Analisis *Malware* Menggunakan Metode Analisis Statis dan Dinamis untuk Pembuatan IOC Berdasarkan STIX Versi 2.1

Yunike Dwi Puji Rahayu[1)], Nanang Trianto[2)]
(1) Rekayasa Keamanan Siber, Badan Siber dan Sandi Negara, yunike.dwi@bssn.go.id
(2) *Rekayasa Keamanan Siber, Badan Siber dan Sandi Negara, nanang.trianto@bssn.ac.id*

Abstrak

Malware atau malicious software adalah kode atau program berbahaya yang dapat menyebabkan kerugian bagi individu ataupun organisasi. Kerugian yang disebabkan oleh malware dapat berupa kerugian finansial maupun material. Pencegahan terhadap insiden malware dapat dilakukan dengan analisis pada malware untuk mengetahui cara kerja dan karakteristik dari malware tersebut. Analisis malware dapat dilakukan dengan dua metode yaitu analisis statis dan analisis dinamis. Menggabungkan kedua metode analisis ini dapat memberikan informasi dan hasil yang lebih lengkap. Informasi yang diperoleh berupa karakteristik dan indikator identik yang menunjukkan adanya keberadaan malware tersebut dalam sistem atau komputer. Selanjutnya, informasi tersebut dapat dimanfaatkan dan didefinisikan pada sebuah Indicator of Compromise (IOC). IOC adalah suatu kumpulan informasi yang dapat digunakan untuk mengidentifikasi sistem atau komputer yang terinfeksi oleh malware. IOC ini disimpan pada suatu sistem Cyber Threat Intelligence (CTI) untuk digunakan sebagai sumber informasi CTI dalam mendeteksi keberadaan dari malware di masa yang akan datang. Pada penelitian ini, dilakukan analisis malware menggunakan metode analisis statis dan dinamis pada 5 spesimen malware. Spesimen malware diperoleh dari sensor Honeynet milik Badan Siber dan Sandi Negara (BSSN) yang dipilih secara acak. Berdasarkan hasil analisis, diketahui bahwa kelima malware tersebut berjenis trojan yang beraktivitas pada latar belakang sistem dan menghubungi beberapa domain berbahaya untuk mengunduh program atau file berbahaya. Indikator dari masing-masing malware selanjutnya didefinisikan ke dalam IOC dan telah divalidasi sehingga dapat disimpan dan digunakan sebagai sumber informasi sistem CTI.

Kata kunci: analisis dinamis (1), analisis *malware* (2), analisis statis (3), IOC (4), *malware* (5)

Abstract

Malware or malicious software is dangerous code or program that causes loss for individual or organization. Loss caused by malware may be financially or materially. Prevention of malware incident can be done with analysis towards malware to figure out the way it works or its characteristics. Malware analysis can be done using two methods which are static and dynamic analysis. Combining these two methods can give thorough information and results. Obtained information may be characteristics and identical indicators that show that malware's existence within the system or computer. Then, the information may be used and defined into a Indicator Of Compromise (IOC). IOC is a set of information that can be used to identify malware infected systems or computers. IOC is stored in a Cyber Threat Intelligence (CTI) system to be used as the source of cti information in detecting the existence of malware in the future. In this research, is done a malware analysis using static and dynamic analysis method to 5 malware specimens which are obtained from honeynet sensor belonged to Badan Siber dan Sandi Negara (BSSN). Based on analysis result, it is known that those 5 malwares are trojans that act on system background and contact some dangerous domains to download malicious programs or files. Indicators of each malware are then defined into IOC and validated so they can be stored and used as information source for CTI system.

Keywords: dynamic analysis (1), IOC (2), malware (3), malware analysis (4), static analysis (5)

1. PENDAHULUAN

Budaya internet atau *cyberculture* adalah budaya yang muncul akibat penggunaan jaringan komputer untuk komunikasi, hiburan dan bisnis. Munculnya budaya ini menjadikan masyarakat ketergantungan terhadap internet [1]. Dikutip dari Laporan *We Are Social and Hootsuite Global Digital Report* 2019, sejak Januari 2018 pengguna internet telah mengalami peningkatan sebanyak 366 juta hingga mencapai 4,39 miliar pada tahun 2019 [2]. Seiring perkembangan budaya internet tersebut maka peluang kejahatan dalam dunia siber akan semakin besar. Dalam Laporan Tahunan ID-SIRTII *Indonesia Cyber Security Monitoring Report* 2018 tercatat Indonesia mengalami 232.447.974 serangan siber, dengan 122.435.215 merupakan aktivitas *malware* [3]. *Malicious software* atau *malware* merupakan segala perangkat lunak yang membahayakan pengguna, komputer ataupun jaringan. *Malware* terbagi menjadi beberapa jenis yaitu *trojan horses*, *worms*, *rootkits*, *scareware* dan *spyware* [4].

