BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Penulis melakukan review pada penelitian sebelumnya yang berkaitan dengan *sanboxing* dan memetakan evolusi *Ransomware*. Kemudian penulis menemukan beberapa paper atau jurnal yang berkaitan dan bisa menunjang penelitian ini. Salah satunya adalah paper yang ditulis oleh Ashwini Balkrushna Kardile dengan judul "*Crypto Ransomware Analysis and Detection Using Process Monitor".*

Paper tersebut membahas tentang *cuckoo sandbox,* yaitu sistem yang di desain untuk melakukan analisis terhadap aktifitas *Ransomware* dalam sebuah sistem. Sistem ini juga mencatat dan menyimpanakses file untuk mengetahui perilaku yang berpotensi merusak. Sistem dalam paper ini memadukan antara analisis statis dan dinamis, dapat melacak log API dan diakses melalui web.

Penulis juga melakukan review terhadap paper yang ditulis oleh Pete Burnap, Richard French,Federic Turner yang berjudul "*Malware classification using self organizing feature maps and machine activity data".* Penelitian dalam jurnal tersebut membahas tentang menggunakan metrik untuk secara otomatis membedakan antara sampel perangkat lunak yang dapat dijalankan dan berbahaya. Dan memungkinkan memetakan sampel baru dan menentukan bahwa perilaku malware berbeda, tetapi cukup mirip dengan perilaku yang sebelumnya yang telah diamati dan diberi label.

2.2 Landasan teori

Bagian ini menjelaskan teori-teori dasar yang berkaitan dengan sandboxing *malware,* tools yang digunakan untuk sandboxing *malware,* memetakan evolusi *malware,* serta tools untuk menganalisis *malware* yang telah berhasil di analisis.

Dari beberapa jurnal yang penulis baca berikut beberapa teori yang penulis dapatkan ditampilkan pada tabel 2.1.

Tabel 2. 1 Tabel jurnal

No	Judul	Penulis	Pencapaian
1.	Detecting fake antivirus	Dae Wook Kim,	Penelitian ini menjelaskan serangan terus menerus untuk

	Software Distribution Webpage	Peiving Yan, Junije Zhang	mencari metode baru unruk mendistribusikan *Ransomware*, seperti penyerangan Antivirus palsu dan meyakinkan Pengguna untuk menginstalnya. Hasil yang dikumpulkan dari halaman web antivirus palsu menunjukkan keakuratan deteksi yang mencapai 90.4% pada tingkat positif palsu
2.	Software-defined networking Based Crypto *Ransomware* Detecting using HTTP traffic characteristics	Krzysztof Cabaj, Marcin Gregorczyk, Wojcieh Mazurczyk	Menjelaskan tentang yang paling berbahaya dari *Ransomware* adalah Yang mengenkripsi data. Dalam makalah ini dijelakan deteksi berbasis Software-defined networking dengan melakukan pengukuran kedua jaringan untuk dua *Ransomware* yaitu, *Cryptowall* da *locky*. Hasil ditemukan membuktikan pendekatan ini mampu mencapai tingkat deteksi 97-98%.
3.	*Ransomware* Classification Using self organizing feature maps and machine activity data	Pete Burnap, Richard French, Frederick Turner, Kevin Jones	Penelitian ini menggunakan matrik aktivitas mesin secara otomatis Membedakan antara sampel perangkat Lunak *portable* yang dapat dijalankan Dan berbahaya.
4.	Probabilistic analisys of Dynamic *Ransomware* traces	Jan Stiborek, Tomas Pevny, Martin Rehak	Makalah ini membahas tentang metode secara otomatis mengelompokkan Biner yang tidak diketahui yang dijalankan sesuai interaksinya

			dengan *system* (file sistem, *registry*) Perbandingan ini menunjukkan bahwa metode yang di usulkan melebihi pendekatan state-of-the-art Terkait karena menghasilkan *cluster* yang individual.
5.	Countering cyber threats of Industrial application: an Automated approach for Malware evasion detection and analysis	Muzzamil Noor, Haider abbas, Waleed B.S	Makalah ini membahas tentang deteksi Dan pencegahan ancaman *malware* dan teknik penghindaran yang Memberikan banyak tantangan bagi para riset *malware*. Selain itu, penanggulangan dilaksanakan oleh Analisys Evasion *malware* sandbox lebih efektif untuk deteksi malware dalam presentase besar.

