Model Monitoring Keamanan

by Slamet A.

Submission date: 03-Apr-2023 10:17PM (UTC+0700)
Submission ID: 2054685657
File name: 116-327-1-PB.pdf (975.79K)
Word count: 4718
Character count: 26281

Model *Monitoring* Keamanan Jaringan untuk *Anomaly Detection* di Jaringan Kampus: Studi Kasus Stikom Surabaya

Slamet
slamet@stikom.edu
Information System Department
Institut Bisnis dan Informatika Stikom Surabaya, Indonesia

Abstrak

Saat ini keamanan jaringan telah menjadi topik penting dalam penelitian di area jaringan komputer. Salah satu implementasinya adalah sebagai sebuah infrastruktur pada layanan informasi universitas, jaringan kampus memainkan peran penting dalam bidang pengajaran, penelitian, dan manajemen. Tidak adanya perhatian terhadap keamanan jaringan, terlebih terhadap data yang dimiliki, membuat layanan informasi rentan terhadap gangguan *intruder*. Penelitian ini bertujuan menyelidiki ancaman keamanan dan kerentanan jaringan pada sistem jaringan kampus. Untuk mengatasi ancaman keamanan dan kerentanan jaringan digunakan Model *Monitoring* Keamanan Jaringan, khususnya untuk mendeteksi *traffic* anomali. Model *monitoring* ini memiliki empat bagian utama yaitu bagian pengambilan data *traffic* di jaringan, bagian pendeteksi data anomali, bagian penganalisis NetFlow dan bagian pelaporan insiden keamanan. *Probe* FlowMon pada bagian pengambilan data *traffic* menghasilkan data NetFlow dan mengekspornya kepada kolektor yang ditangani oleh NFDUMP dan NfSen. Aplikasi ini juga menyediakan penyimpanan dan pengambilan informasi *traffic* yang kemudian dilakukan analisis forensik. Selanjutnya, data NetFlow diproses oleh pendeteksi anomali. Data *traffic* yang mencurigakan atau terlihat anomali di jaringan divisualisasikan pada *interface* pengguna atau dilaporkan melalui email oleh *System* kepada Administrator Jaringan. Hasil dari penelitian ini adalah Administrator jaringan tidak perlu mengamati perilaku jaringan setiap saat, namun seorang Administrator dapat memfokuskan diri pada respon dan resolusi terhadap insiden keamanan.

Kata Kunci: Keamanan Jaringan, *Monitoring* Jaringan, NfSen, NfDump, Deteksi Anomali

1. PENDAHULUAN

Memastikan sistem keamanan dalam sebuah jaringan berkecepatan tinggi adalah tugas Administrator Jaringan yang membutuhkan intensitas dan respek yang tinggi dalam pekerjaannya sehari-sehari. Untuk melakukan pengawasan keamanan, diperlukan Administrator Jaringan yang sangat berpengalaman, yang memiliki wawasan mendalam terhadap perilaku jaringan dan memahami kondisi jaringan dengan baik. Beberapa tugas yang sering dilakukan oleh Administrator Jaringan [1] adalah mengamati grafik dan statistik data *traffic*, mencari puncak (*peak*) pemakaian yang tidak biasa, misalnya dalam volume *byte* dari paket yang ditransfer. Selain itu, Administrator Jaringan juga memeriksa insiden tertentu menggunakan aplikasi seperti penganalisis paket, kolektor aliran, *firewall* dan *log* sistem pada server. Proses melakukan analisis yang lebih mendalam dari paket atau *traffic* tertentu membutuhkan waktu yang lama dan pengetahuan yang sangat baik tentang perilaku jaringan (*network behaviour*).

Demikian pula seseorang yang bertanggung jawab terhadap insiden keamaanan jaringan seperti CERT (Computer Emergency Response Team) atau CSIRT (Computer Security Incident Response Team) selalu melakukan *monitoring* keamanan jaringan dan juga menganalisis perilaku jaringan dalam tugasnya. Berbasis data yang didapatkan, team ini dapat mendeteksi dan mencegah pemakaian yang tidak diinginkan dari pengguna komputer di jaringan. Adanya *system monitoring* keamanan memberikan solusi yang sangat penting dalam melakukan penanganan terhadap insiden keamanan.

