View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repository Proceeding Seminar (Fakultas Ilmu Komputer, Universitas Sriwijaya)

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

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

Klasifikasi Trafik Terenkripsi Menggunakan Metode *Deep Packet Inspection* (Dpi)

Tasmi
Megister Teknik Informatika
Fakultas Ilmu Komputer
Univeristas Sriwijaya
tasmi@ilkom.unsri.ac.id

Sasut Analar Valianta
Megister Teknik Informatika
Fakultas Ilmu Komputer
Univeristas Sriwijaya
valey@mtiseclab.com

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

Abstrak— Tujuan dari penelitian ini adalah untuk mendesain simulasi online network untuk capture paket data terenkripsi dan melakukan proses ekstraksi fitur-fitur paket data yang terenkripsi. Metode yang digunakan adalah Deep Packet Inspection (DPI) dengan menggunakan Regular Expressions dalam proses mengenali pola trafik terenskripsi. Data yang digunakan dalam penelitian ini adalah paket data normal dan paket data terenkripsi dalam jaringan, dimana dari data tersebut data yang akan diolah adalah paket data yang terenkripsi (SSL/TLS). Hasil yang didapatkan menunjukkan bahwa pola dari paket terenskrip (SSL) memiliki Field yang terdiri dari Record Content Type, Protocol Version, Handshake Type dan Service. TLS 1.0 memiliki pola 0x16 0x0100 0x01 0x0b. TLS 1.2 memiliki pola 0x16 0x0300 0x03 0x0b.

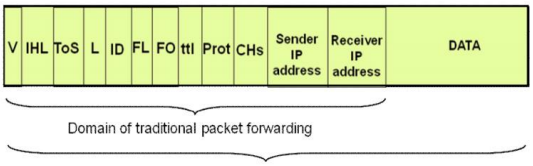
Kata kunci— Deep Packet Inspection (DPI), Regeluar Expression, SSL/TLS, Feature Extraction

I. PENDAHULUAN

Penelitian yang dilakukan oleh [1] menyatakan bahwa tren penggunaan aplikasi-aplikasi HTTPS sudah banyak digunakan untuk menggantikan aplikasi yang menggunakan HTTP seperti *sending tweets*, *messages to mobile*, *posting*, dan *search*, dikarenakan memiliki kelebihan di bidang keamanan (*secure*).

Penelitian [2] menyatakan ada permasalahan dalam proses pengembangan sistem keamanan *network*, karena tidak semua protokol bisa diblok terutama data yang terenkripsi seperti *Facebook* dan *Twiter*, artinya pada proses mengindentifikasi paket data tidak hanya pada jenis protokol dan nomor *port* saja tetapi lebih pada indentifikasi dan regulasi pada *signature* aplikasi.

Mengaju pada penelitian [3] menyatakan bahwa *Deep Packet Inspection* (DPI) berkerja dengan memeriksa paket hingga *application layer* pada OSI Layer untuk mendapatkan informasi jenis trafik, hal ini dapat mengatasi masalah identifikas paket data tidak hanya berdasarkan *port number*. DPI dapat dijadikan salah satu solusi untuk klasifikasi data pada *statefull firewall* yang hanya mengklasifikasikan trafik berdasarkan nomor port dan protokol yang digunakan. Banyak para penelitian menggunakan metode DPI dalam menyelesaikan kasus mengenali pola trafik dan menyelesaikan kasus sistem keamanan jaringan, seperti penelitian yang dilakuka oleh [4] dan [5], mereka menyatakan DPI adalah metode yang digunakan memantau aliran data dan membuat pengelompoan terhadap aliran data tersebut, kemudian penelitian yang dilakukan oleh [2] dan [6], mereka menggunakan metode DPI untuk digunakan dalam sistem keamanan jaringan, sedangkan penelitian yang dilakukan oleh [7] menggunakan DPI dengan *signature* di *payload* untuk mengidentifikasi protokol. Penelitian yang dilakukan oleh [3], [8] dan [6] menggunakan DPI sebagai metode untuk proses klasifikasi paket data yang masuk dan keluar.



Gambar 1. Domain dari DPI [9]

Permasalahan dalam penelitian ini adalah bagaimana melakukan proses *Capture Packet* dan *Feature Extraction* serta bagaimana melakukan proses validasi data. Sedangkan tujuan dari penelitian ini adalah mendesain simulasi *online network* untuk *capture* paket data terenkripsi dan melakukan proses ekstraksi fitur-fitur paket data yang terenkripsi.

