p-ISSN : 2301-8402, e-ISSN : 2685-368X ,dapat diakses melalui https://ejournal.unsrat.ac.id/index.php/elekdankom

Analisa *Malware* Menggunakan Metode *Dynamic Analysis* Pada Jaringan Universitas Sam Ratulangi

Virgiawan A. Manoppo, Arie S. M. Lumenta, Stanley D. S. Karouw
Jurusan Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu, 95115, Indonesia
afidmanoppo@gmail.com, al@unsrat.ac.id, stanley.karouw@unsrat.ac.id
Diterima: 24 Juli 2020; direvisi: 30 September 2020; disetujui: 17 Desember 2020

Abstract — The globalization era brings exponential development to various sectors, one of which is the field of science and technology. Unfortunately, the development is not only on the positive aspect, but also spread to the negative direction. One example of the negative development of technology is the emergence of various kinds of cyber crime, such as the spread of malware. The purpose of this study is to analyze the characteristics of malware found on Sam Ratulangi University's network. The method used in this research is Dynamic Analysis using the Cuckoo Sandbox tool to prevent the infection of malware itself. After analyzing the characteristics of the malware, the researcher concluded that are several signatures, strings, and changes to the registry value.

Keywords — Cuckoo Sandbox; Dynamic Analysis; Malware; Malware Analysis.

Abstrak — Di era globalisasi ini, terdapat berbagai macam sektor yang mengalami perkembangan yang eksponensial, salah satu di antaranya adalah bidang IPTEK. Sayangnya, perkembangan tersebut tidak hanya berdampak positif, tapi juga merambah ke arah negatif. Salah satu perkembangan negatif tersebut adalah munculnya berbagai macam tindak kejahatan siber, seperti penyebaran *malware*.

Tujuan dari penelitian ini adalah untuk menganalisa karakteristik *malware* yang ditemukan pada jaringan Universitas Sam Ratulangi. Adapun metode yang dipakai dalam penelitian ini adalah dengan *Dynamic Analysis* dan menggunakan tool *Cuckoo Sandbox*, sehingga tidak ada resiko untuk terinfeksi *malware*. Berdasarkan analisa yang dilakukan tentang karakteristik dari *malware*, dapat disimpulkan bahwa terdapat beberapa *signature*, *string*, dan perubahan pada *value registry*.

Kata kunci* — Analisa *Malware; Cuckoo Sandbox; Dynamic Analysis; Malware.

I. PENDAHULUAN

Pada era globalisasi saat ini, aktifitas sehari-hari manusia sudah bergantung dengan teknologi. Baik itu dalam bidang sosial, politik, ekonomi maupun pendidikan. Dalam dunia pendidikan, teknologi dimanfaatkan dengan berbagai macam cara, seperti penggunaan *e-book, e-learning, website* dan lain lain. Perkembangan teknologi ini tentu dimanfaatkan oleh berbagai pihak. Selain pemanfaatan dalam sisi positif, terdapat pula pihak-pihak yang memanfaatkan dalam sisi negatif. Dalam hal ini seperti mengembangkan sebuah *software* yang dapat melakukan tindak kejahatan atau yang biasa disebut *cyber crime*. Salah satu bentuk dari kejahatan ini yaitu penyebaran *Malware* dengan begitu mudah. Berdasarkan laporan dari situs Vaksin.com pada kuartal 1 tahun 2019 di dominasi oleh 3 kategori *malware*, pertama kategori *malware* sendiri (44 %) disusul oleh *Adware* (38 %) dan terakhir adalah *Trojan* (18 %).

Malicous software atau yang lebih dikenal sebagai *Malware* merupakan perangkat lunak yang secara eksplisit didesain untuk melakukan aktifitas berbahaya atau perusak perangkat lunak lainnya seperti *Trojan, Virus, Spyware* dan *Exploit* [1]. Sehingga dibutuhkan Analisa untuk menentukan apakah aplikasi dalam komputer itu teridentifikasi adalah sebuah *Malware* dan untuk mengetahui karakteristik dari malware tersebut dan dampak pada sistem setelah *malware* tersebut dieksekusi.

Berdasarkan fakta – fakta terkait *malware* di atas perlu dilakukan metode analisa untuk mengetahui motif – motif dari serangan *malware* sehingga kita mengetahui karakteristik dari malware tersebut.

Analisa *malware* secara umum dapat dilakukan dengan dua metode, yaitu *Dynamic Analysis*, dan *Static Analysis* [5]. Meskipun tujuan dari kedua metode tersebut sama yaitu untuk menjelaskan tentang bagaimana sebuah *malware* berkerja

Gambar 1. Statistik *Malware* Indonesia Kuartal 1 2019
(Sumber : www.vaksin.com)

tetapi dalam proses analisa kedua metode tersebut sangatlah berbeda dalam segi *tools*, waktu dan kemampuan. *Dynamic analysis* dilakukan dengan mengeksekusi contoh *malware* untuk kemudian dipelajari perilaku yang ditimbulkan oleh *malware* tersebut. Berbeda dengan *static analysis* yang pada saat proses analisanya tidak mengeksekusi contoh *malware* melainkan dengan membongkar *source code malware* lalu mempelajari dan memahami kode tersebut.

