TEKNIK PEMBUATAN ANTI VIRUS DENGAN METODE PENCARIAN DATA HEADER FILE MENGGUNAKAN VISUAL BASIC 6.0

Hari Purwanto

Abstract

Technological advances, especially in the field of information technology in the last decade has changed very quickly where the computer in a few decades ago, only used as a word processor and so on.

A computer virus is a computer program that can duplicate or copy itself and spreads by inserting copies of itself into other programs or documents. Computer viruses can be destructive nature for example by destroying the data on the document, make computer users feel disturbed by their presence in a computer system, and does not cause deleterious effects at all.

Antivirus is a type of software that is used to detect and remove computer viruses from the computer system. Antivirus Software Virus Protection is also called, because the application can determine whether a computer system has been infected with a virus or not. In this paper the author tries to raise the issue by making Antivirus. This application is built using Microsoft Visual Basic 6.0.

Keywords: Virus, Anti-Virus, malcode, worms, registry, program

Pendahulan

Selama lebih dari tiga dekade yang lalu, virus komputer telah berkembang dari sekedar riset akademis menjadi masalah yang umum bagi para pengguna komputer di dunia. Masalah terbesar dari virus ini berasal dari penanggulangan efek kerugian yang ditimbulkan oleh penyebarannya. Efek kerugian ini semakin menjadi dengan maraknya penggunaan internet sebagai jalur komunikasi global antara pengguna komputer di seluruh dunia. Seiring dengan perkembangannya, virus komputer mengalami beberapa evolusi dalam bentuk, karakteristik serta media penyebarannya. bentuk evolusi tersebut dikenal dengan *Worms, Spyware, Trojan horse* dan program Malcode lain. Suatu program atau *script* apapun yang bersifat merusak atau merugikan dapat kategorikan sebagai *malcode* termasuk virus komputer, *worm* atau *trojan horse*.

Dengan kemampuan yang dimiliki sebuah virus tersebut, terkadang membuat seorang pengguna komputer awam menjadi panik. Bahkan disebabkan oleh ulah sebuah virus tersebut, seorang pengguna komputer dengan mudahnya melakukan format ulang terhadap sistem operasi yang digunakannya dengan harapan virus tersebut hilang. Namun hal ini juga menyebabkan beberapa data maupun program yang sudah terinstal juga ikut hilang.Berdasarkan observasi yang dilakukan penulis mengenai sebuah aplikasi virus dan antivirus, kinerja dari kedua aplikasi tersebut memiliki sifat dan perilaku yang sama diantara berbagai macam variannya. Dengan berkembangnya teknologi informasi, maka pembuatan aplikasi virus maupun aplikasi antivirus semakin mudah, sehingga diharapkan dengan mengetahui sifat dan perilaku keduanya, aplikasi virus dan antivirus yang akan dibangun dapat semakin memberikan penjelasan yang baik

Perkembangan penyebaran malcode di Indonesia pada awalnya lebih banyak didominasi oleh *worms* dan *virus* yang berasal dari luar negeri. Namun pada bulan Oktober 2005, dominasi ini mulai runtuh dengan menyebarnya virus-virus lokal yang hampir ada disetiap komputer di seluruh Indonesia, virus menyebar dengan sangat cepat dan sangat membuat risih bagi pengguna komputer, dengan demikian dibuatlah anti virus sebagai salah satu solusi mencegah penyebaran.

Berdasarkan latar belakang, maka permasalahan dalam penulisan ini, adalah bagaimana teknik pembuatan anti virus dengan metode pencarian header file data sizeofcode dan AddressOfEntrypoint.

Dari hasi penelitian yang dilakukan, adapun tujuan yang ingin dicapai dalam merancang suatu sistem anti virus yaitu :

1. Untuk membuat sebuah anti virus dengan metode pencarian header file data SizeOfCode dan AddressOfEntrypoint sebagai *pattern virus*.

2. Untuk mengetahui letak kekurangan-kekurangan dari sistem yang sedang berjalan..
3. Untuk mengetahui sejauh mana efektivitas dan efisiensi dari sebuah sistem anti virus yang menggunakan header file sizeofcode dan AddressOfEntryPoint sebagai *pattren virus* yang dirancang, untuk dibandingkan dengan sistem yang menggunakan metode cheksum (CRC-32).

Pengertian Virus Komputer

Istilah virus komputer tak asing lagi bagi kalangan pengguna komputer saat ini. Padahal, sekitar 12 tahun yang lalu, istilah ini telah dikenal oleh masyarakat pengguna komputer. Baru pada tahun 1988, muncul artikel-artikel di media massa yang dengan gencar memberitakan mengenai ancaman baru bagi para pemakai komputer yang kemudian dikenal dengan sebutan "virus komputer". Virus yang terdapat pada komputer hanyalah berupa program biasa, sebagaimana layaknya program-program lain. Tetapi terdapat perbedaan yang sangat mendasar pada virus komputer dan program lainnya.Virus dibuat oleh seseorang dengan tujuan yang bermacam-macam, tetapi umumnya para pembuat virus hanyalah ingin mengejar popularitas dan juga hanya demi kesenangan semata. Tetapi apabila seseorang membuat virus dengan tujuan merusak maka tentu saja akan mengacaukan komputer yang ditularinya.

Kemampuan Dasar Virus Komputer

Definisi umum virus komputer adalah program komputer yang biasanya berukuran kecil yang dapat menyebabkan gangguan atau kerusakan pada sistem komputer dan memiliki beberapa kemampuan dasar, diantaranya adalah :

a. Kemampuan untuk memperbanyak diri
   Yakni kemampuan untuk membuat duplikat dirinya pada *file-file* atau disk-disk yang belum ditularinya, sehingga lama-kelamaan wilayah penyebarannya semakin luas.
b. Kemampuan untuk menyembunyikan diri
   Yakni kemampuan untuk menyembunyikan dirinya dari perhatian user, antara lain dengan cara-cara berikut :
   1) Menghadang keluaran ke layar selama virus bekerja, sehingga pekerjaan virus tak tampak oleh user.
   2) Program virus ditempatkan diluar track2 yang dibuat DOS (misalkan track 41)
   3) Ukuran virus dibuat sekecil mungkin sehingga tidak menarik kecurigaan.
c. Kemampuan untuk mengadakan manipulasi.
   Sebenarnya rutin manipulasi tak terlalu penting. Tetapi inilah yang sering mengganggu. Biasanya rutin ini dibuat untuk :
   1) Membuat tampilan atau pesan yang menggangu pada layar monitor
   2) Mengganti *volume label* disket.
   3) Merusak struktur disk, menghapus *file-file*
   4) Mengacaukan kerja alat-alat I/O, seperti keyboard dan printer
d. Kemampuan untuk mendapatkan informasi.
   Yakni kemampuan untuk mendapatkan informasi tentang struktur media penyimpanan seperti letak boot record asli, letak tabel partisi, letak FAT32, posisi suatu file, dan sebagainya.
e. Kemampuan untuk memeriksa keberadaan dirinya.