Berdasarkan Peraturan Presiden Nomor 53 Tahun 2017, Badan Siber dan Sandi Negara atau BSSN mempunyai tugas melaksanakan keamanan siber secara efektif dan efisien dengan memanfaatkan, mengembangkan, dan mengonsolidasikan semua unsur terkait dengan keamanan siber [5]. Berdasarkan peraturan tersebut, dalam pelaksanaan tugasnya, BSSN melakukan beberapa kegiatan yaitu deteksi, pemantauan, penanggulangan, pemulihan, dan evaluasi atas insiden atau serangan siber. Kegiatan

tersebut diselenggarakan oleh kedeputian dan unit kerja BSSN. Mendeteksi keberadaan *malware* dapat melalui beberapa cara seperti mengunakan *Indicator of Compromise* (IOC), IDS/IPS, *Firewall* atau antivirus. IOC adalah suatu sekumpulan informasi serangan atau insiden siber yang ditulis berdasarkan format atau bahasa tertentu [6]. Informasi yang dituangkan dalam suatu IOC dapat diperoleh melalui kegiatan analisis *malware*. Oleh karena itu, kegiatan analisis *malware* secara efektif dapat membantu organisasi dalam hal tindakan pencegahan dan penanganan terhadap serangan siber khususnya serangan *malware* [7].

Analisis *malware* merupakan kegiatan memeriksa kode berbahaya dalam rangka mengetahui karakteristik *malware* untuk tindakan merespon dan pencegahan insiden [8]. Terdapat dua teknik analisis *malware* yaitu analisis statis dan analisis dinamis dengan dua pendekatan berbeda yaitu pendekatan berbasis *host* dan berbasis jaringan [7]. Pada tahun 2015, [9] melakukan analisis statis dan analisis dinamis sampel *malware* TT.exe. Pada penelitian tersebut, sampel *malware* dieksekusi pada suatu lingkungan virtual sistem operasi Windows. Analisis *malware* menggunakan beberapa aplikasi perangkat lunak untuk mengetahui perubahan registry, aktivitas *malware* di jaringan, dan metadata *malware*. Hasil dari penelitian tersebut menyebutkan bahwasanya dengan teknik analisis tersebut diperoleh informasi yang lebih lengkap terkait sampel *malware*. Pada tahun 2016, [10] melakukan penelitian terhadap aktivitas *malware* pada sistem komputer yang terinfeksi melalui Windows Events Logs, yang selanjutnya dituangkan ke dalam IOC.

Pada penelitian ini, dilakukan kajian terhadap tiga penelitian yang berkaitan dengan analisis malware dan diperoleh perbedaan tahapan dari masing-masing metode yang digunakan. Perbedaan metode analisis terletak pada penerapan analisis statis dan analisis dinamis yang digunakan. Selanjutnya, dilakukan penyusunan metode yang mengkombinasikan tahapan analisis pada tiga penelitian tersebut untuk memperoleh metode analisis yang komprehensif. Metode analisis *malware* yang disusun diharapkan dapat menghasilkan informasi yang lengkap untuk digunakan sebagai bahan pembuatan IoC STIX. IoC STIX dipilih karena merupakan *tools opensource* yang telah banyak digunakan serta mudah untuk dikonversi ke dalam bentuk format lain. Selain itu, IoC STIX juga memiliki objek yang lengkap untuk mendefinisikan informasi indikator malware.

2. LANDASAN TEORI

Bagian ini memuat teori-teori dan penelitian terkait yang digunakan sebagai landasan teori serta acuan selama pelaksanaan penelitian. Teori tersebut diperoleh dengan melakukan kajian terhadap beberapa paper dan dokumen-dokumen yang berkaitan dengan analisis *malware*, IOC dan STIX.

2.1. Analisis *Malware*

Malware dibagi menjadi beberapa macam seperti virus, trojan horse, worm, rootkit, *scareware* dan *spyware* [11]. *Malware* tidak selalu dapat diklasifikasikan berdasarkan fungsinya karena satu *malware* dapat memiliki beberapa fungsi yang bergantung pada tujuan dari pembuatannya [4]. Fungsi dan perilaku dari *malware* dapat diketahui dengan melakukan analisis pada *malware* menggunakan metode analisis statis dan analisis dinamis. Tabel 1 merupakan gambaran tentang perbedaan metode analisis malware dari 3 referensi.

