 ISSN:2302-8149
e-ISSN:2540-9719

Deteksi Malware pada Jaringan Lokal Berbasis Honeypot dan Yara

Malware Detection on Local Network based on Honeypot and Yara

[1]Nur Rohman Rosyid*, [2]Budi Bayu Murti, [3]Brama Prayudha, [4]Arul Ferian Ramadloni, [5]Lukman Subekti
[12345]Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada
Jalan Gedung Herman Yohanes, Sekip Unit I, Bulaksumur, Yogyakarta, Indonesia
*e-mail: nrohmanr@ugm.ac.id

(***received*:** 21 November 2022, ***revised*:** 12 Desember 2022, ***accepted*:** 24 Desember 2022)

Abstrak

Ancaman malware tidak pernah surut, bahkan tren memperlihatkan peningkatan dan bervariasi seiring dengan perkembangan teknologi *hardware* dan *software*. Pengguna akhir mungkin tidak menyadari jika mesin mereka telah terkompromi oleh malware. Hal ini dapat dikarenakan mekanisme anti-malware tidak bekerja dengan baik, misalkan anti-virus yang tidak mutakhir atau memang ada *zero-day attack*. Oleh karena itu dibutuhkan pendeteksian adanya malware pada piranti pengguna akhir pada jaringan lokal yang dapat menemukan malware yang telah terdefinisi maupun yang belum (dimungkinkan *zero-day attack*). Pemanfaatan honeypot sebagai sensor keamanan pengumpul data serangan malware yang berupa file malware dan malware *hash* dapat digunakan sebagai *signature* dalam pemindaian dan pendeteksian malware. Penelitian ini memanfaatkan honeypot sebagai sensor keamanan yang akan menangkap malware, selanjutnya kode hash malware dari honeypot digunakan untuk memindai dan mendeteksi adanya malware pada *end-system* di jaringan local seperti PC atau Server. Selanjutnya Yara akan membantu memperjelas jenis malware yang ditemukan dengan melakukan pemindaian pada file yang terindikasi malware tersebut. Hasil pemindaian dan deteksi malware oleh Yara dilaporkan ke pihak yang berwenang via aplikasi Telegram *channels* sebagai peringatan dini adanya ancaman keamanan. Sistem keamanan ini memberikan kontribusi mengumpulkan kode hash dari malware yang sedang melakukan serangan saat itu.

Kata kunci: Honeypot, Malware, Yara, Keamanan Proaktif.

Abstract

The malware threats have never subsided, even the trend shows an increase and varies along with the development of hardware and software technology. End user may not realize if their machine is compromised by malware. It could be the anti-malware mechanism is not working properly, such as the anti-virus is not updated or there is a zero-day attack. Therefore, it is necessary to detect the presence of malware on end-systems devices or the existence of zero-day attack in the local network. Implementation of honeypot as a security sensor that collects malware attack data in the form of malware files and malware hashes can be used as signatures for scanning and detecting malware. This research utilizes a honeypot as a security sensor to catching malware. The malware hash from the honeypot is used to scanning and detecting the presence of malware on the end-system in a local network such as a PC or server. Furthermore, Yara helps clarify the type of malware found by scanning suspected files. The results of scanning and detecting of malware by Yara will be reported to the appropriate authorities via Telegram application channles. This research contributes by providing early warning of potential security threats to the network and collecting hash code of recently malware attacking to the network.

Keywords: *Honeypot, Malware, Yara, Proactive Security.*

ISSN:2302-8149
e-ISSN:2540-9719

1 Pendahuluan

Ancaman malware Internet selalu ada dan terus bekembang baik metode penyebarannya maupun teknologi perangkat lunaknya. Kemunculan serangan baru dan variasi tipe ancaman yang datang dari malware dapat menurunkan unjuk kerja sistem, pada tahun 2020 dilaporkan bahwa sekitar 230,000 serangan baru setiap harinya[1]. Menurut Microsoft pada tahun 2019 Indonesia menempati peringkat kedua di Asia Pasifik terkait serangan siber mengunakan malware sekaligus menempati peringkat kedua terkait kasus serangan ransomware. Salah satu kasus ransomware yang sangat terkenal dan berbahaya yaitu Ransomware WannaCry yang merupakan malware yang dapat memblokir akses pengguna ke berkas atau sistem, menahan perangkat menggunakan enkripsi sampai korban membayar untuk mendapatkan kunci dekripsi. Kondisi ini meningkatkan tantangan terhadap perlawanan serangan yang juga semakin sulit. Analisis keamanan menjadi rekomendasi nomor satu untuk dapat mengantisipasi banyaknya serangan. Hal ini didasari oleh banyaknya data serangan yang belum dianalisis dan dimanfaatkan secara optimal. Penerapan sensor keamanan seperti Honeypot berfungsi mengumpulkan data serangan yang selanjutnya dapat dimanfaatkan untuk identifikasi malware [2] bahkan di piranti IoT [3]. Honeypot sendiri adalah mesin yang didedikasikan untuk diserang sehingga tahapan-tahapan serangan yang dilakukan oleh penyerang dapat direkam oleh honeypot, sehingga log serangan dapat dianalisis lebih jauh [4]. Data serangan yang masuk ke Honeypot dapat berupa indikasi adanya serangan (indication of compromise (IoC)) dengan variasi tingkat keparahan. Pemanfaatan IoC yang direkam oleh honeypot diharapkan dapat membantu mengidenfitikasi adanya serangan pada jaringan lokal.