Dynamic Analysis bisa dilakukan dengan dua cara yaitu manual dan menggunakan *tool* analisa otomatis yaitu *cuckoo sandbox. Cuckoo sandbox* dapat melakukan analisa *malware* secara otomatis dan menampilkan informasi – informasi yang dilakukan oleh sampel *malware*.

Universitas Sam Ratulangi merupakan sarana penunjang proses pembelajaran dan pelayanan informasi kepada mahasiswa, dosen, dan staff. Universitas ini juga turut menggunakan teknologi dalam lingkungannya. Seperti Portal Akademik, E-Learning, dan Website unsrat yang dapat diakses bukan hanya oleh mahasiswa dan dosen tapi seluruh masyarakat. Hal ini tentu membuat Universitas Sam Ratulangi membutuhkan lingkungan yang aman dari berbagai ancaman serangan malware agar infromasi yang dibutuhkan tidak merugikan pengguna.

Berdasarkan latar belakang masalah di atas malware merupakan perangkat lunak yang sangat merugikan pengguna maka kita perlu mengetahui bagaimana cara kerja dan apa saja dampak yang ditimbulkan oleh malware menggunakan metode dynamic analysis. Maka dapat disimpulkan sebagai berikut :

1) Bagamaina menganalisis malware menggunakan metode Dynamic Analysis.?
2) Bagaimana membangun mesin virtual menggunakan cuckoo sandbox.?

A. *Malware*

Malware (Malicious Software) merupakan program yang dirancang untuk disusupkan ke dalam sebuah sistem dengan tujuan untuk melakukan beraneka ragam aktivitas yang bersifat merugikan pemiliknya. Merugikan dalam arti kata dampak negatif yang ditimbulkan dapat berkisar mulai dari sekedar memperlambat kinerja sistem hingga merusak bahkan menghancurkan data penting yang tersimpan dalam sistem yang dimaksud. Ada tiga jenis *Malware* klasik yang paling banyak ditemui, yaitu: *Virus, Worms,* dan *Trojan Horse.* [6].

1) *Virus*

Virus merupakan program komputer yang bersifat "*malicious*" (memiliki tujuan merugikan maupun bersifat mengganggu pengguna sistem) yang dapat menginfeksi satu atau lebih sistem komputer melalui berbagai cara penularan yang dipicu oleh otorasisasi atau keterlibatan *"user"* sebagai pengguna komputer. Ditinjau dari cara kerjanya, virus dapat dikelompokkan menjadi:

1) *Overwriting Virus* – merupakan penggalan program yang dibuat sedemikian rupa untuk menggantikan program utama (baca: host) dari sebuah program besar sehingga menjalankan perintah yang tidak semestinya;
2) *Prepending Virus* – merupakan tambahan program yang disisipkan pada bagian awal dari program utama atau "host" sehingga pada saat dieksekusi, program virus akan dijalankan terlebih (bereplikasi) dahulu sebelum program yang sebenarnya;
3) *Appending Virus* – merupakan program tambahan yang disisipkan pada bagian akhir dari program host sehingga akan dijalankan setelah program sebenarnya tereksekusi;
4) *File Infector Virus* – merupakan penggalan program yang mampu memiliki kemampuan untuk melekatkan diri (baca: attached) pada sebuah file lain, yang biasanya merupakan file "executable", sehingga sistem yang menjalankan file tersebut akan langsung terinfeksi;
5) *Boot Sector Virus* – merupakan program yang bekerja memodifikasi program yang berada di dalam boot sector pada cakram penyimpan (baca: disc) atau disket yang telah diformat. Pada umumnya, sebuah *boot sector virus* akan terlebih dahulu mengeksekusi dirinya sendiri sebelum proses "*boot - up*" pada komputer terjadi, sehingga seluruh "*floppy disk*" yang digunakan pada komputer tersebut akan terjangkiti pula (perhatikan bahwa dewasa ini, modus operandi sejenis terjadi dengan memanfaatkan media penyimpan USB);
6) Multipartite Virus – merupakan kombinasi dari Infector Virus dan Boot Sector Virus dalam arti kata ketika sebuah file yang terinfeksi oleh virus jenis ini dieksekusi, maka virus akan menjangkiti boot sector dari hard disk atau partition sector dari komputer tersebut, dan sebaliknya; dan
7) Macro Virus menjangkiti program "macro" dari sebuah file data atau dokumen (yang biasanya digunakan untuk "global setting" seperti pada template Microsoft Word) sehingga dokumen berikutnya yang di edit oleh program aplikasi tersebut akan terinfeksi pula oleh penggalan program macro yang telah terinfeksi sebelumnya.

2) *Worms*

Worms merupakan program yang dibangun dengan algoritma tertentu sehingga yang bersangkutan mampu untuk mereplikasikan dirinya sendiri pada sebuah jaringan komputer tanpa melalui intervensi atau bantuan maupun keterlibatan pengguna.

