Detecting A Botnet By Reverse Engineering

Oesman Hendra Kelana dan Khabib Mustofa

Abstract— Botnet malware is a malicious program. Botnet that infects computers, called bots, will be controlled by a botmaster to do various things such as: spamming, phishing, keylogging Distributed Denial of Service (DDoS) and other activities that are generally profitable to the owner of the bot (botmaster) or those who use botnet services. The problem is that many computers have been controlled by botnets without the knowledge of the computer owner. There are many ways to examine botnets, for example by studying the traffic from the botnet network, studying how botnets communicate to each, studying how each robot receives orders to do something, and so forth. Of the many methods, the most frequently and commonly used is the reverse engineering, where researchers study how a botnet works by botnet debugging.

In this study the author tries to understand or research botnets by taking a type of botnet, namely Agobot, using reverse engineering. One of the result of the research is that malware program files in general and in particular botnet has a technique to obscure the way that research using reverse engineering.

Another result also shows that the botnet Agobot runs on computers by using the Windows service, and by changing the Windows registry so that every time the computer starts, Agobot always actively works in the computer memory.

Keywords— *Malware, Bot, Botnet, Botmaster, Agobot, Spam, Distributed Denial of Services, Identity Theft, Computer Security, Reverse Engineering, Debug, Windows Service, the Registry.*

I. PENGANTAR

Botnet adalah sekumpulan komputer yang terhubung ke Internet, yang dikuasai (*compromised*) dan dikendalikan oleh para penyerang (*hacker/cracker*) untuk tujuan yang tidak baik dan melanggar hukum (*illegal*). Kata ini berasal dari kata *robot* atau disingkat *bot*, yaitu suatu program komputer yang bisa berjalan secara otomatis [6]. Kadang bot disebut juga dengan istilah *zombie* [1].

O. Hendra Kelana, Computer Science Study Program, Gadjah Mada University, Yogyakarta, Indonesia 55281.

K. Mustofa, Computer Science Study Program, Gadjah Mada University, Yogyakarta, Indonesia 55281.

Program komputer bot merupakan program yang ada di Internet yang bersifat jahat (*malicious ware*, disingkat *malwar*e). Bot dikatakan jahat karena program ini menggabungkan unsur-unsur yang terdapat dalam program *virus*, *worm*, *spyware* dan program-program malware lainnya [6]. Dengan kemampuan yang dimilikinya tersebut, bot dapat melakukan banyak hal yang jahat, melebihi yang dapat dilakukan oleh malware lain yang disebut di atas.

Orang yang mengendalikan suatu botnet dikenal dengan sebutan *botmaster* atau *bot-herder*. Motivasi botmaster menjalankan botnet adalah semata-mata untuk keuntungan finansial. Botnet sangat menguntungkan bagi pemilik atau pengendalinya, karena botnet dapat digunakan untuk mendapatkan uang lewat kegiatan *spam*, penyebaran sypware yang bekerja untuk mencuri data identitas penting (*user ID* dan *password*), dan bahkan untuk mendapatkan uang dari hasil memeras perusahan-perusahaan yang menggunakan teknologi informasi, dengan cara mengancam akan melakukan *Distributed Denial of Service* (DDoS) [3].

Sebelum adanya botnet, motivasi utama dari penyerangan-penyerangan dalam Internet adalah untuk popularitas dan ketenaran semata. Berdasarkan rancangannya, serangan-serangan ini bersifat *noisy* (menimbulkan kehebohan) dan mudah dideteksi. Contoh-contoh malware yang terkenal: worm email Mellisa, ILOVEYOU, Code Red, Slammer, dan Sasser. Meskipun dampak dari virus-virus dan worm-worm ini cukup hebat, kerusakan yang ditimbulkan tidak berlangsung lama dan biaya yang dikeluarkan untuk mengatasi kerusakan akibat virus/worm ini adalah terutama dari kerugian karena tidak beroperasinya komputer (*lost opportunity cost*) dan biaya tenaga kerja untuk membersihkan komputer dari malware. Setelah virus dihapus dari komputer dan celah kerentanan (*vulnerability*) ditutup, para penyerang tidak dapat mengendalikan lagi komputer tersebut.

Mencermati akan hal tersebut di atas, maka bot/botnet dapat dianggap sebagai suatu ancaman yang cukup serius bagi keamanan sistem komputer. Keberadaan botnet dapat

mengganggu kegiatan yang dilakukan oleh pengguna komputer dan Internet. Oleh karena itu maka perlu dilakukan suatu penelitian untuk meneliti botnet sehingga masalah-masalah yang ditimbulkan botnet dapat diatasi, dan kegiatan-kegiatan yang berhubungan dengan komputer dan Internet tidak terganggu karenanya.

Permasalahan yang timbul adalah bahwa kadang botnet tidak mudah untuk dilacak keberadaannya di dalam suatu sistem komputer. Selain itu, walaupun telah diketahui keberadaannya, bot kadang sulit untuk dihapus dari dalam suatu sistem komputer. Meskipun telah menggunakan program antivirus, kadang bot tidak dapat dihilangkan dengan seketika. Untuk itulah penelitian ini dilakukan, yakni bagaimana mendeteksi suatu botnet di dalam suatu sistem komputer.