Metode pemindaian merupakan metode yang populer diterapkan pada piranti akhir pengguna seperti PC, laptop, server, dll. dalam bentuk anti-virus. Namun efektifitas pemindai pada aplikasi anti-virus yang dipasang pada piranti akhir pengguna tergantung dari kebaruan pustaka malware. Kemutakhiran honeypot dalam menangkap serangan-serangan terkini memberikan harapan besar untuk membantu mengidentifikasi adanya serangan baru atau dikenal dengan *zero-day attack*. Penelitian ini memanfaatkan IoC yang ditangkap honeypot berupa kode hash dari malware yang menyerang honeypot. Kode hash akan digunakan sebagai pustaka acuan dalam proses deteksi malware pada piranti akhir pengguna secara waktu nyata. Metode deteksi dilakukan dengan cara mencocokkan kode hash dari file yang diunduh pengguna akhir dan kode hash malware yang menyerang honeypot. Honeypot yang digunakan adalah Dionaea yang merupakan *low interaction honeypot* dan membangkitkan kode hash malware menggunakan *MD5 hash function*. Selanjutnya file dari pengguna akhir yang terindikasi malware akan dianalisis dan diidentifikasi labih jauh. Metode yang populer digunakan adalah Yara rules untuk melakukan pemindaian files berdasarkan pola string dari file yang memiliki karakteristik varian malware, seperti urutan symbol string, heksadesimal, dan *regular expressions* [5][6][7][8][9].

Artikel ini melaporkan haril pengembangan sistem identifikasi malware pada piranti akhir pengguna dalam jaringan lokal secara waktu nyata. Sistem yang dibangun juga menampilkan hasil analisis file yang teridentifikasi malware berdasarkan hasil peminaian Yara rules. Selain itu sistem juga secara waktu nyata memberikan notifikasi kepada admin melalui platform aplikasi chat Telegram.

2 Tinjauan Literatur

Teknologi honeypot dapat secara proaktif mendeteksi dan merespon usaha penyusupan oleh penyerang pada suatu jaringan. Teknologi ini lebih sederhana dibandingkan dengan mekanisme keamanan lainnya, menawarkan konfigurasi yang fleksibel, penggunaan sumberdaya yang rendah, dan dapat bekerja secara efektif pada lingkungan yang kompleks [10]. Honeypot juga mampu berkamuflase seolah-olah terlihat oleh penyerang sebagai lingkungan system pabrik cerdas berbasis IoT [11][12]. Informasi yang didapat cukup komprehansip untuk secara efektif membatasi penyebaran yang sifatnya agresif pada jaringan.

Metode pendekatan honeypot dengan memberikan sumberdaya yang penting untuk dapat digunakan oleh penyerang sukses dalam penyerangan tanpa pengetahuan penyerang menjadi pembeda dengan metode yang umum seperti IDS. Penyebaran honeypot secara luas dengan metode gabungan antara low interaction dan high interaction honeypot untuk mendapatkan dan menganalisis kerentangan dan exploits [13]. Pendekatan yang dilakukan adalah dengan menjadikan low interaction

