Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

Identifikasi Serangan Port Scanning dengan Metode String Matching

Sasut Analar Valianta
Magister Informatika
Fakultas Ilmu Komputer
Universitas Sriwijaya
valey@mtiseclab.com
a.valeyys@gmail.com

Tasmi
Magister Informatika
Fakultas Ilmu Komputer
Universitas Sriwijaya
tasmi@ilkom.unsri.ac.id
tasmi@mtiseclab.com

Deris Stiawan
Sistem Komputer
Fakultas Ilmu Komputer
Universitas Sriwijaya
deris@ilkom.unsri.ac.id

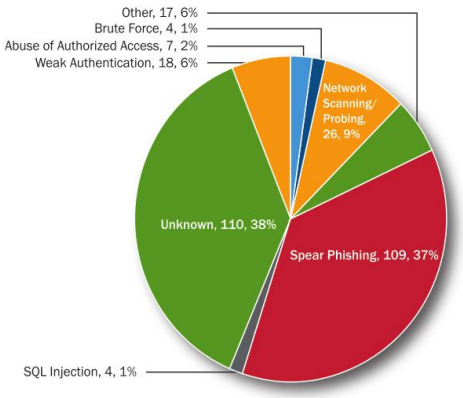
Abstrak*— Port Scanning adalah salah satu serangan yang cukup berbahaya, teknik ini dapat memetakan karakteristik, mendeteksi port yang terbuka bahkan mendapatkan informasi penting pada suatu jaringan atau host untuk kemudian diteruskan ke serangan lebih lanjut. TCP Connect Scan merupakan salah satu teknik port scanning yang sukar untuk dideteksi pada kondisi real-time. Permasalahan yang sering diungkapkan pada proses deteksi gangguan secara real-time yaitu kapasitas sensor terhadap trafik data, efisiensi volume data serta pemilihan metode yang dapat mengatasi besarnya lalu-lintas data. Dalam paper ini kami mengusulkan pendeteksian serangan port scanning secara real-time dengan menggunakan pendekatan metode string matching automaton based. Untuk mengatasi masalah efisiensi pada volume data, proses inspeksi data akan mengikuti stream TCP yang berlangsung. TAP sebagai sensor device digunakan untuk memantau aliran data dan high transfer NIC digunakan pada monitoring device.

Kata kunci*—Port Scanning; Deep Packet Inspection; String Matching; Finite Automata.

I. PENDAHULUAN

Laporan dari departemen *Homeland Security Amerika Serikat [1]* pada akhir tahun 2015 menyatakan bahwa 26,95% dari percobaan serangan berasal dari *Network Scan* atau *Probe*, meningkat dibanding pertengahan tahun 2015 yang hanya sebesar 21,19 %. Secara spesifik dilaporkan bahwa 97.33% target serangan adalah pada sektor industri penting.

Dapat dijelaskan pada gambar 1, bahwa 109 percobaan *(37%)* berasal dari spear phising, 110 percobaan serangan *(38%)* tidak diketahui sumbernya, besar kemungkinan serangan yang tidak terdeksi ini berasal aktifitas terhadap port scanning. Network scanning sendiri terindikasi sebesar *26.9%* sedangkan sisanya *47.5%* adalah aktifitas serangan yang terjadi akibat kelemahan sistem.



Gambar 1. *Insiden dari percobaan serangan [1]*

Mengacu pada terdahulu *[2][3][4]*, dapat disimpulkan bahwa teknik *Port Scanning* adalah tahapan awal untuk mendeteksi port-port yang terbuka dan mendapatkan informasi dari port yang terbuka pada host, versi server dan sebagainya. Sedangkan dalam paper *[5][6]* secara tegas menyatakan bahwa serangan terhadap jaringan yang berawal dari port scanning tidak dapat dihindari, karena port mapping atau port scanning sering teridentifikasi sebagai lalu-lintas rutin pada sebuah jaringan.