Sebelum menyusupi suatu file, virus akan memeriksa keberadaan dirinya di dalam file tersebut dengan mencari ID (tanda pengenal) dirinya. File yang belum tertular suatu virus tentunya tidak mengandung ID dari virus yang bersangkutan. Kemampuan ini mencegah penyusupan yang berkali-kali pada suatu file yang sama.

Perbedaan Virus dan Worms

Dari klasifikasi di atas, karena memiliki aksi yang hampir serupa, terdapat dua tipe malware yang agak sulit dibedakan yaitu, virus dan worms. Untuk mempermudah melihat perbedaan kedua malware tersebut dapat ditinjau dari :

1. Definisi
   a. pada awalnya virus, adalah, sebagai suatu program yang dapat menginfeksi program lain dengan memodifikasinya, termasuk kemungkinan untuk berevolusi dengan menggandakan dirinya sendiri. Seiring dengan perkembangan teknik pemrogramannya, terdapat beberapa bentuk virus yang tidak sesuai dengan definisi tersebut. Sebagai contoh, suatu virus yang sering disebut companion virus memiliki kemampuan menggandakan diri tanpa mengubah program yang diinfeksinya. Sehingga menurut Peter Szor (2005), definisi yang lebih akurat untuk virus pada saat ini adalah, suatu program yang secara berulang

(recursively) dan dengan tegas (explicitly) menggandakan suatu versi dirinya sebagai kemungkinan untuk berevolusi

b. Sedangkan definisi formal untuk worms, adalah suatu program yang berpindah dari satu komputer ke komputer yang lain tanpa mengikatkan dirinya (attach itselft) pada sistem operasi komputer yang. Sejalan dengan perkembangannya, definsi *worms* tersebut sudah tidak begitu tepat. Beberapa worms sering menggunakan teknik untuk menyembunyikan kehadirannya dengan melakukan instalasi atau memodifikasi sistem. Sehingga definisi yang lebih tepat menurut Jose Nazario(2004), adalah suatu agen penginfeksi yang otonom dan independen dalam bereplikasi, serta memiliki kemampuan dalam menginfeksi sistem host baru melalui fasilitas jaringan.

2. Cara penyebaran

a. Virus memerlukan campur tangan pengguna dalam penyebarannya, misalnya dalam proses download, klik ganda pada file yang terinfeksi, dan lain-lain.

b. Sedangkan worms dapat secara otomatis menyebar dengan tanpa atau sedikit campur tangan dari penggunannya. Misalnya dengan satu kali klik pada file lampiran e-mail yang terinfeksi worms, maka satu atau beberapa sistem yang terkoneksi melalui e-mail tersebut akan segera terinfeksi.

Daerah-daerah rawan serangan virus

a. Registry
Registry windows adalah suatu database untuk menyimpan dan mengatur sistem di windows. Caru masuk di regedit yaitu :Klik start => Run => ketik '*regedit*' (tanpa tanda petik).

b. Alamat registry
Registry mempunyai alamat yang berguna untuk mengatur konfigurasi pada windows, misalnya :

1) Alamat registry:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run, Berguna untuk menjalankan suatu aplikasi secara\otomatis

2) Alamat registry:
HKEY_LOCAL_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\ Berguna untuk memanipulasi kode explorer.

3) Alamat registry:
HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\
Berguna untuk memanipulasi drive penginstalan, lisensi pada window

c. System Editor (Sysedit)
System editor atau yang biasa disingkat dengan sysedit adalah file-file tertentu yang di jalankan ketika komputermasuk dalam windows pertama kali, seperti regedit sysedit ini biasanya dipakai masih banyak dipakai disistem operasi windows lama seperti win95, win98 win3.1 jadi walaupun memang sudah lama tapi secara tidak langsung berpengaruh juga pada user yang memakai sistem Operasi windowsXp, Caru masuk di sysedit yaitu :Klik start => Run => ketik 'sysedit' (tanpa tanda petik)

d. Config.sys
Config.sys adalah file yang memuat tentang seluruh konfigurasi windows dan dijalankan ketika pertama kali windows mulai. Letak file ada pada drive C:\ dan mempunyai atribut file system dan di hidden.

e. Autoexec.bat
Autoexec.bat adalah file yang berisi perintah yang ada di komputer dan akan dijalankan pertama kali ketika ketika pertama kali windows mulai. Letak file ada pada drive C:\ dan mempunyai atribut file system dan di hiden

f. Win.ini
Ini juga sebuah file yang dieksekusi pertama kali oleh windows. File yang dieksekusi pertama kali oleh windows. File ini berisi tentang aplikasi 16 bit yang disupport oleh windows.

g. System.ini
System.ini adalah sebuah file yang berguna untuk menyimpan data font yang diakses oleh windows ketika pertama kali.

h. Msconfig
Msconfig sebenarnya adalah sebuah aplikasi , dari aplikasi msconfi, seluruh file system editor (sysedit) tadi dijalankan.

i. Direktory yang sering di incar oleh virus
C:\Windows
C:\Windows\System32

Mempersiapkan Pattern Virus

Berdasarkan pengamatan yang dilakukan penulis adalah cukup efektif menggunakan data SizeOfCode dan AddressOfEntryPoint sebagai pattern virus, karena sulit ditemukan dua executable memiliki SizeOf-

