TAXONOMY BOTNET DAN STUDI KASUS: CONFICKER

Adhitya Nugraha[1], Fauzi Adi Rafrastara[2]
[1]*Faculty of Information and Communication Technology, Universitas of Technical Malaysia Melaka, Malaysia 1752*
E-mail : adhitya_gro@yahoo.com

[2]*Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang 50131*
E-mail : fauzi_adi@yahoo.co.id

ABSTRAK

Botnet merupakan salah satu ancaman yang nyata bagi pengguna komputer saat ini, khususnya bagi mereka yang terhubung pada jaringan, baik itu jaringan lokal maupun global (internet). Botnet tergolong sebagai program berbahaya karena dia merupakan bentuk hybrid dari beberapa malware, dimana dia menggabungkan mekanisme Command & Control (C&C) sekaligus.
Untuk mengatasi ancaman botnet ini memang tidak mudah, mengingat diperlukan pengetahuan dan pemahaman yang mendalam terlebih dahulu tentang botnet itu sendiri. Dalam paper ini, penulis membahas tentang botnet beserta taxonominya. Pembahasan mengenai taxonomi ini akan memberikan gambaran tentang karakteristik yang dimiliki oleh botnet beserta penggolongannya. Selanjutnya, study case dilakukan untuk memperoleh gambaran detail tentang suatu botnet berdasarkan taxonominya. Dalam studi case ini, penulis menggunakan salah satu botnet yang cukup popular saat ini, yaitu Conficker. Dengan demikian, pembaca akan memperoleh gambaran yang lebih jelas tentang karakteristik botnet conficker tersebu.

Kata kunci : Botnet, Taxonomy, Conficker, C&C

1. PENDAHULUAN

Botnet menjadi salah satu ancaman paling serius terhadap keamanan internet. Hal ini disebabkan karena *Botnet* mampu menyediakan *platform* yang dapat didistribusikan pada kegiatan ilegal seperti serangan-serangan di internet, termasuk *spam, phishing, click fraud,* pencurian *password* dan *Distributed Denial of Service(DDoS) attack* [1,2,3].

Salah satu kemampuan dari *Botnet* yang membedakannya dari *malware* yang lain adalah *Botnet* dapat dikendalikan dari jauh oleh seseorang (*Botmaster*) dibawah suatu infrastruktur yang disebut *Command and Control (C & C) channel*. *Host* yang terinfeksi *malware* ini, atau biasa disebut *bot*, tidak secara fisik dimiliki oleh *Botmaster* dan mungkin terletak di beberapa lokasi yang mencakup seluruh dunia [1,3]. Perbedaan zona waktu, bahasa, dan hukum inilah yang membuat sulit melacak keberadaan dan aktivitas berbahaya dari *Botnet*. Karakteristik ini membuat *Botnet* menjadi alat yang menarik untuk kejahatan dan bahkan menimbulkan ancaman besar terhadap *cyber-security.*

Sampai saat ini, botnet pun terus mengalami perkembangan, mulai dari cara menginfeksi, pola perilaku, hingga teknik untuk mendeteksinya pun terus mengalami perkembangan juga. Untuk itulah, di dalam paper ini, penulis mencoba mengkaji penelitian-penelitian yang ada saat ini, untuk kemudian mengembangkan taxonomy yang sebelumnya telah diusulkan oleh [4]. Selanjutnya penulis mencoba menganalisa pola salah satu botnet yang cukup populer, yaitu Conficker, dengan memanfaatkan taxonomy yang penulis susun.

Di dalam paper ini, karakteristik dan siklus hidup botnet akan dijelaskan di bagian 2. Selanjutnya, pada bagian 3, akan dibahas taxonomy dari botnet. Bagian 4 merupakan case study dimana penulis menguji langsung botnet conficker, untuk kemudian menganalisanya. Bagian 5 merupakan kesimpulan sekaligus penutup dari paper ini.

2. KARAKTERISTIK DAN SIKLUS HIDUP BOTNET

Botnet menjadi salah satu ancaman paling serius terhadap keamanan internet. Hal ini disebabkan karena *Botnet* mampu menyediakan *platform* yang dapat didistribusikan pada kegiatan ilegal seperti serangan-

serangan di internet, termasuk *spam, phishing, click fraud,* pencurian *password* dan *Distributed Denial of Service(DDoS) attack* [1,2,5].