3) *Trojan Horse*

Trojan Horse merupakan program *malicious* yang dimasukkan ke dalam sistem melalui sebuah program atau aktivitas yang legal – seperti: melalui proses instalasi perangkat lunak aplikasi, melalui proses "*upgrading*" versi *software* yang baru, melalui proses "*download*" program - program *freeware*, melalui *file - file* multimedia (seperti gambar, lagu, dan video), dan lain sebagainya.

Terdapat beberapa jenis *Trojan Horse,* antara lain:

p-ISSN : 2301-8402, e-ISSN : 2685-368X ,dapat diakses melalui https://ejournal.unsrat.ac.id/index.php/elekdankom

1) *Remote Access Trojan* - kerugian yang ditimbulkan adalah komputer korban serangan dapat diakses secara remote;
2) *Password Sending Trojan* - kerugian yang ditimbulkan adalah password yang diketik oleh komputer korban akan dikirimkan melalui email tanpa sepengetahuan dari korban serangan;
3) *Keylogger* - kerugian yang ditimbulkan adalah ketikan atau input melalui keyboard akan dicatat dan dikirimkan via email kepada hacker yang memasang keylogger;
4) *Destructive Trojan* – kerugian yang ditimbulkan adalah file - file yang terhapus atau hard disk yang terformat;
5) *FTP Trojan* – kerugian yang terjadi adalah dibukanya port 21 dalam sistem komputer tempat dilakukannya download dan upload file;
6) *Software Detection Killer* – kerugiannya dapat program - program keamanan seperti zone alarm, anti - virus, dan aplikasi keamanan lainnya; dan
7) *Proxy Trojan* – kerugian yang ditimbulkan adalah di "settingnya" komputer korban menjadi "proxy server" agar digunakan untuk melakukan "anonymous telnet", sehingga dimungkinkan dilakukan aktivitas belanja online dengan kartu kredit curian dimana yang terlacak nantinya adalah komputer korban, bukan komputer pelaku kejahatan

B. *Model Analisa Malware*

Pada dasarnya *malware* adalah sebuah program, yang disusun berdasarkan tujuan tertentu dengan menggunakan logika dan algoritma yang relevan dengannya. Oleh karena itulah maka model analisa yang biasa dipergunakan untuk mengkaji *malware* sangat erat kaitannya dengan ilmu dasar komputer, yaitu: bahasa pemrograman, algoritma, struktur data, dan rekayasa piranti lunak.

Secara umum, ada 3 (tiga) jenis analisa terhadap sebuah program untuk mendeteksi apakah yang bersangkutan merupakan *malware* atau bukan. Ketiga pendekatan dimaksud akan dijelaskan dalam masing - masing paparan sebagai berikut [6]:

1) *Surface Analysis*

Sesuai dengan namanya, "*surface analysis*" adalah suatu kajian pendeteksian *malware* dengan mengamati sekilas ciri - ciri khas sebuah file program tanpa harus mengeksekusinya. Untuk melihat ciri khas tersebut dapat dilakukan dengan menggunakan bantuan *software* atau perangkat aplikasi pendudukung. Analisa ini memiliki ciri – ciri sebagai berikut:

1) Program yang dikaji tidak akan dijalankan, hanya akan dilihat "bagian luarnya" saja (sebagai analogi selayaknya orang yang ingin membeli buah - buahan, untuk mengetahui apakah buah yang bersangkutan masih mentah atau sudah busuk cukup dengan melihat permukaan kulitnya, membauinya, dan meraba - raba tekstur atau struktur kulitnya). Dari sini akan dicoba ditemukan hal - hal yang patut untuk dicurigai karena berbeda dengan ciri khas program kebanyakan yang serupa dengannya;
2) Sang pengkaji tidak mencoba untuk mempelajari "source code" program yang bersangkutan untuk mempelajari algoritma maupun struktur datanya (sebagaimana layaknya melihat sebuah kotak hitam atau "black box".

2) *Runtime Analysis*

Pada dasarnya ada kesamaan antara *runtime analysis* dengan *surface analysis,* yaitu keduanya sama - sama berada dalam ranah mempelajari ciri - ciri khas yang selayaknya ada pada sebuah program yang normal. Bedanya adalah bahwa dalam *runtime analysis,* dipersiapkan sebuah prosedur dan lingkungan untuk mengeksekusi atau menjalankan program yang dicurigai mengandung atau sebagai *malware* tersebut.

Model analisa ini menghasilkan kajian yang lebih mendalam karena selain dihilangkannya proses "menduga - duga", dengan mengeksekusi *malware* dimaksud akan dapat dilihat "perilaku" dari program dalam menjalankan "skenario jahatnya" sehingga selanjutnya dapat dilakukan analisa dampak terhadap sistem yang ada.

3) *Static Analysis*

Dari ketiga metode yang ada, *static analysis* merupakan model kajian yang paling sulit dilakukan karena sifat analisanya yang *"white box"* alias pengkajian melibatkan proses melihat dan mempelajari isi serta algoritma program *malware* dimaksud, sambil mengamati sekaligus menjalankan/mengeksekusinya.