Analisis *malware* adalah sebuah proses mengekstraksi informasi dari suatu *malware* melalui inspeksi statis dan dinamis menggunakan bantuan perangkat lunak, teknik dan proses [12]. Dengan melakukan kegiatan analisis *malware*, analis akan mengetahui cara kerja suatu *malware*, mencegah, menangani dan menyingkirkan *malware* pada sistem komputer yang terinfeksi [11]. Pada umumnya terdapat dua metode analisis yaitu analisis dinamis dan analisis statis, namun apabila dilihat dari tingkat kompleksitasnya terdapat 4 tingkatan analisis *malware* yaitu *manual code reversing*, *interactive behaviour analysis*, *static properties analysis*, dan *automated analysis* [13]. Dalam pelaksanaannya, dibuat lingkungan khusus yang digunakan untuk proses analisis. Lingkungan khusus ini dapat berbentuk perangkat fisik ataupun mesin virtual. Beberapa perangkat lunak yang dapat digunakan adalah VMWare, Parallels, Xen, dan Microsoft Virtual PC [14].

Tabel 1 Perbandingan metode analisis malware

No.	Tahapan	Joseph Peppers [14]	Ligh et.al [13]	Rudman dan Irwin [6]
1.	Pembuatan lingkungan analisis *malware*	√	√	-
2.	Menjalankan *malware* pada mesin target	√	√	-
3.	Menjalankan *malware* pada lingkungan analisis dinamis otomatis	-	-	√
4.	Menangkap lalu lintas jaringan	√	√	-
5.	Pengumpulan data dan analisis	√	√	√

Pada Tabel 1 dapat dilihat bahwa Joseph Peppers melakukan analisis *malware* dengan menyiapkan lingkungan analisis, kemudian dilanjutkan dengan mengeksekusi *malware* pada mesin yang telah disiapkan dan melakukan pengamatan saat eksekusi *malware* berlangsung. Selanjutnya dilakukan analisis dan pengumpulan data. Metode analisis ini juga digunakan oleh Ligh et.al. Namun pada penelitian [6] yang membedakan adalah lingkungan analisis yang digunakan. Rudman dan Irwin pada penelitiannya menggunakan lingkungan analisis dinamis otomatis yang kemudian hasil dari proses eksekusi tersebut di analisis untuk dijadikan sebagai sumber informasi pembuatan IOC.

2.2. *Structured Threat Information Expression* (STIX)

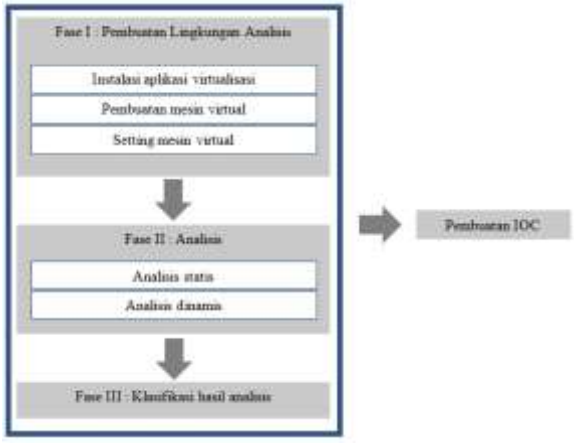
Indicator of Compromise atau IOC adalah suatu informasi yang dapat digunakan untuk mengidentifikasi sistem yang terkompromi. IOC dapat berisi Alamat IP, paket jaringan, nama domain, alamat IP, *file* hash atau *file* mutex yang diduga mencurigakan[6]. *Structured Threat Information Expression* atau STIX merupakan salah satu dari beberapa standar format IOC. STIX adalah sebuah skema yang mendefinisikan taksonomi dari *Cyber Threat Intelligence* (CTI) [15]. STIX tersusun dari beberapa jenis obyek yang dikombinasikan menjadi satu untuk membangun satu buah STIX utuh, antara lain SDO, SCO, dan SRO [14].

STIX *Domain Objects* atau SDO terdiri dari 18 objek yaitu *attack pattern*, *campaign*, *Course of Action*, *grouping*, *Identitiy*, *Indicator*, *Infrastructure*, *Intrusion set*, *Location*, *Malware*, *Malware analysis*, *note*, *Observed data*, *Opinion*, *report*, *Threat actor*, *Tool* dan *Vulnerability*. Masing-masing dari objek tersebut menggunakan konsep yang digunakan pada CTI secara umum. Dari 18 objek tersebut tidak seluruhnya akan digunakan dalam pembuatan STIX CTI karena masing-masing ancaman siber yang dianalisis memiliki karakteristik yang berbeda satu dengan yang lainnya. Pada pembuatannya, disesuaikan dengan karakteristik ancaman siber yang dianalisis.