Salah satu kemampuan dari *Botnet* yang membedakannya dari *malware* yang lain adalah *Botnet* dapat dikendalikan dari jauh oleh seseorang (*Botmaster*) dibawah suatu infrastruktur yang disebut *Command and Control (C & C) channel*. *Host* yang terinfeksi *malware* ini, atau biasa disebut *bot*, tidak secara fisik dimiliki oleh *Botmaster* dan mungkin terletak di beberapa lokasi yang mencakup seluruh dunia [1,3]. Perbedaan zona waktu, bahasa, dan hukum inilah yang membuat sulit melacak keberadaan dan aktivitas berbahaya dari *Botnet*. Karakteristik ini membuat *Botnet* menjadi alat yang menarik untuk kejahatan dan bahkan menimbulkan ancaman besar terhadap *cyber-security*. Dalam rangka untuk memberikan pemahaman yang lebih baik tentang fenomena *Botnet*, karakteristik *Botnet* dan siklus hidup *Botnet* akan dijelaskan secara masing-masing.

2.1 Karakteristik Botnet

Seperti *malware* lainnya, *Botnet* adalah perangkat lunak yang dirancang untuk masuk atau merusak sistem komputer tanpa sepengetahuan pemilik. Hal ini dilakukan dengan cara memanfaatkan dan menginfeksi kelemahan suatu sistem dari sebuah *host*, kemudian mengeksploitasi kinerja dari sistem tersebut untuk keperluan pribadi ataupun memperluas jangkauan mereka.

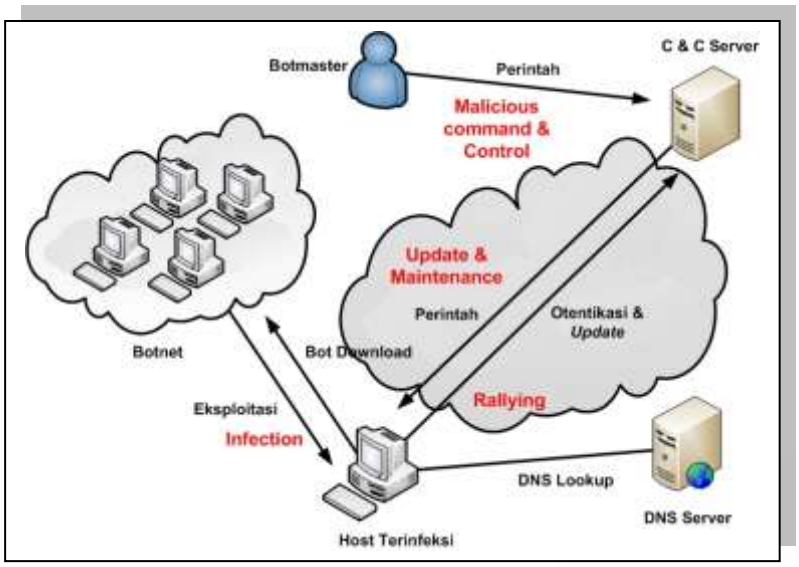
Perbedaan utama antara *Botnet* dan jenis *malware* lainnya adalah adanya infrastruktur *Command-and-Control (C & C)*. *C & C* memungkinkan sejumlah bot untuk dapat menerima perintah untuk melakukan *update* atau bahkan malakukan kejahatan seperti *DDOS attack, spaming* dan lainnya sebagaimana yang diinginkan oleh *Botmaster* [1,2,3,5].

Generasi pertama dari *Botnet* adalah *centralized C & C model* yang memanfaatkan *IRC (Internet Relay Chat)* protokol sebagai jalur komunikasi antara *C & C server* dan bot [1,2,3]. IRC protokol pada awalnya dirancang untuk komunikasi satu ke banyak (*one to many*). Hal ini dimanfaatkan *Botmaster* untuk mengontrol bot dalam jumlah banyak untuk memaksimalkan keuntungan mereka. Dalam implementasinya, C*entralized C & C model* memberikan kemudahan bagi *Botmaster*. Namun, C*entralized C & C model* membuat *Botnet* menjadi lebih rentan saat dideteksi sehingga mudah menemukan *C & C server* untuk kemudian menghancurkannya. Contoh *Botnet* ini adalah AgoBot, SDBot dan Zotob[2].

Dikarenakan adanya kelemahan dari C*entralized C & C model* ini, kemudian diciptakan *Botnet* generasi baru yang dapat menyembunyikan komunikasi *Botnet* dengan *C & C server*, yang disebut *Peer-to-Peer (P2P) model* [1,2,3]. Dibandingkan dengan *Centralized C & C model, P2P model* jauh lebih sulit untuk ditemukan dan dihancurkan, karena sistem komunikasi tidak sangat bergantung pada server tertentu saja. Menghancurkan satu atau bahkan sejumlah bot, tidak akan menyebabkan kerusakan dari seluruh *Botnet*. Beberapa *bot* seperti Phatbot dan Peacomm telah menggunakan komunikasi P2P sebagai alat untuk mengontrol *Botnet* [2].