Dalam sebuah survei *[7]* dijelaskan, DPI berfungsi sebagai inspector yang dapat digunakan pada setiap lapisan (layer) pada komponen jaringan, sehingga dimungkinkan untuk melakukan analisa dan inspeksi lebih mendalam terhadap trafik data. lebih detail dalam sebuah survei *[8]* menyampaikan bahwa algoritma *String Matching* terbukti berguna untuk medeteksi serangan, pendekatan-pendekatan yang digunakan antara lain automaton-based, heuristic-based, atau filtering-based.

Pada penelitian terdahulu *[9][10][11]* dan terkini *[12][13]* secara konsisten menyimpulkan bahwa proses inspeksi data dalam keadaan *Real-time* lebih akurat daripada dalam keadaan offline. Dalam keadaan real-time kerap kali

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

ditemukan masalah dalam menganalisa, mengidentifikasi dan mengenali paket data seperti yang disinggung pada laporan penelitian sebelumnya *[14]*. Hal senada juga disampaikan pada penelitian *[15]*, dalam laporannya menyatakan proses identifikasi fitur sangat berpengaruh pada besarnya lalu-lintas paket data. demikan juga dipaparkan pada penelitian *[16]*, dimana proses capturing atau sniffing data dalam keadaan real-time sering terjadi masalah *PacketDrop* (kehilangan bit data) dan masalah buffering pada *Network Interface Card*.

Dalam paper ini diusulkan penelitian terhadap serangan port scanning. Penelitian akan difokuskan pada serangan *TCP Connect Scan*. Secara terurut akan dijelaskan tahapan dalam melakukan penelitian, dimana akan dibagi menjadi dua bagian utama yaitu; Identifikasi terhadap masalah yang akan dipecahkan, kemudian pendekatan terhadap metodologi yang akan digunakan.

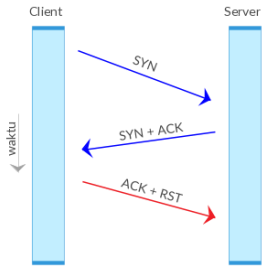
II. IDENTIFIKASI MASALAH

Dari hasil analisa dan survei yang dilakukan pada beberapa penelitian sebelumnya, kemudian dengan membandingkan terhadap literatur keilmuan, secara garis besar dapat ditarik kesimpulan bahwa permasalahan utama yang dihadapi dalam mendeteksi aktifitas port scanning antara lain; aktifitas port scanning, trafik data, metode identifikasi, dan desain topology untuk proses sniffing atau capturing data.

2.1. Aktifitas Port Scanning

Komunikasi dari Port scanning umumnya berada pada layer *Transport Protocol*, pada kasus tertentu port scanning akan terjadi pada *Session Layer*. Pada laporan survei *[2]* memaparkan, port scanning dapat dipisahkan menjadi beberapa bagian yaitu; *Stealth Scan, SOCKS Port Probe, Bounce Scan, TCP Scan,* dan *UDP Scan*. Salah satu serangan port scanning yang paling berbahaya adalah TCP Scan, karena pada TCP scan komunikasi tidak sepenuhnya terhubung. Tanpa melakukan inspeksi lebih mendalam pada komunikasi data, maka TCP Scan akan sukar untuk dideteksi. TCP scan sendiri terbagi menjadi beberapa teknik diantaranya; *TCP connect scanning, TCP half-connect scanning, TCP NULL, TCP FIN, TCP XMAS* dan dimungkinkan kombinasi dari beberapa teknik tersebut.

Pada gambar 2, dapat dijelaskan terdapat tiga tahap dalam komunikasi yang mencirikan signature dari *Tcp Connect Scan*. Pada tahap awal attacker akan melakukan request ke target host dengan flag SYN *(Syndicate)*, kemudian jika port dinyatakan terbuka target akan melakukan reply dengan flag SYN dan ACK *(Acknowledgment*), selanjutnya attacker akan mereply kembali dengan flag ACK dan diakhiri dengan RST *(Reset)*. Dengan kata lain terjadi 3way handshake pada TCP Connect Scan.



