Analisis Malware Botnet Proteus Pendekatan Static dan Dinamic

Indra Gunawan [a*], Andrey Ferriyan [b]

[a] Sekolah Tinggi Teknologi Ronggolawe Cepu
[b] R&D Security Assalam Teknologi Yogyakarta
[*] Penulis Korenspondensi, Alamat Email: igunsttr@gmail.com

Diterima : 29 Maret 2021 - Direview : 29 Juni 2021 - Diterbitkan : 12 Juli 2021

Intisari

Botnet Proteus dianggap sebagai salah satu botnet yang mengerikan dampaknya terhadap dunia internet di kisaran tahun 2017, botnet ini mulai dideteksi akhir tahun 2016 dan terus menyebar setelahnya. Proteus adalah botnet yang sample virusnya sulit didapatkan sehingga sangat sedikit penelitian yang membahasnya. Dalam penelitian ini digunakan pendekatan static dan dinamis untuk proses analisis terhadap botnet, pendekatan static dilakukan dengan analisis file virus secara langsung, sedangkan pada pendekatan dynamic, analisa dilakukan melalui jaringan terhadap paket-paket. Hasil analisis menunjukkan bahwa botnet ini menggunakan fitur-fitur Obfuscated, Cryptology, berbasis .Net, memiliki Command & Center, memodifikasi registry, memodifikasi file sistem windows, hanya aktif jika terdapat koneksi internet, tidak melakukan perusakan terhadap file-file yang ada. Kesimpulan akhir dari analisis tersebut menunjukkan bahwa botnet proteus dapat di kategorikan sebagai malware yang lebih bersifat sebagai malware pencuri data serta malware pembuat dan pengontrol zombie computer.

Kata Kunci proteus, botnet, analisis malware.

Abstract

Botnet Proteus is considered one of the horrible botnets of its impact on the internet world in the range of 2017, this botnet began to be detected at the end of 2016 and continued to spread afterwards. Proteus is a botnet that the virus samples are difficult to obtain so that very few studies discuss it. In this study the Static and dynamic approach was used for the analysis process of botnets, a static approach was carried out by analysis of virus files directly, while in the dynamic approach, the analysis was carried out through the network against packages. The results of the analysis show that this botnet uses obfuscated features, cryptology, .net-based, has a command & center, modifying the registry, modifying Windows system files, only active if there is an internet connection, it does not damage the existing files. The final conclusion of the analysis shows that the Proteus botnet can be categorized as a malware that is more malware as a malware of data thieves and malware makers and controller zombie computers

Keywords proteus, botnet, malware analysis.

1. Pendahuluan

Jumlah botnet yang terdeteksi dalam kurun waktu 2017 ini semakin meningkat secara signifikan, laporan yang dikeluarkan oleh majalah infosecurity menunjukkan adanya peningkatan sebesar 37% dibanding tahun sebelumnya. Terdeteksi sebanyak 9500 botnet server didalam 1122 jaringan yang berbeda. Jika peningkatan ini dibandingkan dengan laporan pada tahun 2014 maka menunjukkan angka sebesar 90% (Seals, 2017). Angka ini cukup mengkhawatirkan sehingga diperlukan lebih banyak lagi penelitian dibidang malware khususnya terhadap botnet.

Analisa terhadap malware secara umum dilakukan mengunakan tiga metode yaitu metode static, metode dynamic dan metode hybrid (Gandotra, Bansal, & Sofat, 2014). Metode analisis yang digunakan dalam penelitian ini adalah metode static dan metode dynamic, penggabungan metode ini tidak banyak dilakukan oleh para peneliti karena luasnya cakupan dan banyaknya perkakas yang digunakan serta membutuhkan waktu yang tidak sedikit, akan tetapi dilain sisi metode ini menghasilkan analisis yang detil dan menyeluruh (Yusirwan, S, Prayudi, & Riadi, 2015).