II. TINJAUAN TEORI

II.I AGOBOT

Agobot sering dikenal juga dengan nama *Gaobot*, muncul pertama kali pada tahun 2002 dengan memiliki desain modular dan fungsi-fungsi yang signifikan [4]. Agobot memiliki desain modular artinya: Agobot tidak memasukkan keseluruhan kode bot-nya sekaligus ke dalam suatu sistem komputer, tetapi hanya modul pertama saja. Agobot memiliki tiga modul:

1. Modul pertama yang dibawa dalam bot berisi klien bot *Internet Relay Chat* (IRC) dan *backdoor* akses jarak jauh.
2. Modul kedua menyerang dan mematikan proses (*service*) dari antivirus.
3. Modul ketiga mencegah pengguna komputer untuk mengakses *website-website* tertentu, biasanya situs pembuat antivirus.

Ketika suatu modul selesai menjalankan tugas utamanya, modul tersebut akan membaca modul berikutnya. Dengan cara ini, bot-herder dapat meng-*update* modul kedua dan ketiga dengan teknik-teknik yang baru atau tambahan daftar situs. Kemampuan untuk meng-update modul-modulnya ini membuat varian Agobot menjadi banyak hingga bisa mencapai ribuan.

Agobot menggunakan IRC untuk melakukan *Command & Control* (C&C), tetapi menggunakan *peer-to-peer* (P2P) untuk aplikasi *file-sharing*-nya. Bot bisa diperintah melalui IRC, tetapi Agobot juga membuka backdoor akses jarak jauh sehingga memungkinkan suatu bot diakses secara langsung.

Agobot memiliki kemampuan-kemampuan [4]. :

a. Memindai (*scan*) celah-celah keamanan tertentu.
b. Meluncurkan serangkaian serangan DDoS.
c. Mencari *CD key* untuk *game-game.*
d. Mematikan proses antivirus dan proses monitoring.
e. Memodifikasi *file-file* host untuk mencegah akses ke situs antivirus.
f. Menyembunyikan diri dengan menggunakan teknik *rootkit*.
g. Menggunakan teknik-teknik tertentu untuk menyulitkan reverse engineering.

Bot ini merupakan program berorientasi objek dan *multi-thread* yang kebanyakan ditulis dalam bahasa C++, dan sebagian kecil ditulis dengan bahasa Assembly. Agobot merupakan contoh dari botnet yang pemakainya membutuhkan sedikit atau bahkan tidak membutuhkan sama sekali pengetahuan pemrograman.

Berbagai versi dan varian baru bot yang berasal dari keluarga Agobot muncul dengan cepat dan berkembang melebihi keluarga bot yang lain. Bot-bot lain yang termasuk keluarg*a* Agobot adalah Forbot, Phatbot, Urxbot, Rxbot, dan Rbot. Kebanyakan turunan dari Agobot yang terkini harus dengan program aplikasi Microsoft Visual Studio. Ukuran Agobot yang siap untuk ditularkan berukuran antara 12 Kilo Byte (KB) hingga 500 KB, tergantung kepada fiturnya, optimisasi compiler dan modifikasi biner.

Suatu modul yang dibuat untuk salah satu anggota dari keluarga Agobot, umumnya dapat di-*porting* dengan mudah untuk digunakan oleh bot lain. Modul yang bisa dipasangkan dari satu bot ke bot lain bisa berbeda-beda, sesuai dengan kebutuhan pemiliknya. Hal inilah yang membuat terbentuknya banyak varian Agobot.

II.II MENGHAPUS BOTNET

Ketika botnet timbul, respons yang dilakukan oleh para pembuat program antivirus adalah sama seperti terhadap *malware* yang pernah ada sebelumnya, yakni para pembuat antivirus membuat tanda (*signature*) dari virus dan membuat teknis penghapus virus bagi setiap bot yang ada. Pendekatan seperti ini awalnya berjalan lancar, tetapi para peneliti belakangan ini mulai mencoba mengeksplorasi metode tingkat lanjut untuk menghapus lebih dari satu bot pada suatu waktu. Hal ini perlu dilakukan mengingat

bahwa menghapus suatu botnet yang memiliki puluhan ribu anggota dengan cara satu per satu (individual) akan memakan waktu lama dan membosankan.

Menghapus bot-bot secara individu umumnya tidak memiliki dampak yang terasa bagi botnet secara keseluruhan, tetapi cara ini merupakan langkah pertama yang peting dalam menangkal botnet. Pendekatan dasar yang digunakan oleh program antivirus, yakni berdasarkan deteksi signature, masih efektif untuk mendeteksi kebanyakan botnet. Selain itu beberapa botnet telah mulai menggunakan polimorfisme, di mana botnet tersebut membuat bentuk unik dari kode tiap bot yang dapat menghindari pendeteksian yang menggunakan signature. Misalnya Agobot, yang memiliki ribuan varian, menyertakan fungsi *built-in* yang mendukung polimorfisme untuk mengubah signature sesuai keinginannya.