Karena sifat dan ruang lingkupnya yang cukup luas dan mendalam, strategi khusus perlu dipersiapkan untuk melakukan kajian ini. Disamping itu, kajian ini juga memerlukan sumber daya yang khusus – misalnya adalah SDM yang memiliki pengetahuan dan pengalaman dalam membuat serta membaca program berbahasa mesin atau rakitan *(assembly language)* serta ahli arsitektur dan organisasi piranti komputasi seperti komputer, PDA, *tablet, mobile phone,* dan lain sebagainya.

4) *Dynamic Analysis*

Analisis Dinamis dilakukan dengan menjalankan sampel *malware* pada sebuah ruang lingkup yang dikontrol dan dimonitor selama ia berjalan. Pada beberapa kasus, analisis statis tidak menampilkan informasi yang banyak dikarenakan *obfuscation*, dan *packing*. Pada kasus seperti ini, Analisis Dinamis adalah cara terbaik untuk mengidentifikasi fungsionalitas *malware*(Firana et al., 2016)

C. *Cuckoo Sandbox*

Cuckoo Sandbox merupakan *tool* analisa *malware* dan dapat memberikan beberapa informasi mengenai *malware* yang sedang berjalan dalam lingkungan yang terisolasi [7].

Hal yang dapat dilakukan *Cuckoo sandbox* adalah sebagai berikut:

1) *Native functions* dan *Windows API Calls Trace* yang dapat mencatat setiap eksekusi kode dari suatu file yang diupload ke dalam *Cuckoo Sandbox*.
2) Melakukan pencatatan pada setiap *file* yang dibuat atau dihapus dari sistem.
3) *Memory dump* dari hasil analisis *malware*.
4) Jejak aliran jaringan dalam format PCAP.
5) *Screenshot desktop* selama aktivitas analisis *malware* berlangsung.
6) *Full memory dump* dari mesin VM.

Cuckoo Sandbox dapat melakukan analisa terhadap berbagai macam sampel *malware*. Beberapa jenis file yang dapat dianalisa dengan *Cuckoo* adalah sebagai berikut. File .exe *(Generic Windows Executable), File DLL,* Dokumen PDF, Dokumen *Microsoft Office,* Halaman *Web* dan *Script PHP*.

Setelah dijalankan, *Cuckoo* melaporkan hasil analisa *malware*. Hasil analisa yang dilaporkan antara lain, *API calls* yang dijalankan *malware, File* yang dibuat, *File* yang dihapus, *File* yang diunduh *malware,* Aktivitas *malware* di *memory (memory dumps),* Trafik jaringan yang diakses *malware* (dalam format PCAP.

D. Penelitian Terkait

Analisa Malware Trojan. Sulhaedir. Teknik Informatika (2016). Penelitian ini membahas tentang analisa *malware trojan* menggunakan analisa statis dan dinamis. Hubungan dengan peneltian ini adalah kesepahaman dalam konsep metode *static analisys* untuk melakukan analisa *malware*, sedangkan perbedaan dengan penelitian ini adalah aplikasi yang terduga *malware* telah diketahui karakteristik *malware* trojan pada penelitian saya belum diketahui.

Analisis *Malware* Dengan Teknik *Static Analysis* (Irman Hariman,Azhar,Syams). Penelitian ini membahas tentang analisa *malware worm* menggunakan metode *static analysis* dan cara mencegah agar *malware* tidak berkembang biak dan pembuatan aplikasi *antimalware*. Hubungan dengan penelitian ini adalah kesepahaman konsep tentang metode *static analysis*, sedangkan perbedaan dengan penelitian ini adalah aplikasi yang terduga *malware* telah diketahui karakteristik *malware worm* dan pembuatan aplikasi *antimalware* pada penelitian saya belum diketahui.

Analisis dan Deteksi *Malware* Menggunakan Metode *Malware* Analisis Dinamis dan *Malware* Analisis Statis (Triawan Adi Cahyanto). Penelitian ini membahas tentang analisa *malware posion ivy* RAT (*Remote Access Trojan*) yang menggunakan metode statis dan dinamis untuk mengetahui *output* dari kedua metode ini sama. Hubungan dengan metode ini adalah kesepahaman tentang konsep *static analysis* yang digunakan untuk mengidentifikasi *malware poison ivy* RAT

Malware Analysis on Windows Operation System to Detect Trojan. Sabam Chandra Yohanes Hutauruk, Fazmah Arif Yulianto, Gandeva Bayu Satrya. Penelitian ini membahas tentang analisa *malware trojan* menggunakan metode statis dan dinamis pada sistem operasi *windows*. Hubangan dengan penelitian ini adalah kesepahaman konsep *static analysis* untuk mengetahui karakteristik dari sebuah *malware*.