 ISSN:2302-8149
e-ISSN:2540-9719

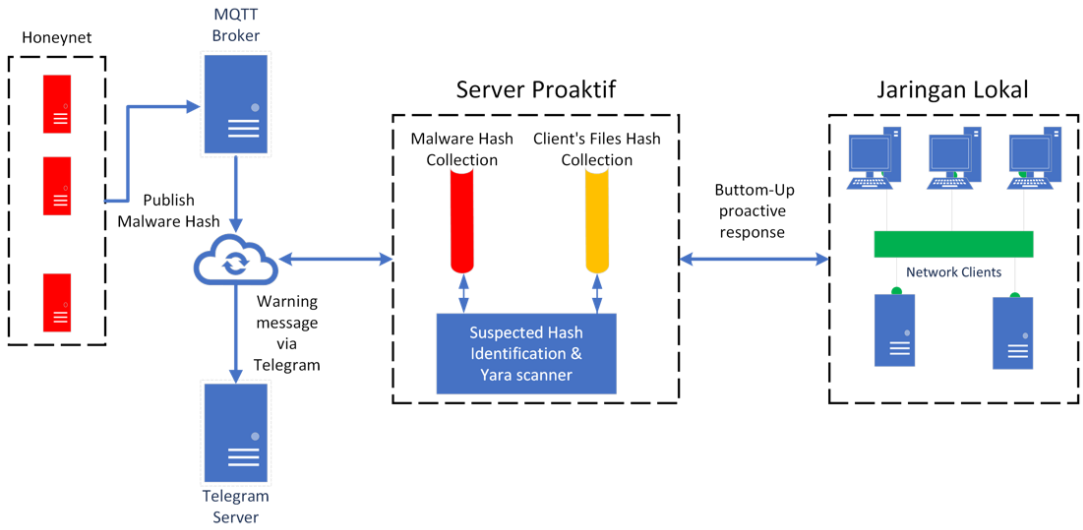
honeypot sebagai penyaring isi di sisi front-end, sedangkan teknik bebasis host akan menangkap informasi lengkap seperti isi paket serangan untuk dilakukan analisis lebih mendalam.

Pemindaian file untuk menemukan kemiripan/kesamaan dengan suatu malware sangat membantu dalam mengklasifikasikan jenis dan varian dari file terindikasi malware, bahkan menemukan varian yang belum teridentifikasi [14]. Pustaka dari Python Bernama Yara mampu untuk memindai file dalam bentuk text maupun biner berdasarkan rules yang disebut dengan Yara rules. Yara rules diciptakan untuk mendeteksi malware pada sistem operasi baik itu Windows maupun Linux/Unix berdasarkan sifat dan ciri khasnya. Yara rules memiliki struktur indikator, yaitu: text strings; hexadecimal strings dan expression strings. Yara rules dapat secara efektif melakukan identifikasi terhadap malware namun efektifitas ini masih tergantung dengan kuantitas dan kualitas Yara rules yang dilibatkan dalam pemindaian file tersebut, sehingga perlu usaha optimasi dalam fase eksekusinya dengan menambahkan Fuzzy rules [15] dan pembangkitan otomatis Yara rules [16].

Keamanan proaktif yang melibatkan kemampuan honeypot mendapatkan kode hash malware dan pemindaian berbasis Yara cukup menjanjikan untuk dapat membantu memitigasi resiko serangan malware yang lebih luas. Artikel ini fokus pada identifikasi file yang diunduh oleh pengguna akhir apakah terindikasi memiliki kesamaan/kemiripan dengan malware atau tidak. Selanjutnya jika terindikasi, maka akan dilakukan pemindaian menggunakan Yara untuk menemukan jenis dan tipe malware.

3 Metode Penelitian

Topologi sistem yang digunakan pada penelitian ini terdiri dari beberapa komponen antara lain jaringan honeypot, MQTT broker, server proaktif, dan jaringan local (pengguna), seperti terlihat pada Gambar 1.



Gambar 1. Rancangan Topologi Sistem.

a. Jaringan Honeypot – sensor node menggunakan Raspberry Pi 4 sebagai *hardware platform* sementara Dionaea *low-interaction* honeypot sebagai emulator beberapa layanan yang didedikasikan untuk diserang oleh penyerang. Data yang dihasilkan oleh honeypot dalam format JSON dan di-*publish* oleh mnodul *harvester* pada honeypot ke modul *collector* pada server proaktif menggunakan protokol MQTT
b. MQTT Broker – bertugas untuk meneruskan data yang dikirim oleh modul *harvester* ke modul *collector*, selanjutnya modul *collector* yang berada pada Server Proaktif akan menyimpan ke dalam basisdata MonggoDB sebagai *malware hash collection*.
c. Server Proaktif – berupa *Virtual Private Server* dengan sistem operasi Linux Ubuntu. Modul-modul yang berada pada server ini antara lain adalah modul basisdata, modul *collector*, modul pemindai Yara, dan modul notifikasi. Modul basisdata menggunakan mongoDB yang berisi dua collections yaitu *malware hash collection* dan *client's file hash collection.* Modul *collector*