Untuk menangani bot-bot yang lebih canggih ini, dan juga malware yang melakukan polimorfisme, pendeteksian dilakukan dengan analisa tingkah laku (*behavior*) dan pengalaman (*heuristic*). Stinson [5], telah mengembangkan suatu pendekatan berdasarkan noda (*taint*) yang dinamakan *BotSwat* yang menandai semua data yang berasal dari jaringan. Jika data ini digunakan sebagai masukkan bagi *suatu system call*, ada kemungkinan besar bahwa tingkah laku tersebut berhubungan dengan bot, karena input pengguna umumnya masuk lewat keyboard atau mouse dan sistem *end-user* pada umumnya.

II.III REVERSE ENGINEERING

Reverse Engineering adalah suatu proses untuk mengekstraksi pengetahuan atau cetak-biru desain dari segala sesuatu yang dibuat oleh manusia [2]. Konsep ini telah ada sejak lama, jauh sebelum komputer atau teknologi modern; diperkirakan reverse engineering telah ada pada waktu revolusi industri terjadi.

Reverse engineering biasanya dilakukan untuk memperoleh pengetahuan, ide dan filosofi dari suatu desain yang hilang diakibatkan oleh tidak tersedianya informasi tentang desain tersebut. Dalam beberapa kasus, informasi dimiliki oleh seseorang yang tidak mau membagikan informasinya kepada orang lain. Dalam kasus lain, informasi yang diperlukan telah hilang atau telah musnah.

Secara tradisional, reverse engineering adalah tentang membuka “bungkus” dari produk yang diteliti dan membongkar (membedah) produk sehingga diketahui desain dan hal-hal yang tersembunyi (rahasia) dari produk tersebut. Rahasia tersebut yang kemudian digunakan untuk membuat produk yang sejenis atau bahkan yang lebih baik. Dalam banyak industri, reverse engineering meliputi pemeriksaan produk di bawah mikroskop atau membongkarnya menjadi bagian-bagian kecil dan meneliti cara kerja dari bagian-bagian tersebut.

Reverse engineering merupakan model dalam software engineering yang berkebalikan dengan model yang umum digunakan yakni model *waterfall*, di mana dalam reverse engineering hasil dari tahap implementasi (kode program) diubah menjadi tahap analisis sistem. Jika dalam model waterfall, tahapan yang dilakukan adalah analisis sistem terlebih dahulu, baru kemudian diikuti dengan tahapan-tahapan desain dan implementasi, maka dalam reverse engineering justru tahap implementasi yang lebih dulu dikerjakan, dan kemudian dilakukan analisis dan desain dari sistem yang akan dibuat.

Reverse engineering memiliki hubungan yang penting dengan aspek-aspek keamanan komputer. Sebagai contoh, reverse engineering digunakan dalam penelitian tentang enkripsi, di mana peneliti me-reverse (membalik) suatu produk enkripsi dan mengevaluasi tingkat keamanannya. Teknik reversing juga sering digunakan dalam kaitannya dengan malware, baik oleh pengembang (pembuat) malware atau oleh pengembang program penghapus malware, seperti pengembang antivirus dan lain sebagainya. Selain itu, reversing sangat populer digunakan oleh *cracker* untuk menganalisis dan bahkan mengatasi berbagai skema proteksi penggandaan (*copy protection*) perangkat lunak.

Reversing sering digunakan baik oleh pengembang malware maupun pengembang anti malware. Pengembang malware acap kali menggunakan reversing untuk melokalisir celah-celah keamanan (*vulnerabilities*) dari software-software sistem operasi dan software-software lainnya. Celah-celah keamanan tersebut dapat digunakan untuk menembus lapisan-lapisan sistem pertahanan komputer yang kemudian digunakan untuk menyebarluarkan malware, yang biasanya dilakukan melalui Internet. Selain penginfeksian malware, para pelaku kejahatan ini kadang melakukan teknik reversing untuk menentukan lokasi celah-celah keamanan perangkat lunak agar membolehkan program malware mendapatkan akses ke informasi sensitif (penting) atau bahkan mendapatkan kendali penuh atas sistem komputer.

Di sisi lain, reversing digunakan para pengembang software antivirus untuk membedah dan menganalisa semua program malware yang mereka terima. Mereka menggunakan teknik reversing untuk melacak setiap langkah yang dilakukan program malware, dan memperkirakan kerusakan yang dapat timbul, memperkirakan tingkat penyebarluasannya, bagaimana malware dapat dihapus dari sistem yang terinfeksi, dan penyebarluasan apa saja yang dapat dihindarkan sekaligus.

III. Cara Penelitian

Ada beberapa cara yang dapat dilakukan untuk mendeteksi keberadaan bot pada umumnya, dan Agobot pada khususnya. Dalam penelitian ini digunakan reverse engineering. Teknik ini telah sering digunakan untuk mendeteksi keberadaan virus, worm, trojan dan lain sebagainya.

Metode yang dilakukan dalam penelitian ini adalah dengan melakukan *disassembly* suatu bot dengan menggunakan program/*tools disassembler* Sesuai dengan namanya, disassembler merupakan program yang digunakan untuk mengubah program yang semula berupa bahasa mesin, menjadi bahasa assembly yang lebih mudah untuk dibaca.