Code dan AddressOfEntryPoint yang sama, kecuali merupakan executable file yang sama, artinya dengan menggunakan data ini dapat mengenali file executable virus, karena jika merupakan file virus yang sama maka, akan memiliki data yang sama juga.

```
Public Function AmbilPatternFile(sFiles)
Dim sBuffer As String * 512
Dim ImageNTHeader As
IMAGE_NT_HEADERS
Dim e_lfanew As Integer
Dim nf As Integer

nf = FreeFile
AmbilPatternFile = ""
On Error GoTo Finally
Open sFiles For Binary Access Read As #nf

Get #nf, , sBuffer

e_lfanew = InStr(sBuffer, "PE" + Chr$(0) +
Chr$(0))
If e_lfanew > 0 Then
  Seek #nf, e_lfanew
  Get #nf, , ImageNTHeader
  AmbilPatternFile =
buatPattern(ImageNTHeader)
End If
Close #nf
Finally:
End Function
```

Potongan program diatas berfungsi mengambil pattern file dari suatu executable :

```
e_lfanew = InStr(sBuffer, "PE" + Chr$(0) +
Chr$(0))
```

ambil posisi e_lfanew dengan mencari substring “PE” dalam sBuffer.

```
If e_lfanew > 0 Then
  Seek #nf, e_lfanew
  Get #nf, , ImageNTHeader
  AmbilPatternFile =
buatPattern(ImageNTHeader)
End If
```

Jika e_lfanew > nol (ditemukan), maka akan dipindahkan file pointer ke offset dimana file header dimulai dengan perintah **Seek #nf, e_lfanew**, dan kemudian dilakukan pembacaan ke variable ImageNTHeader dengan perintah **Get #nf, , ImageNTHeader**, dan selanjutnya akan dipanggil function buatPattern dan melewatkan variable ImageNTHeader sebagai parameter. Variabel ImageNTHeader merupakan variable struktur IMAGE_NT_HEADERS

```
Dim ImageNTHeader As
IMAGE_NT_HEADERS
```

Dan struktur IMAGE_NT_HEADER dapat dilihat pada penjelasan sebelumnya.

```
Public Function
buatPattern(ImageNTHeader As
IMAGE_NT_HEADERS)
buatPattern=Right$("00000000"+Hex$(Imag
eNTHeader.OptionalHeader.AddressOfEntry
Point),8) + Right$("00000000" + Hex$
(ImageNTHeader.OptionalHeader.Size
OfCode), 8)

End Function
```

Function buatPattern akan mengembalikan pattern virus dalam bentuk hexadecimal yang merupakan data AddressOfEntryPoint dan SizeCode.

Menyimpan pattern virus ke file

Bisa saja langsung memasukkan pattern virus kedalam program, tetapi tentu saja pendekatan ini tidak efektif, dimana setiap ada pattern baru, maka harus melakukan modifikasi terhadap source code dan melakukan kompilasi ulang. Pendekatan lain yang dapat lakukan adalah dengan menyimpan pattern virus pada suatu text file terpisah (indoprog.vdf), sehingga setiap adanya pattern virus baru cukup dimasukan kedalam file.

Pattern Virus Nama virus
XXXXXXXXXXXXXXXyyyyyyyyyyyyyyyyyy
yyyyyyyyyy

Dimana 16 digit pertama adalah pola virus, dan diikuti oleh nama virus. Sehingga
00095F0000095000virus1
0000136000003000virus2
0000135800003000virus3
000070E000002000virus4
dst
setiap startup program antivirus cukup membaca semua pattern tersebut ke suatu variable array.

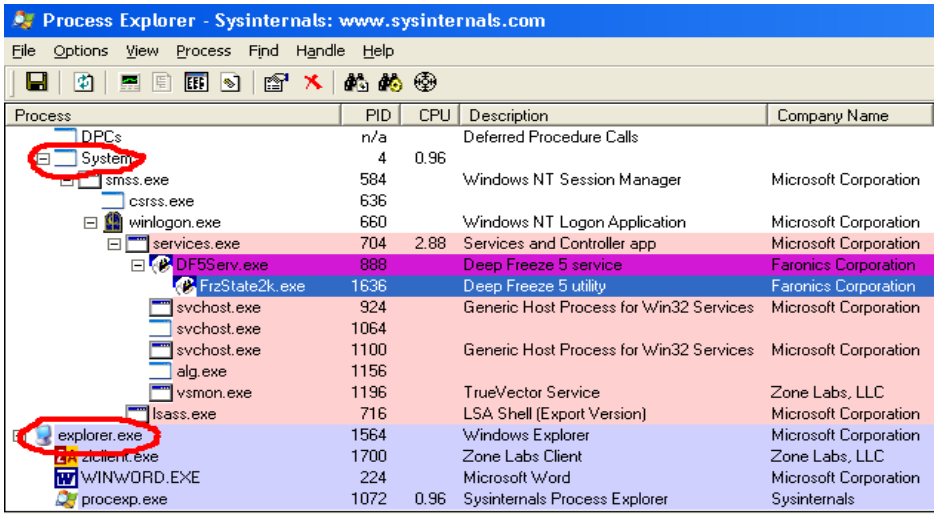
```
Dim PatternCount As Integer
Dim PatternVirus(100) As String

Private Sub loadVdf()
Dim cVDF As String
Dim nf As Integer
cVDF = App.Path + "\indoprog.vdf"
nf = FreeFile
PatternCount = 0
```

```
Open cVDF For Input As #nf
Do While Not EOF(nf)
  Input #nf, PatternVirus(PatternCount)
  Call lstHistory.AddItem("Baca : " +
PatternVirus(PatternCount), 0)
  PatternCount = PatternCount + 1
Loop
PatternCount = PatternCount - 1
Close #nf
End Sub
```

Teknik Mendeteksi proses virus dimemori dan menghentikannya

Sebagaimana dengan program executable lainnya, pada saat runtime program virus juga berupa process yang aktif dimemory.

Jika sistem sudah terinfeksi virus, maka proses dari virus juga akan tampil pada halaman proses pada task manager. Tetapi permasalahannya, banyak virus menggunakan nama file yang menyerupai program internal Windows sehingga menyulitkan untuk membedakan mana yang virus maupun mana yang merupakan program sebenarnya. Untuk dapat melihat data dari process yang lebih terperinci, bisa dengan menggunakan software Process Explorer dari sysinternal. (http://www.sysinternal.com).