2.2 Siklus Hidup Botnet

Dalam penyebarannya, *Botnet* memiliki siklus hidup yang terdiri dari 4 tahap, yaitu: *infection, rallying, malicious command and control, update and maintenance.*



Gambar 1: Siklus Hidup *Botnet*

Bot mampu mendistribusikan dirinya sendiri melalui jaringan internet dengan mencari celah terhadap komputer yang rentan dan tidak dilindungi agar dapat diinfeksi. *Malware* ini umumnya disebar oleh para *Botmaster* dengan menyusupkannya ke situs-situs legal yang telah dieksploitasi; instalasi software dari sumber yang tidak dipercaya; mengirimkan *spam mail* untuk memperdaya korban sehingga mengklik *link* tertentu; dan *backdoor* yang ditinggalkan oleh *virus* [1,3].

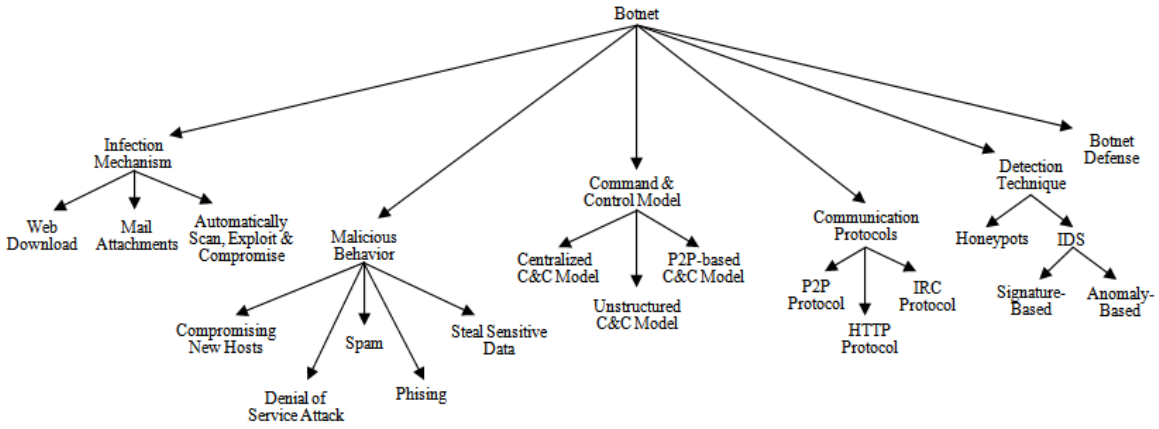
Setelah fase *infection* sukses, *host* yang terinfeksi akan mendownload *script code bot* dari sebuah *remote server* dan diinstal secara otomatis yang kemudian merubah host tersebut menjadi "zombie". Meskipun urutan siklus hidup kadang dapat bervariasi, di beberapa titik awal siklus hidup klien *Botnet* yang baru harus melaporkan diri ke "rumah" (C & C server), proses ini disebut *rallying.*

Proses *rallying* diawali dengan proses pencarian *C & C server* oleh *host* baru yang terinfeksi dengan menggunakan *DNS lookup*. Kebanyakan *Botnet* menggunakan *Internet Relay Chat (IRC) server* sebagai *C & C server* untuk mengontrol *Botnet* mereka. Setelah mendapatkan alamat *C & C server, bot* kemudian melakukan *login* dan mengotentikasikan dirinya sebagai bagian dari *Botnet* tertentu. Pada titik ini, *Botmaster* dapat mengeluarkan perintah kepada bot untuk melakukan serangan seperti *spamming, click fraud, DDoS attack* ataupun menyebarkan *malware* untuk memperluas jaringan mereka. Hal ini yang disebut *malicious command and control.*

Disisi lain, *Botmaster* dapat melakukan perintah *update* terhadap bot untuk melakukan pembaruan dengan cara mendownload *script code* yang baru untuk beberapa alasan seperti menambah fungsionalitas *bot army*, menghindari pendeteksian, atau memperbaharui alamat *C & C server*. Ini akan menjamin bahwa *bot* dapat dikelola dan dapat diamankan dari pendeteksian. *Bot* tetap tersembunyi sampai mereka diinformasikan oleh *Botmaster* untuk melakukan serangan atau tugas.

