TEKNIK EVASIVE MALWARE ANDROID TERHADAP GOOGLE SANDBOX

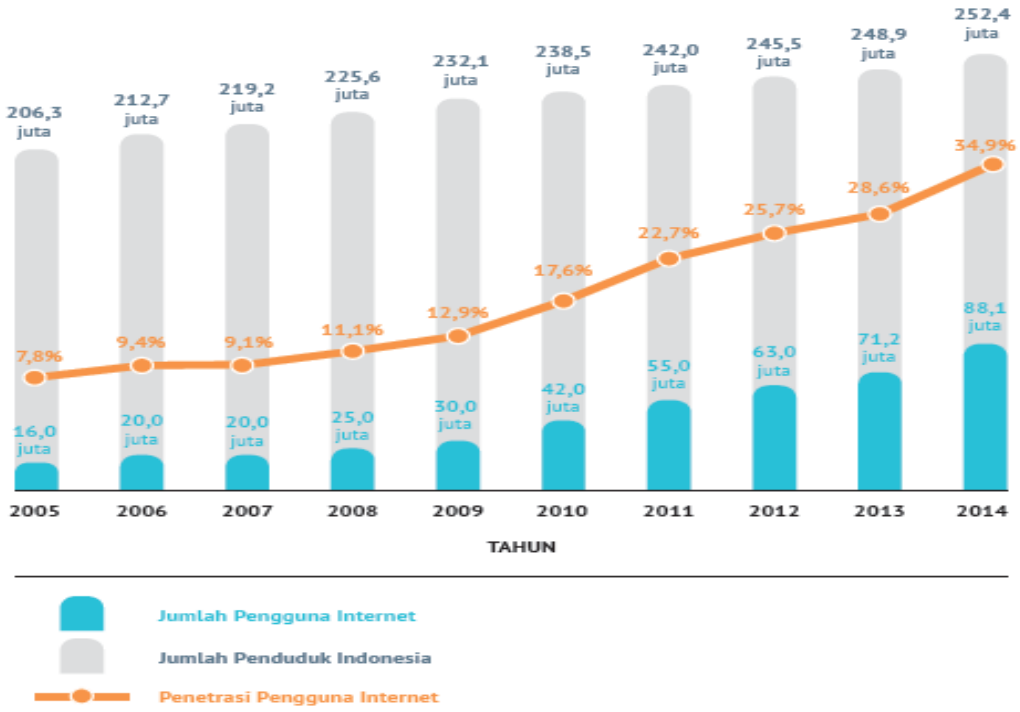
M.Alimuddin (23215022)
Software Security (EL5215)
Rekayasa dan Manajemen Keamanan Informasi
Sekolah Tinggi Ilmu Elektro dan Informatika, STEI
m.alimuddin@s.itb.ac.id

***Abstract*—Peningkatan jumlah malware dalam beberapa tahun terakhir menjadi ancaman yang nyata bagi teknologi informasi. *Sandbox atau blackbox analysis* adalah metode yang potensial digunakan secara efisien dan efektife untuk mengamati perilaku malware. Sampel malware akan dilihat perilakunya didalam lingkungan sandbox yang sudah sedemikian rupa mirip dengan system sebenarnya. Makalah ini akan menjelaskan berbagai metode yang ada untuk mendeteksi dan mengelabui sandbox. Makalah ini juga memberikan rekomendasi yang bisa digunakan untuk menghadapi perilaku malware sehingga dapat dideteksi.**

Keywords—Sandbox; Blackbox analysis

I. PENDAHULUAN

Berdasarkan hasil survey Asosiasi Pengguna Jasa Internet indonesia (APJII) jumlah pengguna internet selalu meningkat dari tahun 2005 sampai dengan tahun 2014. Puncak peningkatan tertinggi terjaddi pada tahun 2013 terhadap 2014 dengan jumlah penambahan sebanyak 6,9 juta pengguna[1]. Secara lebih detail peningkatan jumlah pengguna internet di Indonesia dari tahun 2005 sampai dengan tahun 2014 dapat dilihat pada grafik 1.



Grafik 1. Perkembangan Jumlah Pengguna Internet di Indonesia Tahun 2005-2014.