Gambar 3.8 Process yang ditampilkan dengan Process Explorer

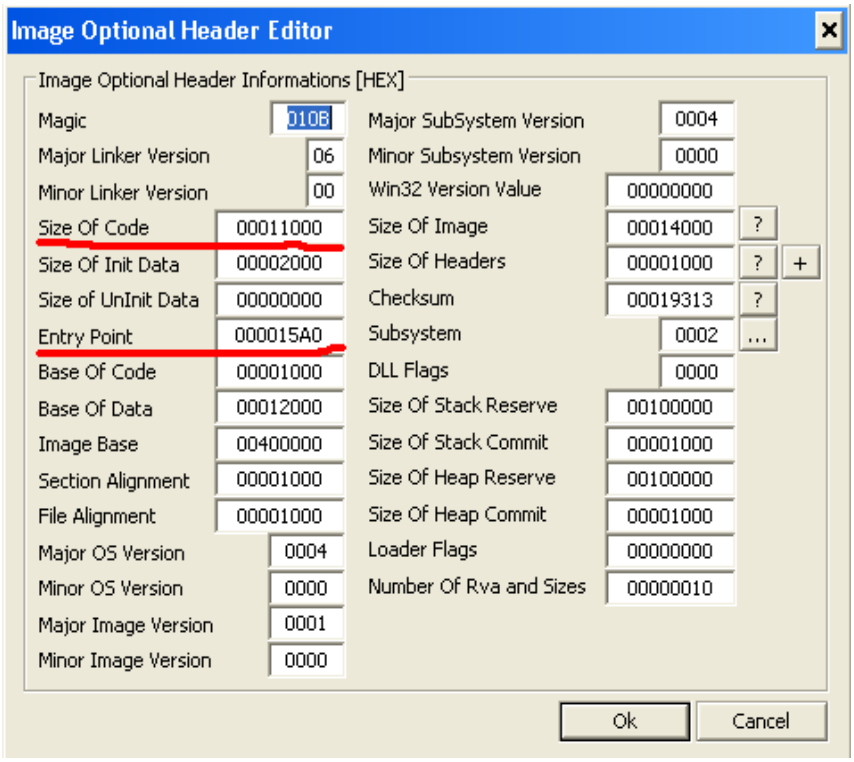
Pada Gambar diatas, dapat langsung dibedakan mana yang merupakan System Service dan mana yang merupakan aplikasi yang berjalan dibawah Explorer.exe, serta Description dan Company Name.

Kadang ada file image dari proses yang sama dengan file virus. Virus menggunakan EXE Packer untuk memperkecil ukurannya pada media penyimpanan maupun transfer data, tetapi pada saat dijalankan tentu saja harus diextract sehingga dalam hal ini image dari process virus akan berbeda dengan file virus. Misalnya program Indoprog Anti Virus.exe normalnya berukuran 86016 bytes Setelah dilakukan Packer menggunakan UPX, ukurannya menjadi lebih kecil.

```
C:\vbVrus>dir indoprogAV.exe
 Volume in drive C is Local Disk
 Volume Serial Number is 6CBD-EFCA

 Directory of C:\vbVrus

11/04/2006  12:01 PM            86,016 IndoprogAV.exe
               1 File(s)         86,016 bytes
               0 Dir(s)     468,287,488 bytes free

C:\vbVrus>upx IndoprogAV.exe

C:\vbVrus>upx IndoprogAV.exe
                       Ultimate Packer for eXecutables
   Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004
UPX 1.25w           Markus F.X.J. Oberhumer & Laszlo Molnar       Jun 29th 2004

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     86016 ->     35840   41.67%    win32/pe     IndoprogAV.exe

Packed 1 file.
```

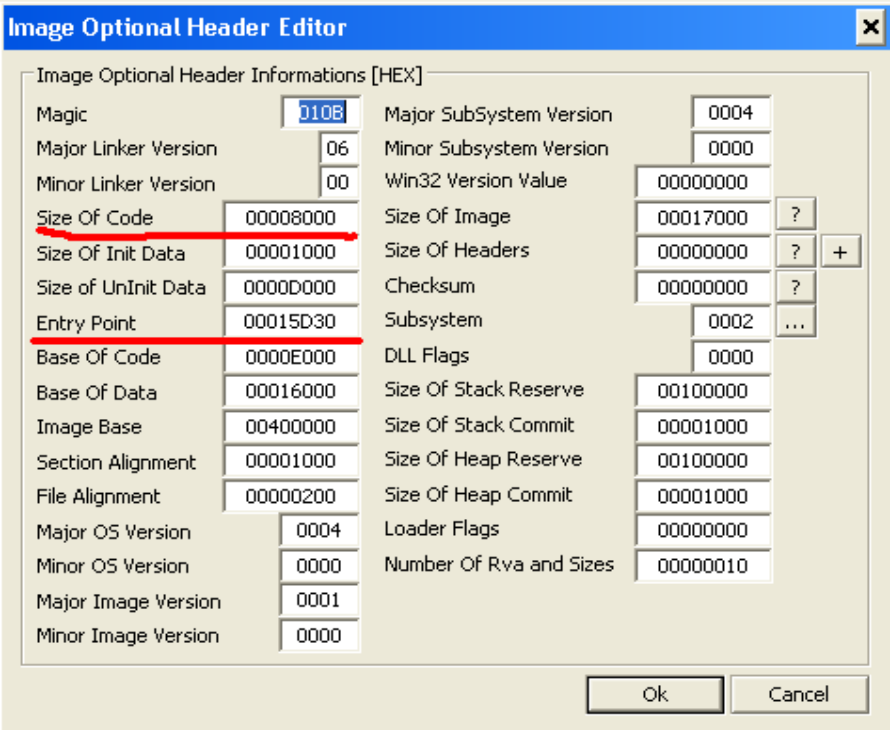
Gambar 3.9 Process UPX

Yaitu menjadi berukuran 35840 bytes. indoprog anti virus.exe sebelum di UPX. Berikut ini adalah Optional Header file