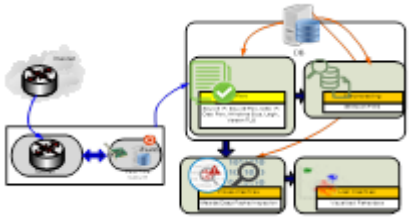
Dalam penelitian ini akan dibahas mengenai proses klasifikasi trafik terenskripsi, dimana tahapan metodologi yang digunakan dalam menyelesaikan permasalahan dijelaskan pada bagian II dan bagian III yang menampilkan data sementara. Kesimpulan akan ditampilkan pada bagian IV

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

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

II. METODOLOGI PENELITIAN

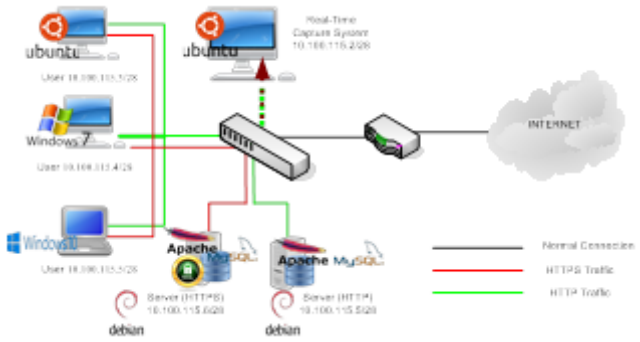
Pada penelitian ini ada beberapa tahapan yang akan dilakukan seperti ditampilkan pada Gambar 2 . Secara umum ada empat tahap, dimana setiap tahapan mempunyai permasalahan-permasalahan yang akan diselesaikan selama proses penelitian ini. Empat tahapan tersebut akan dibagi menjadi dua tahap pengamatan dan percobaan.



Gambar 2. *Roadmap* Penelitian

2.1. Perancangan Sistem

Dalam penelitian ini monitoring trafik akan berpusat dari *network Indonesia Internet Exchange* (IIX) sedangkan mesin target akan ditempatkan dibeberapa tempat seperti USA dan Eropa dengan menggunakan layanan sewa *Virtual Private Server* (VPS) dibeberapa penyedia layanan di Tier-1 dunia. Topologi yang digunakan adalah *start* dengan menggunakan *manage switch*. Penelitian ini menggunakan 6 buah PC yang berfungsi sebagai *Server, Packet Analyzer + IDS, Common User* dan *Malicius User* dan 1 buah *Router* sebagai *forwarder* paket data. *Endpoint* yang digunakan menggunakan jenis distribusi dan sistem operasi seperti pada Gambar 3.

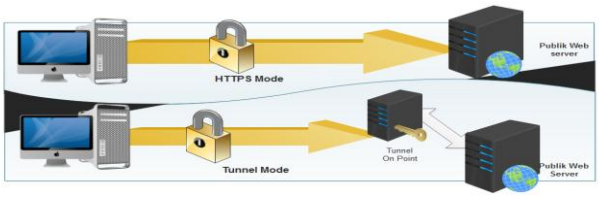


Gambar 3. Topologi Penelitian

.

2.2. Koleksi Data

Tahap pertama, data yang di-*capture* berasal dari trafik di *network*. Data yang digunakan dalam penelitian ini adalah paket data normal dan paket data terenkripsi dalam jaringan, dimana dari data tersebut data yang akan diolah adalah paket data yang terenkripsi (SSL/TLS dan VPN) seperti pada Gambar 4. Selanjutnya data yang didapat akan digunakan untuk proses *Feature Extraction.*



Gambar 4. Model Data SSL/TLS

2.3. Ekstraksi dan Analisis Data

Header TCP/IP bersifat unik dan memiliki *header-header* tersembunyi yang tergantung dari protokol dan proses *encapsulated* yang menyebabkan *raw data* sulit untuk dibaca dan dimengerti. Oleh sebab itu, akan dilakukan proses *features extraction* dengan sebuah algoritma untuk mengekstrak *layer-layer* tersebut sehingga mendapatkan nilai-nilai yang kuat untuk dijadikan parameter-parameter sebagai pola dari paket data terenkripsi tersebut. Hasil dari proses *features extraction* dari *raw data* akan menghasilkan sebuah file *.csv*, dimana tipe file ini mudah diolah dan dapat diterima secara umum.