 ISSN:2302-8149
e-ISSN:2540-9719

bertugas untuk menerima *Indicator of Compromized* (IoC) dari modul *harvester* yang ada pada honeypot untuk disimpan pada *malware hash collection*, dan menerima kode hash dari klien yang berada pada jaringan lokal untuk disimpan pada *client's file hash collection.* Modul pemindaui Yara bertugas untuk memindai file dari klien yang terindikasi malware untuk dianalisis berdasar Yara rules sehingga diketahui jenis dan tipe malware apa. Modul notifikasi bertugas untuk mem-*publish* informasi adanya penemuan file terindikasi malware ke admin keamanan melalui *channel chat* aplikasi Telegram. Seluruh modul menggunakan bahasa pemrograman Golang.

d. Jaringan Lokal – merupakan sekumpulan piranti pengguna akhir dapat berupa PC dan laptop pengguna dengan sistem operasi Linux, MacOS, dan Windows. Pada setiap klien ini terdapat agen yang berupa modul pendeteksi adanya file baru pada suatu folder dan membangkitkan kode hash file baru tesebut. Selanjutnya kode hash akan dikirim ke server proaktif untuk dicocokkan dengan koleksi kode hash malware.

Metode Identifikasi Malware berbasis Honeypot

Honeypot merupakan mesin yang didedikasikan untuk diserang, masin ini mengemulasi beberapa layanan seperti web server, remote login, file sharing, database server, dll. Beberapa emulasi layanan memungkinkan penyerang untuk mengunggah payload yang berupa malware ke dalam mesin honeypot. Unggahan dari penyerang ini disimpan oleh honeypot dengan nama menggunakan kode hash malware tesrebut. Gambar 2 memperlihatkan penamaan pada folder /*binaries* di dalam honeypot.

```
total 2.3G
drwxr-xr-x  2 0 0  56K Nov 18 01:44 .
drwxr-xr-x 13 0 0 4.0K Nov 10 13:09 ..
-rw-------  1 0 0 5.1M Sep 14 16:08 00502c92416865ca5054bc123d388df5
-rw-------  1 0 0 5.1M Sep 19 12:23 0064e2641d419d2c68f9beb18246a297
-rw-------  1 0 0 5.1M May  8  2022 014ae77d9d8126e805c49d8682b8bb37
-rw-------  1 0 0 5.1M Sep 21  2021 017f63d0be693e53bc5b8edd426cfbd1
-rw-------  1 0 0 5.1M Sep 20  2021 01bdc6fb077098f4a3b60f4b0e479a7f
-rw-------  1 0 0 5.1M Apr  6  2022 01d87121a4a589930d580a88e4df3640
-rw-------  1 0 0  70K Sep  8  2021 02c5f1515bf42798728fac17bfe1e4c1
-rw-------  1 0 0 4.2K Oct  7 23:28 0390e6c918755fc055bb02ccac50215b
-rw-------  1 0 0 100K Oct  8 20:34 0395e945a88761280ab01e8376667aa6
-rw-------  1 0 0 5.1M May 31 15:11 03a37c162b997a9f95c1783dba33b0aa
-rw-------  1 0 0 5.1M Oct  7 23:11 03a6ae2e25815ab9d587a69465dd6d32
-rw-------  1 0 0 2.3M Mar  6  2022 042b933bd1497d5cd47341af3c5ea67b
```

Gambar 2. Penamaan file malware pada folder /binaries di dalam honeypot.

Kode hash dari malware tersebut terekam dalam setiap adanya serangan yang disertai unggahan malware. Setiap terjadi koneksi serangan dan unggahan malware, maka modul harvester akan mempublikasikan ke MQTT Broker yang akan diteruskan ke modul *collector* yang berada pada server proaktif untuk disimpan pada basisdata MongoDB. Gambar 3 memperlihatkan dokumen kejadian yang disimpan pada *malware hash collection.*


```
_id: ObjectId('6331543ffaabbd3ffa7481f9')
download: 10432
connection: 3851933
download_url: ""
download_md5_hash: "beb68e9c7ef18f421df8230c032fe02a"
connection_type: "accept"
connection_transport: "tcp"
connection_protocol: "smbd"
connection_root: 3851933
connection_parent: null
local_host: "202.43.92.50"
local_port: 445
remote_host: "123.22.4.246"
remote_hostname: ""
remote_port: 36024
eventid: "download"
timestamp: "2022-09-02T19:45:52.975564Z"
misp_string: "123.22.4.246
              beb68e9c7ef18f421df8230c032fe02a
              "
sensor: "Public-DSSDI-2"
```


Gambar 3. Dokumen kejadian serangan tersimpan pada malware hash collection MongoDB.