3. TAXONOMI BOTNET

Di dalam papernya, [4] melakukan survey terhadap botnet sekaligus mengklasifikasikannya ke dalam beberapa karakteristik, yaitu: infection mechanism, malicious behavior, command & control model, communication protocols, detection technique, dan botnet defense. Dalam bagian ini, akan dibahas penggolongan-penggolongan dari karakteristik botnet tersebut satu per satu.



Gambar 2: Taxonomi Botnet

3.1. Infection Mechanism

Menurut [4,1], ada tiga macam metode yang biasa digunakan oleh penyerang untuk menyebarkan bot mereka, yaitu:

1) *Web Download*: dengan memanfaatkan web-based malware, suatu bot akan bisa masuk ke dalam system computer dengan sangat mudah. Hasil studi google menunjukkan bahwa penyebaran botnet melalui website yang telah terinfeksi malware ini sudah menjadi hal yang biasa dan umum terjadi.
2) *Mail Attachments*: metode ini sangat mudah dan efektif untuk digunakan dalam menyebarkan suatu bot, yaitu dengan mengirimkan e-mail secara masal yang berisikan worm dan bot. Teknik spam semacam ini akan menyederhanakan sekaligus mempercepat proses penyebaran bots.
3) *Automatically Scan, Exploit & Compromise*: sebuah bot akan secara otomatis mencari dan menginfeksi host yang memiliki kelemahan atau celah keamanannya terbuka.

3.2. Malicious Behavior

Botnet dapat digunakan untuk melakukan berbagai macam aktifitas yang tidak baik, seperti yang disampaikan oleh [4,5,6,7], bahwa ada lima aktifitas utama botnet, yaitu:

1) *Compromising New Hosts*: untuk memperkuat jaringan botnetnya, para botmaster biasanya akan selalu merekrut host-host baru untuk dijadikan sebagai komputer zombie, dengan melalui teknik rekayasa sosial (*social engineering*) ataupun melalui penyebaran email-email bervirus.
2) *Denial of Service Attack*: DDoS merupakan fitur standard yang dimiliki oleh botnet. Setiap botnet selalu mengandung satu set mekanisme flooding, mulai dari SYN flood, ICMP flood, hingga HTTP flood, untuk mengirimkan paket-paket tersebut ke jaringan yang dituju, atau hanya sekedar mengirim ribuan http dan ftp yang sah ke suatu situs.
3) *Spam*: spam bots dapat menggunakan SMTP server untuk mengirimkan spam sesuai keinginan attacker. Sebagian besar dari email-email spam yang beredar saat ini merupakan ulah dari botnet. Contoh dari spam bots yang cukup populer saat ini ialah PhatBot.
4) *Phising*: dalam kebanyakan kasus, bots dapat digunakan sebagai *hosting* dari situs-situs phising. Seorang penyerang dapat mengekstrak informasi dari bot-bot mereka, yaitu dengan mengubahnya menjadi web server atau DNS server untuk melakukan phising.
5) *Steal Sensitive Data*: botnet yang berkembang saat ini sudah dilengkapi dengan peralatan canggih seperti *keylogger* dan *network traffic sniffer,* guna membantu pencurian data-data sensitif user. Nantinya, data-data yang telah tertangkap tersebut akan dikirim secara periodik dan otomatis ke botmaster.

3.3. Command & Control Model

Pemahaman tentang mekanisme Command & Control (C&C) pada botnet menjadi hal yang sangat penting dalam upaya kita melawan botnet. Di dalam papernya, [4,8] mengidentifikasi tiga kemungkinan topologi komunikasi C&C serta menginvestigasi kelebihan dan kelemahan masing-masing topologi tersebut.

1) *Centralized C&C Model* [8]: Centralized Model atau Model Tersentralisasi dikarakteriskan dengan titik pusat yang meneruskan pesan antara klien. Model tersentralisasi ini memiliki kelebihan, yaitu implementasi dan kustomisasi yang sederhana. Namun, kemudahan dan kesederhanaan yang menjadi keunggulan dari model ini, ternyata sekaligus jg menjadi titik lemahnya, mengingat model ini mudah untuk dideteksi dan dihancurkan. Ada beberapa botnet yang memanfaatkan kelemahan dari model ini, diantaranya yaitu: AgoBot [9], SDBot [10], Zotob.
2) *P2P-based C&C Model* [8]: melihat kelemahan yang ada di model tersentralisasi, selanjutnya botmaster mulai berpindah ke model P2P-based. Model ini lebih baik daripada model tersentralisasi khususnya dari segi keamanan botnet, dimana botnet akan lebih sulit utk ditemukan untuk kemudian dihancurkan. Namun demikian, model ini lebih sulit dari model sebelumnya, dimana harus mendesain P2P system terlebih dahulu. Contoh bot yang menggunakan model ini, diantaranya adalah: Phatbot [11] dan Peacomm [8].
3) *Unstructured C&C Model*: sebuah bot tidak akan secara aktif menghubungi bot-bot lain atau botmaster, dan dia akan memperhatikan setiap koneksi yang masuk dari botmaster. Seorang botmaster akan secara random mengecek internet dan akan meneruskan pesan terenkripsi ketika mendeteksi bot-bot lain.