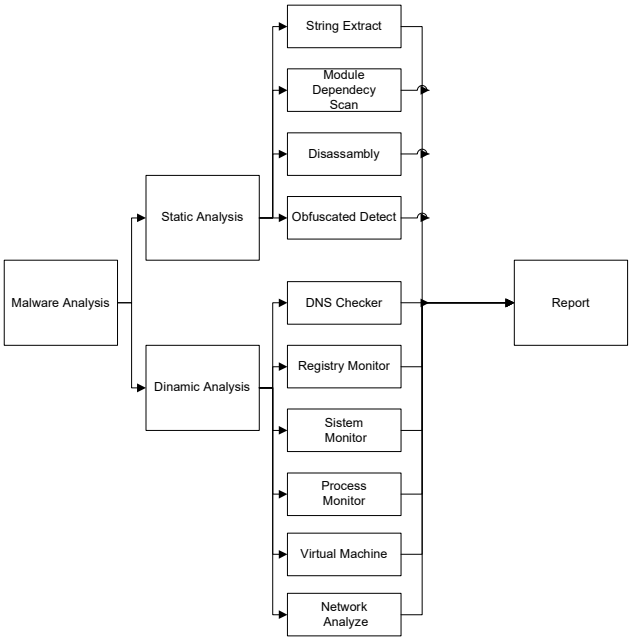
Penelitian ini menggunakan malware botnet proteus sebagai obyek analisis. Metode analisis yang dilakukan meliputi metode static dan metode dynamic. Metode static dilakukan dengan cara melakukan analisis malware secara langsung dengan skenario dibuat sistem virtual yang terinfeksi oleh malware botnet proteus, sedangkan metode dynamic dilakukan dengan cara dilakukan analisis terhadap jaringan pada sistem virtual terhadap paket-paket data lewat pada kartu jaringan. Telah ada beberapa penelitian terkait tentang analisis malware menggunakan metode static,

dynamic dan hybrid. Penggunaan metode static telah dilakukan oleh beberapa penelitian yaitu. Selanjutnya penggunaan metode dynamic telah dilakukan oleh (Afonso, de Amorim, Grégio, Junquera, & de Geus, 2015; Bulazel & Yener, 2017; Yan, Zheng, Jiang, Lou, & Hou, 2015). Sedangkan penggunaan metode static dan dynamic telah dilakukan oleh (Spreitzenbarth, Schreck, Echtler, Arp, & Hoffmann, 2015; YusirwanS, Prayudi, & Riadi, 2015). Selanjutnya metode hybrid telah dilakukan oleh beberapa peneliti yaitu (Abdullah, 2014; Abdullah, Syahirah, Faizal, Noh, & Azri, 2014; Martinelli, Mercaldo, & Saracino, 2017; Obeidat, 2017; Satrya, Cahyani, & Andreta, 2015). Dari semua penelitian diatas belum ada penelitian yang membahas tentang penggunaan metode static dan dynamic yang menggunakan obyek botnet proteus sehingga penelitian ini penting.

2. Metode Analisis

Metode analisis yang digunakan pada penelitian ini adalah metode static dan dynamic. Metode static dilakukan dengan cara menganalisis malware secara langsung yaitu dengan cara membuat system virtual yang diinfeksi dengan malware proteus, selanjutnya dilakukan analisis menggunakan aplikasi aplikasi yang telah diinstal pada sistem virtual yang terinfeksi. Metode dynamic dilakukan dengan cara menganalisis jaringan sistem virtual yang terinfeksi menggunakan aplikasi network analyzer.

Langkah-langkah metode analisis tersebut dapat dilihat pada Gambar 1 dibawah ini.



Gambar 1. Metode analisis malware

2.1 Static Analysis

2.1.1 String Extract

String Extract adalah suatu metode ekstraksi file untuk mencari string-string yang dicurigai tidak lazim dan mengandung informasi yang diperlukan (Gandotra, Bansal, & Sofat, 2014).

