex-rays.com/ida-free/#download) vào Kali Linux. Tải file cài đặt (ví dụ: idafree83\_linux.run):

$sudo chmod +x ./Downloads/idafree83\_linux.run

$./Downloads/idafree83\_linux.run

#Thực hiện các bước theo yêu cầu. IDA Free sẽ được cài đặt ở thư mục cá nhân của người dùng.

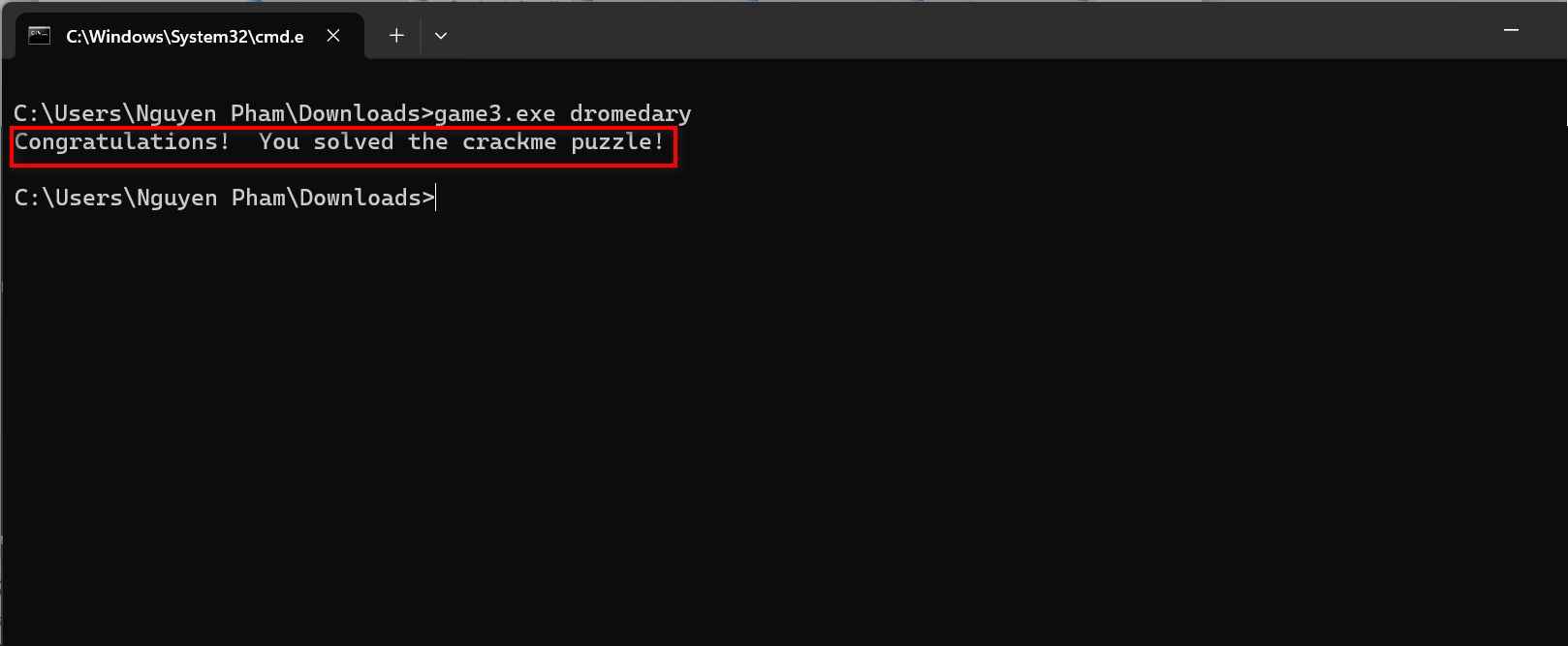
1.2. Trên máy Windows, tải file Lab03.zip. Giải nén sẽ được thư mục crackme chứa 02 file crackme.exe và msvcr100d.dll. Chạy file crackme.exe ở môi trường CMD với vài giá trị đầu vào khác nhau và xem kết quả.

1.3. Di chuyển file crackme.exe vào máy ảo Kali Linux (sử dụng chức năng Shared Folders hoặc Drag and Drop). Chạy IDA Free và mở file crackme.exe trên môi trường IDA Free.

1.4. Trên thanh menu, sử dụng chức năng “Search, Text” tìm kiếm với từ khóa “usage”. Trên cửa sổ “IDA - View A” chọn “Fit window”. Chọn “Text view” hoặc “Graph view” để xem mã Assembly của chương trình.

1.5 Trên thanh menu, sử dụng chức năng “View, Sub view, Generate pseudo code” (F5) để sinh mã giả của crackme.exe. Trên cửa sổ “Pseudocode - A”, sử dụng chức năng “Synchronize with” để đồng bộ với “IDA - View A” và “Hex - View 1”.