3.4. Communication Protocol

Botnet biasanya menggunakan protocol komunikasi yang terdefinisikan dengan baik. Dengan mempelajari protocol komunikasi dapat membantu kita untuk menentukan asal mula serangan botnet dan membongkar aktifitas komunikasi yang dilakukan antara bot dengan botmaster. Di dalam paper [4,5], penulis mengklasifikasikan prokol komunikasi menjadi tiga kategori, yaitu:

1) *IRC Protocol*: protocol ini merupakan protocol komunikasi yang paling banyak digunakan oleh para botmaster untuk berkomunikasi dengan bot mereka. Protocol IRC ini utamanya didesain untuk komunikasi *one to many*, namun protocol ini juga dapat menghandle komunikasi *one to one*, yang mana akan sangat berguna bagi para botmaster untuk mengontrol botnet mereka. Namun demikian, perangkat keamanan computer dengan mudah dikonfigurasi untuk memblok trafik IRC.
2) *HTTP Protocol*: protocol ini juga merupakan protocol favorit botnet karena akan sulit untuk dideteksi. Dengan menggunakan protocol ini, biasanya botnet akan dapat dengan mudah melepati perangkat keamanan computer yang dipasang.
3) *P2P Protocol*: akhir-akhir ini, mulai banyak botnet-botnet yang menggunakan protocol P2P untuk komunikasi mereka. Sebagai contoh adalah varian terbaru dari Phatbot [9] dan Agobot [12], Nugache [13] dan Peacomm.

3.5. Detection Technique

Menurut [14], teknik pendeteksian botnet dibagi menjadi dua kelompok berdasarkan tool yang digunakan, yaitu *Honeypot* dan *Intrusion Detection System (IDS)*.

1) *Honeypot*: merupakan suatu sistem komputer khusus yang digunakan untuk menjebak seseorang yang berniat jahat terhadap suatu sistem. Dengan honeypot ini, selain akan menghadirkan sistem palsu kepada si penyerang, honeypot juga dapat digunakan untuk menggali informasi-informasi penting dari serang tersebut, diantaranya yaitu: (i) Sidik jari dari malware bot yang berguna dalam *content-based detection*. (ii) Informasi tentang botnet C&C server dan mekanismenya. (iii) Lubang keamanan yang belum dikenali yang memungkinkan suatu bot untuk melakukan penyerangan. (iv) Tool dan teknik yang digunakan oleh si penyerang. (v) Motivasi dari si penyerang dalam melakukan serangan tersebut. Meskipun honeypot ini memiliki kemampuan yang cukup baik untuk mengenali karakteristik botnet beserta teknologi yang digunakan, namun honeypot tidak dapat mendeteksi infeksi yang dilakukan oleh botnet setiap waktu. Selain itu, ada beberapa kendala yang muncul ketika menggunakan honeypot ini, yaitu: (i) Hanya aktifitas-aktifitas eksploitasi tertentu yang bisa dilacak. (ii) Honeypot tidak dapat digunakan untuk mencapture suatu bot yang menggunakan metode propagasi selain scanning, yaitu

seperti: spam dan web drive-by download. (iii) Honeypot hanya dapat memberikan laporan tentang infeksi terhadap suatu sistem yang memang telah diantisipasi dan diletakkan di dalam jaringan sebagai sistem palsu atau perangkap.

Oleh karena itu, honeypot tidak dapat memberikan laporan terhadap suatu komputer yang tidak diletakkan ke dalam mesin perangkap, meskipun telah terinfeksi oleh bot di dalam jaringan. Dengan demikian, dapat ditarik kesimpulan bahwa secara umum, dengan menggunakan teknik honeypot ini, kita masih harus menunggu sampai satu bot di dalam jaringan menginfeksi sistem kita, baru selanjutnya kita bisa melacak atau menganalisis serangan tersebut.

2) *Intrusion Detection System (IDS)*: Di dalam papernya, [14] membagi IDS ini menjadi dua grup lagi berdasarkan metode deteksinya, yaitu Signature-Based dan Anomaly-Based.