Paper ini memperkenalkan konsep baru yang dapat mengurangi beban pekerjaan secara signifikan bagi seorang Network Administrator, CERT, CSIRT atau Network Operator. Biasanya seorang Network Operator yang berpengalaman masih membutuhkan waktu yang lama dalam melakukan *network surveillance*. Seorang System Administrator tidak perlu lagi mengamati perilaku jaringan secara terus menerus, tetapi pekerjaannya dapat difokuskan pada respon dan resolusi dari insiden keamanan. System ini mampu membuat laporan insiden yang sistematis dan terklasifikasi terkait *traffic* jaringan yang *untrustfull*. Oleh karena itu, seorang System Administrator dapat menerima laporan tentang insiden keamanan dan kemudian memeriksanya. Jika tidak ada kondisi *false positive* [2] maka ia dapat melakukan tindakan lebih lanjut yang diperlukan.

Penelitian ini terdiri dari beberapa bagian. Diawali dengan membahas Flow Mon berbasis *monitoring traffic* yang didalamnya terdiri dari pembuatan (*generate*) data netflow dan koleksi data netflow. Kemudian digunakan AlienVault OSSIM untuk mendeteksi *traffic* anomali dan teknik *trust modelling* untuk

mengurangi *false positive*. Data yang digunakan sebagai percobaan diambil secara *real time* dari *traffic* jaringan Stikom Surabaya, khususnya data terhadap serangan virus Conficker di jaringan ini.

2. METODE PENELITIAN

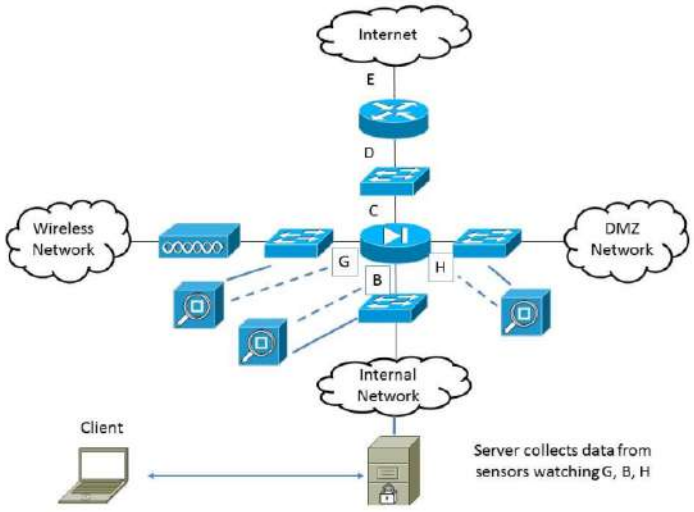
Model *Monitoring* Keamanan Jaringan yang digunakan memiliki empat bagian. Model ini terdiri dari bagian pengambilan data dari *interface* FlowMon, bagian kolektor data / pendeteksi anomali (*anomaly detection*), bagian penganalisis data NetFlow (NfSen) dan bagian pelaporan insiden (*incident reports*), seperti yang ditunjukkan pada Gambar 1.

Pada bagian pengambilan data, *traffic* diambil menggunakan FlowMon dengan menggunakan *hardware* Cisco [3] sehingga menghasilkan data NetFlow. Kemudian data ini dikirim kepada bagian kolektor. Dalam hal ini yang bertugas sebagai kolektor dan menangani data netflow adalah NFDUMP [4] dan NfSen [5]. Aplikasi ini juga melakukan penyimpanan dan pengambilan informasi dari *traffic* data untuk dilakukan analisis forensik. Selanjutnya Data NetFlow diolah oleh pendeteksi anomali (*anomaly detection*). Aliran yang mencurigakan atau adanya anomali di jaringan dapat dilihat pada *web interface* dari pengguna atau dikirim melalui email Administrator Jaringan terkait hasil pelacakan, pelaporan dan pemberitahuan.

Gambar 1: Model Monitoring Keamanan Jaringan

2.1 *Monitoring* Keamanan Jaringan

Untuk menyampaikan fakta terkait kondisi jaringan yang ada, diperlukan statistik dari *traffic* secara detil seperti rekam jejak (*trace route*) paket dan volume statistik *traffic*. Pada penelitian ini, bahan pengukuran diambil dari data *traffic* di dalam jaringan Stikom Surabaya. *Trace route* yang digunakan oleh penganalisis *traffic* memberikan informasi yang rinci tentang statistik *header IP*, *payload*, lamanya waktu dalam berkomunikasi dan dengan siapa IP *address* tertentu berkomunikasi. Volume statistik diperoleh dari data SNMP yang bisa memberikan informasi secara akurat yang kemudian dilakukan analisis lebih lanjut.