Survey tersebut juga mencatat jumlah pengguna internet Indonesia pada 2014 berjumlah 88,1 juta pengguna dan 85 % diantaranya menggunakan telepon seluler sebagai salah satu media akses internet. Dan menurut riset Gartner pada quartal kedua tahun 2015 handphone dengan system operasi android menguasai sebesar 82,2% dari total penjualan telepon seluler[2]. Sehingga berdasarkan data-data tersebut bahwa pengguna internet di Indonesia sebagian besar mengakses internet dengan media akses melalui handphone dengan system operasi android.

Berdasarkan laporan GData Malware report pada tahun 2014 sudah terdapat lebih dari 1,5 juta jenis malware android yang beredar. Dan pada kuartal 1 tahun 2015 terdeteksi sebanyak 440 ribu jenis malware android baru

yang terdeteksi. Jumlah malware android tersebut diperkirakan akan terus meningkat seiring dengan peningkatan jumlah pengguna android[3].

Google mengklaim bahwa android merupakan system operasi yang sangat secure namun tidak dengan penggunanya. Android dilengkapi dengan beberapa layer keamanan yang menghalangi malware untuk mendapatkan akses ke dalam system dan data dari user. Namun android juga sebuah open system yang membebaskan user dan developer aplikasi untuk membangun aplikasi yang bermanfaat bagi user. Sehingga jika user menginginkannya, aplikasi bisa mendapatkan akses terdalam dari system jika user memberikan akses root. Sehingga dalam diskusi security android selalu berakhir dengan kesimpulan bahwa terinfeksinya android oleh malware bukan dikarenakan malware tersebut sangat hebat namun lebih dikarenakan user memberikan akses kepada malware untuk hidup dan berkembang didalam system androidnya. *Android gives us a lot of power, and with great power comes great responsibility*[4].

Novriadi, Novaria dan Saksono (2014) melakukan analisis forensic pada android malware dengan nama *Hongtoutou* yang membaca dan mengirimkan *International Mobile Equipment Identity(IMEI)* dan *International Mobile Subscriber Identity (IMSI)* ke nomor tertentu dari handpone yang di infeksinya. Mereka juga melakukan analisis forensic terhadap malware *Hippo* yang mengirimkan *Short Message Service (SMS)* ke layanan premium tertentu sehingga korban secara otomatis menjadi berlangganan SMS premium tersebut. Kedua malware android di atas bekerja secara otomatis setelah diinstall tanpa autentikasi dari user[5].

II. HISTORY ANDROID

Android merupakan sistem operasi yang berbasis Linux dan dirancang untuk perangkat seluler layar sentuh seperti smartphone serta komputer tablet. Android pada awalnya dikembangkan oleh perusahaan bernama Android, Inc. yang didirikan di Palo Alto, California, pada Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri Wildfire Communications, Inc.), Nick Sears (mantan VP T-Mobile), dan Chris White (Kepala desain dan pengembangan userinterface WebTV), dengan dukungan finansial yang berasal dari Google, yang kemudian Google pun membelinya pada tahun 2005. Sistem operasi android tersebut secara resmi dirilis pada tahun 2007, bersamaan dengan didirikannya sebuah perusahaan Open Handset Alliance, konsorsium dari beberapa perusahaan-perusahaan perangkat keras, perangkat lunak, serta telekomunikasi yang memiliki tujuan untuk memajukan standar terbuka dari perangkat seluler. Ponsel yang berbasis sistem operasi Android pertama dijual pada bulan Oktober 2008.

Google mengakuisisi perusahaan Android Inc. pada tanggal 17 Agustus 2005 dan menjadikannya sebagai anak perusahaan yang dimiliki oleh Google. Pendiri Android Inc. yaitu Rubin, Miner, serta White tetap bekerja pada perusahaan tersebut setelah diakuisisi oleh Google. Di Google, tim yang dipimpin oleh Andy Rubin mulai untuk mengembangkan sebuah platform perangkat seluler dengan menggunakan kernel Linux[6].