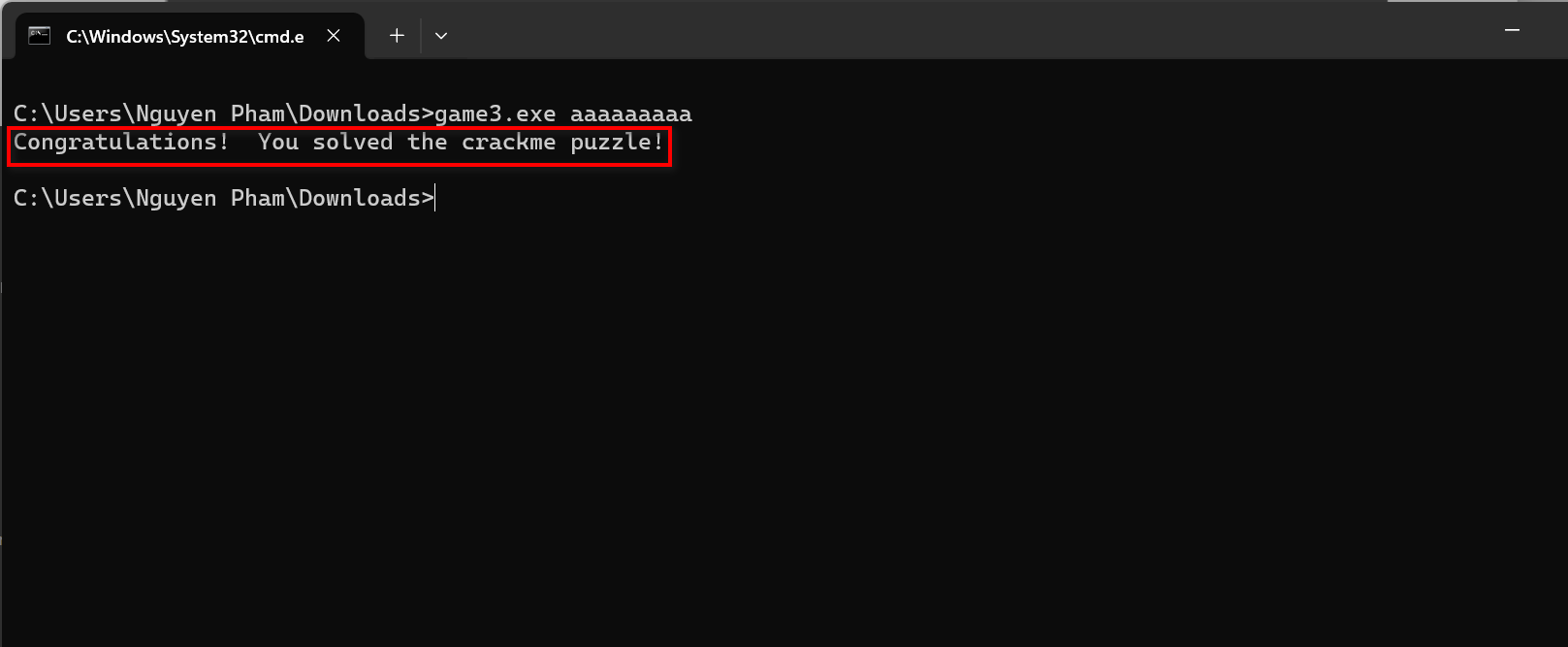
1.6 Phân tích mã giả của crackme.exe để có thể chạy chương trình in ra thông báo "Congratulations! You solved the crackme puzzle!”. Chạy file crackme.exe ở môi trường CMD của máy Windows in ra thông báo trên.



1.7. Cài đặt công cụ Bless hex editor vào Kali Linux:

$ sudo apt update && sudo apt install bless -y

1.8. Sử dụng công cụ Bless hex editor để thay đổi giá trị password cần nhập cho crackme.exe thành “aaaaaaaaa”. Di chuyển file crackme.exe từ Kali Linux qua máy Windows. Chạy file crackme.exe ở môi trường CMD của máy Windows in ra thông báo trên "Congratulations! You solved the crackme puzzle!”



**Câu 2: Thực hiện kỹ thuật Fuzzing với công cụ Spike**

* Cài đặt công cụ [nmap (Zenmap)](https://nmap.org/download.html#windows) trên máy Windows.
* Tắt tường lửa máy Windows. Thực hiện dịch vụ Netcat ở môi trường CMD của máy Windows:

>ncat -vklp 9000

* Thay đổi cấu hình mạng của máy ảo Kali Linux sao cho có thể giao tiếp mạng tới máy Windows. Tạo file simple1.spk với nội dung bên dưới:

$nano simple1.spk

# Nội dung file

s\_string("Hello, world!");

* Thực hiện lệnh sau để fuzzing dịch vụ Netcat. Quan sát kết quả dịch vụ Netcat nhận được.

$ generic\_send\_tcp <IP máy Windows> 9000 simple1.spk 0 0

* Tạo file simple2.spk với nội dung bên dưới:

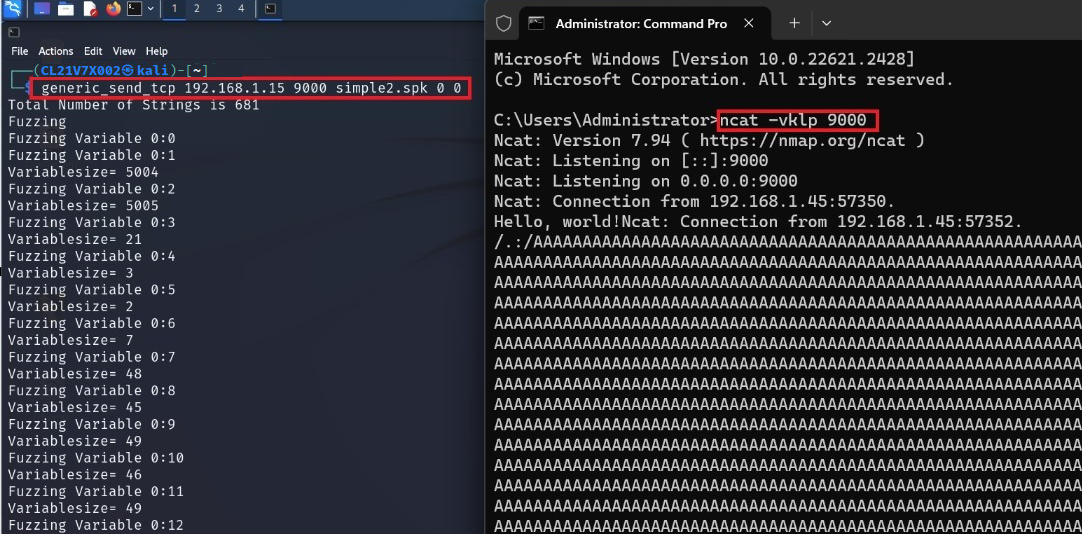
$nano simple2.spk

# Nội dung file

s\_string\_variable("Hello, world!");

* Thực hiện lệnh sau để fuzzing dịch vụ Netcat. Quan sát kết quả dịch vụ Netcat nhận được.