2.2.1 *Ransomware*

Ransomware adalah jenis perangkat lunak berbahaya yang mencegah pengguna mengakses atau membatasi akses mereka ke sistem atau file, baik dengan mengunci layar atau dengan mengenkripsi file, sampai meminta tebusan. Dalam banyaknya kasus, *Ransomware* meninggalkan pengguna dengan sangat sedikit opsi, seperti hanya membolehkan korban untuk berkomunikasi dengan penyerang dan membayar tebusan.

Jenis *Ransomware* yang paling umum menggunakan beberapa bentuk *Ransomware*enkripsi, termasuk berbasis simetris dan dan dengan skema enkripsi. *Ransomware* yang bergantung pada enkripsi publickey sangat sulit dimitigasi kunci enkripsi disimpan dalam perintah jarak jauh dan kontrol (C&C) server.

Biasanya ada batas waktu untuk tebusan yang harus dibayar, para pengguna disediakan dengan khusus situs web untuk menebus dan petunjuk langkah demi langkah tentang cara membayar uang tebusan. cara kerja *Ransomware* modern biasanya terdiri langkah-langkah berikut: distribusi, infeksi, K&C komunikasi, pencarian file, enkripsi file dan permintaan tebusan.

2.2.2 Jenis jenis *Ransomware*

a. Wannacry

Wannacry pertama kali muncul pada tahun 2017 dimana virus ini menyerang beberapa perusahaan besar di Eropa. Wannacry adalah sebuah *Ransomware* yang diciptakan oleh para hacker, yang menyerang sistem komputer melalui celah keamanan. Komputer yang terinfeksi malware atau *Ransomware* tersebut, datanya akan dicuri dan dikunci. Komputer yang terinfeksi malware, akan muncul tampilan berwarna merah yang memiliki pesan dimana pengguna diharuskan untuk mengirimkan uang sejumlah 300 USD apabila pengguna ingin datanya dikembalikan. Jika tidak file mereka yang telah dicuri akan dihapus.