2.1.2 Module Dependency Scan

Module Dependency Scan adalah suatu metode pengecekan suatu file untuk mendapatkan informasi module-module yang dibutuhkan oleh suatu file malware (YusirwanS et al., 2015).

2.1.3 Disassamble

Disassembly adalah metode mengubah suatu file executable menjadi file assembly untuk mencari perintah-perintah yang dicurigai digunakan oleh malware untuk melakukan perusakan terhadap sistem (Bulazel & Yener, 2017).

2.1.4 Obfuscated Detect

Obfuscated adalah metode untuk mendeteksi pengaburan kode program yang dibuat oleh pembuat virus, umumnya pembuat virus akan mengacak kode programnya agar kode program virus tersebut sulit untuk dibaca (Abdullah, 2014).

2.2 Dynamic Analysis

2.2.1 DNS Checker

DNS Checker adalah metode untuk mendeteksi perubahan DNS sistem yang terinfeksi malware, terdapat beberapa malware yang mengubah dan menambah DNS pada sistem yang diinfeksinya untuk memudahkan malware terkoneksi dengan server kontrolnya (Karim, Salleh, & Khan, 2016).

2.2.2 Registry Monitor

Registry Monitor adalah suatu metode untuk mendeteksi perubahan registry sistem berbasis windows yang dilakukan oleh malware, metode ini dilakukan dengan cara membandingkan registry sebelum terinfeksi dan sesudah terinfeksi (Gandotra et al., 2014).

2.2.3 System Monitor

System Monitor adalah suatu metode untuk mendeteksi perubahan pada sistem operasi yang dilakukan oleh malware (Bulazel & Yener, 2017).

2.2.4 Process Monitor

Process Monitor adalah suatu metode untuk mendeteksi proses-proses yang sedang dijalankan oleh suatu malware (Abdullah, 2014).

2.2.5 Virtual Machine

Virtual Machine adalah suatu metode analisis malware yang dilakukan dengan cara membuat suatu sistem virtual yang diinfeksi dengan malware yang sedang diteliti sehingga sehingga didapatkan situasi yang sebenarnya, dan tidak merusak peralatan lainnya (Gandotra et al., 2014).

2.2.6 Network Analyze

Network Analyze adalah suatu metode analisis malware yang dilakukan pada level jaringan, umumnya metode ini dilakukan dengan menggunakan aplikasi Wireshark Network Analyzer. Analisa dilakukan paket per paket setiap data yang lewat dalam jaringan tersebut (Gandotra et al., 2014).

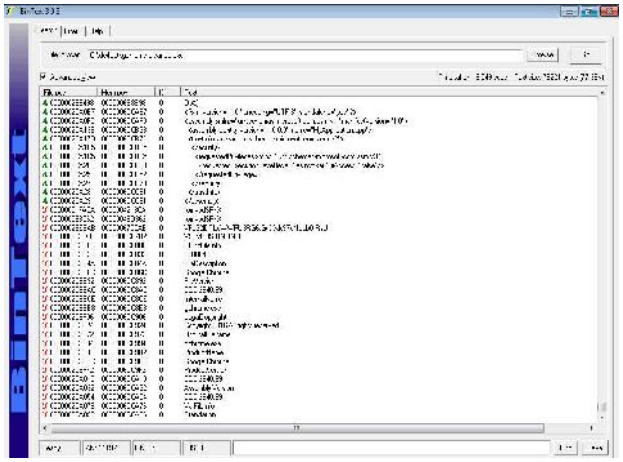
3. Hasil dan Pembahasan

3.1 Static Analysis

Static Analysis dilakukan dengan cara melakukan analisis terhadap file executable malware secara langsung.