Pembangunan *Server Analisis Malware* menggunakan *Cuckoo Sandbox* pada Sistem Operasi berbasis Linux. Rida Firana, Setia Juli Irzal Ismail, Periyadi. Penelitian ini membahas tentang analisa *malware* menggunakan metode dinamis dan mempunyai kesamaan tools untuk melakukan analisa yaitu *Cuckoo Sandbox.*

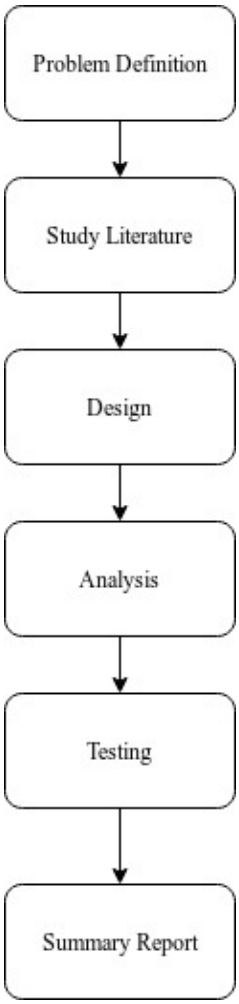
II. METODE

A. Tahapan Penelitian

Pada bagian ini menjelaskan mengenai tahapan penelitian yang digunakan untuk menganalisa *malware* [7].

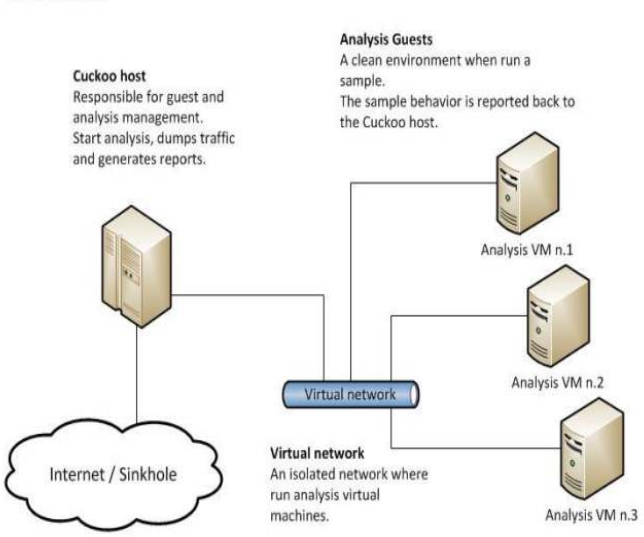
1) *Problem Definition*

Pada tahap ini penulis mengidentifikasi masalah yang dibahas dalam penelitian, latar belakang maupun batasan masalah yang digunakan.



Gambar 2. Tahapan Penelitian

p-ISSN : 2301-8402, e-ISSN : 2685-368X ,dapat diakses melalui https://ejournal.unsrat.ac.id/index.php/elekdankom



Gambar 3. Arsitektur sistem
(Sumber:https://medium.com/@warunikaamali/cuckoo-sandbox-installation-guide-d7a09bd4ee1f)

2) *Study Literature*

Pada tahap ini penulis mempelajari literatur yang berhubungan dengan penelitian, seperti literatur mengenai *malware* dan teknik analisis *malware*, *cuckoo sandbox*. Literatur yang diperoleh berupa dokumentasi penelitian – penelitian sebelumnya seperti jurnal, paper, serta *website* yang terpercaya.

3) *Design*

Pada tahap ini penulis melakukan perancangan sistem yang meliputi tahapan penelitian dan lingkungan yang akan digunakan untuk proses analisa *malware*.

4) *Analysis*

Pada tahap ini penulis melakukan analisis terhadap sampel *file malware*, *tools*, Data yang digunakan adalah data sampel *malware* yang diambil pada *website* universitas sam ratulangi. Tools yang dipakai adalah *Cuckoo Sandbox* untuk proses analisa dinamis.

5) *Testing*

Pada tahap ini penulis mengeksekusi *malware* menggunakan *tool cuckoo sandbox* dan mengamati segala aktvitas yang dilakukan oleh *malware*.

6) *Summary Report*

Pada tahap ini penulis mengevaluasi keseluruhan tahapan penelitian dan mendokumentasikan dalam bentuk laporan.

B. Rencana Pengerjaan.

Hal yang akan dilakukan untuk membangun lingkungan isolasi (*virtual machine*) diantaranya :

1) Instalasi sistem operasi linux Mint 19.3 pada *server host.*

2) Instalasi *software* yang dibutuhkan pada *host* seperti *package*. *dependencies* dan *libraries* untuk *cuckoo sandbox, cuckoo sandbox* 2.0.7 dan *Virtualbox* 6.0.

3) Pembuatan mesin Virtual yang berperan sebagai *guest* atau ruang lingkup pengujian sampel *malware*. Sistem operasi yang digunakan sebagai *guest* adalah *Windows* 7 SP3 32bit.

Gambar 4. Proses Konfigurasi *Virtualbox*

4) Pembuatan *folder sharing* antara *server host* dan mesin virtual.

5) Konfigurasi *server host* meliputi konfigurasi *file cuckoo sandbox* (versi 2.0.7), konfigurasi *Virtualbox* dan konfigurasi *iptables server host*. Pengaturan *iptables* dibuat untuk mengatur IP *Forwarding* dan *filtering* mesin virtual.

6) Konfigurasi mesin virtual meliputi instalasi aplikasi python, python imaging Library, Agent.py, dan aplikasi lain yang biasa dipakai pada personal komputer asli.

C. Arsitektur Sistem

Desain sistem yang digunakan dapat dilihat pada gambar 3.