Gambar 2. 1 tampilan *Ransomware* Wannacry

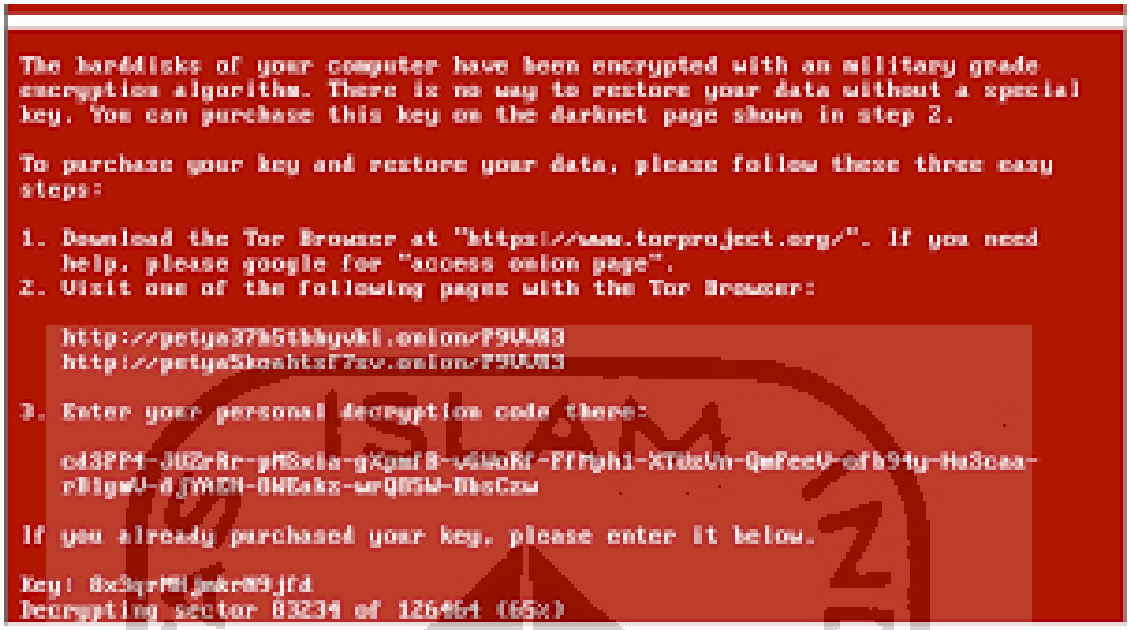
b. Petya

Petya merupakan *Ransomware* pengenkripsi yang pertama kali ditemukan pada tahun 2016. Petya menyerang komputer dengan sistem operasi Windows. Dampak yang ditimbulkan dari perangkat pemeras ini adalah pengguna tidak dapat mengakses komputer dan semua berkas di dalamnya terkunci, kemudian perangkat tersebut meminta sejumlah tebusan yang dibayarkan menggunakan Bitcoin.

Varian dari Petya pertama kali terlihat pada Maret 2016, yang menyebar melalui lampiran surat elektronik. Pada Juni 2017, jenis baru dari Petya digunakan untuk serangan secara global, dengan target utamanya adalah Ukraina. Jenis baru ini menggunakan exploit EternalBlue, yang mana dipercaya sebagai buatan Badan Keamanan Nasional Amerika Serikat (NSA), dan juga digunakan sebelumnya oleh Wannacry. Kaspersky Lab menyebut versi ini sebagai NotPetya untuk membedakan dengan Petya yang dibuat pada tahun 2016.

Virus Petya adalah program *Ransomware* yang mengambil alih komputer pengguna dan mengunci file mereka. *Ransomware* ini mungkin sangat berbahaya dan menginfeksi PC, tapi target utamanya adalah komputer pada perusahaan Jerman. Program jahat ini memasuki komputer korban secara diam-diam dan melakukan kegiatan berbahaya tanpa diketahui dan dicurigai oleh pemilik komputer.

Petya mengenkripsi file dengan sangat kompleks menggunakan algoritma RSA-4096 dan AES-256, enkripsi inilah yang juga digunakan oleh pihak militer. Kode tersebut tidak bisa dibuka tanpa adanya kunci pembukanya tersendiri. Tentu saja, biasanya untuk program *Ransomware* lain seperti virus Locky, virus CryptoWall, dan CryptoLocker, kunci pribadi ini disimpan pada beberapa server yang jauh, yang hanya dapat diakses dengan membayar uang tebusan untuk para pencipta virus. setelah virus ini masuk ke PC anda, maka PC Anda akan segera reboot, dan ketika boot lagi, akan muncul notif "*do not turn off your pc! if you abort this process, you could destroy all of your data! please ensure that your power cable is plugged in!"* setelah itu akan muncul tampilan seperti gambar 2.2