3.1.1 String Extract

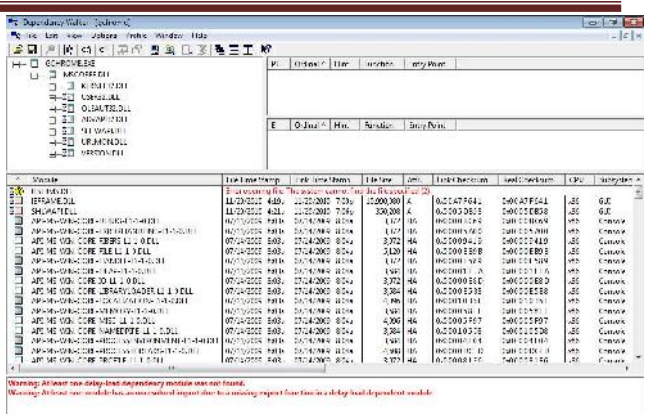
Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi BinText untuk mendapatkan string-string yang dicurigai. Hasil analisis tersebut dapat dilihat pada Gambar 2 dibawah ini.



Gambar 2. String Extract menggunakan BinText

3.1.2 Module Dependency Scan

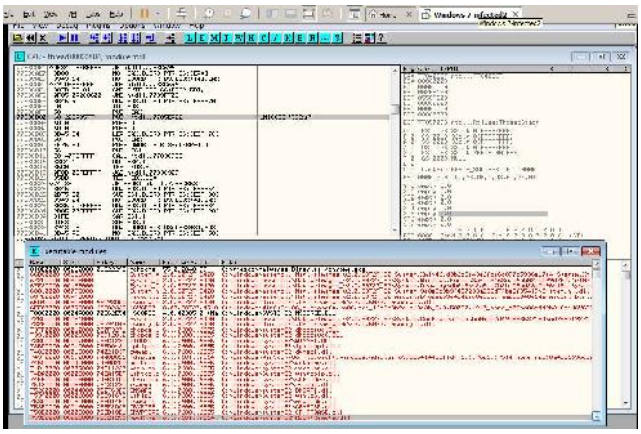
Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi Dependency Walker untuk mendapatkan module-module yang digunakan malware. Hasil analisis tersebut dapat dilihat pada Gambar 3 dibawah ini.



Gambar 3. Dependency Check menggunakan Dependency Walker.

3.1.3 Disassembly

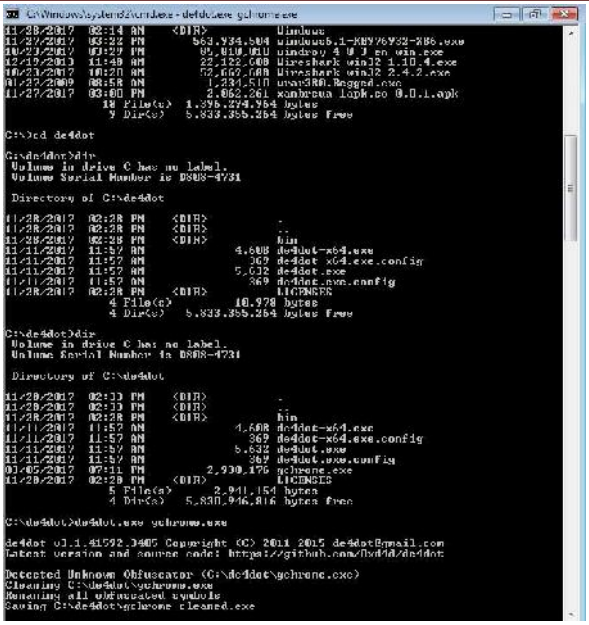
Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi Odbg untuk mendapatkan perintah-perintah assembly yang dicurigai. Hasil analisis tersebut dapat dilihat pada Gambar 4 dibawah ini.



Gambar 4. Disassembly menggunakan Odbg.

3.1.4 Obfuscated Detect

Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi De4dot untuk mendeteksi pengaburan pada file malware. Hasil analisis tersebut dapat dilihat pada Gambar 5 dibawah ini.