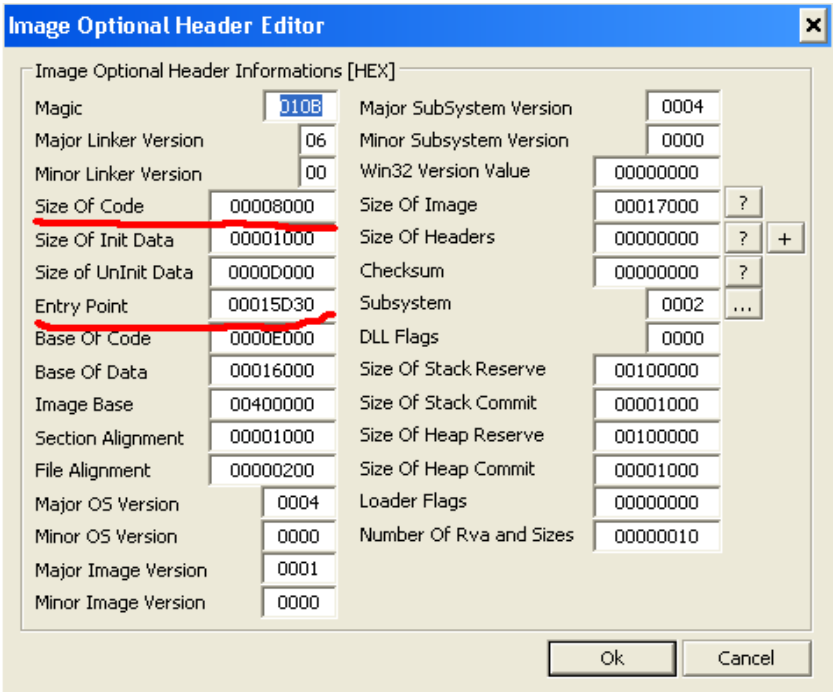
Gambar 3.10 Optional Header file Indoprog Anti Virus.exe sebelum UPX

Berikut ini adalah Optional Header file Indoprog Anti Virus.exe setelah di UPX.



Gambar 3.11 Optional Header file Indoprog Anti Virus.exe setelah UPX

Berikut ini adalah Optional Header dari image process Indoprog Anti Virus pada saat runtime.



Gambar 3.12 Optional Header image process Indoprog Anti Virus.exe (UPX)

Tetapi berdasarkan pengamatan yang dilakukan oleh penulis, ternyata data AddressOfEntryPoint dan SizeOfCode tidak diubah pada image proses, sehingga teknik pendeteksian dengan pola tersebut layak dilakukan. Pada prinsipnya keberadaan process dimemory memiliki PId (Process Id), dan masing-masing process terdiri dari Module-module (terdiri dari executable itu sendiri, dll, dan ocx yang diimport oleh executable tersebut).

Mengambil semua process yang aktif

Untuk mengambil process yang sedang aktif, membutuhkan fungsi API seperti :

```
Public Sub periksaProcesses()
Dim hSnapShot As Long
Dim ProcessEntry As PROCESSENTRY32
Dim NextEnumExists As Boolean
Dim Pos As Long
Dim pId As Long
Dim fileName As String
Dim baseName As String

hSnapShot =
CreateToolhelp32Snapshot(TH32CS_SNAP
PROCESS, 0)
If hSnapShot = 0 Then
   MsgBox "Failed to create Module and
Thread snapshot"
   Exit Sub
End If
ProcessEntry.dwSize = Len(ProcessEntry)
NextEnumExists = 0 <>
Process32First(hSnapShot, ProcessEntry)

While NextEnumExists
   pId = ProcessEntry.th32ProcessID

   Pos = InStr(ProcessEntry.szExeFile,
Chr(0))

   If Pos > 1 Then
      fileName =
Left(ProcessEntry.szExeFile, Pos - 1)
      baseName =
extractFilename(fileName)
   Else
      fileName = ""
      baseName = ""
   End If

   Call lstHistory.AddItem("Periksa (" +
Hex$(pId) + ") " + fileName, 0)

   Call periksaModules(pId)

   NextEnumExists = 0 <>
Process32Next(hSnapShot, ProcessEntry)
Wend

Call CloseHandle(hSnapShot)
End Sub
```

Dimana deklarasi dari masing-masing

fungsi API untuk CreateToolhelp32Snapshot, Process32First, Process32Next, CloseHandle adalah sebagai berikut :

```
Public Declare Function CreateToolhelp32Snapshot Lib "kernel32" _
  (ByVal dwFlags As Long, ByVal th32ProcessID As Long) As Long

Public Declare Function CloseHandle Lib "kernel32" _
  (ByVal hObject As Long) As Long

Public Declare Function Process32First Lib "kernel32" _
  (ByVal hSnapShot As Long, ByRef ThreadStruct As PROCESSENTRY32) As Long

Public Declare Function Process32Next Lib "kernel32" _
  (ByVal hSnapShot As Long, ByRef ThreadStruct As PROCESSENTRY32) As Long
```

sedangkan TH32CS_SNAPPROCESS merupakan konstanta dengan nilai &H2

```
Public Const TH32CS_SNAPPROCESS  As Long = &H2&
```

Dan PROCESSENTRY32 adalah struktur yang dideklarasikan sebagai berikut :

```
Public Type PROCESSENTRY32
   dwSize As Long
   cntUsage As Long
   th32ProcessID As Long
   th32DefaultHeapID As Long
   th32ModuleID As Long
   cntThreads As Long
   th32ParentProcessID As Long
   pcPriClassBase As Long
   dwFlags As Long
   szExeFile As String * 260
End Type
```