STIX *Relationship Objects* atau SRO adalah bentuk hubungan dari sesama STIX *Domain Objects* (SDO), sesama STIX *Cyber-observable Objects* (SCO), atau antara suatu SDO dengan SCO yang menunjukkan cara keduanya saling berhubungan. Terdapat dua macam SRO yaitu *generic* SRO dan *Sighting* SRO. *Generic* SRO adalah jenis "*relationship"* yang paling banyak digunakan dalam merepresentasikan hubungan objek STIX. Contohnya antara *SDO Indicator* dan *SDO Malware*, keduanya dapat dihubungkan dengan menggunakan properti "relationship_type" yang berarti bahwa nilai yang ada pada SDO *Indicator* dapat digunakan untuk mendeteksi *malware* yang didefinisikan pada SDO *Malware*. *Sighting* SRO adalah jenis "*relationship*" yang digunakan apabila pihak lain menemukan indikator *malware* yang sama di sistemnya dengan STIX yang diunduh dari TAXI *server* [15].

3. METODE PENELITIAN

Dari hasil telaah kepustakaan yang dilakukan, pada tabel 1 disajikan perbandingan dari metode analisis *malware* yang digunakan pada tiga referensi yang dikaji. Gambar 1 merupakan skema metode penelitian pada penelitian ini. Skema tersebut diperoleh dari hasil kajian paper dengan mengekstraksi setiap langkah penelitian pada paper terkait.



Gambar 1. Skema Metode Penelitian

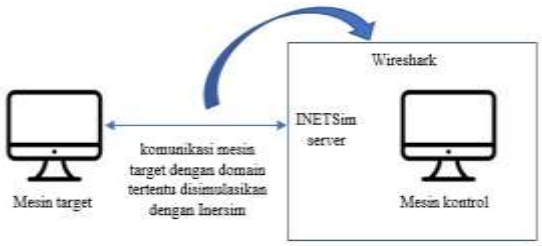
3.1. Fase I: Pembuatan Lingkungan Analisis

Pada fase ini, akan dilakukan pembuatan lingkungan analisis *malware*. Terdapat dua jenis lingkungan analisis *malware* yaitu berupa lingkungan analisis fisik dan lingkungan analisis berbasis virtual. Pada penelitian ini digunakan lingkungan analisis virtual yang terdiri dari dua buah mesin yaitu mesin target dan mesin kontrol.

3.2. Fase II: Analisis

Pada fase 2 ini, akan dilakukan dua metode analisis yaitu analisis statis dan analisis dinamis. Analisis statis adalah langkah awal yang dilakukan pada kegiatan analisis *malware*. Hal-hal yang dapat dilakukan adalah mengidentifikasi target dari *malware*, pemindaian anti-virus, ekstraksi string, fungsi dan metadata *malware*, mengidentifikasi teknik obfuskasi *malware*, serta mengklasifikasikan dan membandingkan sampel *malware*. Pada penelitian ini, analisis statis dilakukan dengan memindai *malware* menggunakan Virustotal untuk mendapatkan informasi awal dari sampel *malware*.

Setelah analisis statis, selanjutnya dilakukan analisis dinamis. Analisis dinamis ini dilakukan dengan mengeksekusi *malware* pada lingkungan analisis yang telah dibuat sebelumnya. Berikut ini adalah ilustrasi skenario analisis dinamis yang dilakukan.



Gambar 2 Skenario analisis dinamis

Gambar 2 merupakan ilustrasi skenario dari analisis dinamis yang akan dilakukan pada penelitian ini.

a. Mengeksekusi *malware*
   Lima buah spesimen *malware* yang diperoleh akan dijalankan pada mesin virtual yang bertindak sebagai mesin target. Mengeksekusi *malware* ini dilakukan dengan tujuan untuk memperoleh informasi tentang *malware* terutama pada aktivitasnya di jaringan. Saat dijalankan, mesin virtual yang bertindak sebagai mesin kontrol akan melakukan *monitoring* dan menangkap seluruh aktivitas yang terjadi.
b. Simulasi jaringan dan penangkapan paket jaringan
   Inetsim atau *Internet Simulation* merupakan aplikasi yang digunakan untuk pencatatan log aktivitas jaringan. Tidak hanya untuk itu, Inetsim dapat digunakan untuk mensimulasikan layanan yang akan dihubungi oleh suatu *malware*. Ketika mengeksekusi *malware* pada komputer target, Inetsim akan melakukan pencatatan log dari setiap aktivitas. Wireshark merupakan aplikasi yang dapat dijalankan pada sistem operasi Windows, Linux, MAC OS dan *platform* lainnya. Aplikasi ini memiliki fungsi untuk menangkap paket jaringan dan dapat digunakan untuk melakukan inspeksi mendalam protokol jaringan dan mengekspor hasilnya dalam bentul pcap *file*, CSV dan XML.