 ISSN:2302-8149
e-ISSN:2540-9719

Klien pada jaringan lokal akan terus memantau adanya file baru pada folder yang sudah ditentukan. Setiap terdeteksi adanya file baru, maka aplikasi agen yang tertanam di klien akan membuat kode hash dari file tersebut. Selanjutnya kode *hash file* akan diunggah ke server proaktif untuk dicocokkan dengan kode-kode *hash* malware yang ada. Apabila terjadi kesamaan, maka hal ini mengindikasikan bahwa file dari klien adalah terindikasi malware.

Metode pemindaian malware menggunakan Yara

File yang terindikasi malware pada klien selanjutnya diunggah ke server proaktif dan dipindai menggunakan Yara untuk menemukan spesifikasi malware. Pemindaian menggunakan Yara ini dimaksudkan untuk mengidentifikasi kesamaan/kemiripan spesifikasi file terindikasi malware dengan pustaka malware yang dimiliki oleh Yara *rules*. Terdapat 409 Yara *rules* yang digunakan untuk pemindaian menggunakan Yara. Gambar 4 menunjukkan conbtoh Yara *rules* yang digunakan dalam peminaian menggunakna Yara.

```
rule with_sqlite : sqlite
{
        meta:
                author = "Julian J. Gonzalez <info@seguridadparatodos.es>"
                reference = "http://www.st2labs.com"
                description = "Rule to detect the presence of SQLite data in raw image"
        strings:
                $hex_string = {53 51 4c 69 74 65 20 66 6f 72 6d 61 74 20 33 00}
        condition:
                all of them
}
```

Gambar 4. Yara rule deteksi kemiripan malware.

Yara *rules* akan memindai seluruh isi file dan mencari string yang sesuai dengna apa yang tertulis pada *field* string dari Yara *rules* tersebut. Pencarian akan mengikuti aturan sesuai *condition* dalam hal ini pada Gambar 4 adalah *all of them* berarti semua himpunan string hexadecimal harus ada pada file yang dipindai. Jika kondisi tersebut ditemukan pada suatu file, maka file tersebut terindikasi malware sqlite.

4 Hasil dan Pembahasan

Pengujian sistem dilakukan menggunakan 2 sensor honeypot yang dipasang pada jaringan publik, sementara server proaktif menggunakan 409 Yara *rules* malware. Aplikasi agen pada klien diinstall pada sistem operasi Windows dan MacOS masing-masing memantau folder /*Downloads.* Data kode hash malware yang dikumpulkan oleh 2 sensor honeypot sebanyak 4.2 juta dokumen trersimpan pada *malware hash collection* basisdata MongoDB

Aplikasi klien atau agen pada *debug mode* menampilkan notifikasi apa yang terjadi dan dikerjakan. Notifikasi dimulai saat terdapat indikasi adanya file baru sampai file tersebut terdeteksi ada atau tidak adanya kesamaan/kemiripan malware. Informasi pada saat tidak ditemukan indikasi kesamaan malware tidak terlalu banyak, seperti yang terlihat pada Gambar 5.

```
2022/11/19 18:59:29 /Users/nrohmanr/Downloads/20221119_094150.mp4.crdownload RENAME|CHMOD
2022/11/19 18:59:29 /Users/nrohmanr/Downloads/20221119_111427.mp4 CREATE
2022/11/19 18:59:29 New file detected. We are running scan the file...
2022/11/19 18:59:34 No Malware Detected
2022/11/19 18:59:34 /Users/nrohmanr/Downloads/20221119_094150.mp4 CHMOD
```

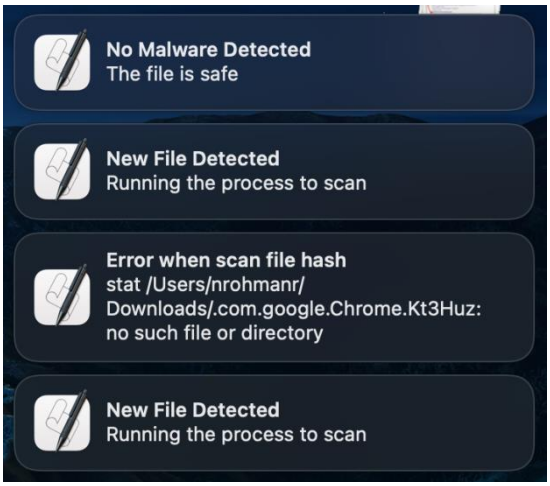
Gambar 5. Tampilan notifikasi pada terminal klien saat tidak ditemukan adanya kesamaan malware.