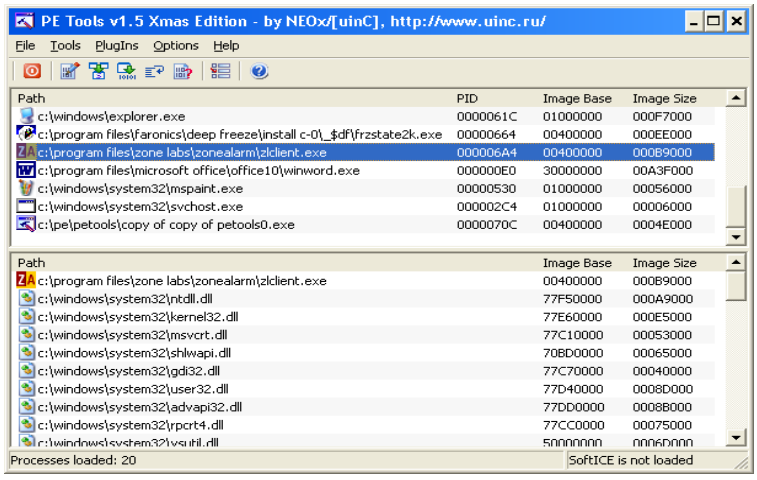
Potongan kode di atas akan menjelaskan bagaimana program akan melakukan looping mulai dari process yang pertama **NextEnumExists = 0 <> Process32First(hSnapShot,ProcessEntry)**, dan seterusnya dengan **NextEnumExists = 0 <> Process32 Next (hSnapShot, ProcessEntry)**. Selanjutnya dalam masing-masing looping akan diambil pId = ProcessEntry.th32ProcessID, dan diextract nama file exe image process tersebut

```
   Pos = InStr(ProcessEntry.szExeFile, Chr(0))

   If Pos > 1 Then
      fileName = Left(ProcessEntry.szExeFile, Pos - 1)
      baseName = extractFilename(fileName)
   Else
      fileName = ""
      baseName = ""
   End If
```

Dan masing-masing PId akan dilewatkan sebagai argument pada fungsi **Call periksaModules(pId)**

Memeriksa Masing-masing Module dalam Process

Setelah mendapatkan process, maka perlu memeriksa masing-masing module dalam process untuk mencocokan memory image dari masing-masing module dengan pattern virus yang telah persiapkan, jika ternyata ada memory image yang menyerupai pattern virus, maka process pemilik module tersebut harus dihentikan.



Gambar 3.13 Module dari process zlclient.exe

```
Public Sub periksaModules(pId As Long)
Dim ModuleEntry As TMODULEENTRY32
Dim hProcess As Long
Dim Proceed As Long
Dim hSnapShot As Long

Dim lWritten As Long

Dim sBuffer As String * 512
Dim ImageNTHeader As
IMAGE_NT_HEADERS
Dim e_lfanew As Integer
Dim Pattern As String

Dim i As Integer

If GetVersion =
VER_PLATFORM_WIN32_NT Then
  If Not SetPrivilege("SeDebugPrivilege",
True) Then Exit Sub
End If

hProcess =
OpenProcess(PROCESS_ALL_ACCESS,
False, pId)

If hProcess <> 0 Then
  hSnapShot =
CreateToolhelp32Snapshot(TH32CS_SNAP
MODULE, pId)

  If hSnapShot <> -1 Then
    ModuleEntry.dwSize = Len(ModuleEntry)
    Proceed = Module32First(hSnapShot,
ModuleEntry)

    Do While Proceed

      sBuffer = Space(1024)

      If ReadProcessMemory(hProcess,
ByVal ModuleEntry.modBaseAddr, ByVal
sBuffer, 512, lWritten) Then
        If lWritten > 0 Then
          e_lfanew = InStr(sBuffer, "PE" +
Chr$(0) + Chr$(0)) - 1
          If e_lfanew > 0 Then
            If
ReadProcessMemory(hProcess, ByVal
(ModuleEntry.modBaseAddr + e_lfanew),
ByVal ImageNTHeader,
Len(ImageNTHeader), lWritten) Then
              Pattern =
buatPattern(ImageNTHeader)
              i = 0
              Do While i <= PatternCount
                If Pattern =
Left(PatternVirus(i), Len(Pattern)) Then Exit
Do
                i = i + 1
              Loop
              If i <= PatternCount Then
                adanti virusirus = True
                Call
TerminateProcess(hProcess, 0)
                Call
lstHistory.AddItem("Found :" +
Mid$(PatternVirus(i), Len(Pattern) + 1), 0)
                Call
lstHistory.AddItem("Action: Kill process", 0)
              End If
            End If
          End If
        End If
      End If
      Proceed = Module32Next(hSnapShot,
ModuleEntry)
    Loop
  End If

  CloseHandle (hSnapShot)

End If

CloseHandle (hProcess)

If GetVersion() =
VER_PLATFORM_WIN32_NT Then
  Call SetPrivilege("SeDebugPrivilege",
False)
End If
End Sub
```

Sebagaimana dengan pengambilan process, pengambilan module-module berdasarkan PId juga membutuhkan fungsi WIN API seperti :OpenProcess, CreateToolhelp 32Snapshot, Module32First, Module32Next. Sesuatu yang menjadi permasalahan dalam pengambilan module adalah pada system operasi berbasis NT seperti NT4/2000/XP yang membutuhkan suatu priviledge SeDebugPrivilege.

```
If GetVersion =
VER_PLATFORM_WIN32_NT Then
  If Not SetPrivilege("SeDebugPrivilege",
True) Then Exit Sub
End If
```

Setelah mendapatkan modul, maka selanjutnya adalah melakukan ReadProcess Memory yang bertujuan membaca image dari modul.

```
sBuffer = Space(1024)

If ReadProcessMemory(hProcess, ByVal
ModuleEntry.modBaseAddr, ByVal sBuffer,
512, lWritten) Then
```

Dimana persiapkan string buffer yang berukuran 1024 bytes, dan melakukan proses pembacaan image ke variable sBuffer. Sesuatu hal yang perlu diperhatikan adalah pembacaan memori proses didasarkan pada nilai pointer ModuleEntry.modBaseAddr. lWritten akan mengembalikan ukuran byte yang terbaca, sehingga dengan memeriksa nilai lWritten akan diketahui apakah pembacaan berhasil dilakukan. Selanjutnya akan mencara posisi offset file Header (e_lfanew), dengan mencari **e_lfanew = lnStr(sBuffer, "PE"+Chr$(0)+Chr$(0))–1**