Gambar 2. *Signature dari TCP Connect Scan*

2.2. Trafik Data

Masalah mengenai trafik data seringkali diungkapkan pada beberapa penelitian. Pada penelitian *[17]* menyatakan terdapat permasalahan dalam identifikasi packet data, dimana kurangnya efisiensi terhadap besarnya volume data yang akan diolah. Hal senada juga disampaikan oleh *[18]* dimana kapasitas yang dimiliki sensor merupakan suatu masalah yang harus dipecahkan dalam jaringan real-time. Dalam penelitian *[15]* juga menyinggung mengenai trafik data dalam proses identifikasi fitur, serta sulitnya melakukan deteksi secara real-time karena lalu-lintas dari paket data yang instesif. Dari pernyataan-pernyataan yang disebutkan dari penelitian terdahulu permasalahan pada trafik data dapat di simpulkan sebagai berikut :

- Efisiensi volume data.
- Kapasistas sensor terhadap lalu-lintas realtime.
- Kelemahan proses identifikasi fitur terhadap paket data dalam keadaan realtime.

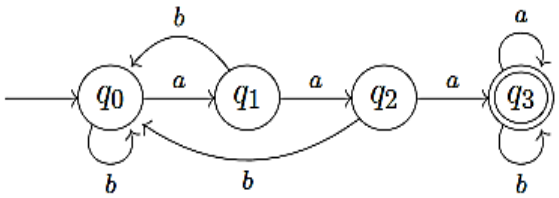
2.3. Metode Identifikasi Port Scanning

Permasalahan lainnya dalam melakukan identifikasi port scanning adalah metode yang akan digunakan. Terdapat banyak cara dalam melakukan identifikasi, untuk mendapatkan signature dan pola yang, penggunaan metoda Deep Packet Inspection (DPI) adalah salah satu cara yang tepat, karena DPI dapat melakukan inspeksi lebih mendalam dengan memeriksa setiap lapisan (layer) pada packet data. DPI sendiri terbagi menjadi beberapa metoda secara umum yaitu; *Port-Based, Protocol Decoding, Patern Matching, Statistical analisys.* Untuk metoda petern matching sendiri dibagi lagi menjadi dua kelas yaitu; *String Matching* dan *Regular Expression.*

Komunikasi pada TCP Scan khususnya TCP Connect scan berisi fitur-fitur yang dapat di ektraksi untuk mendapatkan signature data, signature yang akan dianalisa dan dilakukan klasifikasi dapat diolah dengan menggunakan pendekatan metoda string matching. Sebuah artikel *[8]* menjelaskan, dalam menggunakan string matching terdapat beberapa pendekatan algoritma yaitu; *Automaton-Based, Heuristic-Based,* dan *Filtering-Based.*

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

Dengan membandingkan kelebihan dan kekurangan pada algortima yang digunakan dalam pendekatan string matcing maka akan digunakan metoda string matching dengan pendekatan algoritma *Automaton-Based*. Gambar 3 merupakan sebuah contoh yang diberikan dalam sebuah lecture notes *[19]* untuk state-transisition diagram pada algoritma string matching automaton based.



dimana : *States :* q_0, q_1, q_2, q_3
Start state : q_0
Transitions: as indicated above.
Input symbols: $a, b = \sum$
Accepting state(s) : q_3
Read character : babaabaaaba
Search character : aaa = i

Gambar 3. *Contoh State-transition Finite Automata [19]*

Dari state transition diagram diatas dapat dijelaskan sebagai berikut; karakter yang akan dicari adalah *aaa* , sedangkan karakter yang akan dibaca adalah *babaabaaaba*

- pencarian dimulai dari state q_0
- selanjutnya pada karakter pertama tidak ditemukan awalan karakter pencarian, dilanjutkan ke karakter kedua, state tetap pada q_0
- pada karakter kedua ditemukan awalan pencarian karakter *a* maka pindah ke state q_1 dan dilanjutkan ke pencarian pada karakter ketiga
- karakter ketiga tidak ditemukan karakter pencarian pada state $q_1$maka kembali ke state q_0
- karakter keempat ditemukan ditemukan pencarian karakter q_0 kemudian dilanjutkan kembali ke state q_1
- karakter kelima ditemukan pencarian karakter pada state q_2
- pada karakter keenam tidak ditemukan pencarian karakter pada state q_3, maka state kembali ke awal pencarian untuk karakter berikutnya
- pada karakter ketujuh ditemukan pencarian karakter pada state q_1 , dilanjutkan ke state q_2
- pada karakter kedelapan ditemukan karakter pada state q_3, dengan ditemukannya pencarian karakter pada state q_1, q_2, q_3, maka pencarian dianggap selesai.