(i) *Signature-Based* menggunakan *signature* atau sidik jari dari botnet-botnet yang ada saat ini untuk mendeteksinya. Sebagai contoh, Snort [15] memiliki kemampuan untuk memonitor lalu lintas dalam suatu jaringan, sekaligus mampu menemukan sidik jari dari bot-bot yang ada. Namun, metode ini hanya mampu memdeteksi botnet-botnet yang sudah dikenal saja. Kelemahan yang paling nyata adalah ketidakmampuan metode ini ketika digunakan untuk mendeteksi *zero-day bots attack*. Salah satu contoh dari metode deteksi berdasarkan sidik jari ini ialah Rishi [16], dimana dia akan mencocokkan pola-pola nickname yang sudah dikenal dan digunakan oleh bot-bot IRC. Mengingat Rishi ini tergolong sebagai *signature-based detection technique*, maka teknik ini tidak mampu mendeteksi adanya nickname-nickname baru yang dimiliki oleh bot IRC yang belum dikenal.

(ii) *Anomaly-Based* jauh berbeda dari teknik sebelumnya, yaitu Signature-Based. Teknik ini dibangun untuk mengatasi masalah yang ada di teknik-teknik sebelumnya. Secara teknis, model ini mencoba mendeteksi keberadaan suatu botnet dengan mengamati anomali pada lalu lintas data di jaringan. Menurut [1], anomali-anomali yang perlu diperhatikan di sini, yaitu: lalu lintas data di jaringan terlalu padat, adanya lalu-lintas pada port-port yang tidak biasa, dan juga perilaku ganjil dari sistem. [17] mengusulkan anomaly-based sytem yang mengombinasikan *IRC Statistics* dengan *TCP Work Weight* untuk mendeteksi botnet IRC. Selain itu, [18] juga mengembangkan *anomaly-based passive analysis algorithm*, dimana algoritma ini mampu mendeteksi *IRC botnet controller* yang berjalan di port manapun (random) tanpa perlu mengetahui sidik jari atau bahkan binary-nya terlebih dahulu.

3.6. Botnet Defense

Berdasarkan pada struktur botnet, seorang defender atau user bisa mempertimbangkan metode berikut ini untuk mencegah serangan botnet. Yang pertama yaitu dengan menghubungi pemilik dari host yang terinfeksi botnet, untuk membersihkan komputer sekaligus mengupdate sistemnya supaya lebih aman. Namun, jumlah malware bot yang sudah terlalu banyak membuat cara pertama tersebut menjadi tidak praktis dan cenderung sia-sia. Metode berikutnya yang dapat digunakan oleh defender atau user, ialah bertujuan untuk menghancurkan infrastruktur nyata yang digunakan oleh botnet, seperti C&C server. Metode selanjutnya yaitu bertujuan untuk mengontrol botnet dan mengambil alih peran botmaster, untuk kemudian mematikan botnet. Sementara itu, penulis di paper [19] mengusulkan "a distributed content independent spam classification system" untuk melindungi diri dari serangan botnet dengan menggunakan spam.

4. CASE STUDY: CONFICKER

Berdasarkan [20], conficker bonet merupakan ancaman serius dalam dunia *cyber*. Conficker telah menjadi salah satu malware botnet baru yang memiliki kemampuan untuk menyembunyikan maupun meloloskan diri dari sitem keamanan bahkan oleh *honeynet* sekalipun sehingga menjadi sangat sulit untuk dideteksi dan dihindari. Tercatat sejak akhir November sampai Desember, conficker botnet telah menginfeksi lebih dari 1,5 juta alamat IP yang terinfeksi dari 206 negara. Berdasarkan studi kasus yang penulis lakukan, penulis akan secara ringkas membahas kemampuan dari conficker botnet itu sendiri, khsuusnya mengenai infection mechanism, malicious behavior, dan Command & Control (C&C).

A. Infection Mechanism

Botnet menyebar dengan cara mengeksploitasi kelemahan dari sistem komputer maupun jaringan. Adapun mekanisme infeksi yang dilakukan oleh conficker botnet adalah sebagai berikut.