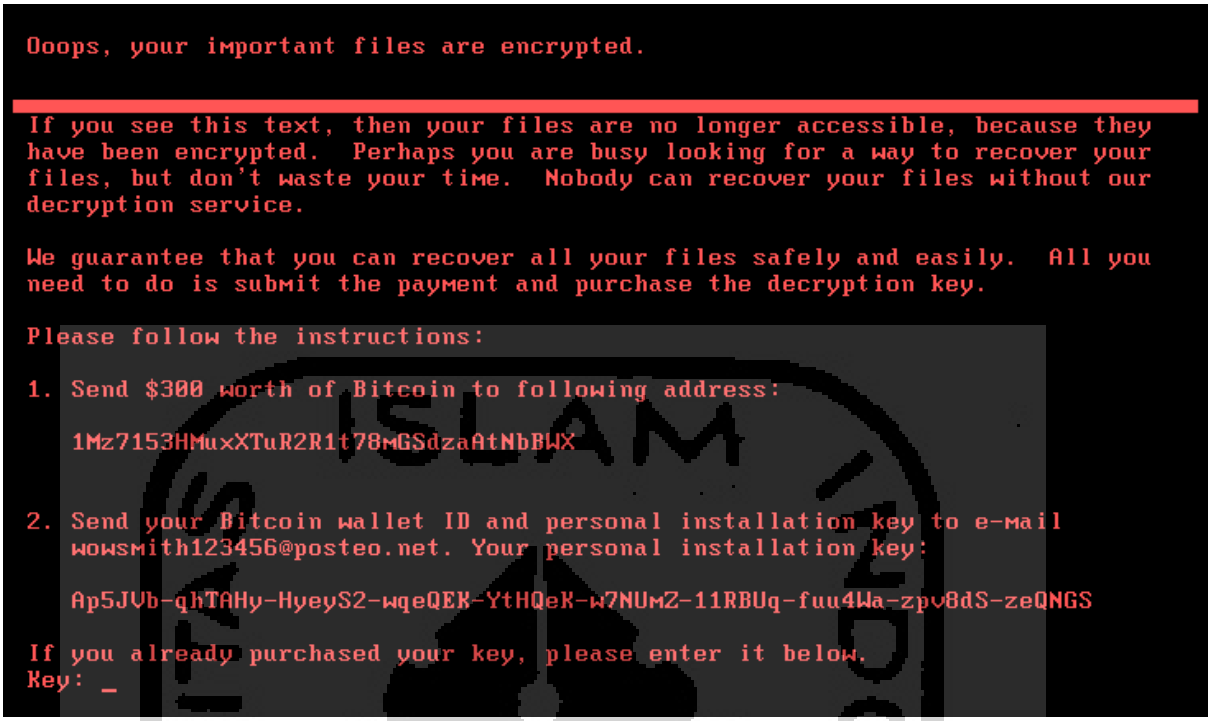
Gambar 2. 2 Tampilan *Ransomware* Petya

c. NotPetya

NotPetya merupakan virus perangkat pemeras (*Ransomware*) seperti Petya pada tahun 2016. Dampak yang ditimbulkan dari perangkat pemeras ini adalah pengguna tidak dapat mengakses komputer dan semua berkas di dalamnya terkunci, kemudian perangkat tersebut meminta sejumlah tebusan yang dibayarkan menggunakan Bitcoin.

Ransomware ini berpotensi lebih merusak daripada Wannacry, karena *Ransomware* ini tidak perlu membuka patch untuk menginfeksi PC dan jaringan local, NotPetya mengambil data SMB dan kredensial dari PC yang etrinfeksi dan menggunakan kredensial tersebut untuk terhubung ke sistem lain menggunakan jaringan lokal untuk menyebarkan dirinya, oleh karena itu ransomeware ini berpotensi menginfeksi tidak hanya 1 PC namun dapat menginfeksi seluruh PC dalam 1 jaringan local

Kapersky sebuah perusahaan keamanan Rusia mengatakan bahwa NotPetya dirancang sebagai wiper yang berpura pura menjadi *Ransomware* yang menjanjikan akan memberi kode unik untuk membuka file yang terenkripsi, namun berdasarkan analisa yang dilakukan Kapersky kode unik itu hanyalah kode unik biasa yang tidak bisa digunakan untuk membuka file yang terenkripsi



Gambar 2. 3 Tampilan PC yang terinfeksi Petya

d. Badrabbit

Bad Rabbit dibangun dari source code yang sama dengan Petya, namun Badrabbit tidak melakukan eksploitasi terhadap celah keamanan seperti yang dilakukan oleh Wannacry dan Petya, pada umumnya Badrabbit melakukan rekayasa sosial di mana ia memalsukan dirinya sebagai update / installer Adobe Flash dan di injeksikan pada situs-situs yang tidak terkonfigurasi dengan aman sehingga pengakses situs yang tidak menyadari hal ini mengira mereka mengunduh Adobe Flash dan menjalankannya.