Secara sederhana pencarian string dari $\sum = a,b$, dapat diterjemahkan sebagai berikut :

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \text{ ..... } (\textit{Search char found}) \quad (1)$$

$$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \text{ .....} (\textit{Search char not found}) \quad (2)$$

Dari state transition tersebut dapat dituangkan dalam bentuk algoritma sederhana seperti terlihat pada algoritma berikut:

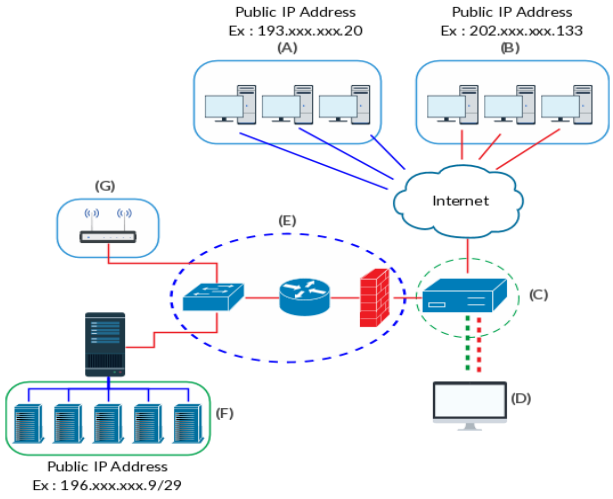
```
State = 0
for (i = 1,2, … , n)
    state = δ(state, T_i)
    if (state == m)
    Match success in pos i − m + l; break;
    Else Match failed;
```

Gambar 4. *Algoritma Finite Automaton*

2.4. Desain Topology

Pada penelitian ini, topologi jaringan didesign seefisien mungkin dengan kondisi real-time menggunakan akses secara public. Teknik sniffing dilakukan untuk memperoleh data trafik dengan menggunakan TAP *(Test Accses Point).*

Pada gambar 4, dapat dilihat bahwa target host akan diberikan beberapa serangan dan akses normal secara serentak. Trafik data akan dicapture sebagai tahap preprocessing maupun analisa dan klasifikasi secara real-time.



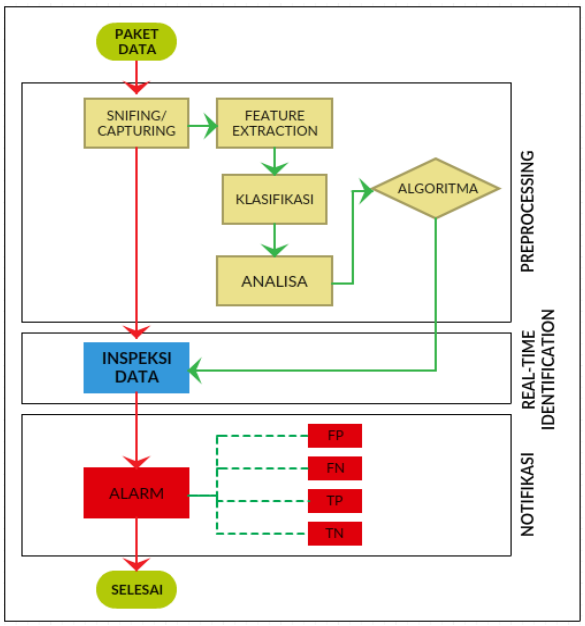
Gambar 5. *Sekenario desain Topology Real-time Network, (A) Normal user, (B) Attacker, (C) Sensor TAP, (D) Monitoring, (E) Firewall Gateway dan Switch, (F) Target host-Virtualization Server, (G) Perangkat tambahan lain.*

Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

III. PENDEKATAN METODOLOGI

Pada bagian ini diusulkan metode untuk mengidentifikasi serangan port scanning menggunakan algoritma string matching dengan pendekatan automaton-based secara real-time. Secara garis besar proses identifikasi dibagi menjadi tiga kategori utama yaitu; Preprocessing, Inspeksi data dan klasifikasi, dan notifikasi, seperti terlihat pada diagram flowchart pada gambar 6.