Gambar 2: Implementasi Model Traffic Monitoring di Jaringan Kampus Stikom

Pada gambar 2 terlihat bahwa Arsitektur *Traffic Monitoring System* bertumpu pada router utama yang juga berfungsi sebagai titik *monitoring* utama. Titik yang dimonitor digunakan sebagai *probe monitoring* dengan cara mengkonfigurasi beberapa *interface* pada router ini. Dengan kata lain, beberapa *interface* yang telah dikonfigurasi pada router ini dijadikan sumber dalam *monitoring* aliran data.

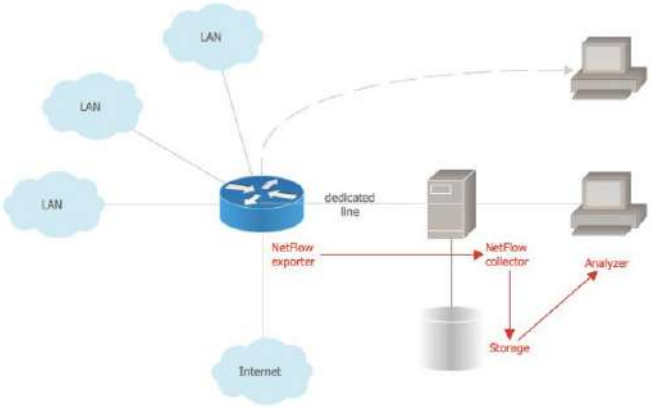
Untuk menjaga skalabilitas *system* digunakan aliran data berisi informasi yang sudah terenkripsi dari NetFlow pada router CISCO. Proses *monitoring* dilakukan untuk mengamati aliran data pada *link* yang digunakan oleh pengguna sampai dengan *link* pada *backbone* jaringan.

2.2 NetFlow Generator

Secara umum, Flow atau aliran data adalah seperangkat paket yang berbagi properti secara umum. Sifat-sifat yang paling penting dari Flow adalah titik akhir dari aliran data ini. Sedangkan NetFlow artinya aliran data yang terjadi pada jaringan komputer. Jenis NetFlow yang paling sederhana adalah 5-*tuple*, dimana semua paketnya memiliki *IP Address* sumber dan tujuan, nomor *port* dan protokol yang sama. Pergerakan aliran data biasanya tidak searah namun semua paketnya bergerak ke arah yang sama. Paket pertama diamati pada titik dimana aliran data mulai mengalir. Aliran data akan berhenti ketika tidak ada *traffic* baru yang perlu diamati (*inactive timeout*) atau pada kondisi koneksi berakhir (misalnya koneksi TCP ditutup). *Inactive timeout* adalah periode waktu setelah arus data akan diekspor. Arsitektur FlowMon dapat dilihat pada gambar 3.

Statistik *traffic* menyediakan informasi tentang: siapa berkomunikasi dengan siapa, kapan komunikasi dilakukan, berapa lama komunikasi terjadi, protokol apa yang digunakan, layanan apa yang digunakan dan juga berapa banyak data yang ditransfer.

Untuk memperoleh statistik router dapat digunakan NetFlow atau *probe* tertentu [6]. Proses mengalirkan data *traffic* pada router dapat mengkonsumsi 50% - 60% dari kinerja router sehingga mempengaruhi *performance* jaringan. Namun dengan menggunakan aliran data secara *dedicated* pada saat *monitoring traffic* data, maka *performance* jaringan tidak akan terpengaruh.



Gambar 3: Arsitektur FlowMon menggunakan Cisco Router

NetFlow dikembangkan oleh Cisco Systems. Saat ini standar industri telah didukung oleh banyak vendor dan telah merilis beberapa versi protokol, tetapi yang paling umum digunakan adalah Netflow versi 5 dan 9.