Pengunduhan file berukuran besar biasanya tidak langsung utuh namun bertahap dengan membentuk *intermediate file* dengan ekstensi *.crdownload* seperti file yang diunduh melalui browser *Chrome.* Pembentukan file yang berlangsung singkat maka agen klien akan mendeteksi adanya file baru dengan menampilkan notifikasi "*New file detected. We are running scan the file…"*Agen klien akan

 ISSN:2302-8149
e-ISSN:2540-9719

menginformasikan berupa pesan "*No Malware Detected,"* jika kode hash dari file tersebut tidak cocok atau tidak ditemukan di database server proaktif pada *malware hash collection*

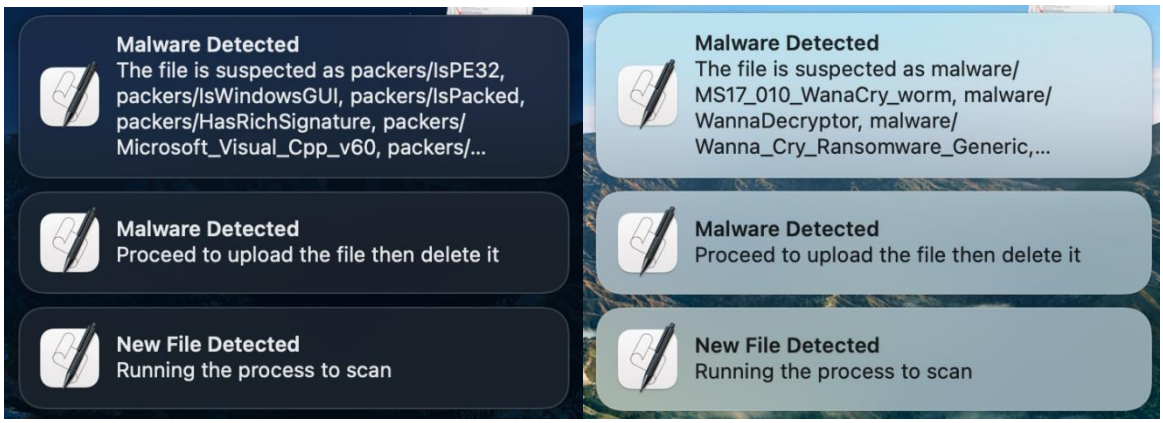
Pengguna akhir yang mengunduh file juga mendapatkan notifikasi berupa pesan yang muncul seketika sehingga dapat dilihat oleh pengguna akhir. Notifikasi akan muncul pada sisi pengguna akhir secara berurutan sesuai kejadian seperti terdeteksinya file baru dengan menampilkan pesan "*New File Detected*," dan kejadian-kejadian lainnya seperti terlihat pada Gambar 6.



Gambar 6. Notifikasi pada piranti klien (MacOS) saat tidak ditemukan kesamaan malware.

Notifikasi pada Gambar 6 secara berurutan adalah dari bawah ke atas sesuai urutan kemunculan notifikasi. Pesan paling bawah menginformasikan adanya file baru dan sistem melakukan proses scan kode hash, namun diikuti pesan berikutnya adalah "*Error when scan file hash*" hal ini terjadi karena aplikasi masih secara seketika merespon adanya file baru dengan proses pembuatan kode hash. Sementara file yang diproses merupakan *intermediate file* yang masih dalam proses pengunduhan dan hilang setelah file terunduh secara penuh. Selanjutnya muncul kembali pesan "*New File Detected*" yang mendeteksi file utuh yang telah lengkap diunduh, dan berhasil dipindai dengan memunculkan pesan "*No Malware Detected*" menyatakan file aman. Aman berarti file baru tersebut tidak terindikasi adanya kesamaan/kemiripan malware dengan daftar pustaka malware yang berada di server proaktif.

Kasus adanya file baru terindikasi kemiripan/kesamaan malware akan membangkitkan notifikasi pada piranti pengguna akhir dan *channel chat Telegram* yang dapat dilihat oleh admin/user yang tergabung dalam *channel* tersebut. Gambar 7 memperlihatkan notifikasi pesan muncul pada piranti pengguna akhir.