3.3. Fase III: Klasifikasi Hasil Analisis

Hasil *file* pcap dari setiap *malware* pada tahap sebelumnya, dianalisis lebih lanjut pada tahap ini. Analisis ini dilakukan untuk memperoleh indikator jaringan yang menjadi karakteristik dari masingmasing *malware*. Segala informasi yang diperoleh didokumentasikan dan dikumpulkan untuk selanjutnya dipetakan dan disusun menjadi suatu IOC

3.4. Pembuatan IOC dan Validasi

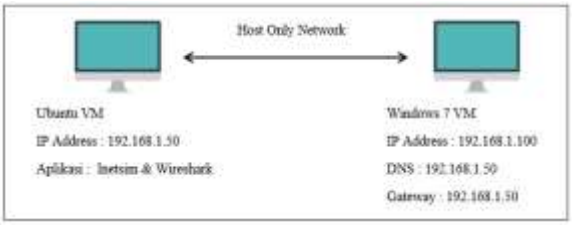
Pembuatan IOC akan dilakukan dengan menggunakan Python STIX API. Python STIX API ini adalah *library* yang menyediakan sebuah API untuk mengembangkan dan membuat konten STIX. Selanjutnya, setelah menghasilkan output berupa STIX dengan bahasa pemrograman json, langkah selanjutnya adalah melakukan validasi menggunakan STIX Visualizer untuk memeriksa validitas dan ketepatan struktur STIX.

4. HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas mengenai proses analisis *malware* yang diawali dari pembuatan lingkungan analisis, menjalankan *malware* pada mesin target, menangkap lalu lintas jaringan dengan menggunakan Wireshark, pengumpulan data dalam bentuk analisis dan ekstraksi informasi dari hasil tangkapan lalu lintas yang selanjutnya informasi tersebut akan digunakan sebagai bahan pembuatan IOC.

4.1. Fase I: Pembuatan Lingkungan Analisis

Pada penelitian ini digunakan Oracle VM Virtualbox untuk pembuatan mesin virtual yang akan digunakan sebagai lingkungan analisis. Dibutuhkan dua mesin virtual yang akan bertindak sebagai mesin target dan mesin kontrol. Pada mesin target akan dipasang sistem operasi Windows 7 Ultimate dan pada mesin kontrol akan dipasang sistem operasi Linux Ubuntu LTS 16.04. Gambar 3 merupakan konfigurasi yang akan diterapkan pada mesin virtual yang dibuat.



Gambar 3. Gambaran konfigurasi lingkungan analisis

Mesin virtual dan mesin target diatur dalam satu *subnet* jaringan untuk menjaga *malware* tidak menyebar dan menginfeksi komputer lain di jaringan yang sama. DNS dan *Gateway* mesin target adalah alamat IP dari mesin kontrol. Pengaturan ini dimaksudkan agar mesin kontrol menjadi pintu dari semua lalu lintas jaringan mesin target sehingga aktivitas tersebut dapat ditangkap oleh aplikasi Wireshark. Mesin target menggunakan sistem operasi Windows 7 Ultimate dengan RAM 2Gb dan alokasi memori sebesar 32 Gb. Alamat IP mesin target adalah 192.168.1.100 dengan alamat mesin kontrol sebagai DNS dan Gateway. Terdapat beberapa aplikasi yang harus dipasang pada mesin target diantaranya adalah WinRar, Mozilla Firefox, HxD, exeinfo, pestudio dan Process Monitor.

Mesin kontrol menggunakan sistem operasi Ubuntu LTS 16.04 dengan beberapa aplikasi yang dibutuhkan untuk penelitian. Alamat IP dari mesin kontrol adalah 192.168.1.50 dengan alokasi RAM sebesar 2Gb dan memori sebesar 10 Gb. Inetsim pada mesin kontrol berguna untuk mensimulasikan protokol jaringan dan Wireshark berguna sebagai penangkap lalu lintas jaringan yang ada selama *malware* dieksekusi pada mesin target. Fitur *snapshot* merupakan fungsi pada virtualbox yang berguna untuk mengembalikan kondisi mesin virtual kembali ke kondisi sebelumnya, dengan begitu pada penelitian

ini hanya dibutuhkan satu buah mesin target yang digunakan untuk mengeksekusi *malware.*

4.2. Fase II : Analisis

Fase kedua pada penelitian ini adalah fase analisis yang terdiri dari dua yaitu analisis statis dan analisis dinamis. Analisis statis dilakukan untuk memperoleh informasi terkait *malware* yang akan dijalankan pada mesin target. Dengan analisis statis akan diperoleh informasi terkait nilai hash dari *malware*, nama lain *malware*, target mesin, dan beberapa karakteristik *malware* lainnya.

a. Analisis statis
Hasil pemindaian menggunakan Virustotal dapat membantu peneliti dalam mengetahui target mesin yang diinfeksi oleh *malware*. Tabel 2 merupakan informasi spesimen *malware* 2 yang diperoleh dari hasil pemindaian menggunakan Virustotal. *Malware* 2 ini lebih dikenal dengan nama "my*file*.exe" dan "wkinstall.exe" dan memiliki *magic number* MZ (*executable file*).