III.I System-Level Reversing Dan Code-Level Reversing

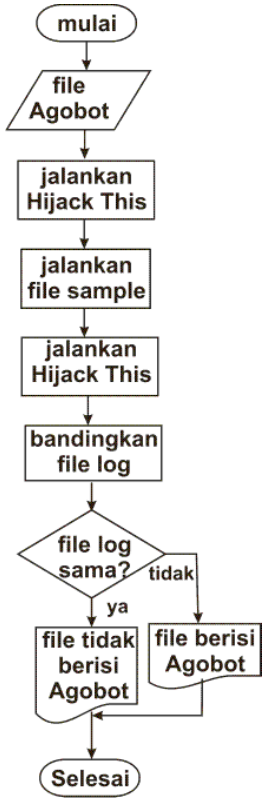
Setelah Virtual Machine (VMWare Player) diinstall, kemudian dilakukan penginstallan sistem operasi di dalam Virtual Machine tersebut, yakni Microsoft Windows XP. Sesudah sistem operasi guest diinstall, pada sistem operasi guest diinstall program PE Editor, PE Explorer, dan IDA Pro, selanjutnya di-copy-kan file sample yang telah terinfeksi Agobot (dalam penelitian ini, filenya bernama wincrt32.exe) ke suatu direktori kerja.

Dalam proses analisis terhadap Agobot ini dilakukan dua pendekatan reversing, yakni *system-level reversing* dan *code-level reversing*. System-level reversing dilakukan dengan menggunakan tool tertentu untuk mendapatkan informasi mengenai semua aktivitas yang dilakukan oleh Agobot. Sedangkan code-level reversing dilakukan dengan mendisassembly file program dan mempelajari perintah-perintah atau baris-baris program yang ada di dalamnya.

III.I.I System-Level Reversing

Pada system-level reversing peneliti tidak perlu mengetahui secara detail tentang jalannya program. Jadi peneliti tidak secara langsung melihat kepada kode program atau bahasa assembly dari file. Dengan analisis ini penulis berusaha memahami sifat dari file dengan cara menjalankannya dalam suatu lingkup yang terkendali. System-level reversing menggunakan tool (program bantu) Hijack This untuk yang bermanfaat untuk mendeteksi aktivitas-aktivitas yang dilakukan Agobot.

Proses-proses yang terjadi dalam system-level reversing dapat dilihat dalam diagram alir pada Gambar 1.



Gambar 1. Diagram alir proses system-level reversing

Dari gambar 1, bisa dilihat proses-proses di dalam system-level reversing. Proses yang dilakukan mula-mula adalah menjalankan program HiJack This, yang berguna untuk mengetahui keadaan sistem komputer sebelum file sampel dijalankan. Kemudian file sampel dijalankan. Setelah file Sampel dijalankan, program HiJack This dijalankan lagi untuk mengetahui keadaan sistem komputer pasca file sampel dijalankan.

Dari hasil file log HiJack This sebelum dan sesudah file sampel Agobot dijalankan, ada beberapa perbedaan yang bisa diamati:

1. Baris 0029: **C:\WINDOWS\system32\wincrt32.exe** Menunjukkan bahwa Agobot telah menyalin file program (wincrt32.exe) ke direktori C:\Windows\system32 dan menjalankannya.

2. Baris 0043: **HKLM\..\Run: [Configuration Loader] wincrt32.exe** Baris 0043: **HKLM\..\RunServices: [Configuration Loader] wincrt32.exe** Menunjukkan bahwa Agobot telah membuat dua bua key di Registry Windows di "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows\ CurrentVersion\Run" dan "HKEY_LOCAL_MACHINE\SOFTWARE\ Microsoft\Windows\CurrentVersion\RunServices", yang masing-masing berisi value "wincrt32.exe".

3. Baris 0056: **Service: Configuration Loader (bF) - Unknown owner - C:\WINDOWS\system32\wincrt32.exe** Menunjukkan bahwa Agobot telah menjalankan file program suatu proses/service.

III.I.II CODE LEVEL-REVERSING

Setelah system-level reversing selesai, proses selanjutnya yang akan dilakukan adalah code-level reversing. Hal ini dilakukan karena dalam system-level reversing, hanya dilihat aktivitas Agobot semata, sedangkan karakteristik yang dimiliki belum terlihat. Code-level reversing dilakukan untuk mengetahui karakter dan tingkah laku dari Agobot dengan cara melihat ke dalam baris-baris perintah dari file program Agobot. Untuk melakukan code-level reversing penulis menggunakan software IDA Pro. Tetapi sebelum melakukan disasssembly suatu file, terlebih dahulu file tersebut harus diperiksa apakah file tersebut memiliki format PE (Portable Executable) dan apakah file tersebut telah dikompresi terlebih dahulu. Proses-proses yang dilakukan di dalam code-level engineering bisa digambarkan seperti dalam Gambar 2.

Gambar 2. Diagram alir proses code-level reversing

Dalam penelitian ini akan dilakukan pengamatan-pengamatan terhadap aktivitas yang sama yang dilakukan Agobot seperti yang terjadi pada system-level reversing, yakni mengamati proses penyalinan dan pengeksekusian file, pembuatan key di registry, pengaktifan service agobot.