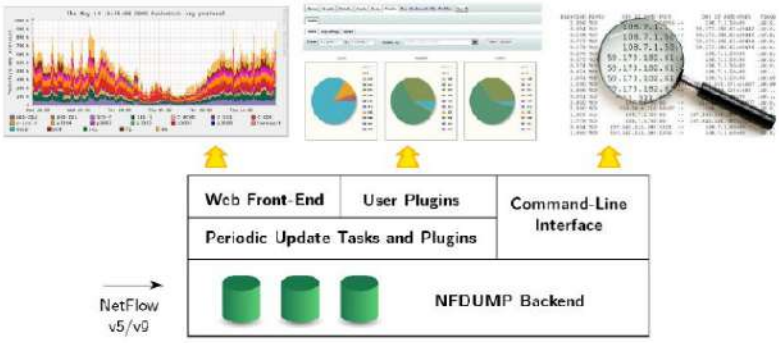
Komponen-komponen yang digunakan untuk mengumpulkan informasi tentang *traffic* IP pada protokol Netflow adalah:

1. Sensor, digunakan untuk mengumpulkan statistik *traffic* jaringan yang melewati switch dan router.
2. Kolektor, digunakan untuk mengumpulkan data yang datangnya dari sensor dan menempatkannya ke media penyimpanan.
3. Analyzer, digunakan untuk menganalisis data yang dikumpulkan dan membuat laporan kepada orang yang diperbolehkan, misalnya Administrator.

2.3 Kolektor *NetFlow*

Tugas utama Kolektor adalah menyimpan paket yang masuk ke dalam *database*, dalam hal ini menggunakan SQL *database*. Di dalam Kolektor terdapat grafik yang merupakan representasi dari *traffic* jaringan, filtrasi aliran data, agregasi data dan evaluasi statistik yang menggunakan IP *Address* sumber dan IP *Address* tujuan, port dan protokol.

Gambar 4 menunjukkan diagram blok dari kolektor NfSen (Netflow Sensor). NfSen adalah *front-end* berbasis web dari NFDUMP. Pada NfSen terdapat *interface* bagi pengguna untuk menampilkan grafik NetFlow. Selain itu terdapat *interface* baris perintah yang digunakan untuk memproses aliran data.



Gambar 4: Diagram Blok Kolektor NfSen / NFDUMP

Profile NfSen adalah *profile* yang bisa dibentuk dari data NetFlow. *Profile* yang terbentuk ditentukan berdasarkan nama, jenis, dan filter *profile*, dimana *profile* merupakan filter yang bisa diterima oleh NfSen. Administrator jaringan menentukan *profile* dan grup *profile* sendiri untuk membuat grafik aliran data.

Interface kolektor memberikan *tool* untuk peringatan (*alert*) secara otomatis. Administrator Jaringan menentukan sekumpulan notifikasi berdasarkan kondisi jaringan, tergantung dari data yang dibentuk dari NetFlow. *Alert* yang ada melakukan tindakan secara otomatis berdasarkan kondisi tertentu, misalnya mengirim email kepada Administrator ketika terjadi gangguan dalam *traffic* jaringan.

Kolektor NfSen mendukung API (*Application Programming Interface*) untuk mengintegrasi ekstensi *plugin-plugin*. *Interface* ini terdiri dari dua bagian, bagian *front-end* diakses melalui web dan melakukan interaksi langsung dengan pengguna. Sedangkan bagian *back-end* melakukan pemrosesan data yang diminta dan mempersiapkannya untuk *interface plugin*. Bagian *front-end* diimplementasikan dalam pemrograman PHP, sedangkan bagian *back-end* diimplementasikan dalam bahasa Perl.

2.4 Deteksi Anomali dan Analisis *Behavior* Jaringan

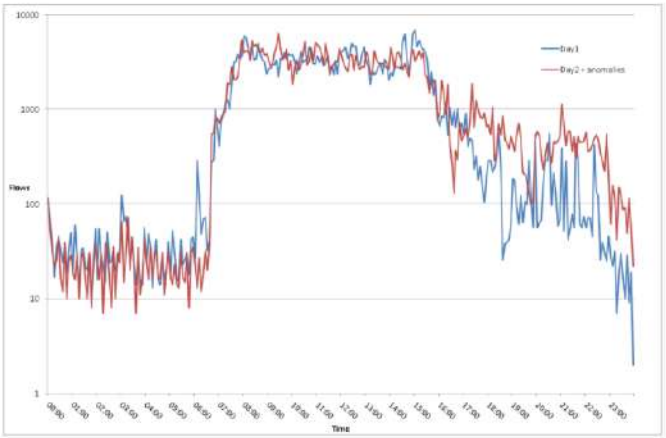
Network Behavior Analysis System mendeteksi serangan yang terjadi, *traffic* anomali, ancaman, dan perilaku yang tidak diinginkan berdasarkan pada evaluasi otomatis secara terus-menerus di jaringan. Selain itu *system* ini juga melakukan analisis statistik dari *traffic* jaringan (NetFlow / IPFIX) yang dihasilkan oleh *Probe* NetFlow, *device* aktif (seperti switch, router) atau alat lainnya (misalnya firewall).