Tabel 2. Hasil analisis statis

Spesimen *malware* 2	
Nama lain	my*file*.exe wkinstall.exe
Target mesin	Intel 386 or later processors and compatible processors
Tipe *file*	Win32 EXE
MD5	741f83e8cec69b1fe32dc48eb147e6cd
SHA-256	28e1b6cf6980b009867b8919f68f41203e6796e1f95c3f82c17cf1c5d8ac5b68
Ukuran *file*	257.35 kb
Titik masuk	0x1cec9
Jenis *malware*	trojan

b. Analisis dinamis
Berdasarkan hasil analisis statis, *malware* 2 termasuk dalam kategori trojan yang biasa digunakan oleh pelaku kejahatan sebagai umpan awal serangan pada perangkat korban. *Malware* trojan ini merupakan *malware* yang beraktivitas dengan melakukan pemasangan dan pengunduhan aplikasi di latar belakang sehingga aktivitas ini tidak diketahui oleh pemiliknya. Terdapat 3 domain yang dihubungi oleh *malware* antara lain www.download.windowsupdate.com , www.teredo.ipv6.microsoft.com dan www.down.ctosus.ru.

4.3. Fase III: Klasifikasi Hasil Analisis

Tabel 3 merupakan tabel yang berisi daftar dari domain yang dihubungi oleh mesin target selama *malware* 2 dijalankan. Domain-domain yang tercatat dalam Tabel 3 sudah dipindai menggunakan aplikasi pemindai online Virustotal dan URLvoid untuk mengetahui reputasinya. Dari 3 domain hanya 1 domain yang tercatat sebagai situs berbahaya.

Tabel 3. Domain yang dihubungi malware

Resolved DNS	**Status domain**
www.download.windowsupdate.com	Tidak berbahaya
teredo.ipv6.microsoft.com	Tidak Berbahaya
down.ctosus.ru	Domain berbahaya

Berdasarkan hasil dari URLvoid, tercatat domain "down.ctosus.ru" dideteksi berbahaya oleh 9 pemindai virus online. Salah satu dari pemindari virus tersebut Bitdefender, melaporkan bahwa pada situs down.ctosus.ru terdapat *malware* yang dapat menyebabkan kerugian apabila host terinfeksi *malware* tersebut.

Tabel 4. Daftar *HTTP Request Object*

Situs Web	**HTTP Objek**	**Jenis Konten**	**Status Objek**
down.ctosus.ru	wget.exe	x-msdos-program	Berbahaya
watson.microsoft.com	0.htm?LCID=1033&OS=6.1.7601.20010100.1.0.1.17514&SM=innotek%20GmbH&SPN=VirtualBox&BV=VirtualBox	text/html	Tidak berbahaya

Tabel 4 merupakan daftar dari semua objek yang diminta oleh mesin target selama eksekusi *malware* 2. Setelah dilakukan pemindaian menggunakan Virustotal "wget.exe" tercatat sebagai program berbahaya.

4.4. Pembuatan IOC dan Validasi

Pada bagian ini akan dilakukan pemetaan terhadap semua data yang didapatkan pada tahap analisis statis dan analisis dinamis.Pemetaan informasi disesuaikan dengan masing-masing kebutuhan Objek STIX. Dengan melakukan pemetaan ini, maka akan mempermudah proses pembuatan IOC. Pembuatan IOC akan dilakukan dengan menggunakan STIX Python API, sekaligus sebagai validasi kesesuaian format penulisan STIX. Pada tahap sebelumnya telah dilakukan pemetaan informasi ke properti objek STIX, selanjutnya yang akan dilakukan adalah memasukkan informasi yang sudah dipetakan ke dalam format STIX yang ditulis menggunakan sumber kode Python. *File* akan di-*compile* dan akan menghasilkan keluaran berupa baris kode json sesuai dengan format penulisan STIX. Keluaran hasil kompilasi *file* python yang telah dibuat selanjutnya akan di validasi menggunakan STIX Visualizer. STIX Visualizer akan memeriksa skema json yang dimasukkan dan memberikan output berupa visualisasi dari IOC yang dimasukkan.

Tabel 5. Pemetaan hasil analisis ke SDO Indicator

SDO *INDICATOR*	
type	*indicator*
name	*Malicious site hosting downloader*
description	*"Downloading malicious program wget.exe"*
indicator_type	*"malicious-activity"*
pattern_type	stix
pattern	"[url:value='http://down.ctosus.ru/wget.exe']"

Tabel 5 merupakan tabel yang berisi pemetaan informasi hasil analisis *malware* ke dalam properti objek *SDO Indicator*. Pada objek ini didefinisikan URL yang dihubungi oleh *malware* pada tahap analisis dinamis sebagai nilai dari properti "pattern".