2.4. Proses Klasifikasi

Tahap terakhir dalam penelitian ini adalah proses pelatihan (*training*) dan klasifikasi menggunakan metode *Deep Packet Inspection* (DPI). Merujuk pada penelitian yang dilakukan oleh [5] menyatakan bahwa metode DPI menggunakan empat metode yaitu *Port-Based, Protocol Decoding, Pattern Matching* dan *Statistical Analysis*. Pada metode *Pattern Matching* dibagi lagi menjadi dua bagian yaitu *String Matching* dan *Regular Expression Matching.* Pada penelitian ini proses *training* dan klasifikasi menggunakan *Regular Expression Matching.* Berdasarkan penelitian [6] menyebutkan bahwa ada dua metode yang digunakan dalam *Regular Expression Matching* yaitu metode (NFA/DFA). Perbedaan dari kedua metode ini adalah kecepatan, dimana DFA lebih baik dibandingkan dengan NFA karena DFA dapat membandingkan semua karakter secara bersamaan. Tabel 1.1 merupakan pola-pola yang dipakai *Regular Expressions* dalam melakukan proses *scanning* paket *payload*. Pada penelitian [10] menggunakan *Regular Expressions* untuk mendeteksi pola traffik yahoo *"^(ymsg|ypns|yhoo).?.?.?.?.?.?.?[lwt].*\xc0\x80?*". Tahapannya adalah pencocokan semua paket *payload* dengan karakter *ymsg,ypns,yhoo* diikuti oleh karakter hexa, dan huruf *l, w atau t* , dan yang terakhir adalah karater ASCII "*xc0"* dan "x*80?"*

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

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

Tabel 1.1. Pola dari *Regular Expressions* [11]

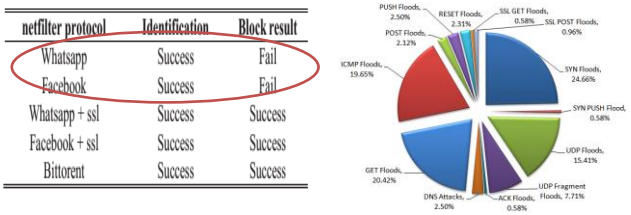
Sintak	Keterangan
^	Jika diletakkan di depan pattern, akan cocok dengan awal sebuah string
[]	ekspresi kurung. cocok dengan satu karakter yang berada dalam kurung
?	cocok dengan nol atau satu karakter sebelumnya
[^]	cocok dengan sebuah karakter yang tidak ada dalam kurung, berlawanan dengan yang diatas
*	cocok dengan nol atau lebih karakter sebelumnya

III. HASIL

Pada bagian ini akan dijelaskan tahap-tahap dalam proses analisis permasalahan, hasil dari *capture raw data*, proses *features extraction*, dan juga kebutuhan-kebutuhan sehingga mendapat hal yang baik

3.1. Analisis Permasalahan

Mengaju pada penelitian [2] ditemukan adanya kelemahan dalam melakukan proses *filter* paket data. Kelemahannya adalah belum dapat melakukan proses *filtering* paket data yang terenkripsi. Penelitian [10] juga belum dapat menghasilkan pola dari paket data yang terenkripsi



netfilter protocol	Identification	Block result
Whatsapp	Success	Fail
Facebook	Success	Fail
Whatsapp + ssl	Success	Success
Facebook + ssl	Success	Success
Bittorent	Success	Success

(a) (b)

Gambar 5. (a) Hasil penelitian [2], (b) Distribusi Tipe Serangan [12]