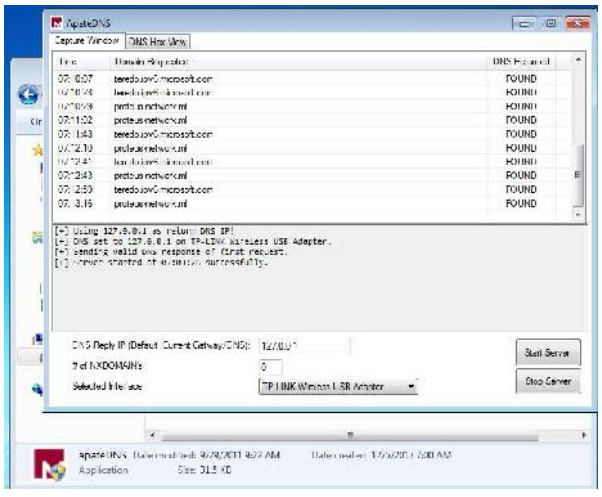
Gambar 5. Deobfuscated menggunakan De4dot.

3.2 Dynamic Analysis

Dynamic Analysis dilakukan dengan cara menjalankan malware pada sistem virtual sehingga sistemnya terinfeksi, setelah sistem terinfeksi kemudian dilakukan analisis dan pengamatan terhadap perubahan-perubahan yang terjadi pada sistem tersebut.

3.2.1 DNS Checker

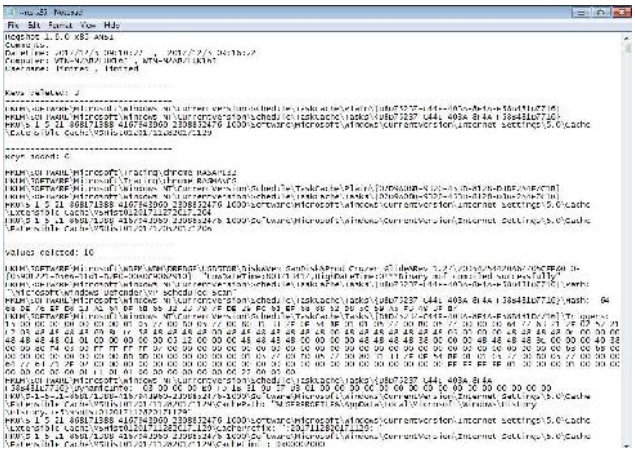
Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi ApateDNS untuk mendeteksi perubahan pada DNS Client sistem yang terinfeksi. Hasil analisis tersebut dapat dilihat pada Gambar 7 dibawah ini.



Gambar 6. DNS Check menggunakan ApateDNS.

3.2.2 Registry Monitor

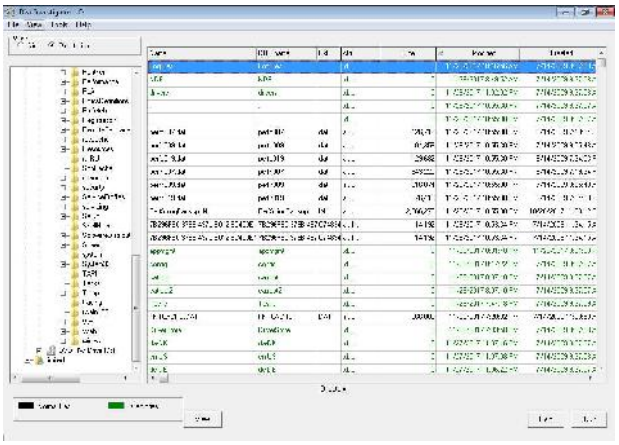
Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi Regshot untuk mendeteksi perubahan pada registry setelah sistem virtual terinfeksi. Hasil analisis tersebut dapat dilihat pada Gambar 7 dibawah ini.