Selanjutnya, untuk mengetahui bagian program yang melakukan pengaktifan service Windows Agobot, dilakukan pencarian string

1. Penganalisisan file PE

PEditor merupakan suatu program bantu (tool) yang bermanfaat untuk menampilkan struktur dari suatu file. Penulis menggunakan tool ini untuk memeriksa apakah file yang diteliti merupakan file yang valid/sahih. PEditor akan menunjukkan semua bagian dari file PE. PEditor memiliki bagian fungsi yang dapat menampilkan string atau data yang ditambahkan ke dalam file. Kita bisa mengetahui dengan sekilas bahwa suatu data ditambahkan ke file atau ada file yang diikutsertakan (*embedded*) dalam bagian resource dari file. Penambahan data atau file ke dalam file merupakan teknik yang sering digunakan oleh trojan. Data yang ditambahkan berisi kode jahat (malicous) yang biasanya terletak setelah akhir dari objek yang ada dalam suatu file PE. Entry point (titik mula) dari program berisi perintah untuk melompat (jump) ke kode tambahan, sehingga kode jahat bisa dilaksanakan terlebih dahulu untuk kemudian akan kembali ke program yang sesungguhnya.

Suatu file dikatakan memiliki format PE jika file tersebut dapat dibuka oleh PEditor dan menampilkan informasi-informasi tentang file tersebut. Jika suatu file yang tidak berformat PE, maka ketika dibuka oleh PEditor, program akan memberitahukan bahwa file tersebut bukan file berformat PE.

2. Peng-unpacking-an file

Packer biasanya digunakan untuk menyamarkan kode program atau untuk memadatkan file. Trojan umumnya menggunakan packer agar terhindar dari deteksi signaturenya oleh antivirus. Penulis akan menggunakan tool PE Explorer yang akan dipakai untuk membuka (unpack) dari file sample. PE explorer mampu membuka hampir semua jenis packer yang biasanya digunakan, di mana tool akan menemukan ciri (signature) dari packer di dalam file dan kemudian menggunakan unpacker yang sesuai dengan signature tersebut.

Dari hasil pemeriksaan yang dilakukan PE Explorer, diketahui bahwa file sampel yang akan kita analisa ternyata telah di-packing dan dienkripsi dengan packer UPX (*Ultimate Packer eXecutable*).

Agar sampel bisa dianalisis lebih lanjut, kemudian dilakukan proses unpack dari file sample untuk kemudian disimpan dalam suatu file. File ini akan digunakan untuk proses-proses selanjutnya, yakni untuk men-disassembly file sampel Agobot.

3. Pen-disassembly-an file sampel

Setelah file sampel di-unpack dengan program PE Explorer, selanjutnya file tersebut dibuka oleh program IDA Pro untuk di-disassembly guna melihat baris-baris perintah dari file sampel. Dari hasil disassembly terhadap file sampel, kemudian penulis mencari bagian-bagian program yang melakukan hal-hal berikut:

a. Penyalinan dan pengeksekusian file sampel Agobot

Mula-mula Agobot melakukan penyalinan file sampel Agobot di direktori sistem, yakni di C:\Windows\System32\. Untuk mendapatkan direktori sistem tersebut, Agobot menjalankan suatu rutin seperti yang tertera di Gambar 3.

```
. . .
004057D1        call        ds:GetSystemDirectoryA
. . .
004057FF        call        ds:GetModuleHandleA
. . .
00405806        all         ds:GetModuleFileNameA
. . .
0040580C        push        offset LibFileName ; "kernel32.dll"
00405811        call        ds:LoadLibraryA
. . .
```

Gambar 3. Bagian program yang mencari direktori sistem

Dari Gambar 3 terdapat baris perintah:

004057D1 call ds:GetSystemDirectoryA

Perintah ini memanggil fungsi GetSystemDirectoryA dari Windows API (kernel32.dll). Kernel32.dll adalah file yang menangani manajemen memori, operasi input/output (I/O) dan proses/thread dari sistem operasi Windows.

Selanjutnya program memuat modul-modul lain yang ada di Windows API, dengan perintah-perintah berikut:

```
0004057FF    call    GetModuleHandleA
00405806     all     ds:GetModuleFileNameA
0040580C     push    offset LibFileName;"kernel32.dll"
00405811     call    ds:LoadLibraryA
```

Setelah direktori sistem ditemukan, Agobot kemudian menyalin dan menjalankan file program Agobot. Fungsi/subrutin yang melakukan penyalinan dan pengeksekusian file dapat dilihat pada Gambar 4.

```
. . .
00405AB6        call        ds:CreateFileA
. . .
00405AD4        call        ds:WriteFile
. . .
00405B5F        call        ds:CreateProcessA
. . .
```

Gambar 4. Potongan program untuk menyalin dan menjalankan Agobot

Dari Gambar 4 terdapat baris perintah:

00405AB6 call ds:CreateFileA

yang memanggil fungsi CreateFileA Windows API untuk membuat sebuah file. Selanjutnya Agobot menyalin file dan menyimpan file dengan perintah:

00405AD4 call ds:WriteFile

Terakhir, Agobot memanggil fungsi API CreateProcessA untuk menjalankan file hasil penyalinan dengan perintah:

00405B5F call ds:CreateProcessA

Nama file program yang mengandung Agobot bisa bermacam-macam. Secara default nama file program disimpan di dalam suatu variabel. Gambar 5 memperlihatkan penyimpanan nama file dari program yang mengandung Agobot.

```
. . .
0041E20C aWincrt32_exe  db 'wincrt32.exe',0  ; DATA XREF:
. . .
```

Gambar 5. Potongan program berisi nama file dari Agobot

Dari Gambar 5 terdapat baris perintah:

0041E20C aWincrt32_exe db 'wincrt32.exe'

Merupakan bagian program yang mendeklarasikan variabel aWincrt32_exe dengan isi variabel "wincrt32.exe".

b. Pembuatan key di dalam registry Windows

Setelah menyalin dan menjalankan file programnya, Agobot kemudian membuat key di dalam registry Windows. Gambar 6 menampilkan potongan proram Agobot yang membuat key-key di dalam registry Windows. Key-key tersebut adalah di dalam "HKEY_LOCAL_MACHINE\ SOFTWARE\Microsoft\Windows\CurrentVersion\Run" dan "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ RunServices" dengan masing-masing key adalah "Configuration Loader" dan value "wincrt32.exe".

```
. . .
00405B89            push        offset SubKey ;
"Software\\Microsoft\\Windows\\CurrentVersi"...
. . .
.text:00405B93      call        edi ; RegCreateKeyExA
. . .
00405BB9            mov         ebx, ds:RegSetValueExA
. . .
00405BC3            call        ebx ; RegSetValueExA
. . .
00405BDC            push        offset aSoftwareMicr_0 ;
"Software\\Microsoft\\Windows\\CurrentVersi"...
. . .
00405BE6            call        edi ; RegCreateKeyExA
. . .
00405C10            call        ebx ; RegSetValueExA
. . .
```

Gambar 6. Potongan program untuk membuat key di registry

Dari Gambar 6 terdapat baris-baris perintah:

```
00405BE6      call      edi ; RegCreateKeyExA
00405B89      push      offset SubKey  ; "Software\\
                        Microsoft\\Windows\\CurrentVersi"...
```

Perintah-perintah ini berfungsi untuk memanggil Windows API (Advapi32.dll) untuk membuat suatu key di dalam registry Windows dengan key:
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ Run".

Perintah-perintah:

00405BB9 mov ebx, ds:RegSetValueExA
00405BC3 call ebx ; RegSetValueExA

Merupakan bagian program yang membuat key registry:
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" dan kemudian mengisinya dengan value "wincrt32.exe".

Begitu juga dengan baris-baris perintah:

```
00405BDC      push      offset aSoftwareMicr_0 ; "Software\\
                        Microsoft\\Windows\\CurrentVersi"...
00405BE6      call      edi ; RegCreateKeyExA
00405C10      call      ebx ; RegSetValueExA
```

Merupakan bagian program yang membuat key registry:
"HKEY_LOCAL_MACHINE\SOFTWARE\ Microsoft\Windows\CurrentVersion\ RunServices" dan kemudian mengisinya dengan value "wincrt32.exe".

c. Pengatifkan service dari file sampel Agobot

Setelah membuat key dalam registry Windows, selanjut Agobot menjalankan/mengaktifkan program Agobot sebagai sebuah service. Gambar 7 memperlihatkan potongan program yang mengaktifkan service dari file sampel Agobot.

```
. . .
00405CC2    call    ds:OpenSCManagerA ; Establish a connection to the service
. . .
00405CF3    call    ds:GetSystemDirectoryA
. . .
00405D5A    call    ds:CreateServiceA
```

Gambar 7. Potongan program untuk mengaktifkan service Agobot

Dari Gambar 7. terdapat baris perintah:

```
00405CC2        call        ds:OpenSCManagerA ; Establish a
                            connection to the service
00405CF3        call        ds:GetSystemDirectoryA
00405D5A        call        ds:CreateServiceA
```

Merupakan bagian program yang memanggil Windows API: OpenSCManagerA yang berfungsi untuk membuat koneksi dengan *service control manager* (SCM). Setelah koneksi terhubung, Agobot mencari direktori sistem dengan memanggil fungsi: GetSystemDirectoryA untuk mendapatkan direktori tempat menyimpan file "wincrt32.exe", yakni direktori C:\Windows\System32. Setelah file didapatkan, Agobot kemudian menjalankan file wincrt32.exe sebagai suatu service dengan memanggil fungsi: CreateServiceA.