Sejak tahun 2008, Android mulai secara bertahap melakukan sejumlah pembaruan atau update untuk meningkatkan kinerja dari sistem operasi tersebut dengan menambahkan fitur baru, memperbaiki bug pada versi android yang sebelumnya. Setiap versi yang dirilis dinamakan secara alfabetis dengan berdasarkan nama sebuah makanan pencuci mulut, seperti cupcake, donut, dan sebagainya. Berikut nama-nama versi android:

1. Android Apple Pie

Google dan OHA merilis setidaknya ada 2 versi sebelum Android beta dirilis pada November 2007. Versi Alpha memiliki codename yaitu : Astro Boy, Bender dan R2-D2.

2. Android Banana Bread

Android beta dirilis pada tanggal 5 November 2007, sedangkan software development kit atau SDK telah dirilis pada tanggal 12 November 2007. Pada tanggal 5 November kemudian ditetapkan sebagai hari “ulang tahun”.

3.Android 1.5 Cupcake

Android 1.5 Cupcake dirilis pada tanggal 30 April 2009. Cupcake adalah versi android pertama yang menggunakan nama dari sebuah makanan. Konon katanya versi android cupcake seharusnya versi 1.2, namun Google telah memutuskan untuk membuat revisi yang besar serta membuatnya menjadi versi 1.5. Cupcake adalah kue yang berbentuk kecil dan dipanggang dalam cetakan berbentuk cup.

4. Android 1.6 Donut

Android 1.6 Donut dirilis pada tanggal 15 September 2009. pada versi ini telah diperbaiki beberapa kesalahan reboot, perubahan pada fitur foto dan video serta integrasi pencarian yang lebih baik. Donut merupakan makanan yang berbentuk cincin.

5. Android 2.0/2.1 Éclair

Android 2.0/2.1 Eclair dirilis pada tanggal 26 Oktober 2009. Eclair merupakan makanan penutup yang berupa kue berbentuk persegi panjang dan ada krim di tengah serta lapisan cokelat di atasnya.

6. Android 2.2 Froyo

Android 2.2 Froyo dirilis pada tanggal 20 Mei 2010. Froyo merupakan makanan penutup yang berasal dari sebuah nama merk produk yang terbuat dari yoghurt. Froyo adalah yoghurt yang dingin sehingga seperti es krim. Froyo merupakan singkatan dari Frozen yoghurt.

7. Android 2.3 Gingerbread

Android 2.3 Gingerbread dirilis pada tanggal 6 December 2010. Gingerbeard merupakan sejenis kue kering yang memiliki rasa jahe. Gingerbread biasanya dibuat pada saat perayaan libur akhir tahun di benua Amerika.

8. Android 3.0 Honercomb

Android 3.0 Honeycomb dirilis pada tanggal 22 February 2011. Honeycomb merupakan sereal sarapan manis yang sudah pernah dibuat tahun 1965 oleh Posting Sereal. Honeycomb atau sarang lebah, sereal ini terbuat dari beberapa potongan jagung yang kemudian dibentuk seperti sarang lebah dengan rasa madu.

9. Android 4.0 Ice Cream Sandwich

Android 4.0 Ice Cream Sandwich dirilis pada tanggal 19 Oktober 2001. Ice cream sandwich merupakan lapisan es krim yang biasanya terdapat rasa vanila terjepit di antara dua kue cokelat, berbentuk persegi panjang.

10. Android 4.1 Jelly Bean

Android 4.1 Jelly Bean dirilis pada tanggal 9 Juli 2012. Jelly bean adalah nama sejenis permen dalam beraneka macam rasa buah-buahan. Ukurannya seperti kacang merah. Permen ini keras di luar namun lunak di dalam dan lengket apabila digigit.

11. Android KitKat

Android 4.4 KitKat dirilis pada tanggal 31 Oktober 2013. KitKat merupakan merk cokelat yang dikeluarkan oleh Nestle.

12. Android 5.0 lollipop

Android 5.0 Lollipop dirilis pada tanggal 15 Oktober 2014. Lollipop merupakan sebuah permen manis dalam stickt yang biasanya berbentuk lingkaran atau bulat.