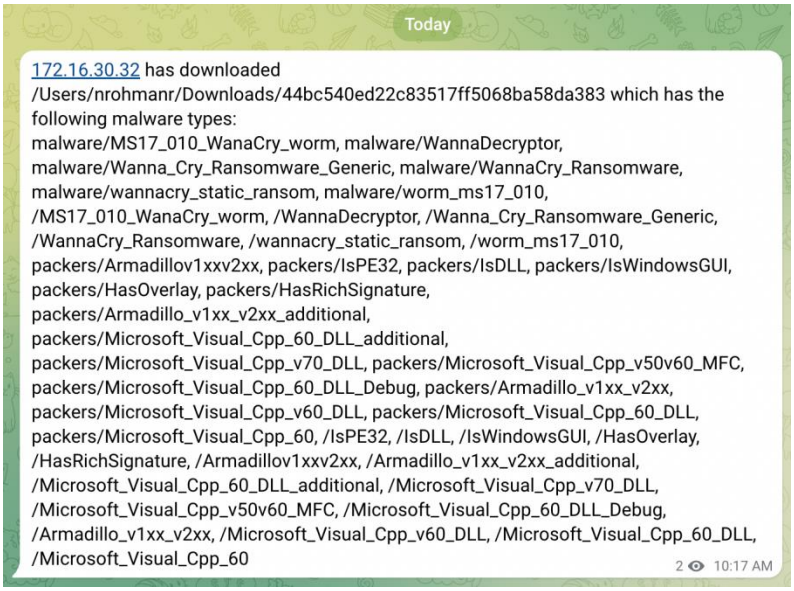


Gambar 7. Notifikasi pada piranti klien (MacOS) saat ditemukan kesamaan malware.

Notifikasi pesan muncul diawali dari urutan paling bawah menginformasikan adanya file baru beserta sub-pesan "*Running the process to scan.*" Server Proaktif menemukan kecocokan kode hash dari file

 ISSN:2302-8149
e-ISSN:2540-9719

yang diunduh klien dengan kode hash malware pada malware hash collection. Hal ini membangkitkan notifikasi "Malware Detected" pada sisi klien dan meminta agen klien untuk mengunggah file terdeteksi kesamaan/kemiripan malware ke Server Proaktif untuk dipindai menggunakan Yara. Pemindaian Yara akan menginformasikan jenis dan varian kemiripan/kesamaan dengan beberapa malware sekaligus. Aksi yang dilakukan pada sisi klien adalah file terindikasi malware ini langsung dihapus. Terlihat pada Gambar 7 bahwa file dari klien terindikasi kemiripan/kesamaan dengan malware WanaCry dengan berbagai variannya. Pada saat yang bersamaan Server Proaktif mengirimkan notifikasi ke pengguna/admin keamanan melalui *channel chat Telegram* yang berisi alamat IP, *Full Path* dari file tersebut, dan daftar jenis serta varian malware. Gambar 8 memperlihatkan tampilan notifikasi yang diterima pengguna akhir/admin keamanan pada *channle chat telegram*.



Gambar 8. Tampilan notifikasi pada *channel chat* Telegram Apps.

Akurasi sistem pemindai malware berbasis honeypot dan Yara ini sangat bergantung pada kemutakhiran dari 2 faktor, pertama yaitu kode hash malware dan kedua adalah pustaka Yara rules. Faktor pertama yaitu kemutakhiran dari kuantitas jumlah kode hash malware yang dibangkitkan oleh honeypot. Hal ini dapat dipenuhi dengan memberpanyak sebaran penempatan sensor honeypot sehingga merepresentasikan area-area blok jaringan. Pada percobaan ini hanya menggunakan 2 honeypots, pengembangan lanjut dengan pemasangan dalam jumlah banyak dan penempatan dengan sebaran luas secara geografis diharapkan akan mampu memutakhirkan baik kuantitas dan kualitas kode hash malware yang ditangkap oleh jaringan honeypot tersebut. Mengingat *Unknown-malware* (atau bisa jadi *zero-day attacks*) lebih banyak ditemukan/diunduh daripada well-known malware oleh honeypot [17], sehingga potensi efektifitas proteksi jaringan lokal dari serangan malware terkini sangat besar. Faktor kedua yaitu kemutakhiran pustaka Yara rules. Pustaka Yara rules masih menjadi kunci dalam menjelaskan jenis dan varian malware. Karena Yara rules dibangun menggunakan *string* dari tubuh kode malware maka keterbatasannya adalah tidak bisa menjelaskan apabila file terindikasi malware yang dipindai adalah *unknown-malware*

5 Kesimpulan

Pengembangan sistem deteksi malware pada jaringan lokal berbasis Honeypot dan Yara telah berhasil dilakukan dan memiliki potensi sebagai mekanisme keamanan proaktif yang membantu meningkatkan kualitas mitigasi resiko keamanan siber di era industri 4.0. Sistem dapat mendeteksi keberadaan file baru pada folder yang ditentukan dan mencocokkan adanya kesamaan/kemiripan malware. Sistem juga mampu mendifinisikan jenis dan tipe kesamaan/kemiripan file terhadap malware. Akurasi dari sistem tergantung dari kemutakhiran jumlah dan sebaran Honeypot pada jaringan publik dan kemutakhiran Yara rules.