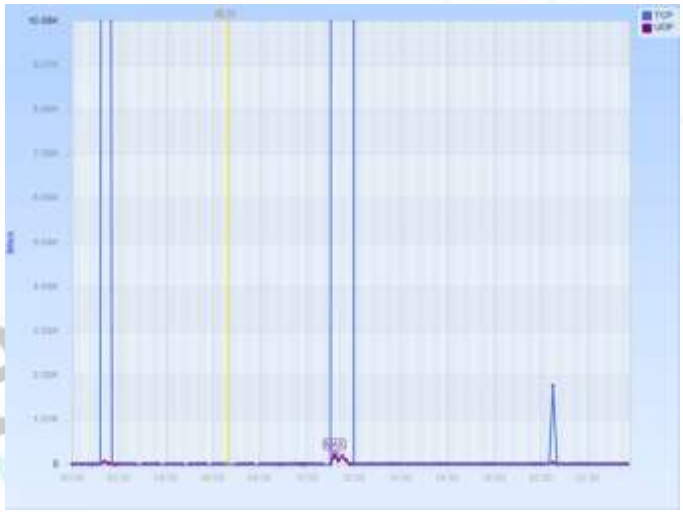
- Mengekploitasi kelemahan *Microsoft Windows Server Services (MS08-067)*. Malware ini mencoba untuk mengekploitasi *SMB (Server Message Block)* di port 445/TCP pada komputer korban. Malware akan menutup jalur yang biasa dilakukan *service sever* dan kemudian memaksa untuk memproses paket yang sudah diekploitasi sebelumnya. Setelah itu, komputer korban secara otomatis akan men*download dynamically linked library (DLL)* yang merupakan *script code bot.*
- Mengekploitasi *NetBIOS share* dengan cara memanfaatkan *weak password* untuk dapat menggandakan dirinya pada *admin share* ataupun *IPC (interprocess communication)*. Hal ini mengakibatkan conficker dapat menyebar melalui *network share* dalam jaringan.
- Mengekploitasi USB dengan cara menggandakan dirinya sebagai *autorun.inf* melalui *removable media*.

B. Malicious Behavior

Conficker memiliki kemampuan untuk melakukan *spaming* dan pencurian data ataupun informasi berharga dari computer yang terinfeksi.

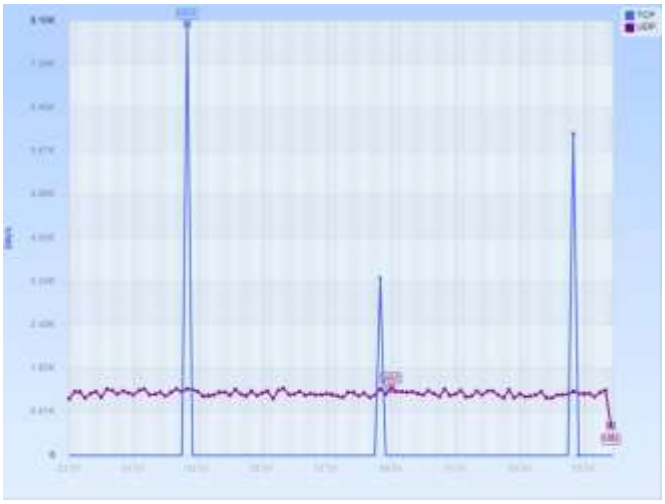
C. Command and Control

Conficker merupakan botnet yang tergolong dalam *P2P based C&C model. Signature* yang ditunjukan oleh botnet model ini adalah peningkatan jumlah lalu lintas paket pada DNS query dan HTTP (port 80) oleh satu IP address. Botnet ini juga memiliki pola untuk aktif tiap 5 menit, 10 menit, 30 menit, dsb. Saat aktif, botnet ini akan menghubungi alamat C&C server dan men*download bot binary* dari *remote server.* P2P model ini juga memungkinkan sejumlah *host* yang terinfeksi untuk dapat melakukan *peer-to-peer* komunikasi dan mengirimkan file yang ter-*update* antar sesama *host*. Hal ini berdampak meningkatnya lalu lintas paket dalam jaringan khususnya pada port UDP dan TCP. Gambar dibawah ini menunjukan perbandingan antara keadaan normal dan anomali botnet pada jaringan.



Gambar 3: lalu lintas normal pada UDP dan TCP

Pada gambar 3, paket TCP menunjukan aktifitas yang aktif walaupun hanya pengiriman paket kecil. Hal ini menunjukan tidak adanya pola khusus untuk aktifitas TCP. Saat TCP mencapai keadaan maximun, dimungkinkan dikarenakan banyak *host* melakukan browsing dan aktifitas download disaat yang bersamaan. Dalam gambar ini pula, UDP tidak menunjukan aktifitas pengiriman paket yang besar dan terus menerus. Data ini kemudian akan kita bandingkan dengan anomali jaringan pada botnet yang terinfeksi.