Setelah berhasil menginfeksi melalui internet, Bad Rabbit memiliki kemampuan dictionary attack simpel pada akun administrator. Jika akun administrator berhasil dikuasai, dia akan menyebarkan dirinya melalui intranet.

Bad Rabbit awalnya disebarkan melalui situs-situs populer seperti situs berita yang berbasis di Rusia. Situs-situs ini diretas dan pengaksesnya diarahkan ke situs utama yang beralamat di 1dnscontrol.com guna mengunduh file malware yang telah dipersiapkan oleh pembuat malware.

Gambar 2. 4 Tampilan PC yang terinfeksi Badrabbit

2.2.3 Sandbox

Sandbox adalah sebuah mekanisme keamanan untuk memisahkan program yang sedang berjalan. Sandbox ini sering digunakan untuk mengeksekusi kode yang belum teruji, atau program yang tidak dipercaya dari pihak pihak ketiga yang tidak diverifikasi, pemasok, pengguna yang tidak dipercaya dan situs yang tidak dipercaya (Bremer, 2018). Konsep ini juga berlaku untuk analisis *malware* sandboxing yang bertujuan untuk menjalankan aplikasi atau file yang tidak dikenal dan tidak dipercaya didalam lingkungan yang terisolasi dan mendapatkan informasi tentang apa yang dilakukannya.

Malware sandboxing adalah implementasi dari pendekatan dynamic analysis, program akan benar-benar dijalankan dan dipantau secara real-time

Teknik sandboxing ini tentunya memiliki kelebihan dan kekurangan, tetapi teknik ini adalah teknik yang baik untuk mendapatkan detail tambahan tentang *malware*, seperti perilaku jaringannya. Dan akan lebih baik jika sandboxing ini dilakukan bersamaan dengan analisis statis terhadap *malware* untuk mendapatkan pemahaman yang lebih dalam.

Teknik sandboxing ini tentunya memiliki kelebihan dan kekurangan, tetapi teknik ini adalah teknik yang baik untuk mendapatkan detail tambahan tentang *malware*, seperti perilaku

jaringannya. Dan akan lebih baik jika sandboxing ini dilakukan bersamaan dengan analisis statis terhadap malware untuk mendapatkan pemahaman yang lebih dalam.