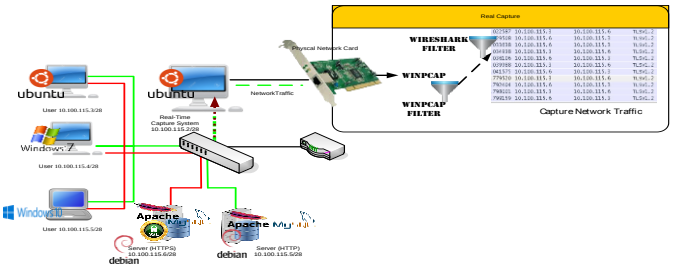
Menurut penelitian [12], distribusi tipe serangan untuk paket data yang terenkripsi sebesar 0.58 persen untuk SSL GET *Floods*, dan 0.9 persen untuk SSL POST *Floods*. Begitupula pada penelitian [13] yang menyatakan bahwa terdapat kesulitan dalam mengenali pola *significant shares of client* menggunakan metode *host-based pairing.*

3.2. Analisis Capture Raw Data

Packet capture (Pcap) adalah program yang digunakan untuk meng-*capture* trafik di *network*. Pada OS *windows* dikenal dengan nama wincap dan di OS linux dikenal dengan lipcap. Gambar 9 menampilkan struktur dari *file* PCAP, dimana pada sebuah *file* yang dihasilkan dari *capture traffic* akan memiliki *global hader* yang berisi tentang *global information* dan disertai oleh banyak *record* untuk setiap paket yang di *capture.*

Gambar 6. Struktur file PCAP

Pada penelitian ini, dasar dari proses validasi data akan melihat kebenaran dari *packet capture (raw data*), dimana proses *SSL Transaction* masih dilakukan secara *offline* dengan *ip user* 10.100.115.3/28, ip 10.100.115.2/28 digunakan sebagai mesin untuk *capture traffic*, sedangkan *server* SSL/TLS menggunakan ip 10.100.115.6/28. Kemudian user mengakes sebuah *website* dengan alamat https://10.100.115.6/.



Gambar 7. Proses *Capture* Trafik Paket Data

Dari Gambar 7 menampilkan hasil *capture* dengan cara memfilter paket "SSL" yang akan ditampilkan saat *user* (10.100.115.3/28) mengakses https://10.100.115.6/28 adalah protokol TSLV1.2, dimana menurut penelitian [14] menjelaskan bawah TSLV merupakan pengembangan dari SSL.

3.3. Validasi data SSL/TLS

Penelitian [15] menyatakan bahwa spesifikasi nilai *header* dari TLS ada empat komponen, yaitu *Content Types, Alert, ApplicationData, ChangeCipherSpec or Handshake* dimana setiap *record* harus memenuhi salah satu dari empat komponen tersebut.

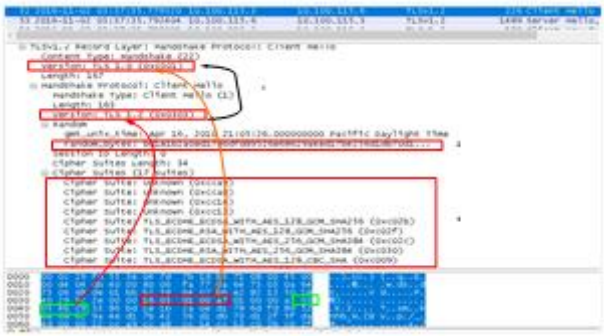
Tabel 1.2. Komponen dan Nilai dari TLS *Header*

No	Komponen	Nilai
1	Content Type	ChangeCipherSpec = 0x14 Alert = 0x15 Handshake = 0x16 ApplicationData = 0x17
2	Protocol Version	SSL v3 = 0x0300 TLS v1.0 = 0x0301 TLS v1.1 = 0x0302 TLS v1.2 = 0x0303
3	Handshake Message Type	ClientHello = 0x01 ServerHello = 0x02 Certificate = 0x0B ServerKeyExchange = 0x0C ServerHelloDone = 0x0E ClientKeyExchange = 0x10 Finished = 0x14
4	Alert Message Type	close_notify 0x00 unexpected_message 0x0A bad_record_mac 0x14 decryption_failed 0x15 record_overflow 0x16 decompression_failure 0x1E handshake_failure 0x28 no_certificate 0x29 SSLv3 only bad_certificate 0x2A unsupported_certificate 0x2B certificate_revoked 0x2C certificate_expired 0x2D certificate_unknown 0x2E illegal_parameter 0x2F unknown_ca 0x30 access_denied 0x31 decode_error 0x32 decrypt_error 0x33 export_restriction 0x3C protocol_version 0x46 insufficient_security 0x47 internal_error 0x50 user_canceled 0x5A no_renegotiation 0x64