Gambar 6. Registry Monitor menggunakan Regshot.

3.2.3 System Monitor

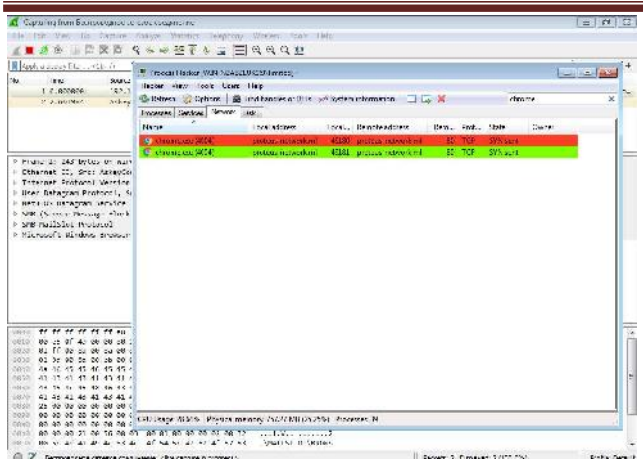
Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi Disk Investigator untuk mendeteksi perubahan pada sistem windows. Hasil analisis tersebut dapat dilihat pada Gambar 7 dibawah ini.



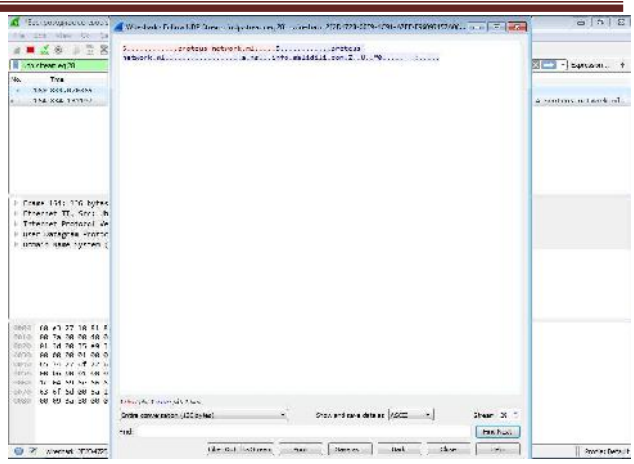
Gambar 7. System Monitor menggunakan Disk Investigator.

3.2.4 Process Monitor

Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi Process Hacker untuk mendeteksi proses yang sedang dijalankan oleh malware. Hasil analisis tersebut dapat dilihat pada Gambar 8 dibawah ini.



Gambar 8. Process Monitor menggunakan Process Hacker



Gambar 1. Network Analyze menggunakan Wireshark.

3.2.5 Virtual Machine

Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi VMWare. Dengan mengunakan aplikasi VMWare dapat dibuat sistem virtual yang dapat disimulasikan seperti keadaan nyata, sistem guest yang diinfeksi tidak akan berdampak pada sistem lainnya. Hasil analisis tersebut dapat dilihat pada Gambar 9 dibawah ini.

Gambar 9. Virtual Machine menggunakan VMWare.

3.2.6 Network Analyze

Analisis yang dilakukan pada tahap ini adalah menggunakan aplikasi Wireshark untuk mendapatkan string-string dan akses-akses data yang dicurigai. Hasil analisis tersebut dapat dilihat pada Gambar 10 dibawah ini.

4. Simpulan

Kesimpulan yang didapatkan dari analisa yang telah dilakukan adalah sebagai berikut: Malware botnet proteus dapat dikategorikan sebagai malware pencuri data dan remote botnet, dimana malware ini tujuan dibuatnya adalah bukan untuk merusak sistem yang diinfeksi melainkan malware ini mengontrol sistem korban agar dapat diremote dari jarak jauh yang pada akhirnya akan digunakan oleh pembuat malware untuk mendapatkan manfaat yang berhubungan dengan kekayaan. Termasuk didalamnya adalah fungsi pencurian data, bitcoin mining, Keylogging, dan pengaksesan kartu kredit. Fungsi-fungsi lainnya dapat dilakukan lebih jauh oleh pembuat malware dengan cara pengontrolan terhadap malware dari jauh.