13. Android 6.0 Marshmallow

Android 6.0 Marshmallow secara resmi dirilis google pada 17 Agustus 2015. Beberapa kelebihan Marshmallow dibandingkan versi sebelumnya yaitu adanya mode "Doze". Mode Doze yaitu fitur terbaru dari android yang mampu memperkecil konsumsi daya baterai ketika smartphone berada pada kondisi standby atau moe sleep yaitu dengan cara menonaktifkan sebagian besar aplikasi background. Kelebihan lainnya yaitu pada App permissions dimana marshmallow memperkenalkan mmetode app permissions yang didesain ulang. Aplikasi tidak lagi secara otomatis diberikan semua akses yang ditentukan pada waktu instalasi. Pada marshmallow pengguna dapat memberikan atau menolak permission dari aplikasi terhadap servise tertentu misalnya kemampuan aplikasi mengakses kamera atau mikrofon.

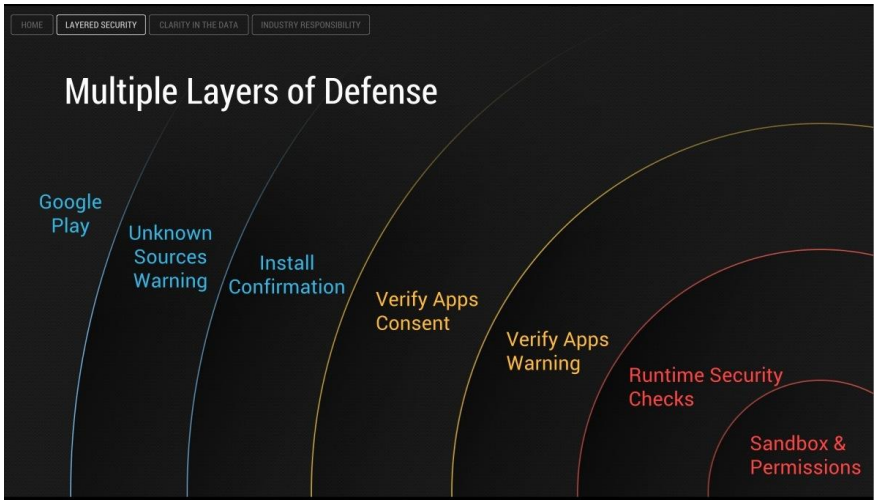
III. LAYER SECURITY ANDROID

Android di amankan oleh tujuh layer of defense yaitu *Google Play, Unknown Sources Warning, Install Confirmation, Verify Apps Consent, Verify Apps Warning, Runtime Security Checks, dan Sandbox&Permissions*. Untuk menginstall aplikasi, file instalasi aplikasi bisa diletakkan pada Google playstore. Jika aplikasi tidak disimpan didalam playstore maka user hanya bisa melakukan install aplikasi jika user memberikan akses untuk instalasi aplikasi dari "unknown sources warning" dan user memberikan izin instalasi (confirm the installation). Disisi lain ketika developer aplikasi ingin aplikasinya di terima dan dapat diakses melalui playstore maka aplikasi harus melewati "Verify Apps" dimana akan dilakukan cek terhadap file APK terhadap keberadaan malware yang sudah ada di dalam database malware yang dimiliki Google. Tahap selanjutnya yaitu "sandboxing" terhadap aplikasi tersebut untuk melihat bagaimana aplikasi berjalan di dalam system[4].

Jika user menggunakan android secara normal dan hanya melakukan instalasi dari sumber yang terpercaya dan secara selektif menentukan aplikasi yang akan di install didalam system androidnya maka semua proses anomaly yang akan terjadi sudah dapat di prediksi[4].

Namun dalam kenyataannya banyak user yang tidak terlalu peduli dengan hak akses(permission) yang diminta oleh aplikasi ketika melakukan instalasi. User biasanya baru merasa dirugikan setelah kehilangan data, melambatnya kinerja system, atau berkurangnya pulsa.

Gambar 1. layer Security Android



Berdasarkan skema layer of defense dari android di atas Sandboxing adalah layer tertinggi dari system security pada android. Sandbox adalah lingkungan virtual tertutup yang dibuat untuk menguji coba aplikasi/software bahkan malware untuk melihat perilaku, kerusakan yang dibuat dan juga mencari cara menangani malware tersebut. Metode Sandboxing tidak hanya digunakan oleh Google untuk menyeleksi aplikasi android terhadap sisipan malware. Namun juga digunakan secara umum oleh perusahaandan developer aplikasi untuk menganalisis kode baru dan juga sifat serta perilaku malware.