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

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

- *ClientHello*

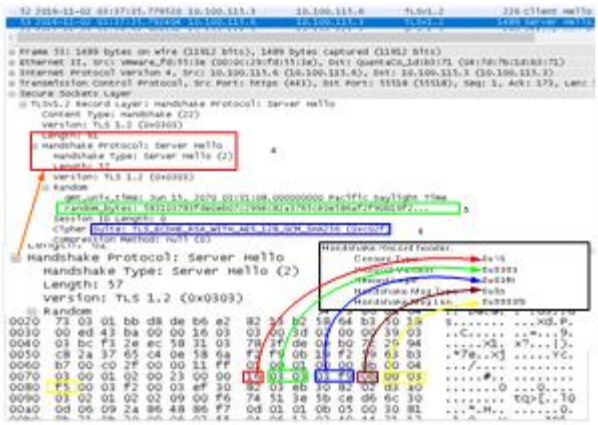


Gambar 8. Proses *ClientHello*

Pada Gambar 8 menjelaskan proses *Handshake* yang dimulai user (10.100.115.3/28) mengirimkan pesan ke *sever* (10.100.115.6/28), dimana pesan ini berisi versi SSL dari user(1), kemudian sebuah nilai acak yang akan digunakan untuk penurunan kunci(2), (3) daftar *cipher suite offer*.

- *ServerHello*

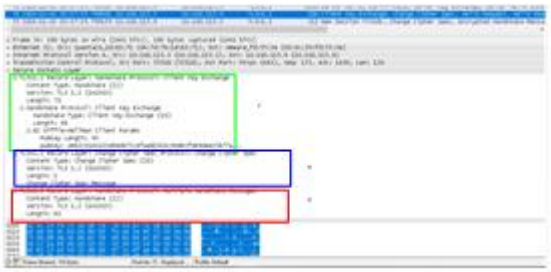
Step selanjutnya, *server* akan membalas pesan yaitu *ServeHello*, dimana pesan ini berisi tentang versi SSL yang disepakati(4), pada proses ke (6) *cipher* yang terkuat yang akan dipilih oleh server dari *offer cipher* user, dan proses (5) adalah bilangan acak yang dimiliki oleh server.



Gambar 9. Proses *ServerHello*

- *Certificate*

Gambar 10 ini adalah proses sertifikat, dimana proses point(7) adalah *Client* melakukan verifikasi *server* dan melakukan proses ekstraksi kunci publik *server* melalui pesan *client key exchange*, proses point(8) adalah *change cipher spec*, dan bagian terakhir proses *encrypted handshake message* (9).



Gambar 10. Proses sertifikat Client

Selajutnya *server* akan menerima *premaster* dari *client*, maka *server* dan *client* secara bersama-sama akan membangkitkan kunci simetri yang sama menggunakan *premaster*, selanjutnya dengan menggunakan TLS *pseudo-random function* (PRF), *server* dan *client* akan secara bersama-sama membangkitkan bilangan acak yang telah dipertukarkan sebelumnya. *Server* akan mengirim *change cipher spec* yang berisi rahasia dan beberapa data.

- *AplicationData*

Proses *Aplication Data protocol* dapat dilakukan setelah *handshake* selesai dilakukan oleh *server* dan *client*, kemudian data yang dikirim akan aman karena sudah terenkripsi.

- *Features extraction*

Merupakan proses pencarian ciri-ciri yang unik dari *raw data* yang akan diolah. Adapun manfaat dari proses ini adalah dapat memperkecil jumlah *raw data*, dapat meningkatkan ketelitian dalam pengolahan *raw data* dan juga digunakan untuk memberikan informasi-informasi yang penting data *raw data*. Dalam penelitian ini proses *feature extraction* menggunakan metode *Correlationbased Feature Selection* (CFS). Motede CFS ini adalah bagian dari metode *heuristic* dengan cara melihat fungsi-fungsi dari setiap field yang digunakan untuk prediksi kelas antar field dengan korelasi antar level. Persamaan dari *Correlationbased Feature Selection* (CFS) adalah sebagai berikut