Setelah mendapatkan posisi e_lfanew, maka dapat melakukan pembacaan ke variable ImageNTHeader dengan perintah **ReadProcessMemory(hProcess, ByVal (ModuleEntry.modBaseAddr + e_lfanew), ByVal ImageNTHeader, Len (ImageNTHeader), lWritten)** yang secara kongkrit dapat dilihat pada potongan program berikut :

```
If ReadProcessMemory(hProcess, ByVal
ModuleEntry.modBaseAddr, ByVal sBuffer,
512, lWritten) Then
        If lWritten > 0 Then
          e_lfanew = InStr(sBuffer, "PE" +
Chr$(0) + Chr$(0)) - 1
          If e_lfanew > 0 Then
            If
ReadProcessMemory(hProcess, ByVal
(ModuleEntry.modBaseAddr + e_lfanew),
ByVal ImageNTHeader,
Len(ImageNTHeader), lWritten) Then
              Pattern =
buatPattern(ImageNTHeader)
              i = 0
              Do While i <= PatternCount
                If Pattern =
Left(PatternVirus(i), Len(Pattern)) Then Exit
Do
                i = i + 1
              Loop
              If i <= PatternCount Then
                adanti virusirus = True
                Call
TerminateProcess(hProcess, 0)
                Call
lstHistory.AddItem("Found :" +
Mid$(PatternVirus(i), Len(Pattern) + 1), 0)

 Call lstHistory.AddItem("Action: Kill
process", 0)
              End If
            End If
          End If
        End If
      End If
```

Setelah dibaca memori proses mulai dari offset e_lfanew ke-variabel ImageNTHeader maka dapat dilakukan proses generater PatternVirus, dan selanjutnya dibandingkan dengan pola virus yang dipersiapkan terlebih dahulu.

Menghentikan proses virus

Jika ditemukan pola virus tersebut, berarti proses tersebut merupakan process virus, sehingga harus dihentikan dengan

Call TerminateProcess(hProcess, 0)

Perancangan Program

Rancangan sistem ini terdiri hanya satu form saja yang sudah mencakup semua proses.

1. Rancangan input dan proses

Gambar 4.1 Form input proses

Algoritma dari program

Metode pencarian virus yang paling sering di pakai oleh anti virus yaitu metode CRC-32 (*Cyclic Redundancy Code*). Metode CRC-32 merupakan teknik yang semulanya digunakan untuk mengecek kerusakan pada file. Metode ini yang sering digunakan oleh anti virus lokal untuk mengecek signature dari virus, tetapi teknik ini tidak efesien apabila diterapkan pada malware yang sudah mengimplementasikan teknik polymorph.. Kasus virus lokal sudah ditemukan penggunaan teknik *polymorph*. Baik itu secara sederhana maupun kompleks. Cara yang biasa digunakan yaitu:

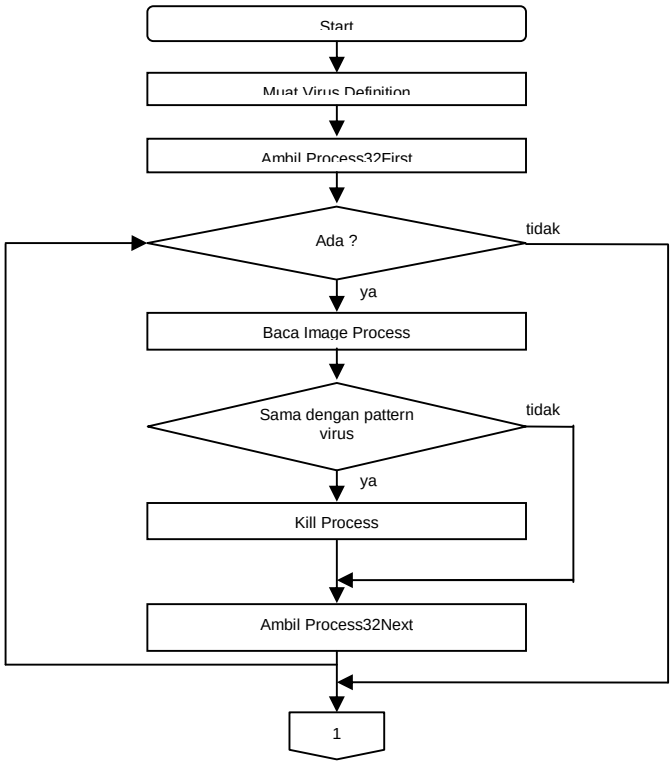
a. Merubah atau mengenkripsi nama variabel dan string.
b. Menambah atau mengurangi byte-byte tertentu di virus
c. Menggunakan engine polymorph tertentu

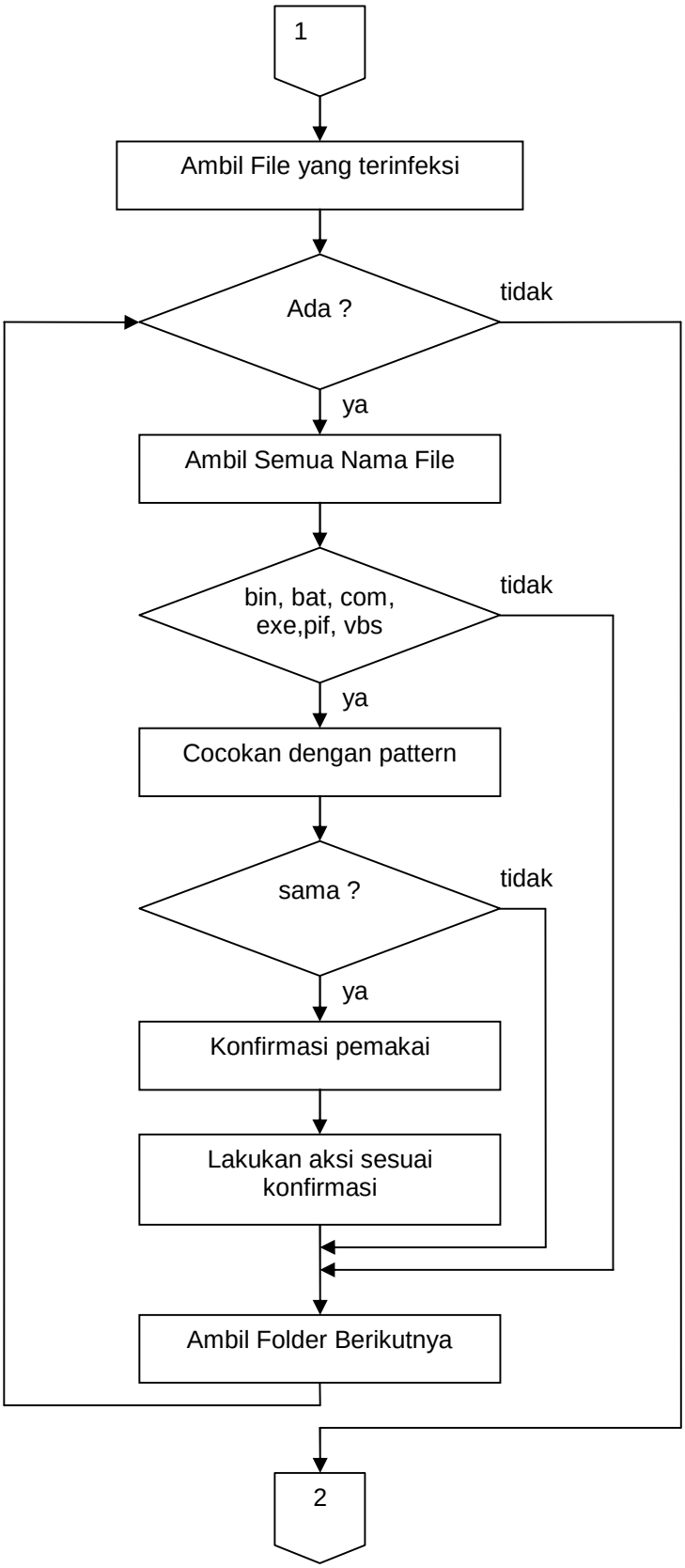
Jika secara normal metode Crc-32 ini sangat gampang untuk dikelabui, hal ini dikarenakan perubahan 1 bit kode pada program maka akan menyebabkan perubahan hasil pengecekan CRC-32.