Arsitektur *Cuckoo Sandbox* terdiri dari sebuah pusat manajemen perangkat lunak yang menangani eksekusi sampel *malware* dan analisis *malware*. Tiap analisis dijalankan di dalam sebuah mesin virtual yang terisolasi (dari jaringan luar). Infrastruktur *cuckoo* tersusun dari sebuah mesin *host* dan beberapa mesin *guest*.

Mesin host menjalankan komponen inti dari *sandbox* yang mengelola seluruh proses analisis, sedangkan mesin *guest* adalah lingkungan yang terisolasi dimana sampel *malware* dengan aman untuk proses analisis

III. HASIL DAN PEMBAHASAN

A. Implementasi

Pada tahap ini menjelaskan tentang pembangunan *server* analisa *malware* meliputi konfigurasi *Virtualbox* 6.0, instalasi *Cuckoo Sandbox* 2.0.7, *package dependencies*, mesin virtual dan konfigurasi seluruh sistem.

1) Konfigurasi Virtualbox

Untuk melakukan analisa *malware* dengan metode dinamik kita perlu lingkungan yang aman (terisolasi) dari komputer *host* dimana *malware* akan dijalankan. Didalam *virtualbox* kita akan membuat mesin virtual atau *guest* dengan sistem operasi *windows* 7 sp1.

2) *Instalasi Package Dependencies dan Cuckoo Sandbox.*

Cuckoo Sandbox merupakan aplikasi yang sangat modular sehingga membutuhkan *package* dan *libraries* untuk menjalankan setiap syntax-nya. Instalasi dilakukan via *Terminal* Linux Mint.

3) *Konfigurasi Cuckoo Sandbox*

Setelah semua ter-install perlu melakukan konfigurasi pada aplikasi *cuckoo sandbox* agar bisa saling terkoneksi. Konfigurasi dilakukan via *terminal* linux.

B. Pengujian

Pada tahap ini kita mulai menguji *cuckoo sandbox* dan file malware untuk proses analisa dinamik dan melihat karakteristik dari malware tersebut.

```
[virtualbox]
# Specify which VirtualBox mode you want to run your machines on.
# Can be "gui" or "headless". Please refer to VirtualBox's official
# documentation to understand the differences.
mode = headless

# Path to the local installation of the VBoxManage utility.
path = /usr/bin/VBoxManage
# If you are running Cuckoo on Mac OS X you have to change the path as follows:
# path = /Applications/VirtualBox.app/Contents/MacOS/VBoxManage

# Default network interface.
interface = vboxnet0

# Specify a comma-separated list of available machines to be used. For each
# specified ID you have to define a dedicated section containing the details
# on the respective machine. (E.g. cuckoo1,cuckoo2,cuckoo3)
machines = Cuckoo

# If remote control is enabled in cuckoo.conf, specify a port range to use.
# Virtualbox will bind the VRDP interface to the first available port.
controlports = 5000-5050


[Cuckoo]
# Specify the label name of the current machine as specified in your
# VirtualBox configuration.
label = Cuckoo
```

Gambar 5. Konfigurasi Cuckoo Sandbox ke *Virtualbox*

File *SizMLX*

Summary	Download / Resubmit sample
Size	6.9MB
Type	ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, stripped
MD5	5447cf2d4b1f61ed74d254c48461a08a
SHA1	5dbd07809d224103d87518521449f6344f04c38c
SHA256	0bf7c36ec904a63d6a567f19998f34c321e1905b4c5548731e9cc2886b236703
SHA512	Show SHA512 efba09307ec8b95c693004dc7186bae151a46fcdb8495c3620e5383fd2b351de94a55635524219e77816625b6169be7a501a4fca9325d0c7b9a7f74a87a6a9f8
CRC32	912DED40
ssdeep	None
Yara	• embedded_pe - Contains an embedded PE32 file • embedded_win_api - A non-Windows executable contains win32 API functions names

Gambar 6. Informasi *File Malware via cuckoo web*

Gambar 7, Merupakan proses menjalankan *Cuckoo Sandbox* via *terminal* Linux agar bisa di akse menggunakan browser.

Gambar 8, Merupakan analisa *Cuckoo Sandbox* menampilkan sampel *malware* yang menjalankan *file wscript.exe* yang

Gambar 9, Merupakan tampilan utama *Cuckoo Sandbox* digunakan untuk mengupload sampel *malware,* dimana ada beberapa fitur pada tampilan awal tersebut seperti *system information, system info, usage statistic.*

Gambar 10, Meruapakan tampilan ketika sampel *malware* telah dimasukkan/di *upload* kedalam *Cuckoo Sandbox.* Setelah itu proses dilakukan secara otomatis dan akan ditampilkan hasilnya seperti *strings, binary code,* hingga *registry value.*

Gambar 11,Merupakan analisa *Cuckoo Sandbox* yang menampilkan hasil *strings.*

Gambar 12, Merupakan tampilan dimana sampel *malware* mengakses data *registry* hingga melakukan duplikasi diri.