Sandboxing adalah teknik analisis yang efektif untuk ancaman yang ditimbulkan malware. Konsep Sandboxing atau biasa juga disebut *blackbox analysis* yaitu untuk menangkap sampel program jahat dalam lingkungan pengujian yang terkontrol untuk dipelajari dan dianalisis[6]. Lingkungan otomatis dari Sandbox memudahkan untuk mengikuti perilaku malware, mengelompokkannya kedalam beberapa jenis malware dan menyiapkan databasenya. Beberapa teknik malware menyembunyikan diri yaitu pembuatan *hidden executable files*, malware *executable files* dibuat mirip *system files* asli, *auto starting files* pada saat *boot*, mengganti *image* dengan proses yang berbeda, dan *refusing to debug*[7].

IV. TEKNIK EVASIVE MALWARE

Malware menggunakan berbagai macam teknik untuk mengalahkan teknologi Sandbox. Menurut Lakhani (2015), terdapat 4 teknik dasar yang digunakan malware[8], yaitu:

- *Configuration-specific techniques* seperti : s*leep calls, time triggers, fast flux*, dan *process hiding*.
- *human interaction techniques* seperti : *mouse click* dan *dialog boxes*.
- *environment specific techniques* seperti: *version, embedded iframes*, dan *DLL loaders*.
- *VMware-specific techniques* seperti : *system service lists, unique files*, dan *the VMX port*.

Dari beberapa metode diatas *extended sleep* dan *fast flux* adalah metode yang paling banyak digunakan. Berikut pembahasan dari beberapa metode tersebut:

4.1 Extended Sleep

Metode evasive yang paling banyak digunakan oleh malware yaitu extended sleep. Teknik ini sangat efektif karena penggunaan lingkungan sandbox sangat *computationally effective*. Kebanyakan sandbox yang berjalan menggunakan device yang minim dan virtual machine yang terbatas. Mereka menjalankannya hanya dalam beberapa menit dan jika tidak ada reaksi yang menyimpang sandbox maka dianggap tidak ada aktifitas yang berbahaya dari sampel tersebut. Pada beberapa malware extended sleep dikombinasikan dengan deteksi mouse, keyboard dan interaksi manusia untuk mengelabui sandbox.

Berikut adalah contoh script untuk extended sleep yaitu beberapa perintah sleep durasi yang panjang. Bekerja dengan memanipulasi secara dinamis clock system sehingga membuat malware berjalan dalam jangka waktu yang panjang.

```
Script

    $waitTimeInSeconds = 120*1000

    _JBSetSystem("xp")

    ; if sleep is bigger than 50s replace it with 5s
    _JBShortenSleepsGreaterThan(50,5)

    _JBRandomVMDetectionArtifacts()

    _JBHideTools()

    _JBDelAnalysisState()

    _JBStartSniffer()
    _JBStartAnalysis()

    _JBStartScreenDumper()

    _JBLoadProvidedBin()

    Sleep($waitTimeInSeconds)

    _JBStopScreenDumper()

    _JBStopSniffer()
    _JBStopAnalysis()

    _JBCleanUp()

EndScript
```

Contoh malware nya yaitu Khelios Botnet yang dinyatakan sudah mati pada tahun 2011. Untuk menghindari deteksi sandbox berbasis file, salah satu sampel Khelios baru (dikenal dengan TrojanNap) ditemukan pada 2013memanggil SleepEx() API dalam waktu 10 menit. Karena sandbox sebagian besar menguji sampel dalam waktu singkat, sampel Nap tersebut menunda aktifitas berbahayanya diluar waktu yang dipantau dari sandbox. Sampel dipanggil ke undocumented Windows API pada fungsi NtDelayExecution() untuk melaksanakan extended sleep.