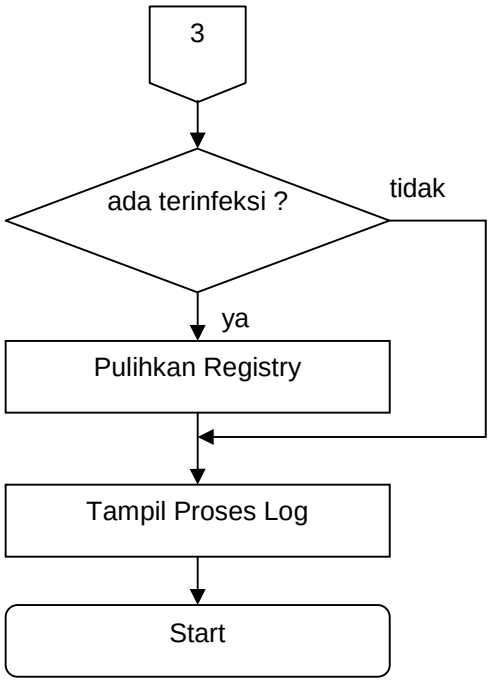
Algoritma dari aplikasi penghapus virus adalah sebagai berikut :

a. Mula-mula program akan memuat suatu daftar definisi virus yang berisi pola-pola virus untuk pendeteksian keberadaan process worm di system komputer maupun pada file.
b. Kemudian program akan mengambil semua PID dari semua yang sedang aktif di system komputer, dan selanjutnya mengambil image process berdasarkan PID, dan image dari masing-masing module process ini akan diperiksa dengan mencocokkan dengan masing-masing pola worm yang telah dipersiapkan sebelumnya,
c. jika ternyata image tersebut sama dengan salah satu pola, maka program akan menghentikan process berdasarkan PID, dan memberikan pesan kepada pemakai.
d. Tahapan selanjutnya adalah melakukan pencarian file-file yang berada pada system komputer, dengan melakukan proses pengambilan nama file. Berdasarkan nama file tersebut, program akan mengambil data dari file untuk dicocokan dengan masing-masing pola virus yang telah dipersiapkan, jika ternyata image tersebut sama dengan salah satu pola, maka program akan menampilkan informasi kepada user dan menghapus file tersebut. Selanjutnya program akan membersihkan registry yang diekploitasi oleh virus, dengan menghapus maupun mengembalikan nilai defaultnya.

Bagan Alir(Flowchart) Logika Proses Program Anti virus Sederhana




1
Ambil File yang terinfeksi
Ada ?
tidak
ya
Ambil Semua Nama File
bin, bat, com,
exe,pif, vbs
tidak
ya
Cocokan dengan pattern
sama ?
tidak
ya
Konfirmasi pemakai
Lakukan aksi sesuai
konfirmasi
Ambil Folder Berikutnya
2



Gambar, 4.2 Flowchart logika proses anti virus sederhana

Kesimpulan dan Saran

Dari hasil teknik uji coba dari sistem anti virus ini dapat di ambil kesimpulan bahwa:

1. Teknik penggunaan header file addres-Ofentrypoint dan sizeofcode, Sangat akurat dalam mengenal virus, walaupun virus telah merubah header filenya tapi datanya tetap sama.
2. *Engine scanernya* juga cepat dan ringan tidak terlalu memberatkan memori.

Anti virus ini juga perlu di tingkatkan sensitifitasnya, karena bila terdapat section dummy pada urutan section kedua atau tidak ada pada section, atau pembelokan pada entrypointnya maka data yang di teliti bisa saja salah, Fungsi ini juga masih rawan dari kesalahan analisa.

Maka dari itulah butuh penelitian lebih lanjut, dengan menambahkan algoritma baru fungsi ataupun prosedur, agar dapat mengembangkan dan memajukan kualitas aplikasi anti virus.

DAFTAR PUSTAKA

Aat Shadewa, 2006, Rahasia Membuat Anti virus Menggunakan Visual Basic, Yogyakarta: Penerbit DSI Publishing.

Gordon, A., Lawrence et. al., (2009), CSI/FBI Computer Crime and Security Survey 2008, CSI Publication, Washington DC, http://www.GoCSI.com/,1 November 2008.

Nazario, Jose, et. al., (2004), Defense and Detection Srategies Againts Internet Worms, Artech House inc., Norwood MA.

Pietrek, Mat; Peering Inside the PE A tour of the PE: A Tour of the Win32 Portable Executable File format; MSDN;1994

Szor, Peter (2005), The Art of Computer Virus Research and Defense, Addison Wesley Proffesional, New Jersey.

Yohanes Nugroho (2005), Analisis Lengkap Virus Brontok, http://www.compactbyte.com /brontok/Analisis Lengkap Virus Brontok.html, 11 September 2006

_____(2006), PT Vaksincom, http://vaksin.com/, 9 Oktober 2006