Gambar 4: lalu lintas abnormal pada UDP dan TCP

Pada gambar 2, paket TCP memiliki pola aktif setiap 6 jam dan setelah itu kembali ke keadaan "*sleep*". Keadaan "*sleep*" menunjukan tidak adanya aktifitas/paket transfer di *host* dikarenakan komputer host sudah sepenuhnya diekploitasi oleh *botnet* sehingga koneksi jaringan hanya aktif apabila *botnet* aktif. Terlihat pula pada gambar, paket UDP terus menunjukan adanya aktifitas yang aktif. Hal ini dikarenakan *host* yang melakukan *peer-to-peer connection* dengan *host* lainnya.

5. KESIMPULAN

Dalam paper ini, penulis telah mengkaji beberapa literature terbaru guna menyusun taxonomi botnet yang lebih detail. Dalam taxonomi ini, botnet diklasifikasikan ke dalam enam kategori, yaitu infection mechanism, malicious behavior, command & control model, communication protocols, detection technique, dan botnet defense. Selanjutnya, berdasarkan taxonomy tersebut, penulis melakukan studi kasus dengan menggunakan botnet conficker, dimana botnet ini merupakan salah satu botnet yang populer saat ini. Dari hasil pengujian yang penulis lakukan, akhirnya penulis memfokuskan analisa pada bagian infection mechanism, malicious behavior, dan command & control model, dimana diperoleh data bahwa botnet conficker, merupakan botnet dengan model infeksi *Automatically Scan, Exploit & Compromise*. Selanjutnya, diperoleh data bahwa botnet conficker ini memiliki kemampuan spamming dan mencuri data-data sensitive. Botnet conficker juga terdeteksi sebagai botnet yang menggunakan model *P2P based C&C*.

DAFTAR PUSTAKA

[1] Saha B, Gairola A. "Botnet: An Overview." *CERT-In White Paper, CIWP-2005-05*. 2005.

[2] Bacher P, Holz T, Kotter M, Wicherski G. "Know your enemy: Tracking Botnets." *The Honeynet Project*. 2005:1-21.

[3] Zhu Z, Lu G, Chen Y, et al. "*Botnet* Research Survey." *32nd Annual IEEE International Computer Software and Applications Conference*. 2008:967-972.

[4] Li C, Jiang W, Zou X. "Botnet: Survey and Case Study." *In: Proceeding of the Fourth International Conference on Innovative Computing Information and Control*. IEEE; 2009

[5] Trend Micro. "Taxonomy of Botnet threats." *Trend Micro White Paper, Tech. Rep*. 2006; (November):1-15.

[6] Rajab MA, Zarfoss J, Monrose F, and Terzis A. "A Multifaceted Approach to Understanding the Botnet Phenomenon," *IMC'06.* ACM. 2006

[7] The Honeynet Project & Research Alliance, "Know your enemy: Tracking botnets," http://www.honeynet.org, March 2005.

[8] Cooke E, Jahanian F, and McPherson D. "The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets," *Usenix Workshop on Steps to Reducing Unwanted Traffic on the Internet*, July 2005.

[9] Sophos. Troj/Agobot-A. http://www.sophos.com/virusinfo/analyses/trojagobota.html, 2002.

[10] Sophos. Troj/SDBot. http://www.sophos.com/virusinfo/analyses/trojsd bot.html, 2002.

[11] Phatbot Trojan Analysis. http://www.secureworks.com/research/threats/phatbot
[12] Nazario J. "Nugache: TCP port 8 Bot", May, 2006. http://asert. arbornetworks.com/2006/05/nugache-tcp-port-8-bot/.
[13] Suenaga M and Ciubotariu M. "Symantec: Trojan.peacomm." http://www.symantec.com/securityresponse/writeup.jsp?docid=2007-011917-1403-99, February 2007.
[14] Zeidanloo HR, Shooshtari MJZ, Amoli PV, Safari M, Zamani M. "A Taxonomy of Botnet Detection Techniques." *In: Proceeding of the International Conference on Computer Science and Information Technology*. IEEE. 2010.
[15] Snort IDS web page. http://www.snort.org, March 2006
[16] Goebel J and Holz T. "Rishi: IdentifY bot contaminated hosts by IRC nickname evaluation". *In Proceedings of USENIX HotBots'07*, 2007.
[17] Binkley JR and Singh S. "An algorithm for anomaly-based botnet detection", *In Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI),* July 2006, pp. 43–48.
[18] Karasaridis A, Rexroad B, Hoeflin D. "Wide-scale Botnet detection and characterization." *In: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets. USENIX Association*; 2007:7–7.
[19] Brodsky A and Brodsky D. "A distributed content independent method for spam detection", *In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007).* 2007.
[20] http://community.norton.com/t5/Norton-Protection-Blog/Cybercrime-News-Conficker-is-Spamming-Weak-Economy-Drives-Crime/ba-p/120346