$ generic\_send\_tcp <IP máy Windows> 9000 simple2.spk 0 0



* Tải và giải nén file Lab03.zip được thư mục vulnserver. Thực thi file vulnserver.exe ở môi trường CMD của máy Windows.
* Trên máy Kali Linux, nối kết vulnserver trên máy Windows:

$nc <IP máy Windows> 9999

HELP

EXIT

* Tạo file trun.spk với nội dung bên dưới:

$nano trun.spk

# Nội dung file

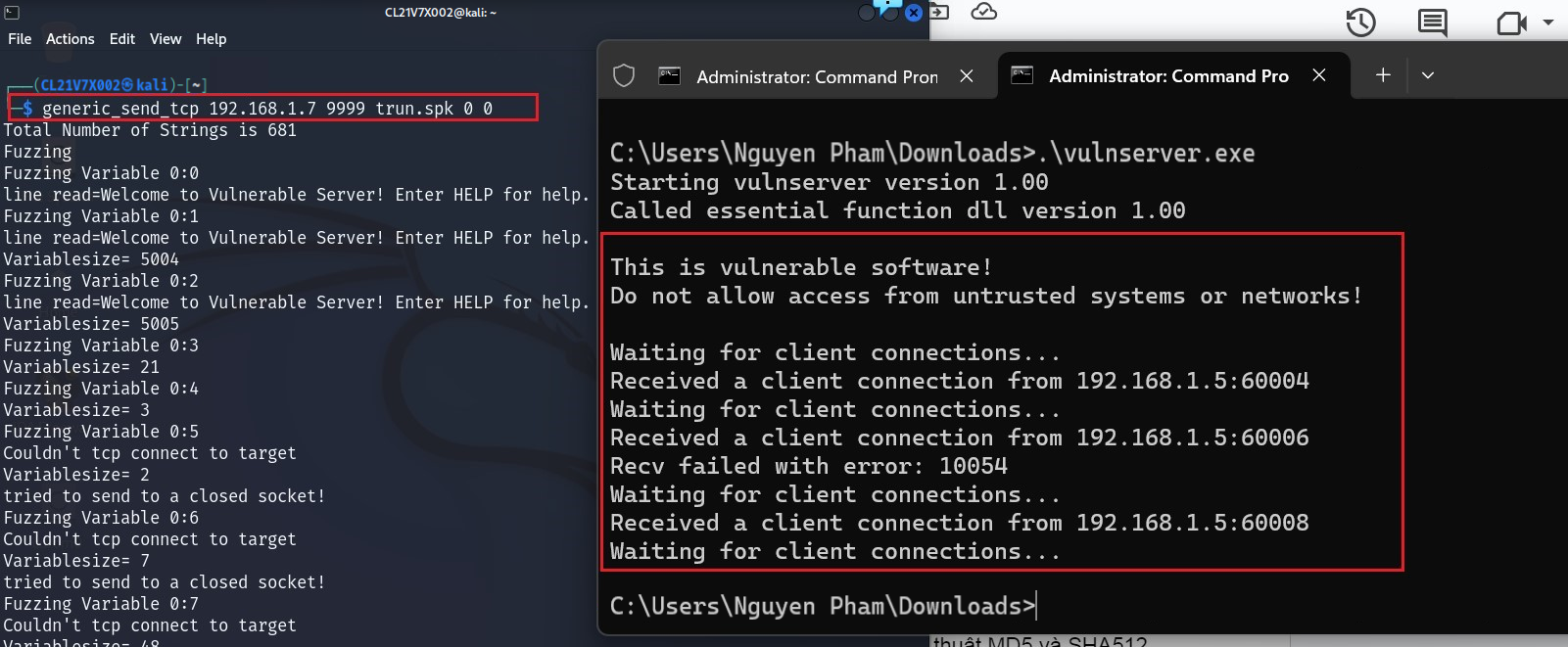
s\_readline();

s\_string("TRUN ");

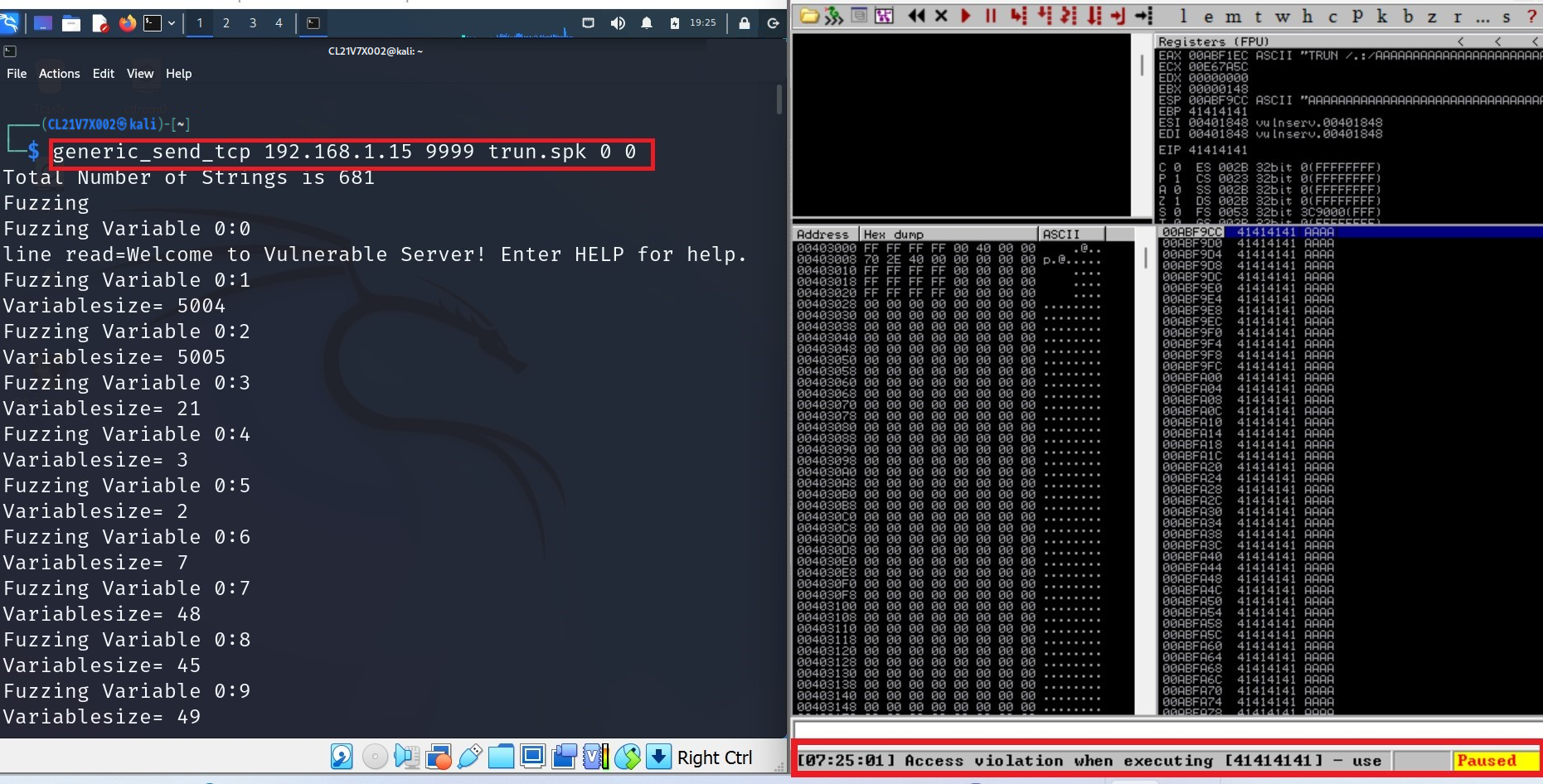
s\_string\_variable("COMMAND");