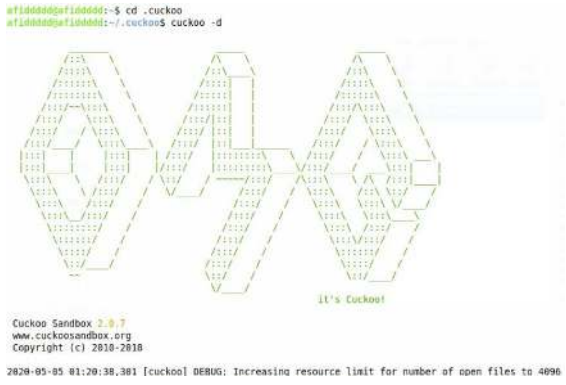
Gambar 13, Merupakan tampilan dimana *malware* mengakses diri sendiri. Proses akses diri sendiri seperti ini dapat mengakibatkan *resource memory* yang menjadi lebih besar

Gambar 14, Merupakan tampilan dimana *malware* melakukan perubahan *value registry* :

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers\Levels

Gambar 15, Merupakan tampilan dimana *malware* melakukan akses ke *registry :*

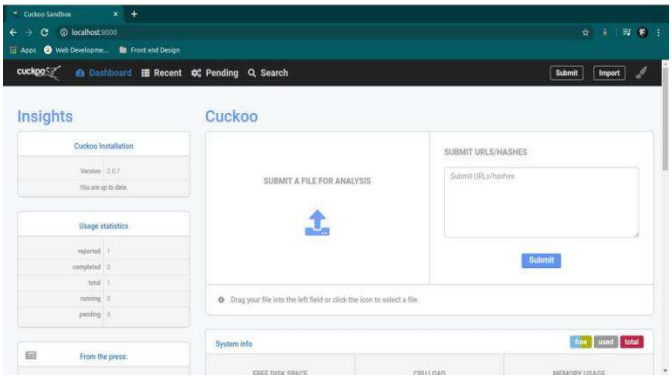
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers\Levels



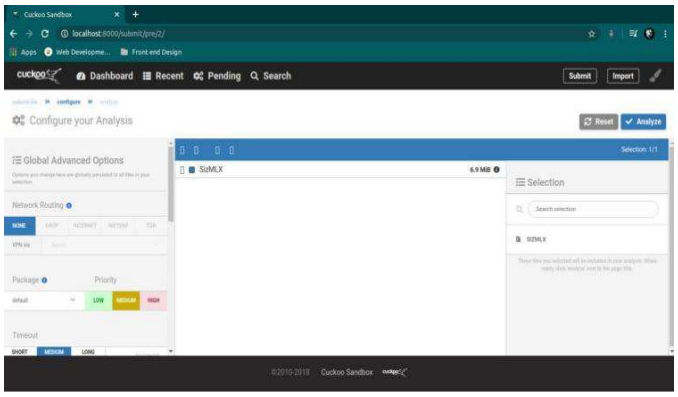
Gambar 7. Proses Menjalankan *Cuckoo Sandbox*

Gambar 8. Sampel *Malware* Menjalankan *wscript.exe*

p-ISSN : 2301-8402, e-ISSN : 2685-368X ,dapat diakses melalui https://ejournal.unsrat.ac.id/index.php/elekdankom



Gambar 9. Halaman Utama *Cuckoo Sandbox*



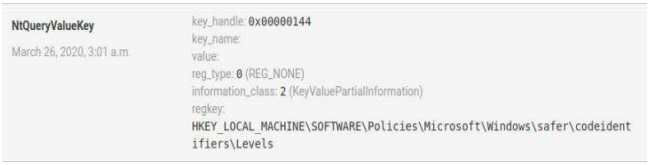
Gambar 10. Proses *Upload* Sampel *Malware* ke *Cuckoo Sandbox*

```
MoYxxE1lw1fXwbYVoYL-/Y7K8RnoD_Y5azr-Ujwqv/qoUaxXZH0mAD2v_3m_OC/M9vODtAusDYIWS2Xellb
singu?
runtu/
tX;ChuY
|$$9;u
|$D9;u
2006u1
2006u5
=WITAt
|$H9;u
|$L9;u
|$09;u
|$89;u
|$09;u
|$$9;u
|$$9;u
|$09;u
|$(9;u
|$,9;u
|$ 9;u
|$D9;u
|$@9;u
|$,9;u
|$,9;u
|$,9;u
|$H9;u
```

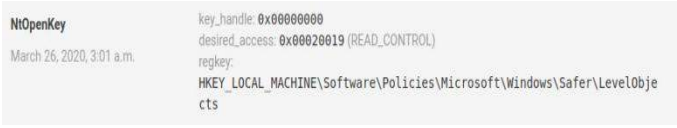
Gambar 11. *Strings* dari sampel *malware*

Gambar 12. Sampel Melakukan Duplikat

Gambar 13. Sampel Mengakses Diri Sendiri



Gambar 14. Sampel Melakukan Perubahan *Value Registry*



Gambar 15. Sampel Mengakses *Registry*

TABEL I *SIGNATURES*

Engine	Mesin Virtual	Sampel	Signatures
Cuckoo Sandbox	*Windows 7 SP1*	A-W	1) Membaca data *image binary* milik dirinya sendiri 2) Melakukan beberapa *request UDP* 3) Menambahkan beberapa *Value* pada beberapa *registry* 4) Teridentifikasi sebagai *malware* oleh 35 dari 57 *Antivirus* pada *VirusTotal* 5) Terdapat anomali – anomali karakteristik *biner*