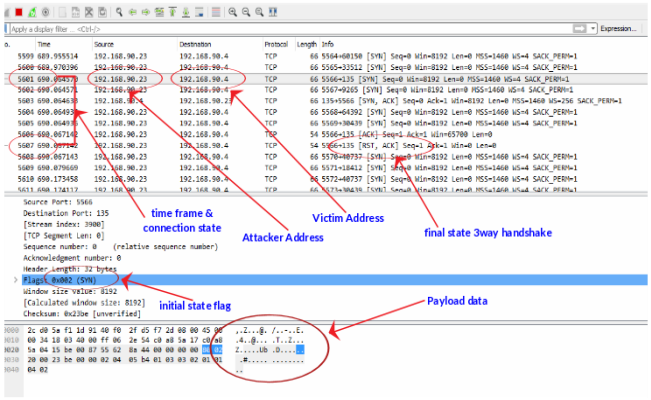


Gambar 6. *Tahapan proses Identifikasi Port Scanning*

3.1. Preprocessing

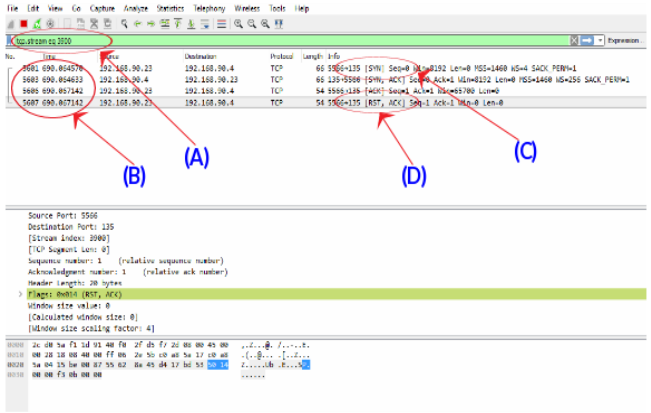
Pada tahap preprocessing, dilakukan pengambilan data awal dengan mengcapture trafik kemudian dilakukan ekstraksi data berdasarkan fitur-fitur yang terdapat didalam protocol TCP.

Proses analisa terlihat pada gambar 7, dapat dijelaskan bahwa jarak dan waktu antara komunikasi three-way handshake terpaut cukup jauh dalam kondisi pengujian off-line port scanning, akan lebih jauh lagi jika kondisi terjadi pada keadaan real-time. Untuk menghindari penumpukan data pada saat inspeksi, dapat digunakan teknik TCP follow stream, sehingga data yang akan diinspeksi lebih lebih pendek jika dibandingkan tanpa mengikuti stream data, seperti terlihat pada gambar 8.



Gambar 7. *Analisa terhadap TCP Connect Scan*

Setelah hasil analisa diperoleh dengan membandingkan antara aktifitas normal dan aktifitas port scanning maka dibuat pendekatan algoritma untuk tahap inspeksi data dan klasifikasi data secara real-time.



Gambar 8. *Stream Flow dari Port Scanning, (A) TCP Stream Follow, (B)Connection Sequences, (C) Initial State, (D) Final State.*

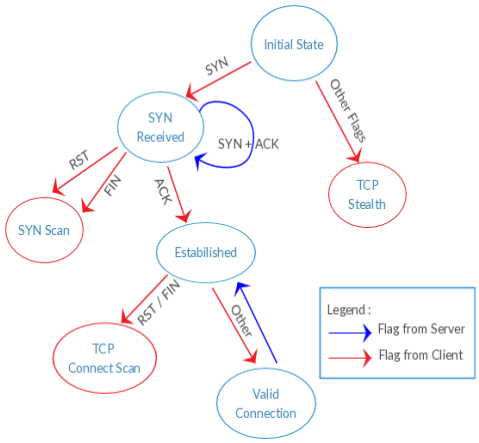
3.2. Inspeksi Data

Dari hasil analisa sementara, didapatkan pola serangan TCP Connect Scanning, dimana attacker akan melakukan request terhadap port target dengan mengirimkan flag SYN, kemudian jika port yang diminta tersedia maka target akan mengirimkan umpan balik kepada attacker dengan mengirimkan flag SYN + ACK. Kemudian attacker akan kembali mengirimkan flag ACK yang berfungsi untuk mengelabui sistem target sehingga terlihat adanya established connection antara target dan attacker. Selanjutnya attacker akan kembali mengirimkan flag ACK+RST yang menandai akhirnya komunikasi.