* Thực hiện lệnh sau để fuzzing dịch vụ vulnserver. Quan sát kết quả sẽ thấy vulnserver dừng hoạt động sau một thời gian ngắn:

$ generic\_send\_tcp <IP máy Windows> 9999 trun.spk 0 0



* Cài đặt công cụ [Immunity Debugger](https://debugger.immunityinc.com/ID_register.py) vào máy Windows. Sử dụng công cụ để mở file và thực thi file vulnserver.exe. Tiếp tục fuzzing dịch vụ vulnserver. Quan sát giao diện để thấy thanh ghi EIP chứa giá trị “41414141” (có lỗi buffer overflow).



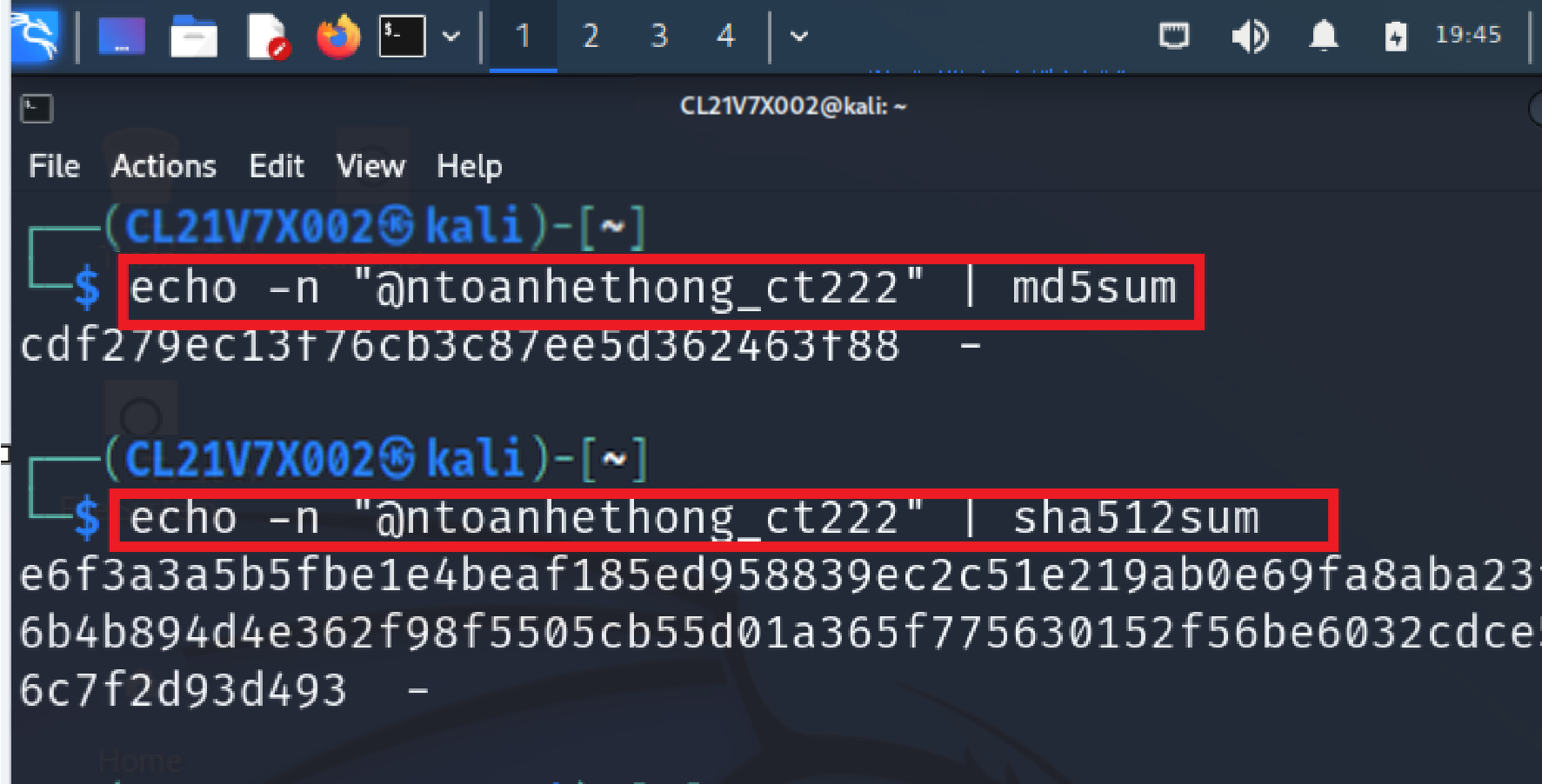
* Khai thác lỗi buffer overflow trên vulnserver.exe theo [hướng dẫn](https://smarinovic.github.io/posts/Vulnserver-KSTET-procmon/). (Không bắt buộc thực hiện)