Tujuan dari *Network Behavior Analysis System* adalah untuk mengidentifikasi masalah keamanan, masalah operasional dan untuk meningkatkan keamanan jaringan. Keuntungan utama dari mendeteksi dan mencegah serangan adalah berfokus pada perilaku keseluruhan *device* di jaringan komputer, sehingga dapat menanggapi ancaman atau *signature* spesifik yang tidak diketahui.

Paradigma deteksi anomali dapat digunakan untuk pemrosesan data NetFlow. Hal ini karena sifatnya yang efektif dan berdimensi rendah dalam rangka memproses karakteristik *traffic* jaringan, baik dalam karakteristik volume [7] maupun karakteristik distribusi parameter [8].

Metode deteksi anomali menggunakan pengamatan *history traffic* untuk membangun model dengan karakteristik yang relevan dengan perilaku jaringan. Metode ini juga dapat memprediksi karakteristik untuk *traffic* jaringan di masa depan dan dapat mengidentifikasi sumber perbedaan antara prediksi dan nilai yang sebenarnya diamati sebagai kemungkinan serangan.

Keuntungan utama dari pendekatan ini adalah kemampuannya untuk mendeteksi serangan yang sulit dideteksi oleh *Intrusion Detection System* (IDS) pada umumnya. IDS pada umumnya mendeteksi dan mengidentifikasi pola-pola dari *malicious* yang diketahui di dalam paket, seperti eksploitasi *zero-day*, *malware* yang dimodifikasi sendiri, serangan di dalam *traffic* jaringan, penyalah-gunaan *resource* atau kesalahan konfigurasi.



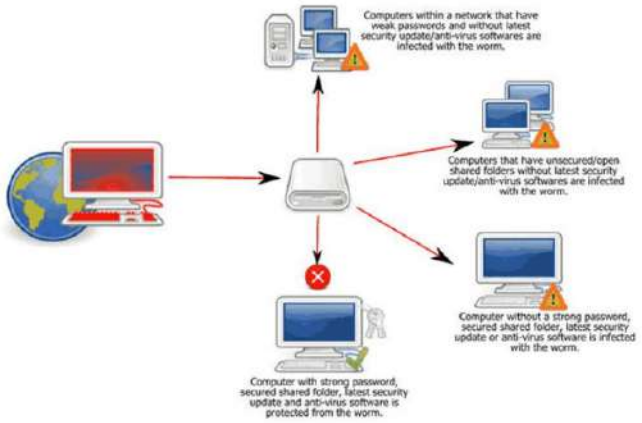
Gambar 5. Metode Deteksi Anomaly Traffic dan Legitimate Traffic

Masalah dari *Intrusion Detection Systems* (IDS) berbasis anomali adalah pada tingkat kesalahannya, yang terdiri dari dua jenis kesalahan, yaitu *false positive* dan *false negative*. *False positives* adalah *legitimate traffic* yang diklasifikasikan sebagai anomali, sedangkan *false negatives* adalah *traffic malicious* yang diklasifikasikan sebagai *traffic* normal. Kebanyakan deteksi anomali atau metode *Network Behaviour Analysis* (NBA) mempunyai tingkat *false positives* yang sangat tinggi sehingga dilakukan modifikasi untuk *deployment*. Sistem NBA menambahkan berbagai pendekatan seperti proses kolaborasi *multistage* dari deteksi, *trust modelling*, atau adaptasi *autonomous* untuk membuat deteksi anomali dapat diterapkan secara operasional.