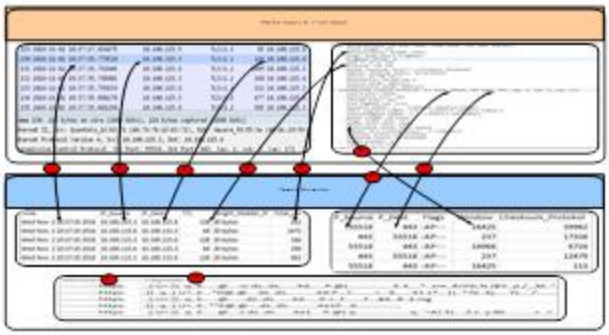
$$r_{sk} = \frac{kr_{cf}}{\sqrt{k+k(k-1)r_{ff}}} \qquad (1)$$

$$CFS = max_{sk}\left[\frac{r_{cf1}+r_{cf2}+\cdots+r_{cfk}}{\sqrt{k+k(k-1)r_{ff}}}\right] \qquad (2)$$

Dimana, rsk = Hubungan antara field, K = jumlah field, rcf = rata-rata hubungan sub field, dan rff = rata-rata korelasi antara bagian Field

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

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



Gambar 11. Hasil Ekstraksi dan *Raw Data*

Gambar 11 menampilkan hubungan antara hasil proses *feature extraction* dan hasil *capture* paket data, (1) mendeskripsikan waktu dan tanggal *capture*, (2) mendeskripsikan IP Source, (3) IP Dest, (4) mendeskripsikan *Time To Live* (TTL), (5) mendeskripsikan *Total Length*, (6) mendeskripsikan *IP Source*, (7) mendeskripsikan *IP Dest*, (8) mendeskripsikan *Windows Size*, (9) mendeskripsikan Service, (10) *Packet Content* yang merupakan konten atau *payload* dari paket data (ASCII dan *hexadecimal*).

3.4. Algortima

Pada penelitian ini, proses klasifikasi trafik terenkripsi akan menggunakan metode *Pattern Matching* (*Regular Expression). Regex* nantinya digunakan untuk mendeskripsikan himpunan karakter *string* yang terdiri dari konstanta yang menampilkan himpunan-himpunan *string* dan operasi antar *string* secara berurutan. Hasil dari beberepa percobaan (*training*) untuk pengenalan pola-pola paket terenskripsi seperti ditampilkan pada proses validasi data SSL/TLS didapat hasil pada Tabel 1.3 dan 1.4

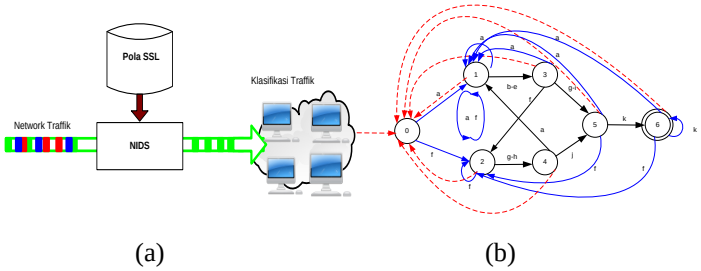
Tabel 1.3 Regex untuk TLS 1.0 dan TLS 1.2

No	Jenis Paket	Regex String (ServerHello)
1	TLS 1.0	"^(.\?.\?\16\03.*\16\03|.\?.\?\01\03\01\?.*\0B)"
2	TLS 1.2	"^(.\?.\?\16\03.*\16\03|.\?.\?\03\03\01\?.*\0B)"

Tabel 1.4 Keterangan Nilai Regex String

Field	Nilai		Keterangan
	TLS1.0	TLS1.2	
Record Content Type	0x16	0x16	Handshake
Protocol Version	0x0100	0x0300	SSL version untuk server
Handshake Type	0x01	0x03	Client Hello
service	0x0b	0x0b	Protokol

Selanjutnya akan dilakukan proses pencarian/pencocokan string dengan memanfaatkan algoritma *Deterministic Finite Automata* (DFA).



(a) (b)

Gambar 12. (a) Sistem Deteksi Paket Terenskripsi, (b) DFA untuk *Regex*

DFA merupakan sebuah fungsi yang terdefinisi untuk semua *state* dalam Q X ∑, mempunyai sifat determinstik yang bermakna bahwa setiap *automata* tidak bisa berada dalam *state-state* dalam keadaan yang sama, DFA difenisikan dalam persamaan berikut:

$$M = (Q,\Sigma, \delta,q0,F) \quad (3)$$

Dimana, Q merupakan himpunan terbatas dari *state*, Σ merupakan himpunan terbatas alphabet, δ: Q ×Σ, q0 ∈ Q adalah *state* awal, dan F⊆Q adalah *final state*. Pada Gambar 12 (b) merupakan contoh penerapan algortima DFA untuk pencocokan string.