Dari hasil analisis yang dilakukan perlu adanya proses monitoring secara lebih mendalam dan waktu yang lebih lama sehingga dapat diketahui fungsi-fungsi malware lebih luas, dan dapat diidentifikasi dampak yang akan terjadi.

Daftar Pustaka

Abdullah, R. S. (2014). Tracing the P2P Botnets Behaviours via Hybrid Analysis Approach, *118*(1), 75–85.

Abdullah, Syahirah, R., Faizal, Noh, M., & Azri, Z. (2014). EBSCOhost | 98259213 | Enhanced P2P Botnets Detection Framework Architecture with Hybrid Analysis Approach. *Journal of Information Assurance & Security*, *9*(2), 62–71.

Afonso, V. M., de Amorim, M. F., Grégio, A. R. A., Junquera, G. B., & de Geus, P. L. (2015). Identifying Android malware using dynamically obtained features. *Journal of Computer Virology and Hacking Techniques*, *11*(1), 9–17. https://doi.org/10.1007/s11416-014-0226-7

Bulazel, A., & Yener, B. (2017). A Survey On Automated Dynamic Malware Analysis Evasion and Counter-Evasion. *Proceedings of the 1st Reversing and Offensive-Oriented Trends Symposium on - ROOTS*, 1–21. https://doi.org/10.1145/3150376.3150378

Gandotra, E., Bansal, D., & Sofat, S. (2014). Malware Analysis and Classification: A Survey. *Journal of Information Security*, *05*(02), 56–64. https://doi.org/10.4236/jis.2014.52006

Karim, A., Salleh, R., & Khan, M. K. (2016). SMARTbot: A behavioral analysis framework augmented with machine learning to identify mobile botnet applications. *PLoS ONE*, *11*(3), 1–35. https://doi.org/10.1371/journal.pone.0150077

Martinelli, F., Mercaldo, F., & Saracino, A. (2017). BRIDEMAID: An Hybrid Tool for Accurate Detection of Android Malware. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security - ASIA CCS '17*, 899–901. https://doi.org/10.1145/3052973.3055156

Obeidat, A. A. (2017). Hybrid Approach for Botnet Detection Using K-Means and K-Medoids with Hopfield Neural Network, *9*(3), 1–9.

Satrya, G. B., Cahyani, N. D. W., & Andreta, R. F. (2015). The Detection of 8 Type Malware botnet using Hybrid Malware Analysis in Executable File Windows Operating Systems. In *Proceedings of the 17th International Conference on Electronic Commerce 2015 - ICEC '15* (pp. 1–4). New York, New York, USA: ACM Press. https://doi.org/10.1145/2781562.2781567

Spreitzenbarth, M., Schreck, T., Echtler, F., Arp, D., & Hoffmann, J. (2015). Mobile-Sandbox: combining static and dynamic analysis with machine-learning techniques. *International Journal of Information Security*, *14*(2), 141–153. https://doi.org/10.1007/s10207-014-0250-0

Yan, Q., Zheng, Y., Jiang, T., Lou, W., & Hou, Y. T. (2015). PeerClean: Unveiling peer-to-peer botnets through dynamic group behavior analysis. *Proceedings - IEEE INFOCOM*, *26*, 316–324. https://doi.org/10.1109/INFOCOM.2015.7218396

YusirwanS, S., Prayudi, Y., & Riadi, I. (2015). Implementation of Malware Analysis using Static and Dynamic Analysis Method. *International Journal of Computer Applications*, *117*(6), 11–15. https://doi.org/10.5120/20557-2943