4.2 Fast Flux

Fast Flux adalah teknik DNS yang digunakan malware untuk menyembunyikan phishing dan pengiriman situs malware tersembunyi dan bahkan merubah network yang terinfeksi dan bertindak sebagai proxy. Dengan adanya metode ini, blokir IP address menjadi tidak mungkin dilakukan karena IP dan nama DNS berubah secara cepat. Yang lebih berbahaya lagi yaitu domain DNS berbahaya selalu di generate secara dinamis.bebapa oraganisasi keamanan computer berhasil melakukan *reverse engineering* terhadap the Dynamic-Domain Generation Algorithms(DGAs).

4.3 Human Interaction

Trojan.APT.BaneChan adalah salah satu dari sekian banyak jenis Trojan yang menunggu aktifitas sejumlah klik dari mouse sebelum mulai beroperasi. Beberapa malware menghitung dan menangkap pola waktu gerakan mouse, klik kanan pada mouse dan interaksi lainnya.malware mencoba untuk mendeteksi system otomatis sandbox. System sandbox harus semakin ditingkatkan kompleksitasnya dan bagaimana mereka membentuk perilaku acak dari user sehingga tidak bisa ditebak malware.

4.4 Environment-specific

Pada umumnya orang beranggapan mereka dapat mencegah website mengenali mereka dengan disable cookies pada web browser nya. Namun tidak sesederhana itu. Ketika kita mengakses website, kita memberikan aksses kepada website untuk mengakses berbagai informasii dari konfigurasi computer kita. Kombinasi tersebut akan menciptakan “fingerprint” atau tanda digital yang dapat dikenalai sebagai user dan computer dari user

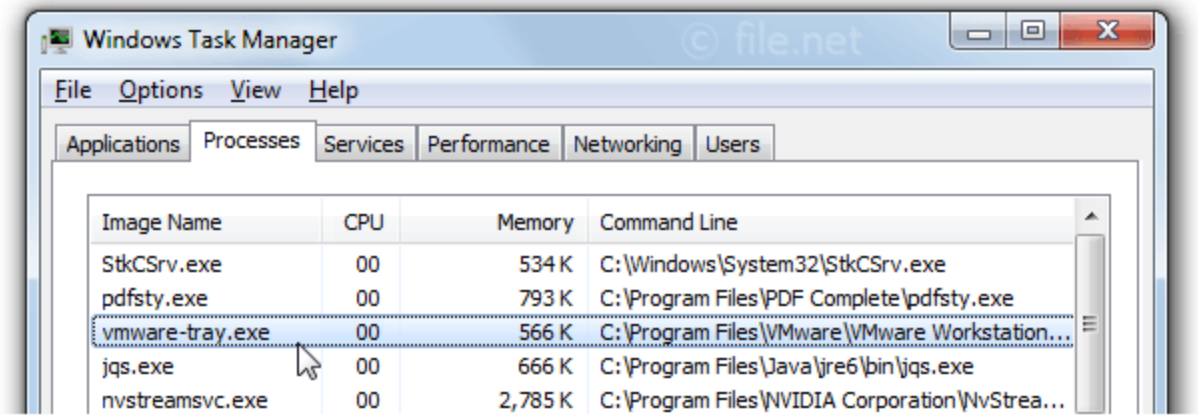
tersebut. Walaupun sampai saat ini belum ada malware yang diketahui menggunakan teknik ini. Namun teknik ini bisa diterapkan untuk malware melihat "fingerprint" unik tersebut apakah dari sandbox atau bukan.

4.5 VMware-specific

Malware bisa scan registry dan harddisk untuk deteksi berbagai tanda bahwa lingkungan tersebut adalah sandbox atau virtual machine. Malware dapat mendeteksinya melalui installasi VWware, registry entry atau proses yang mungkin sedang berjalan. Qemu, CWSandboc,Anubis, VMWare Fusion, VMWare Workstation dan Virtual Box adalah beberapa vmware yang sudah dikenali teknik anti VMware nya. Malware mendeteksi mereka berdasarkan nama proses yang mereka jalankan seperti vmusrvc.exe, vboxtray.exe, vmtoolsd.exe, df5serv.exe, and vboxservice.exe dan lain-lain. Rutin seperti Process32First ()/Process32next () juga sudah digunakan untuk mengidentifikasi keberadaan vmware. Sedangkan registry yang dijadikan bukti keberadaan sandbox pada HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Disk\Enum. Dengan subkey bernilai "0" merupakan parsing untuk mengidentifikasi keberadaan string atau substring seperti: qemu, Xen, dan jenis VMware lainnya[9].