**Câu 3: Giải thuật băm và tấn công mật khẩu**

3.1. Tìm giá trị băm của chuỗi "@ntoanhethong\_ct222" sử dụng giải thuật MD5 và SHA512 sử dụng công cụ md5sum và sha512sum:

$ echo -n "@ntoanhethong\_ct222" | md5sum

$ echo -n "@ntoanhethong\_ct222" | sha512sum



Sử dụng một trang web online cho phép thực hiện giải thuật băm, ví dụ :<https://www.pelock.com/products/hash-calculator> để kiểm tra kết quả.

3.2. Sử dụng công cụ john và hashcat để dịch ngược giá trị băm "b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86"

# Xác định giải thuật băm

$ hashid "b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86"

$ echo -n "b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86" > hash

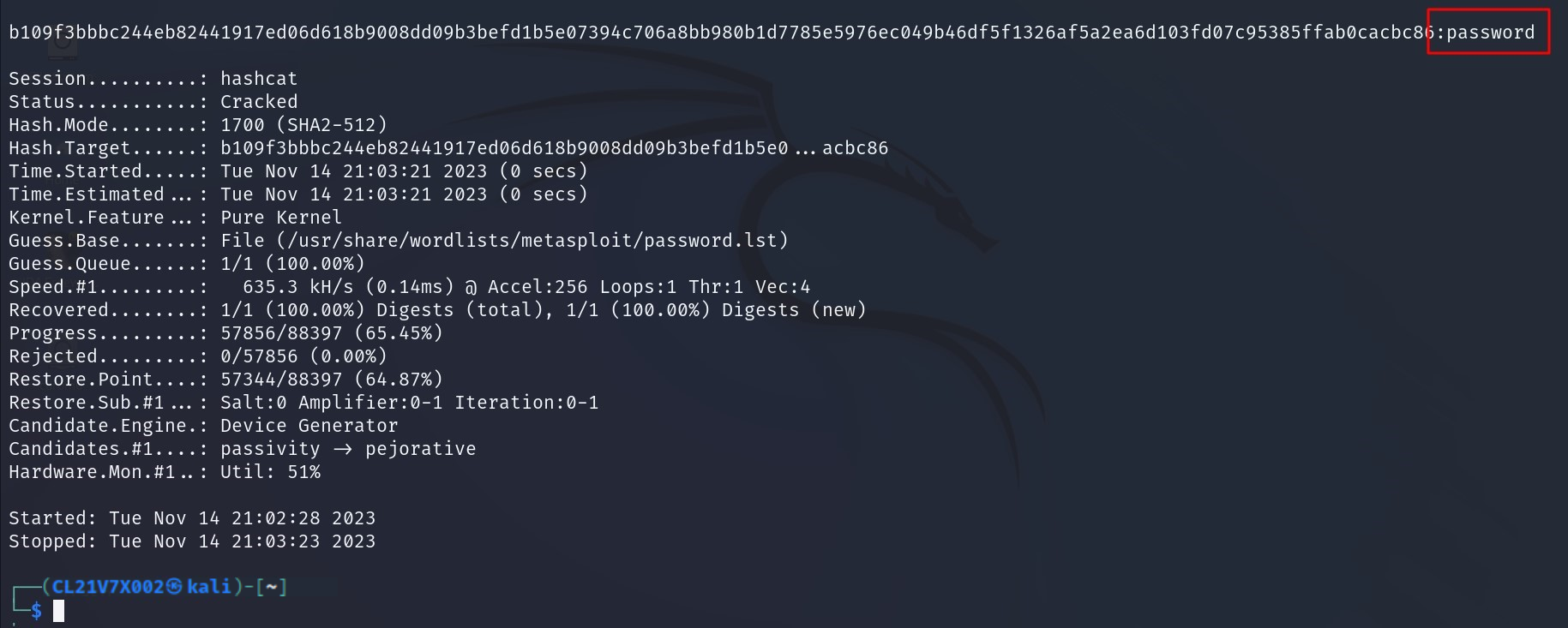
# Sử dụng John the Ripper

$john --format=raw-sha512 \

--wordlist=/usr/share/wordlists/metasploit/password.lst hash

# Sử dụng hashcat

$hashcat -a 0 -m 1700 ./hash /usr/share/wordlists/metasploit/password.lst



Sử dụng một trang web cho phép dịch ngược giá trị băm, ví dụ: <https://crackstation.net/>, để kiểm tra kết quả.