III.II SISTEM PENDETEKSIAN DAN PENGHAPUSAN AGOBOT

Dari hasil analisis di atas, penulis kemudian membuat suatu sistem untuk mendeteksi dan sekaligus menghapus Agobot yang ada dalam suatu komputer. Adapun rancangan sistemnya adalah sebagai berikut:

1. Pemeriksaan service Windows
   Proses pendeteksian dan pencarian Agobot dimulai dengan mencari Windows service terlebih dahulu. Hal ini dilakukan mengingat bahwa Windows service Agobot adalah program botnet yang sedang berada di memori dan aktif bekerja memantau semua kegiatan yang ada di dalam komputer. Service ini akan selalu memantau keberadaan Agobot di dalam komputer, baik itu di dalam media penyimpanan maupun dalam registry Windows. Jika penghapusan Agobot dilakukan terlebih dahulu pada file Agobot atau registry Windows, maka service Agobot akan melakukan perbaikan dengan cara menambah/mengubah registry dan file Agobot agar keberadaannya di dalam komputer tersebut tetap berjalan. Sehingga dapat dikatakan akan sia-sia jika menghapus terlebih dahulu file dan registry, Seharusnya Windows service dari Agobot yang harus terlebih dahulu dimatikan, sehingga setelah itu proses lain akan dengan lancer dijalankan, dan tanpa akan mengalami hambatan.
   Pada tahap awal ini, sistem akan mencari dan memeriksa di dalam sistem operasi, apakah ada service yang bernama *Configuration Load* (bF). Jika ada maka sistem akan menghapus service tersebut. Jika tidak ditemukan maka berarti sistem komputer tersebut belum tertular Agobot.

2. Pemeriksaan registry Windows
   Jika service dari Agobot telah dihilangkan dari memori komputer, langkah selanjutnya adalah menghapus registry yang menjalankan file program Agobot setiap kali komputer dinyalakan. Jika pembersihan komputer hanya berhenti sampai menghapus Windows service dari Agobot, maka ketika komputer dinyalakan ulang, program Agobot akan aktif kembali. Tetapi jika registry yang menjalankan program Agobot dihapus, maka ketika direstart komputer tidak akan menjalankan kembali program Agobot, sehingga komputer sudah bisa dikatakan terbebas dari Agobot.
   Pada tahap ini, mula-mula sistem akan mencari nama file dari Agobot yang disimpan di registry Windows dengan alamat "HKEY_LOCAL_MACHINE\ SOFTWARE\Microsoft\Windows\CurrentVersion\Run dan key "Configuration Loader". Value yang didapat merupakan nama file dari program Agobot.
   Selanjutnya sistem akan menghapus kunci (*key*) dan nilainya yang ditulis oleh Agobot di dalam register sistem operasi. Jika kunci ditemukan, maka sistem akan langsung menghapus key beserta nilainya, jika key tidak ditemukan, maka register tidak akan diubah-ubah. Pada tahap ini sistem akan menghapus key yang ada di tiga tempat:

   - HKEY_LOCAL_MACHINE\SOFTWARE\ Microsoft\Windows\ CurrentVersion\Run
     Dalam lokasi ini akan dihapus kunci: Configuration Loader yang berisi nilai: wincrt32.exe

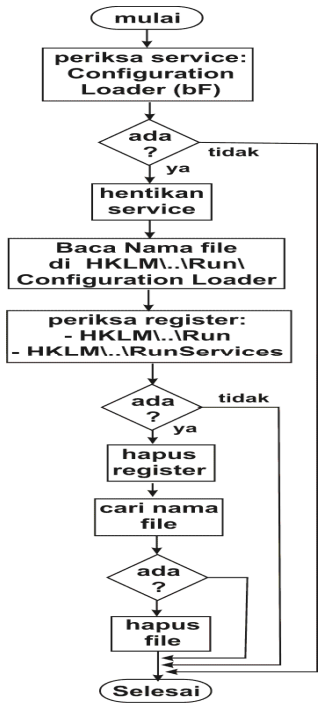
   - HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows\CurrentVersion\RunOnce
     Dalam lokasi ini akan dihapus kunci: Configuration Loader yang berisi nilai: wincrt32.exe.

   -HKEY_LOCAL_MACHINE\SOFTWARE Microsoft\Windows\CurrentVersion\RunOnce
     Dalam lokasi ini akan dihapus kunci: Configuration Loader yang berisi nilai: wincrt32.exe.

- HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\Services
  Dalam lokasi ini akan dihapus kunci: bF yang berisi nilai: C:\Windows\system32\wincrt32.exe.

3. Penghapusan file Agobot
   Selanjutnya, sistem akan mencari file dari Agobot, yang dalam hal ini bernama wincrt32.exe. File ini disimpan oleh Agobot di dalam directory C:\Windows\System32\. Walaupun service dan register telah dibersihkan dari Agobot, tetapi keberadaan file dari Agobot akan menjadi suatu yang berbahaya jika tidak dihapus. Agobot bisa menular kembali ke dalam sistem komputer jika suatu hari dengan tanpa sengaja file ini dieksekusi. Untuk menghindari itu file dari Agobot harus dihapuskan.

Agar lebih jelas, proses-proses pendeteksian dan penghapusan Agobot dapat dilihat urutannya di dalam Gambar 8.



Gambar 8. Flowchart sistem pendeteksian dan penghapusan Agobot

IV. HASIL DAN PEMBAHASAN

Untuk mengimplementasikan hasil analisa dan perancangan, penulis membuat program komputer untuk mendeteksi dan menghapus Agobot dengan menggunakan bahasa pemrograman C/C++. Pemilihan bahasa C dilakukan mengingat bahasa pemrograman cocok digunakan untuk pemrograman yang berhubungan dengan sistem operasi komputer. Dengan alasan tersebut maka kemudian penulis menggunakan bahasa pemrograman ini.