Tabel 6 merupakan tabel yang berisi pemetaan informasi hasil analisis *malware* ke dalam properti objek *SDO Malware*. Pada objek ini didefinisikan deskripsi singkat aktivitas *malware* 2 hasil analisis dinamis pada properti "description" dan nilai SHA256 *malware* 2 hasil analisis statis pada properti "name".

Tabel 6. Pemetaan hasil analisis ke SDO Malware

	SDO *MALWARE*
type	*malware*
is_family	*false*
*malware*_type s	*remote-access-trojan , backdoor*
name	*SHA-256 Malware:*
	28e1b6cf6980b009867b8919f68f4 1203e6796e1f95c3f82c17cf1c5d8 ac5b68
description	*This malware attempts to download remote files after establishing a foothold as a backdoor.*
kill_chain_na me	*mandiant-attack-lifecycle*
phase_name	*establish-foothold*

Setelah dilakukan pemetaan, maka berikut ini disajikan hasil akhir dari IOC yang telah berhasil dibuat. Pembuatan IOC ini dilakukan dengan membuat sumber kode python yang selanjutnya akan di-*compile* untuk menghasilkan IOC dalam bentuk bahasa json. Setelah dijalankan, dan menghasilkan sumber kode json, selanjutnya untuk memvalidasi STIX tersebut akan dilakukan dengan menggunakan STIX Visualizer. STIX Visualizer ini akan memeriksa skema json yang dimasukkan dan memberikan output berupa visualisasi dari IOC yang dimasukkan. Berikut ini adalah sumber kode STIX IOC *Malware* dalam bahasa pemrograman json. *Bundle* merupakan objek STIX yang merupakan metode untuk menggabungkan objek-objek yang saling berhubungan. STIX di bawah ini terdiri dari SDO *Indicator*, SDO *Malware* dan SRO. SDO *Indicator* merupakan objek yang merepresentasikan indikator atau karakteristik dari *malware* yang dilengkapi oleh informasi pada SDO *Malware*. Selanjutnya, hubungan antara kedua objek ini harus di definisikan dalam bentuk SRO atau STIX *Relationship Object.*

4.5. Hasil IOC

IOC *Malware* terdiri dari SDO *Indicator*, SDO *Malware* dan SRO yang dibungkus ke dalam bentuk STIX *Bundle Objects*. Bagian pertama dari IOC *Malware* 2 adalah SDO *Indicator*. Pada objek ini, didefinisikan indikator yang menunjukkan keberadaan dari *malware* 2. Baris pertama SDO *Indicator* diawali dengan properti "type" dan baris akhirnya adalah properti "valid_from". Properti "type" menunjukkan identitas dari objek yaitu SDO *Indicator*. Properti "name" dan "description" berisi deskripsi singkat yang menggambarkan aktivitas *malware* ketika menginfeksi suatu perangkat. Aktivitas tersebut kemudian didefinisikan sebagai aktivitas yang berbahaya pada properti "indicator_types" karena berdasarkan hasil analisis dinamis, diketahui bahwa *malware* berusaha mengunduh program berbahaya "wget.exe" dari url "down.ctosus.ru". Aktivitas anomali tersebut dijadikan sebagai nilai dari properti "pattern" dan ditulis sesuai dengan format STIX. SDO *Indicator* dari *Malware* 2 ditampilkan pada Gambar 4.

```
{
            "type": "indicator",
            "spec_version": "2.1",
            "id": "indicator--e729a414-8187-
46f8-ab10-
17fee3a90506",
            "created": "2020-07-
10T15:10:44.879Z",
            "modified": "2020-07-
10T15:10:44.879Z",
            "name": "Malicious site hosting
downloader",
            "description": "Downloading
wget.exe",
            "indicator_types": [
                "malicious-activity"
            ],
            "pattern": "[url:value =
'http://down.ctosus.ru/wget.exe']",
            "pattern_type": "stix",
            "pattern_version": "2.1",
}       "valid_from": "2020-07-
10T15:10:44.879824Z"
```

Gambar 4. SDO *Indicator* dari *Malware* II

Selanjutnya, berikut ini merupakan sumber kode SDO *Malware* yang mendefinisikan informasi spesifik *malware* 2. Properti "description" berisi deskripsi singkat terkait aktivitas *malware* secara umum yang selanjutnya didefinisikan pada properti "kill_chain_phases" sebagai *establish foothold*. Sumber kode SDO *Malware Malware* 2 ditampilkan pada Gambar 5.

```
{
            "type": "malware",
            "spec_version": "2.1",
            "id": "malware--838000b7-
f320-4c99-8dbbc3d076c1b874",
            "created":        "2020-07-
10T15:10:44.879824Z",
"modified":                   "2020-07-
10T15:11:34.288444Z",           "name":
" SHA-256 Malware:
28e1b6cf6980b009867b8919f68f41203e6796e
1f95c3f82c17cf1c5d8ac5 b68",
            "description": "This
malware attempts to download remote
files after establishing a foothold as
a backdoor.",
            "malware_types": [
                "backdoor",
                "remote-access-trojan"
            ],
            "is_family": false,
            "kill_chain_phases": [
                {
                    "kill_chain_name":
"mandiant-attacklifecycle-model",
                    "phase_name":
"establish-foothold"
                }
            ]
        }
```


Gambar 5. SDO *Malware Malware* II

Bagian ketiga dari IOC *Malware* adalah SRO yang mendefinisikan adanya hubungan antara SDO *Indicator* dan SDO *Malware*.