3.3. Tạo một file zip “abc.zip” với mật khẩu mở file là "abc123". Sử dụng john và hashcat để dò mật khẩu của file:

* Trích xuất giá băm của mật khẩu

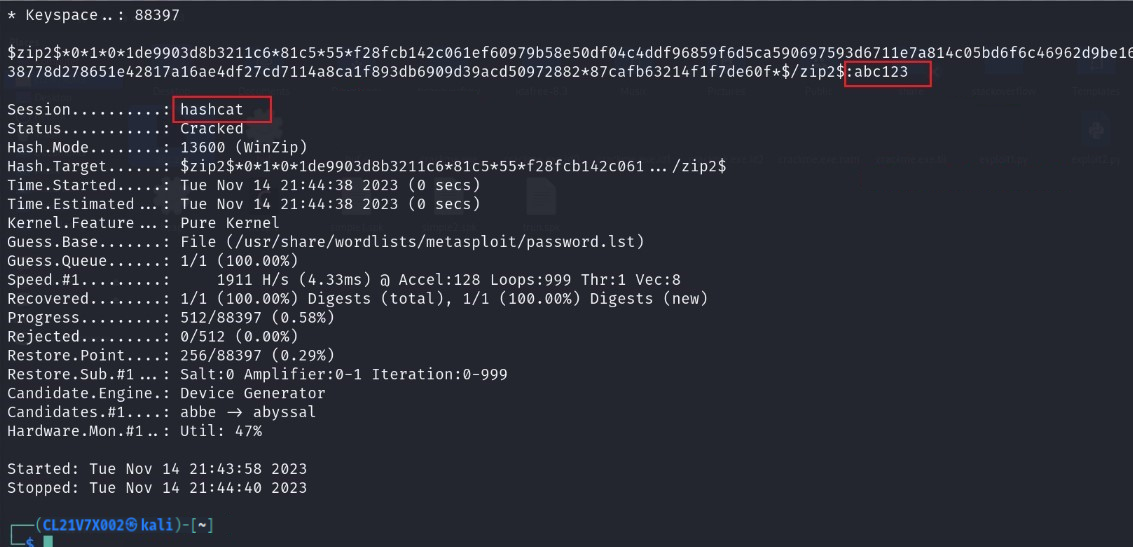
$ zip2john abc.zip > hash

* Dò mật khẩu:

$ john --wordlist=/usr/share/wordlists/metasploit/password.lst hash

# Chỉnh sửa file hash cho phù hợp định dạng của hashcat

# hashcat -a 0 -m 13600 hash /usr/share/wordlists/metasploit/password.lst



3.4. Sử dụng công cụ john dò mật khẩu người dùng:

* Tạo người dùng mới “newuser” với mật khẩu “qwerty”

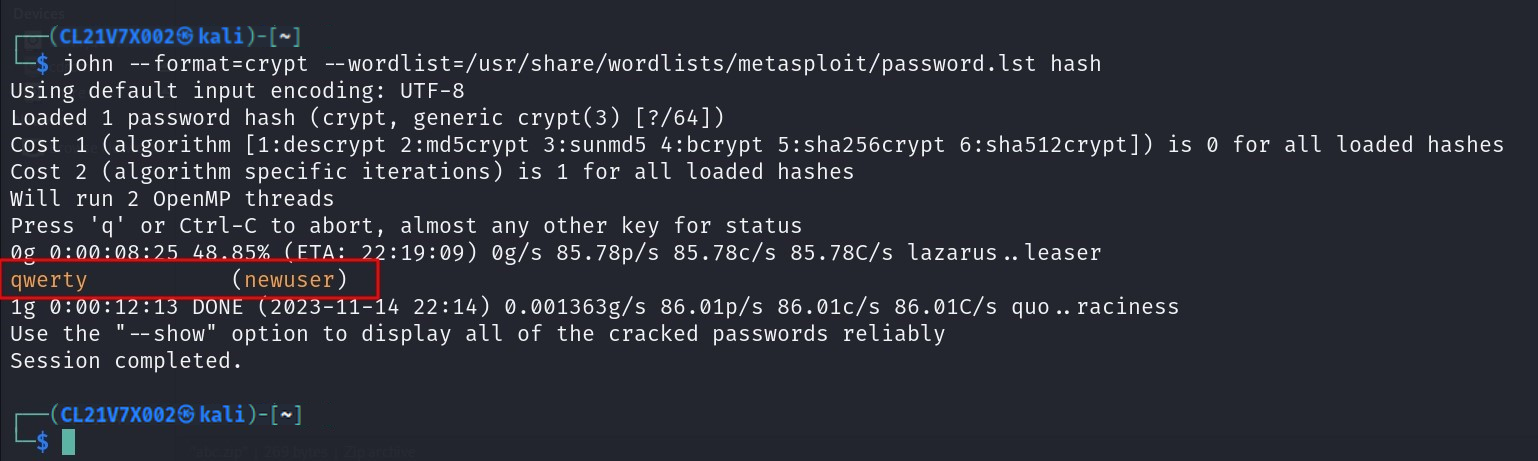
$ sudo adduser newuser

* Trích xuất giá trị băm của mật khẩu người dùng

$ sudo cat /etc/shadow | grep "newuser" > hash

* Dò mật khẩu:

$ john --format=crypt --wordlist=/usr/share/wordlists/metasploit/password.lst hash



**Câu 4: Tìm hiểu giải thuật AES**

* Cài và tạo môi trường ảo cho python:

$ sudo apt update && sudo apt install python3-venv -y

$ python -m venv lab03\_04

* Cài đặt module pycryptodome:

$ ./lab03\_04/bin/pip install pycryptodome

* Tải và giải nén file Lab03.zip được thư mục aes.
* Tạo tập tin aes\_ecb.py để mã hóa tập tin tux.bmp theo giải thuật AES-ECB. Thực thi aes\_ecb.py:

$./lab03\_04/bin/python aes\_ecb.py

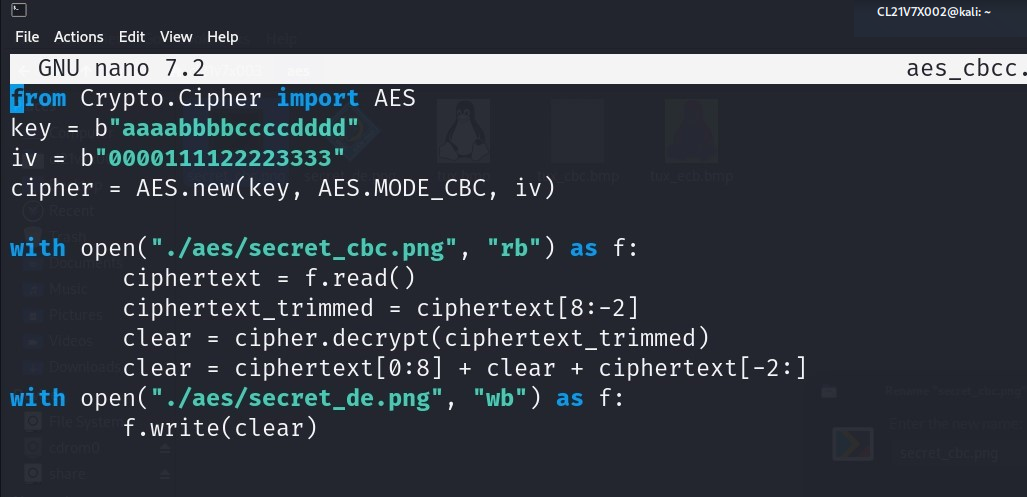
|  |
| --- |
| from Crypto.Cipher import AES key = b"aaaabbbbccccdddd" cipher = AES.new(key, AES.MODE\_ECB)  with open("./aes/tux.bmp", "rb") as f:  clear = f.read()  clear\_trimmed = clear[54:-12]  ciphertext = cipher.encrypt(clear\_trimmed)  ciphertext = clear[0:54] + ciphertext + clear[-12:] with open("./aes/tux\_ecb.bmp", "wb") as f:  f.write(ciphertext) |