Dengan metoda string matching menggunakan pendekatan automaton based maka proses pencarian string pada

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

komunikasi data dapat digambarkan dengan state transition diagram seperti terlihat pada gambar 9.



Gambar 9. *Flowchart state-Transition TCP Scan*

Untuk menghidari besarnya data, maka pada saat inspeksi data hanya akan mengikuti TCP Stream. Sehingga data yang akan diinspeksi hanya berupa fitur data yang terindikasi sebagai state awal dari TCP Connect Scan. Proses metode identifikasi dapat dilihat pada pseudo code berikut :

```
INPUT :
        State 0 ← SYN;
        flag;

METHOD : begin

IF state  0 ← 1
    Follow tcp stream
    IF flag ←  SYN
        GET SourceAdd
                SourceAdd ← S1
        GET DestAdd
                DestAdd ← D1
        GO TO State 1
    ELSE mark AS Suspect_StealhScan
ELSE RETURN 0;

State 1;
GET SourceAdd
GET DestAdd
        IF SourceAdd ← D1 AND DestAdd ← S1
                GET flag
                IF flag ← SYN AND ACK
                GO TO State 2
        ELSE RETURN 0;
```

```
Cont..

State 2;
GET SourceAdd
GET DestAdd
        IF SourceAdd ← S1 AND DestAdd ← D1
                GET flag
                IF flag ← ACK
                GO TO State 3
        ELSE mark AS Suspect_SYNScan

State 3;
GET SourceAdd
GET DestAdd
        IF SourceAdd ← S1 AND DestAdd ← D1
                GET flag
                IF flag ← ACK AND RST
                PRINT ALERT TCPConnet_Found
        ELSE RETURN 0;
```

Gambar 10. *Pseudo-code Identifikasi Port Scanning*

3.3. Notifikasi

Proses notifkasi dilakukan untuk memberikan peringatan terhadap indikasi terjadinya serangan port scanning. Notifikasi dibagi menjadi empat kriteria antara lain; *False Positif, False Negatif, True Positif* dan *True Negatif.* detail kriteria tersebut adalah sebagai berikut ;

- *False Positive* : komunikasi pada state awal (initial state), jika ditemukan request tanpa ataupun flag selain SYN maka sistem akan memberikannotifikasi melakukan marking pada data sebagai suspect scan
- *False Negative* : komunikasi pada state kedua *(reply host/ target),* jika ditemukan reply host selain SYN+ACK maka sistem memberikan notifikasi.
- *True Positif* : komunikasi pada state ketiga (TCP Connect scan terdeteksi), sistem akan memberikan notifikasi / warning alarm bahwa ditemukannya aktifitas TCP Connect scan
- *True Negatif* : komunikasi pada sate kedua, reply client, sitem akan memberikan notifikasi dan melakukan marking terhadap paket mencurigakan yang terindikasi TCP port scanning.

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

IV. KESIMPULAN DAN PERENCANAAN MENDATANG

Dalam paper ini kami mengajukan penelitian untuk mengidentifikasi port scanning pada paket data secara real-time. Solusi yang ditawarkan adalah mengikuti stream pada transport protocol dengan menggunakan pendekatan metode string matching automaton-based untuk mendapatkan pola serangan pada paket data. Solusi ini diharapkan mampu untuk mendeteksi aktifitas port scanning yang sukar terdeteksi pada penelitian-penelitian sebelumnya. Perencanaan mendatang, penelitian akan lebih fokus pada algoritma string matching dalam mengklasifikasikan dan mengidentifikasi aktifitas port scanning pada paket data secara real-time.