Sandbox berkembang dengan cepat, begitu juga dengan attack terhadapnya. Pembuat malware juga melakukan reverse engineering terhadap sandbox system dan membangun metode untuk mendeteksi keberadaannya.

Gambar 2. Keberadaan vmware diketahui dari proses yang brejalan.



4.6 Teknik Evasive Terdepan

Dyre malware adalah salah satu malware yang dapat menghindari sandbox. Malware ini menghindari sandbox dengan cara memeriksa jumlah core dalam processor[10]. Setelah mendeteksi hanya ada satu core maka malware akan berhenti beroperasi sehingga tidak terdeteksi. Banyak sandbox di konfigurasi dengan processor tunggal untuk memungkinkan efektifitas mekanisme pengujian. Tidak dapat dipungkiri ide sederhana dan dikombinasikan dengan metode yang ada bisa menjadi cara yang ampuh untuk mengelabui sandbox.. Adanya kasus ini menandakan ketidakmampuan sandbox sebagai sebauh mekanisme mandiri yang mampu bekerja secara otomatis menghalau malware.

V. TEKNIK YANG DI ANJURKAN

5.1 Behavioral Observation

Solusi untuk mendeteksi dan mencegah malware yaitu harus mengkombinasikan pengamatan perilaku (Behavioral Observation) dengan analisis kodeseperti reverse engineering untuk bisa mengungkap ancaman malware jenis baru dan teknik adaptif penghindaran sandbox.

5.2 Treatment of Environment sensitive programs as malware

Setiap program/aplikasi pada lingkungan-sensitif harus diperlakukan seperti mengandung malware. Karena malware yang mengenali sandbox akan berhenti beroperasi sebelum terdeteksi. Meskipun sandbox adalah yang sangat penting dan efektif dalam menganalisis sejumlah besar malware. sandbox memiliki beberapa vulnerability yang dapat menjadi masalah serius jika sudah di pahami metodenya oleh pembuat malware.

5.3 Concentrating on servers issuing commands

Sebuah fakta bahwa sampel malware dapat merubah perilakunya berdasarkan informasi yang diterima dari C&C servers untuk analisis sandbox yang secara otomatis menghasilkan script untuk berinteraksi dengan remote server bukan sampel malware yang sebenarnya. Dengan memanfaatkan dummy client bukan dengan sampel

malware sebenarnya, maka observability dapat diulang untuk mengamati berbagai remote server secara bersamaan.

5.4 Gunakan Honeypots

Sampel malware dapat dikumpulkan menggunakan program antivirus, security appliances, atau menggunakan honeypot, yang bertindak sebagai umpan untuk memikat malware menyerang host rentan. Perangkap Honeypot dan kemudian aksploit kode malwarenya. Analisis malware bertujuan untuk mengumpulkan informasi lengkap tentang perilaku dan mekanisme operasional malware untuk pengembangan penanggulangan yang efektif. Hal ini bisa menjadi analisis dinamis maupun statis (white box). Analisis bisa dilakukan dengan membongkar binary dan menganalisis perilaku dan fungsi hingga pada level assembly.

5.5 Unpacking malware before analysis

Pembuat malware telah menemukan cara untuk mengaburkan malware dengan menggunakan the Ultimate Packer for eXecutables (UPX). Oleh karena itu para analis diharapkan dapat membongkar sampel malware yang dikemas terlebih dahulu dengan menggunakan metode seperti memori dump[9]. Analisis sandbox mengeksekusi sampel malware dalam sandbox di mana perilaku dari malware yang dianalisis dan dipantau. Metode ini tidak memiliki kesulitan menangani pengepakan dan teknik pengaburan kode.

5.6 Analysis malware while offline

Perangkap terbesar dari analisa sandbox adalah bahwa ia hanya dapat memantau dan menganalisis perilaku dari malware hanya ketika sedang sandbox running. Sandbox dapat benar-benar terisolasi atau hanya tanpa koneksi internet.