IV. KESIMPULAN

Hasil yang didapatkan menunjukkan bahwa pola dari paket untuk proses *Client Hello* adalah dua versi TLS, yaitu TLS 1.0 dan TLS 1.2. TLS 1.0 dan TLS 1.2 memiliki *Field* yang terdiri dari *Record Content Type, Protocol Version, Handshake Type dan Service*. TLS 1.0 memiliki nilai *Record Content Type* 0x16, nilai *Protocol Version* untuk *ServerHello* adalah 0x0100, nilai *Handshake Type* 0x01, dan nilai protokol dari *Service* adalah 0x0b. TLS 1.2 memiliki nilai *Record Content Type* 0x16, nilai P*rotocol Version* untuk S*erverHello* adalah 0x0300, nilai *Handshake Type* 0x03, dan nilai protokol dari *Service* adalah 0x0b.

REFERENSI

[1] L. Deri, M. Martinelli, and A. Cardigliano, "nDPI : Open-Source High-Speed Deep Packet Inspection."

[2] H. Nazief, T. Sabastian, A. Presekal, and G. Guarddin, "Development of University of Indonesia Next Generation Firewall Prototype and Access Control With Deep Packet Inspection," *ICACSIS*, pp. 47–52, 2014.

[3] D. Smallwood and A. Vance, "Intrusion Analysis with Deep Packet Inspection Increasing Efficiency of Packet Based Investigations," pp.

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

342–347, 2011.

[4] R. Bendrath, "Ralf Bendrath Global technology trends and national regulation : Explaining Variation in the Governance of Deep Packet Inspection," *Int. Stud. Annu. Conv. New York City, 15-18 Febr. 2009*, no. February, p. 32, 2009.

[5] C. Xu, S. Chen, J. Su, S. M. Yiu, and L. C. K. Hui, "A Survey on Regular Expression Matching for Deep Packet Inspection : Applications , Algorithms and Hardware platforms," vol. 13, no. 9, 2016.

[6] thaksen j Parvat and P. Chandra, "A Novel Approach to Deep Packet Inspection for Intrusion Detection," *Int. Conf. Adv. Comput. Tecnol. Appl.*, vol. 45, pp. 506–513, 2015.

[7] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, "A Survey on Encrypted Traffic Classification," *Appl. Tech. Inf. Secur.*, pp. 73–81, 2014.

[8] M. Najam, U. Younis, and R. ur Rasool, "Speculative parallel pattern matching using stride-k DFA for deep packet inspection," *J. Netw. Comput. Appl.*, vol. 54, pp. 78–87, 2015.

[9] G. M. Wandhare, P. S. N. Gujar, and V. M. Thakare, "RESEARCH ARTICLE NETWORK INTRUSION DETECTION TECHNIQUE FOR REGULAR EXPRESSION DETECTION USING DPI IN AD-HOC," 2015.

[10] D. Stiawan, A. H. Abdullah, and M. Y. Idris, "Classification of Habitual Activities in Behavior- based Network Detection," no. October 2016, 2010.

[11] G. M. Wandhare and P. S. N. Gujar, "A Survey on DPI Techniques for Regular Expression Detection in Network Intrusion Detection System," vol. 2, no. 9, pp. 270–278, 2014.

[12] S. Bhople, "Server based DoS vulnerabilities in SSL / TLS Protocols Master Thesis," no. August, 2012.

[13] M. Husák, M. Cermák, T. Jirsík, and P. Celeda, "Open Access HTTPS traffic analysis and client identification using passive SSL / TLS fingerprinting," *EURASIP J. Inf. Secur.*, 2016.

[14] C. Meyer, "20 Years of SSL / TLS Research An Analysis of the Internet ' s Security Foundation," 2014.

[15] T. Dierks and E. Rescorla, "RFC 5246 - The transport layer security (TLS) protocol - Version 1.2," in *Network Working Group, IETF*, 2008, pp. 1–105.