TABEL II *SUMMARY SIGNATURES*

Malscore	Kategori	Virustotal Ratio	Definisi Virustotal Ratio	Jenis Malscore
37	Malicious	35/37	67% Terdeteksi sebagai *malware*	*Trojan*

IV. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan penelitian ynag sudah dilakukan untuk menganalisis karakteristik dari sampel *malware* yang diperoleh dari jaringan universitas sam ratulangi maka kesimpulan dari tugas akhir ini adalah :

Metode *dynamic analysis* dapat mempermudah kita untuk analisa *malware* karena kita menjalankan *malware* tersebut dilingkungan yang terisolasi.

Hasil analisa *malware* menggunakan *tool cuckoo sandbox* yang didapatkan adalah informasi *malware*, karakteristik dari *malware*, *behaviour analysis*, *static analysis* dan tingkat *maliciousness malware* berdasarkan hasil yang di dapat dari *VirusTotal*.

B. *Saran*

Untuk penelitian berikutnya diperlukan pemahaman analisa *malware* menggunakan metode *static analysis* agar informasi lebih mendalam mengenai karakteristik *malware* bisa didapatkan.

Untuk penelitian berikutnya dapat dilakukan pada *platform* selain *windows* dan linux, misalnya *MacOS* atau *Android*.

V. KUTIPAN

[1] S. Kramer and J. C. Bradfield, "A general definition of malware," *J. Comput. Virol.*, vol. 6, no. 2, pp. 105–114, 2010, doi: 10.1007/s11416-009-0137-1.

[2] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *Justindo, J. Sist. Teknol. Inf. Indones.*, vol. 2, no. 1, pp. 19–30, 2017, [Online]. Available: http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/1037.

[3] I. Hariman and A. Syams, "Analisis Malware Dengan Teknik Static Analysis," 2015.

[4] S. Chandra, Y. Hutauruk, F. A. Yulianto, and G. B. Satrya, "Malware Analysis Pada Windows Operating System Untuk Mendeteksi Trojan," *e-Proceeding Eng.*, vol. 3, no. 2, pp. 3590–3595, 2016.

[5] Amali, W. "*Cuckoo Sandbox Installation Guide"* Medium, 9 Juli 2017. [Online]. Available: https://medium.com/@warunikaamali/cuckoo-sandbox-installation-guide-d7a09bd4ee1f

[6] Indrajit, R. E. 2011. *"Pengantar Konsep Keamanan Informasi di Dunia Siber"* APTIKOM. Jakarta.

[7] Sibarani, F. R., Sibuea, C. A., Situmorang, P. A. 2019 "*Pendekteksian Malware Dengan Menggunakan Analisis Dinamik Terhadap Aktivitas Jaringan".* http://ri.del.ac.id:8080/xmlui/handle/123456789/429

[8] Novrianda, R., Kunang, N. Y., Shaksono, P. H. 2014. "*Analisis Forensik Malware Pada Platform Android".* Referensi: http://eprints.binadarma.ac.id/2192/

[9] Sulhaedir. 2016. "*Analisa Malware Trojan".* Yogyakarta: STIMIK AKAKOM.

[10] Firana, R., Ismail, S. J. I., Periyadi. 2016. *"Pembangunan Server Analisis Malware Menggunakan Cuckoo Sandbox Pada Sistem Operasi Berbasis Linux".* Bandung: Universitas Telkom.

Virigiawan Arshad Manoppo, lahir di Kotamobagu pada tanggal 12 Agustus 1997 dari pasangan Rusli Manoppo dan Mira Liana Mokodompit.Penulis merupakan anak kedua dari tiga bersaudara. Penulis sekarang bertempat tinggal di Desa Kosio Timur, Kecamatan Dumoga Barat, Kabupaten Bolaang Mongondow.

Penulis menyelesaikan pendidikan Sekolah Dasar di SDN 2 Kosio pada tahun 2009, kemudian melanjutkan pendidikan di SMP 2 Dumoga lulus pada tahun 2012, dan pendidikan Sekolah Menengah Atas di SMK Cokroaminoto Kotamobagu pada tahun 2015. Setelah lulus SMA, penulis melanjutkan pendidikan di salah satu perguruan tinggi Manado yaitu Universitas Sam Ratulangi dengan mengambil Program Studi Teknik Informatika Jurusan Teknik Elektro dan menyelesaikan studi S1 pada tahun 2020. Pada bulan februari 2019 Penulis mengajukan proposal skripsi untuk memenuhi syarat meraih gelar sarjana (S1) dengan judul Analisa *Malware* Menggunakan Metode *Dynamic Analysis* Pada Jaringan Universitas Sam Ratulangi yang kemudian disetujui dan dibimbing oleh dua dosen pembimbing, yaitu Arie S. M. Lumenta, ST, MT dan Stanley D. S. Karouw, ST, MTI. Pada 30 Juni 2020, Penulis resmi menyelesaikan skripsi dengan menyandang gelar sarjana komputer.