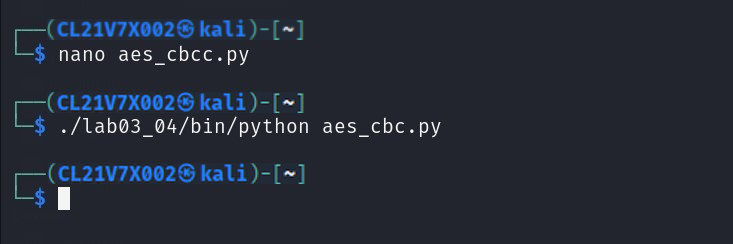
* Tạo tập tin aes\_cbc.py để mã hóa tập tin tux.bmp theo giải thuật AES-CBC. Thực thi aes\_cbc.py:

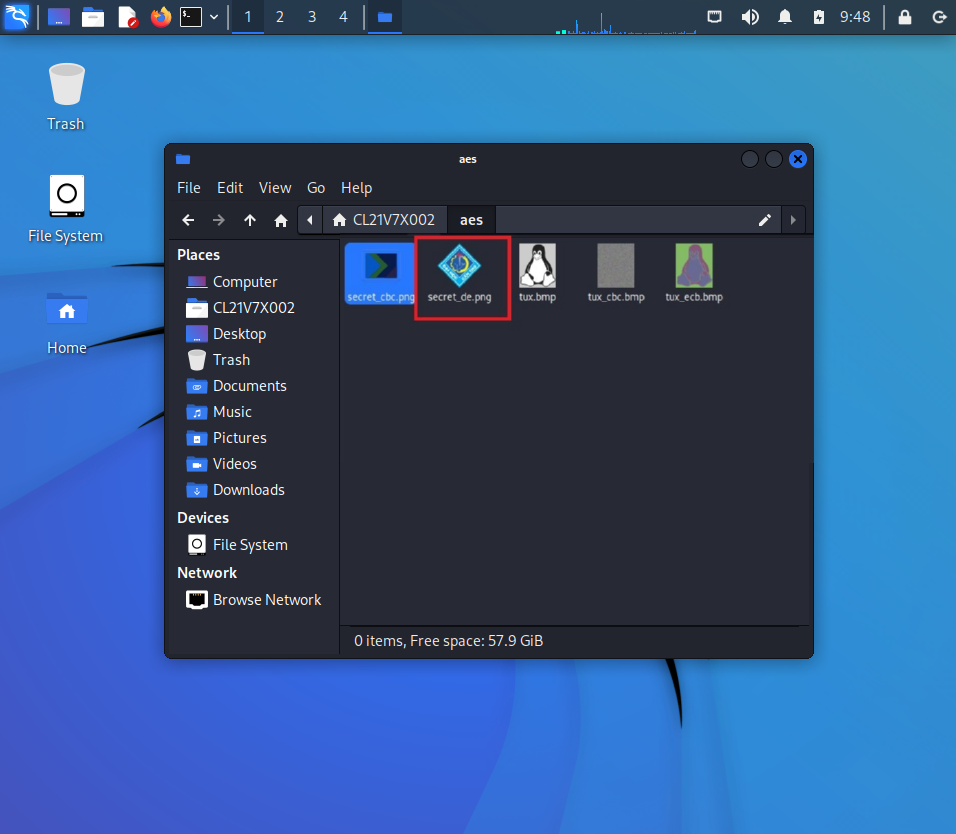
$./lab03\_04/bin/python aes\_cbc.py

|  |
| --- |
| from Crypto.Cipher import AES key = b"aaaabbbbccccdddd" iv = b"0000111122223333" cipher = AES.new(key, AES.MODE\_CBC, iv)  with open("./aes/tux.bmp", "rb") as f:  clear = f.read()  clear\_trimmed = clear[54:-12]  ciphertext = cipher.encrypt(clear\_trimmed)  ciphertext = clear[0:54] + ciphertext + clear[-12:] with open("./aes/tux\_cbc.bmp", "wb") as f:  f.write(ciphertext) |

* Viết code giải mã tập tin secret.png đã được mã hóa theo giải thuật AES CBC. Biết key mã hóa là "aaaabbbbccccdddd" và IV là "0000111122223333". Lưu ý: phần header của tập tin không được mã hóa.







**Câu 5: Tìm hiểu giải thuật Diffie-Hellman và RSA**

5.1. Tìm hiểu giải thuật Diffie-Hellman trao đổi khóa bí mật qua [video](https://www.youtube.com/watch?v=M-0qt6tdHzk). Tìm khóa bí mật sk, với g = 23, p=5, a=4 (khóa bí mật của Alice), b=10 (khóa bí mật của Bob).

Khóa bí mật sk= 1

* **Alice = 4^4 mod 5 = 1 Bob = 1^10 mod 5 = 1**
* **kết quả cho thấy Khóa bí mật sk = 1**

Có thể tìm hiểu thêm về giải thuật Diffie-Hellman qua [video](https://www.youtube.com/watch?v=NmM9HA2MQGI) sau.

5.2. Truy cập đến địa chỉ <http://people.cs.pitt.edu/~kirk/cs1501/notes/rsademo/index.html> để tạo khóa, mã hóa, giải mã sử dụng giải thuật RSA. Sau đó mô tả ngắn gọn cách RSA tạo khóa, mã hóa và giải mã.