ISSN:2302-8149
e-ISSN:2540-9719

Referensi

[1] O. R. M. B. L. L. M. Christos D, "Main incidents in the EU and worldwide ENISA Threat Landscape," Greece, Apr. 2020.

[2] A. Vetterl and R. Clayton, "Honware: A Virtual Honeypot Framework for Capturing CPE and IoT Zero Days," in *eCrime Researchers Summit, eCrime*, 2019, vol. 2019-November. doi: 10.1109/eCrime47957.2019.9037501.

[3] A. Tambe *et al.*, "Detection of threats to IoT devices using scalable VPN-forwarded honeypots," in *CODASPY 2019 - Proceedings of the 9th ACM Conference on Data and Application Security and Privacy*, 2019. doi: 10.1145/3292006.3300024.

[4] G. H. P. Wibawa, I. G. M. A. Sasmita, and I. M. S. Raharja, "Analisis Data Log Honeypot Menggunakan Metode K-Means Clustering," *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, 2020, doi: 10.24843/jim.2020.v08.i01.p02.

[5] N. Naik, P. Jenkins, N. Savage, L. Yang, K. Naik, and J. Song, "Augmented YARA Rules Fused with Fuzzy Hashing in Ransomware Triaging," in *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, 2019. doi: 10.1109/SSCI44817.2019.9002773.

[6] P. Arntz, "Explained: YARA rules," *malwarebytes.com*, 2020.

[7] M. Brengel and C. Rossow, "YARIX: Scalable YARA-based malware intelligence," in *Proceedings of the 30th USENIX Security Symposium*, 2021.

[8] D. Regeciova, D. Kolar, and M. Milkovic, "Pattern Matching in YARA: Improved Aho-Corasick Algorithm," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3074801.

[9] C. Culling, "Which YARA Rules Rule: Basic or Advanced?," *GIAC (GCIA) Gold Certification and RES 5500*, 2018.

[10] J. Bao, C. P. Ji, and G. Mo, "Research on network security of defense based on honeypot," in *ICCASM 2010 - 2010 International Conference on Computer Application and System Modeling, Proceedings*, 2010, vol. 10. doi: 10.1109/ICCASM.2010.5622780.

[11] S. Lee, A. Abdullah, N. Jhanjhi, and S. Kok, "Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning," *PeerJ Comput Sci*, vol. 7, 2021, doi: 10.7717/PEERJ-CS.350.

[12] L. Seungjin, A. Abdullah, and N. Z. Jhanjhi, "A review on honeypot-based botnet detection models for smart factory," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, 2020, doi: 10.14569/IJACSA.2020.0110654.

[13] K. Chawda and A. D. Patel, "Dynamic & hybrid honeypot model for scalable network monitoring," in *2014 International Conference on Information Communication and Embedded Systems, ICICES 2014*, 2015. doi: 10.1109/ICICES.2014.7033844.

[14] P. Black, I. Gondal, A. Bagirov, and M. Moniruzzaman, "Malware Variant Identification Using Incremental Clustering," *Electronics (Basel)*, vol. 10, no. 14, 2021, doi: 10.3390/electronics10141628.

[15] N. Naik, P. Jenkins, N. Savage, L. Yang, K. Naik, and J. Song, "Embedding Fuzzy Rules with YARA Rules for Performance Optimisation of Malware Analysis," in *IEEE International Conference on Fuzzy Systems*, 2020, vol. 2020-July. doi: 10.1109/FUZZ48607.2020.9177856.

[16] N. Naik, P. Jenkins, R. Cooke, J. Gillett, and Y. Jin, "Evaluating Automatically Generated YARA Rules and Enhancing Their Effectiveness," in *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, 2020. doi: 10.1109/SSCI47803.2020.9308179.

[17] N. R. Rosyid, M. Ohrui, H. Kikuchi, P. Sooraksa, and M. Terada, "A discovery of sequential attack patterns of malware in botnets," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2010. doi: 10.1109/ICSMC.2010.5641914.