5.7 Observing, containing and efficiently giving back result

Analisis Sandbox malware memiliki tiga sifat penting: observability, containment, dan efificiency. Observasi yaitu menganalisis perilaku malware. Containment mengumpulkan kebocoran informasi penting malware dan mencegah sampel dieksekusi dari menginfeksi atau menyerang host remote di luar lingkungan sandbox. Efficiency melibatkan penyediaan hasil dengan analisis yang memadai dalam jangka waktu yang wajar.

Generate malware signature - paket software open source seperti ClamAV dan networkbased ID seperti Snort menganalisis sampel malware, ekstrak karakteristik string ketika generate signature untuk mendeteksi malicious kode. Pembuat malware biasanya menggunakan teknik packing techniques and polymorphic shellcodes untuk menghindari detection-based signature, program antivirus yang memanfaatkan mesin deteksi malware berbasis perilaku.

5.8 Using non-signature based detection

Banyak sistem operasi seperti Windows memiliki firewall dan software antivirus untuk mendeteksi dan menetralisir ancaman malware. Sehingga, beberapa malware mencari dan menghentikan proses software tersebut untuk menghindari deteksi. Perlindungan software ini menggunakan signature-based detection engines untuk deteksi malware.

VI. KESIMPULAN

Virtualisasi selama bertahun-tahun telah menjadi alat yang efektif untuk mengamati semua perilaku malware tanpa membahayakan sistem yang sebenarnya. Saat ini, attacker telah selangkah lebih maju ketika malware mereka mampu mendeteksi dan dan bersembunyi dari deteksi sandbox. Ancaman malware saat ini menuntut tidak hanya dorongan sepihak dari penggunaan sandbox namun juga beberapa beberapa metode keamanan sehingga dapat bekerja secara bersinergi. Kombinasi ini akan dapat mendeteksi malware baik dalam keadaan aktif atupun tersembunyi. Malware akan mudah terdeteksi, dicegah dan dinon-aktifkan.

VII. REFERENSI

[1] APJII and PUSKAKOM UI. (2015): Profil Pengguna Internet Indonesia 2014, Jakarta. 2015.

[2] Gartner.inc,"Gartner Says Worldwide Smartphone Sales Recorded Slowest Growth Rate Since 2013." Available: http://www.gartner.com/newsroom/id/3115517. [Accessed: 25-Apr-2016].

[3] G Data Software AG Germany: *G Data Mobile Malware Report, Threat Report :Q1/2015*, White paper Mobile Malware Report USV206-2015.2110290615, pp.4. 2015.

[4] A. Henry, "How Secure Is Android, Really?," *Lifehacker*. [Online]. Available: http://lifehacker.com/how-secure-is-android-really-1446328680. [Accessed: 25-Apr-2016].

[5] Novriadi. R, Novaria K. Y, Shaksono. P.H, (2014): *Analisis Forensik Malware Pada Platform Android*, Konferensi Nasional Ilmu Komputer (KONIK), *2014, pp. 377-385. 2015*.

[6] Kasama, Takahiro. (2014). A study on malware analysis leveraging sandbox evasive behaviors. Available http://kamome.lib.ynu.ac.jp/dspace/bitstream/10131/8598/1/kasama_takahiro-thesis.pdf

[7] Kruegel, C. (2015). Evasive Malware Exposed and Deconstructed | USA 2015 RSA Conference. Rsaconference.com. Available https://www.rsaconference.com/events/us15/agenda/sessions/2022/evasive-malwareexposed-and-deconstructed.

[8] Lakhani, Aamir. (2015) Malware Sandbox and breach detection evasion techniques. Dr. Chaos.Com, May 2015. Web. 17 August 2015. Available http://www.drchaos.com/malware-sandbox-and-breach-detection-evasion-techniques

[9] Singh, Sudeep. (2015). Breaking the sandbox. Exploit-db.Com. Available https://www.exploit-db.com/docs/34591.pdf

[10] Raff, Aviv. (2015) New dyre version –yet another malware evading sandboxes. Seculert, 30 April 2015. Available http://www.seculert.com/blog/2015/04/new-dyre-versionevades-sandboxes.html