3. HASIL DAN PEMBAHASAN

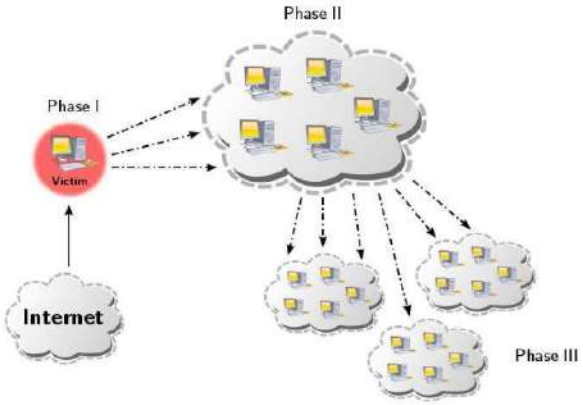
Pada bagian ini menggambarkan *use case* yang menunjukkan kemampuan deteksi anomali dari perspektif pengguna (misalnya, seorang Administrator Jaringan atau Analis Insiden Keamanan). Percobaan penelitian dilakukan untuk mendeteksi *malware* - Win32 / Conficker *worm* [9].

Pada penelitian ini, dilakukan percobaan yang membandingkan deteksi anomali menggunakan *command line* dari NFDUMP dan *dedicated detection system*. Proses mendeteksi *worm* Conficker dilakukan di jaringan kampus Stikom Surabaya.



Gambar 6: Cara Penyebaran Conficker

Win32 / Conficker, juga dikenal sebagai Downup, Downadup dan Kido, adalah *worm* komputer yang menargetkan sistem operasi Microsoft Windows sebagai target serangan. *Worm* ini mengeksploitasi kerentanan yang diketahui terdapat pada jaringan lokal Microsoft Windows. Eksploitasi yang dimaksud adalah menggunakan *Remote Procedure Call* (RPC) melalui port 445 / TCP, sehingga menyebabkan eksekusi kode segmen acak tanpa memerlukan otentikasi. Conficker dilaporkan telah menginfeksi hampir 9 juta PC [10]



Gambar 7: Worm Conficker menyebarkan serangan pada jaringan yang sedang dimonitor

Gambar 6 mengilustrasikan bagaimana Conficker menyebar, sedangkan gambar 7 memperlihatkan bagaimana *worm* ini menyebar dengan cepat di dalam jaringan kampus yang sedang dimonitor. Korban pertama (*victim*) terinfeksi *worm* pada fase I. Fase utama atau fase II dimulai pada pukul 11:00:12 dimana *victim* melakukan aktivitas *scanning* secara masif terhadap komputer di jaringan lokal dan jaringan internet. Tujuannya adalah untuk menemukan dan menginfeksi *host* yang sedang rentan. *Port* yang dituju adalah *port* 445, karena port ini adalah target dari celah keamanan Microsoft Windows. Satu jam kemudian, banyak komputer kampus terinfeksi oleh *worm* ini, dan sekali lagi *conficker* melakukan *scanning* untuk menyebarkan serangan di fase III ke komputer-komputer lainnya, baik di jaringan lokal maupun internet kampus.

3.1 Analisis NetFlow Menggunakan NFDUMP

Pada penelitian ini ditunjukkan bagaimana *worm* berkembang di jaringan kampus dan menyebar kepada banyak komputer akibat adanya *host* yang telah terinfeksi. Untuk melindungi privasi pengguna, semua alamat IP dan nama *domain* diubah (dianonimkan). *Host* yang terinfeksi (IP Address 172.25.88.48) mulai berkomunikasi di dalam jaringan kampus pada pukul 11:00:12.