5. KESIMPULAN

Pada penelitian ini telah dilakukan analisis statis dan dinamis pada 5 sampel *malware*. Setelah dilakukan analisis diketahui bahwa kelima *malware* berjenis trojan dan apabila menginfeksi suatu komputer, *malware* akan menghubungi domain berbahaya dan mencoba untuk mengunduh *file* atau program berbahaya. Informasi yang diperoleh dari hasil analisis selanjutnya digunakan untuk pembuatan IoC dengan format STIX Versi 2.1. Pembuatan IoC dilakukan dengan menggunakan STIX Python API sekaligus sebagai validasi terhadap struktur penulisan format STIX. Objek STIX yang digunakan hanya STIX *Core Objects* yaitu SDO *Malware* dan Indicator, keduanya didefinisikan hubungannya dengan STIX *Relationship Objects*. *File* python yang dibuat, dikompilasi dan akan menghasilkan keluaran berupa sumber kode json sesuai dengan format STIX. Sumber kode json ini selanjutnya divalidasi kembali untuk memeriksa kesesuaian dengan struktur bahasa json menggunakan STIX Visualizer. Kesesuaian sumber kode json ditunjukkan dengan keluaran berupa visualisasi dari sumber kode yang telah dibuat.

REFERENSI

[1] Tim Direktorat Proteksi Ekonomi Digital BSSN, "Tips Singkat dan Praktis di Dunia Siber," 2019.

[2] *We Are Social & Hootsuite, "Digital 2019: Essential Insights Into How People Around The World Use The Internet, Mobile Devices, Social Media, and E-Commerce*," *We Are Soc. Hootsuite*, p. 76, 2019.

[3] IDSIRTII,"*Indonesia Cyber Security Monitoring Report* 2018." Jakarta, 2018.

[4] M. K. A., *Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware*. Birmingham: Packt Publishing Ltd, 2018.

[5] Pemerintah Indonesia, "Peraturan Presiden Republik Indonesia Nomor 53 Tahun 2017 Tentang Badan Siber dan Sandi Negara," p. Lembaran RI Tahun 2007 No. 100, 2017.

[6] L. Rudman and B. Irwin, "*Dridex: Analysis of the traffic and automatic generation of IOCs*," *2016 Inf. Secur. South Africa - Proc. 2016 ISSA Conf.*, pp. 77–84, 2016.

[7] Solutionary, "*How Malware Analysis Benefits Incident Response*," no. 866, 2014.

[8] M. Del, C. P. Tixteco, L. P. Tixteco, G. Sánchez Pérez, L. Karina, and T. Medina, "*Intrusion Detection Using Indicators of Compromise Based on Best Practices and Windows Event Logs*," in *ICIMP 2016: The Eleventh International Conference on Internet Monitoring and Protection*, 2016, pp. 29–37.

[9] S. YusirwanS, Y. Prayudi, and I. Riadi, "*Implementation of Malware Analysis using Static and Dynamic Analysis Method*," *Int. J. Comput. Appl.*, vol. 117, no. 6, pp. 11–15, 2015.

[10] NIST, "NIST Special Publication 800-83 *Guide to Malware Incident Prevention and Handling for Desktops and Laptops NIST Special Publication 800-83 Guide to Malware Incident Prevention and Handling for Desktops and Laptops*," p. 7, 2013.

[11] A. Sikorski, Michael; Honig, *Practical Malware Analysis: The Hands-On Guide ti Dissecting Malicious Software*, vol. 53, no. 9. San Fransisco: William Pollock, 2013C. C. Elisan, *Advanced Malware Analysis*. United States: McGraw-Hill Education, 2016.

[12] M. Ligh, Michael Hale;Adair,Steven;Hartstein, Blake;Richard, *Malware Analyst's Cookbook and DVD:Tools and Techniques for Fighting Malicious Code*. Indianapolis: Wiley Publishing Inc., 2018.

[13] J. Peppers, "Creating a *Malware* Analysis *Lab and Basic Malware* Analysis," Iowa State University, 2018.

[14] OASIS, "*STIX ™ Version 2.1*," United States, 2020.

[15] OASIS, "*STIX Version 2.0. Part 1*: STIX *Core Concepts*," United States, 2017.