1. Tạo Khóa:

a. Chọn Hai Số Nguyên Nguyên Tố Lớn:

- Chọn hai số nguyên tố lớn p và q.

b. Tính N và Φ(N):

- Tính N = p \* q.

- Tính Φ(N) = (p - 1) \* (q - 1).

c. Chọn Khóa Công Khai (e) và Tính Khóa Bí Mật (d):

- Chọn e sao cho 1 < e < Φ(N) và e là số nguyên tố cùng nhau với Φ(N).

- Tính d sao cho (d \* e) % Φ(N) = 1 (mod Φ(N)).

d. Cặp Khóa:

- Khóa công khai là (N, e).

- Khóa bí mật là (N, d).

2. Mã Hóa:

a. Chuẩn bị Dữ Liệu:

- Chia dữ liệu thành các khối nhỏ.

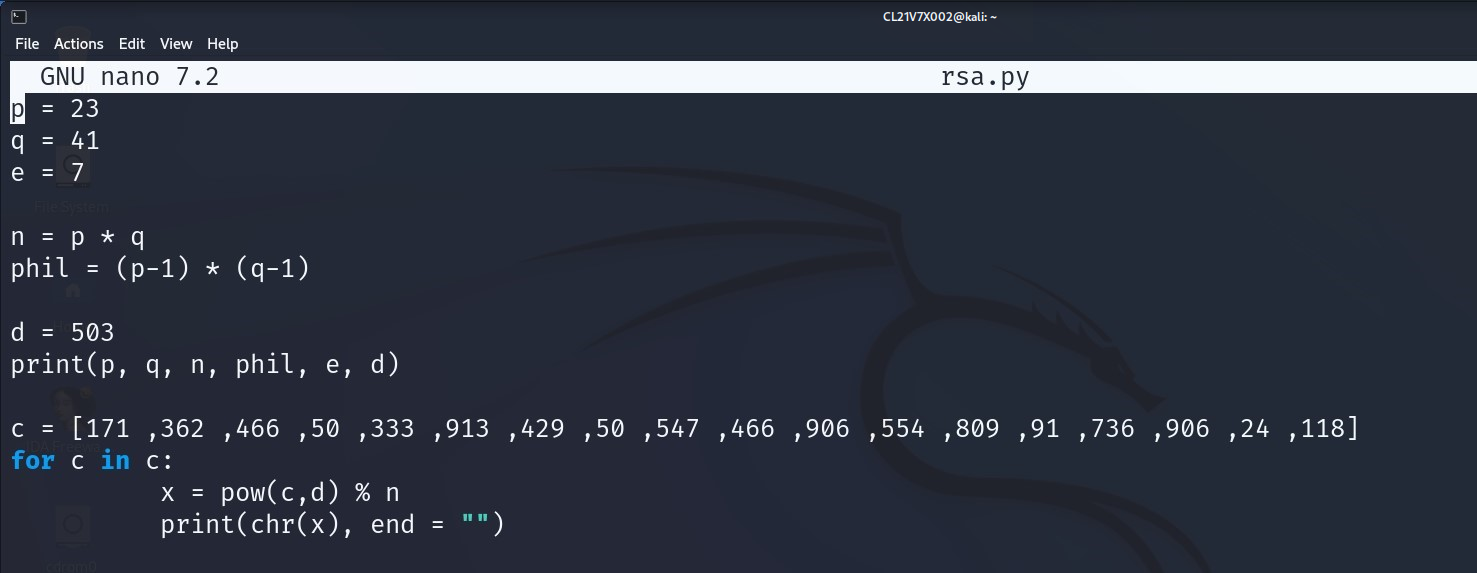
b. Áp Dụng Thuật Toán Mã Hóa:

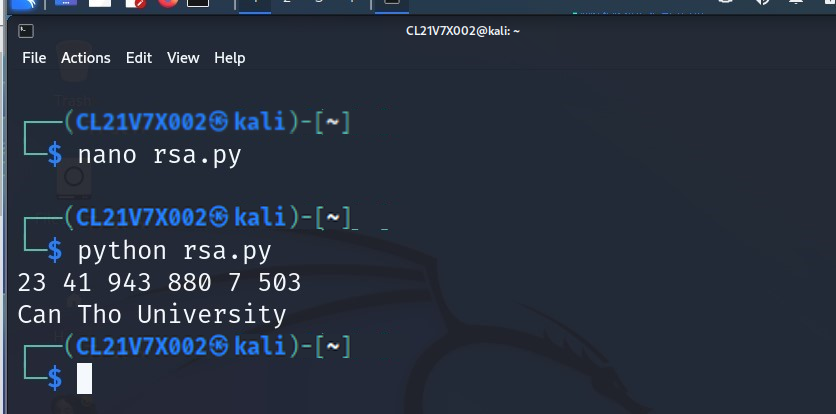
- Sử dụng khóa công khai (N, e) để mã hóa dữ liệu thành dạng mã hóa.

3. Giải Mã:

- Sử dụng khóa bí mật (N, d) để giải mã dữ liệu mã hóa thành dạng gốc.

5.3. Với p = 23, q = 41, e = 7 giải mã dữ liệu đã được mã sử dụng RSA sau: ”171 ,362 ,466 ,50 ,333 ,913 ,429 ,50 ,547 ,466 ,906 ,554 ,809 ,91 ,736 ,906 ,24 ,118”. Trong đó các con số là mã unicode của các kí tự được mã hóa.





---HẾT---

|  |
| --- |
| from Crypto.Cipher import AES key = b"aaaabbbbccccdddd" iv = b"0000111122223333" cipher = AES.new(key, AES.MODE\_CBC, iv)  with open("./aes/secret.png", "rb") as f:  ciphertext = f.read()  ciphertext\_trimmed = ciphertext[8:-2]  clear = cipher.decrypt(ciphertext\_trimmed)  clear = ciphertext[0:8] + clear + ciphertext[-2:] with open("./aes/secret\_de.png", "wb") as f:  f.write(clear) |