2.2.4 Cuckoo Sandbox

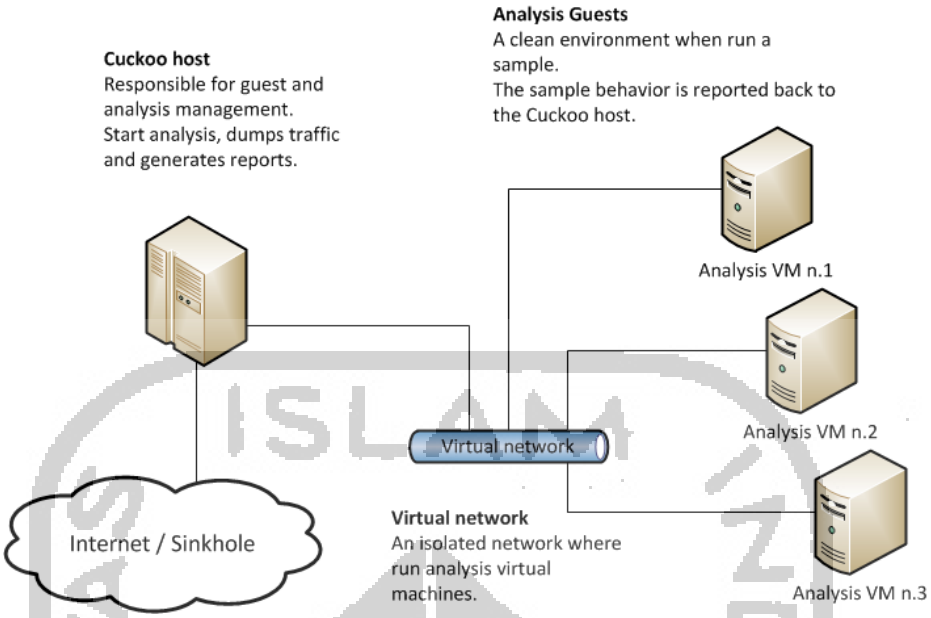
Cuckoo Sandbox adalah tool untuk melakukan analisis *malware* secara otomatis. Digunakan untuk menjalankan dan menganalisa file secara otomatis dan mengumpulkan hasil analisis komprehensif yang menguraikan apa yang dilakukan *malware* saat berjalan didalam sistem operasi. Pada penelitian ini, penulis menggunakan Cuckoo Sandbox, yang berfungsi untuk menganalisis *malware* yang berjalan pada perangkat windows yang terinstall menggunakan virtualbox. Beberapa hasil yang didapatkan oleh Cuckoo Sandbox adalah:

a. Jejak panggilan API Win32 yang dihasilkan oleh semua proses yang dilakukan oleh *malware.*
b. File yang dibuat, dihapus dan diunduh oleh *malware* selama proses eksekusi
c. Penumpukan memori dari proses *malware*
d. Jejak lalu lintas jaringan dalam format PCAP
e. Screenshots dari desktop yang diambil selama proses eksekusi *malware*
f. Seluruh memori dari perangkat.

Setiap analisis dilakukan pada mesin virtual baru yang terisolasi. Infrastruktur Cuckoo terdiri dari mesin host (management software) dan sejumlah mesin guest (mesin virtual yang melakukan analisis). Mesin host menjalankan komponen inti dari sandbox yang mengelola seluruh proses analisis, sementara mesin guest adalah lingkungan terisolasi dimana sampel *malware* dieksekusi dengan aman dan kemudian dianalisis.

Setiap guest terdiri dari mesin virtual linux yang menjalankan android yang dikendalikan oleh modul mesin. Komponen tambahan yang dipasang didalam mesin linux untuk mendukung proses analisis adalah:

a. Python 2.7
b. Script python Agent.py
c. Komponen analisis yang dikirim ke mesin guest pada awalan analisis



Gambar 2. 5 Alur Analisis Cuckoo Sandbox

Cuckoo sandbox dapat diunduh dari situs web resmi, dimana paket yang stabil dirilis, atau dapt dikloning dari repository git.

2.2.5 Volatility

Volatility framework adalah kumpulan tools yang diimplementasikan dengan python dibawah lisensi public-GNU, untuk ekstrak artefak digital dari sampel volatile memori (RAM). Teknik ektraksi dilakukan sepenuhnya independen dari sistem yang sedang tetapi menawarkan visibilitas ke status runtime sistem. Framework ini dimaksudkan untuk memperkenalkan orang kepada orang kepada teknik dan kompleksitas yang terkait dengan ekstraksi artefak digital dari sampel memori yang mudah menumpuk dan menyediakan platform untuk bidang penelitian ini di masa depan.

Volatility mendukung berbagai format file sampel dan kemampuan untuk mengkonversi antara format ini, antara lain.

a. Raw linear sample (dd)
b. Hibernation file (dari windows 7 dan sebelumnya)
c. Crash dump file
d. VirtualBox ELF64 core dump
e. Vmware saved state dan file snapshot
f. EWF format (E01)
g. LiME format
h. Mach-O format file
i. QEMU virtual machine dumps
j. Fireware
k. HPAK (FDPro)