Ketika program dijalankan, program memeriksa service dari Agobot, dengan menggunakan fungsi cek_servis. Tampilan awal program ketika mencari service Agobot tampak seperti Gambar 9.

```
Mencari Windows service Agobot
Harap tunggu sebentar ...
```

Gambar 9. Tampilan program yang sedang mencari & menghapus service Agobot

Jika service ditemukan, maka program kemudian akan menampilkan pesan bahwa service Agobot sudah dimatikan. Gambar 10 menampilkan tampilan pesan pemberitahuan bahwa service telah dimatikan dan sedang melakukan proses berikutnya yakni mencari nama file dari program Agobot dan menghapus key Agobot dari registry.

```
Mencari Windows service Agobot
Harap tunggu sebentar ...

Service Agobot telah dihentikan!

Nama file program Agobot:
"C:\Windows\System32\wincrt32.exe"

Menghapus key registry yang menjalankan
Agobot
Harap tunggu sebentar ..
```

Gambar 10. Tampilan program yang berhasil menghapus service Agobot

Setelah menghapus key Agobot dari registry, program kemudian akan mencari file program Agobot, dalam hal ini file: wincrt32.exe yang terletak di folder: C:\Windows\System32\. Gambar 11 memperlihatkan tampilan program yang sudah menghapus key Agobot di dalam registry dan sedang mencari file Agobot.

```
---------------------------------------
 Anti Agobot, program untuk mencari
 menghapus botnet Agobot
 (c) 2010 by Oesman H. Kelana
---------------------------------------

Mencari Windows service Agobot
Harap tunggu sebentar ...

Service Agobot telah dihentikan!

Nama file program Agobot:
"C:\Windows\System32\wincrt32.exe"

Menghapus key registry yang menjalankan Agobot
Harap tunggu sebentar ..

Program telah menghapus value Agobot di registry

Mencari file Agobot
Harap tunggu sebentar ...
```

Gambar 11. Tampilan program berhasil menghapus key Agobot di registry

Jika file yang dicari ditemukan, akan keluar tampilan seperti yang ada pada Gambar 12.

```
Mencari Windows service Agobot
Harap tunggu sebentar ...

Service Agobot telah dihentikan!

Nama file program Agobot:
"C:\Windows\System32\wincrt32.exe"

Menghapus key registry yang menjalankan Agobot
Harap tunggu sebentar ..

Program telah menghapus value Agobot di registry

Mencari file Agobot
Harap tunggu sebentar ...

Program menemukan file Agobot!
Program telah menghapus file Agobot

Selesai
```

Gambar 12. Tampilan program yang berhasil menghapus file Agobot

V. KESIMPULAN DAN SARAN

Setelah semua rangkaian proses penelitian yang dilakukan, akhirnya penulis akan menutup hasil penelitian ini dengan beberapa kesimpulan:

1. Seperti halnya malware yang lain, Agobot dapat juga diteliti dengan menggunakan metode reverse engineering.

2. Dengan reverse engineering terhadap Agobot, ditemukan adanya service Windows yang dijalankan Agobot untuk mengaktifkan dirinya. Demikian pula di dalam registry Windows, ditemukan pula adanya key-key baru yang dibuat Agobot untuk mengaktifkan dirinya setiap kali komputer dinyalakan. File Agobot yang diaktifkan, disimpan di dalam direktori/folder Windows\System32\.

Hasil dari penelitian yang dilakukan terbatas hanya pada jenis botnet Agobot. Padahal di dalam dunia nyata, sesungguhnya ada banyak bot-bot lain yang berkembang dengan jumlah yang makin hari makin bertambah banyak dan dengan karakter dan tingkah laku yang berbeda-beda. Untuk itu penulis menyarankan beberapa hal:

1. Perlu dilakukan penelitian terus menerus tentang botnet secara umum dan senantiasa mencari varian-varian bot yang baru agar penanggulangan dari masalah yang ditimbulkan botnet dapat dilakukan sesegera mungkin.

2. Perlu ada kerja sama antara para peneliti di bidang botnet pada khususnya, dan malware pada umumnya dengan pengembang program antivirus atau antimalware untuk bersama-sama menghadapi ancaman dan kerusakan yang ditimbulkan oleh malware maupun botnet.

VI. DAFTAR PUSTAKA

[1] Aycock, J., 2006, *Computer Viruses and Malware*. Springer, New York.

[2] Eilam, E., 2005, *Reversing: Secrets of Reverse Engineering*. Wiley, Indianapolis.

[3] Holz, T., 2005, "A Short Visit to the Bot Zoo", *IEEE Security and Privacy*, 3, 3, 76-79.

[4] Schiller, C. A., J. Binkley, D. Harley, G. Evron, T. Bradley, C. Willems & M. Cross , 2007, *Botnets: The Killer Web App*. Syngress, Burlington.

[5] Stinson, E. & J. C. Mitchell, 2008, *Characterizing Bots' Remote Control Behavior*, Lee W. *et al*., *Botnet Detection: Countering the Largest Security Threat*, Springer, New York.

[6] Wang X. dan D. Ramsbrock, 2009, *The Botnet Problem*, J. R. Vacca, *Computer and Information Security Handbook*, Morgan Kaufman, Burlington.