Flow start	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Packets	Bytes	Flows
11:00:12.024	UDP	172.25.88.48:49417	->	224.0.0.252:5355	.....	2	102	1
11:00:12.537	UDP	172.25.88.48:60435	->	224.0.0.252:5355	.....	2	102	1
11:00:14.446	ICMP	172.25.83.1:0	->	172.25.88.48:3.10	.....	25	3028	1
11:00:14.446	UDP	172.25.88.48:137	->	172.25.88.255:137	.....	25	2238	1
11:00:21.692	UDP	172.25.88.48:60436	->	172.25.83.1:53	.....	2	162	1
11:00:21.692	UDP	172.25.83.1:53	->	172.25.88.48:60436	.....	2	383	1
11:00:21.763	UDP	172.25.88.48:5353	->	224.0.0.251:5353	.....	9	867	1
11:00:24.182	UDP	172.25.88.48:60438	->	239.255.255.250:3702	.....	6	6114	1
11:00:24.470	UDP	172.25.88.48:138	->	172.25.88.255:138	.....	3	662	1
11:00:26.069	UDP	172.25.88.48:60443	->	239.255.255.250:1900	.....	14	2254	1
11:00:39.635	UDP	172.25.88.48:55938	->	224.0.0.252:5355	.....	2	104	1
11:00:40.404	UDP	172.25.88.48:60395	->	172.25.83.1:53	.....	1	50	1
11:00:40.405	UDP	172.25.83.1:53	->	172.25.88.48:60395	.....	1	125	1
11:00:40.407	UDP	172.25.88.48:52932	->	224.0.0.252:5355	.....	2	100	1
11:00:42.134	UDP	172.25.88.48:51504	->	224.0.0.252:5355	.....	2	104	1
11:00:42.160	UDP	172.25.88.48:52493	->	224.0.0.252:5355	.....	2	102	1
11:00:42.461	UDP	172.25.88.48:55260	->	224.0.0.252:5355	.....	2	102	1
11:00:43.243	UDP	172.25.88.48:64291	->	172.25.83.1:53	.....	1	62	1
11:00:43.244	UDP	172.25.88.48:50664	->	172.25.83.1:53	.....	1	62	1
11:00:43.244	UDP	172.25.83.1:53	->	172.25.88.48:64291	.....	1	256	1
11:00:43.246	UDP	172.25.83.1:53	->	172.25.88.48:50664	.....	1	127	1
11:00:43.246	**TCP**	**172.25.88.48:49158**	**->**	**207.46.131.206:80**	**A.RS.**	**4**	**172**	**1**

11:00:43.437	**TCP**	**207.46.131.206:80**	**->**	**172.25.88.48:49158**	**AP.SF**	**3**	**510**	**1**
11:00:43.631	UDP	172.25.88.48:63820	->	172.25.83.1:53	.....	1	62	1
11:00:43.673	UDP	172.25.83.1:53	->	172.25.88.48:63820	.....	1	256	1
11:00:44.374	UDP	172.25.88.48:51599	->	224.0.0.252:5355	.....	2	104	1
11:00:45.170	UDP	172.25.88.48:137	->	172.25.88.255:137	.....	11	858	1
11:00:45.876	UDP	172.25.88.48:61423	->	224.0.0.252:5355	.....	2	102	1
11:00:45.881	UDP	172.25.88.48:54743	->	224.0.0.252:5355	.....	1	51	1
11:00:52.792	UDP	172.25.88.48:52975	->	224.0.0.252:5355	.....	2	104	1
11:00:54.719	UDP	172.25.88.48:62459	->	172.25.83.1:53	.....	1	62	1

30 menit kemudian, pada pukul 11:30:42, *worm* memulai aktivitas *scanning* secara *massive* terhadap komputer-komputer lain, baik di jaringan lokal maupun internet. Tujuannya adalah untuk menemukan dan menginfeksi *host* lainnya yang rentan, dengan port tujuan 445.