REFERENSI

[1] N. C. and C. I. C. ICS-CERT, "Ics-Cert," no. December, pp. 1–10, 2015.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," *Comput. J.*, vol. 54, no. 10, pp. 1565–1581, 2011.

[3] X. Zhang, J. Knockel, and J. R. Crandall, "Original SYN: Finding machines hidden behind firewalls," *Comput. Commun. (INFOCOM), 2015 IEEE Conf.*, pp. 720–728, 2015.

[4] B. Soniya and M. Wiscy, "Detection of TCP SYN scanning using packet counts and neural network," *SITIS 2008 - Proc. 4th Int. Conf. Signal Image Technol. Internet Based Syst.*, pp. 646–649, 2008.

[5] K. Shah, S. Bohacek, and a. Broido, "Feasibility of detecting TCP-SYN scanning at a backbone router," *Proc. 2004 Am. Control Conf.*, vol. 2, pp. 988–995, 2004.

[6] J. Wiebelitz, C. Kunz, S. Piger, and C. Grimm, "TCP-AuthN: TCP inline authentication to enhance network security in Grid environments," *8th Int. Symp. Parallel Distrib. Comput. ISPDC 2009*, pp. 237–240, 2009.

[7] T. AbuHmed, A. Mohaisen, and D. Nyang, "A Survey on Deep Packet Inspection for Intrusion Detection Systems," *Inf. Secur.*, vol. 24, p. 10, 2008.

[8] P.-C. Lin, Y.-D. Lin, Y.-C. Lai, and T.-H. Lee, "Using String Matching for Deep Packet Inspection," *IEEE Comput. Soc.*, vol. 41, no. 4, pp. 23–28, 2008.

[9] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline / Realtime Traffic Classification Using Semi-Supervised Learning," no. October, pp. 2–5, 2007.

[10] F. Zhang, W. He, X. Liu, and P. G. Bridges, "Inferring users' online activities through traffic analysis," *Proc. fourth ACM Conf. Wirel. Netw. Secur. - WiSec '11*, p. 11, 2011.

[11] G. SunilKumar, "Real Time and Offline Network Intrusion Detection using Improved Decision Tree Algorithm," *Int. J. Comput. Appl.*, vol. 48, no. 25, pp. 1–6, 2012.

[12] A. Malaguti *et al.*, "Comparison of online and offline methods for measuring fine secondary inorganic ions and carbonaceous aerosols in the central mediterranean area," *Aerosol Air Qual. Res.*, vol. 15, no. 7, pp. 2641–2653, 2015.

[13] T. S. Pawar, R. G. Sawant, P. S. Bothe, and Sh. A. Chopade, "A Study on Real Time Big Data Analytics," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 3, no. 10, pp. 10131–10138, 2015.

[14] D. Stiawan, A. H. Abdullah, and M. Y. Idris, "The trends of Intrusion Prevention System network," *ICETC 2010 - 2010 2nd Int. Conf. Educ. Technol. Comput.*, vol. 4, pp. 217–221, 2010.

[15] C. A. Catania and C. García, "Automatic network intrusion detection : Current techniques and open issues q," *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1062–1072, 2012.

[16] J. Svoboda, "Network Traffic Analysis with Deep Packet Inspection Method," *Fac. Informatics Masaryk Univ.*, no. Master's Thesis, 2014.

[17] D. Smallwood and A. Vance, "Intrusion analysis with deep packet inspection: Increasing efficiency of packet based investigations," *Proc. - 2011 Int. Conf. Cloud Serv. Comput. CSC 2011*, pp. 342–347, 2011.

[18] D. Stiawan, A. Y. I. Shakhatreh, M. Y. Idris, A. B. Kamarulnizam, and H. A. Abdul, "Intrusion prevention system: A survey," *J. Theor. Appl. Inf. Technol.*, vol. 40, no. 1, pp. 44–54, 2012.

[19] A. M. Pitts, "Regular Languages and Finite Automata," *Lect. Notes Part IA Comput. Sci. Tripos*, no. Cambridge University Computer Laboratory, 2008.