Flow start	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Packets	Bytes	Flows
11:30:42.963	TCP	172.25.88.48:49225	->	100.9.240.76:445	...S.	1	48	1
11:30:42.963	TCP	172.25.88.48:49226	->	209.13.138.30:445	...S.	1	48	1
11:30:42.963	TCP	172.25.88.48:49224	->	71.70.105.4:445	...S.	1	48	1
11:30:42.964	TCP	172.25.88.48:49230	->	150.18.37.52:445	...S.	1	48	1
11:30:42.965	TCP	172.25.88.48:49238	->	189.97.157.63:445	...S.	1	48	1
11:30:42.965	TCP	172.25.88.48:49235	->	46.77.154.99:445	...S.	1	48	1
11:30:42.965	TCP	172.25.88.48:49237	->	187.96.185.74:445	...S.	1	48	1
11:30:42.965	TCP	172.25.88.48:49234	->	223.62.32.43:445	...S.	1	48	1
11:30:42.966	TCP	172.25.88.48:49236	->	176.77.174.109:445	...S.	1	48	1
11:30:42.966	TCP	172.25.88.48:49239	->	121.110.84.84:445	...S.	1	48	1
11:30:42.966	TCP	172.25.88.48:49243	->	153.34.211.79:445	...S.	1	48	1
11:30:42.967	TCP	172.25.88.48:49244	->	59.34.59.14:445	...S.	1	48	1
11:30:42.967	TCP	172.25.88.48:49245	->	172.115.82.70:445	...S.	1	48	1
11:30:42.967	TCP	172.25.88.48:49246	->	196.117.5.44:445	...S.	1	48	1
11:30:42.968	TCP	172.25.88.48:49258	->	78.33.209.5:445	...S.	1	48	1
11:30:42.968	TCP	172.25.88.48:49248	->	28.36.5.3:445	...S.	1	48	1
11:30:42.968	TCP	172.25.88.48:49259	->	91.39.4.28:445	...S.	1	48	1
11:30:42.968	TCP	172.25.88.48:49254	->	112.96.125.115:445	...S.	1	48	1
11:30:42.969	TCP	172.25.88.48:49262	->	197.63.38.5:445	...S.	1	48	1
11:30:42.969	TCP	172.25.88.48:49268	->	36.85.125.20:445	...S.	1	48	1
11:30:42.969	TCP	172.25.88.48:49261	->	170.88.178.77:445	...S.	1	48	1
11:30:42.969	TCP	172.25.88.48:49260	->	175.42.90.106:445	...S.	1	48	1
11:30:42.969	TCP	172.25.88.48:49263	->	15.70.58.96:445	...S.	1	48	1

30 menit kemudian, banyak komputer kampus yang terinfeksi, dan sekali lagi *worm* ini mencoba untuk melakukan *scanning* dan menularkan *virus* ke komputer lainnya yang ada di jaringan lokal maupun internet kampus.

Flow start	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Packets	Bytes	Flows
12:00:10.983	TCP	172.25.88.31:50076	->	145.107.246.69:445	AP.S.	30	1259	1
12:00:25.106	UDP	172.25.88.49:63593	->	38.81.201.101:445	.....	6	1408	1
12:00:25.894	TCP	172.25.88.47:51875	->	169.41.101.97:445	AP.S.	29	1298	1
12:00:26.001	TCP	172.25.88.49:63778	->	43.28.146.45:445	AP.S.	18	906	1
12:00:26.948	TCP	172.25.88.50:52225	->	104.24.33.123:445	AP.S.	10	537	1
12:00:27.466	TCP	172.25.88.35:55484	->	109.18.23.97:445	AP.SF	102	146397	1
12:00:28.443	TCP	172.25.88.37:53098	->	102.124.181.67:445	AP.S.	15	804	1
12:00:28.473	TCP	172.25.88.38:60340	->	222.50.79.96:445	AP.S.	23	4549	1
12:00:28.797	TCP	172.25.88.37:53174	->	212.82.132.58:445	AP.S.	19	861	1
12:00:29.267	TCP	172.25.88.34:64769	->	34.56.183.93:445	AP.S.	17	1696	1
12:00:29.409	TCP	172.25.88.34:64756	->	89.109.215.111:445	AP.S.	17	3037	1
12:00:29.492	TCP	172.25.88.44:57145	->	32.113.4.81:445	AP.S.	15	2562	1
12:00:29.749	TCP	172.25.88.43:52707	->	138.8.147.38:445	AP.S.	16	1725	1
12:00:30.159	TCP	172.25.88.49:63902	->	203.101.75.18:445	AP.S.	22	2316	1
12:00:31.116	TCP	172.25.88.31:50766	->	194.125.49.68:445	...S.	2	96	1
12:00:31.117	TCP	172.25.88.31:50768	->	193.114.216.37:445	...S.	2	96	1
12:00:31.117	TCP	172.25.88.31:50769	->	37.107.5.111:445	...S.	2	96	1
12:00:31.117	TCP	172.25.88.31:50770	->	126.96.239.95:445	...S.	2	96	1
12:00:31.118	TCP	172.25.88.31:50776	->	43.87.170.91:445	...S.	2	96	1
12:00:31.119	TCP	172.25.88.31:50778	->	103.13.70.122:445	...S.	2	96	1
12:00:31.127	TCP	172.25.88.31:50784	->	200.68.202.35:445	...S.	2	96	1
12:00:31.129	TCP	172.25.88.31:50791	->	56.39.208.87:445	...S.	2	96	1

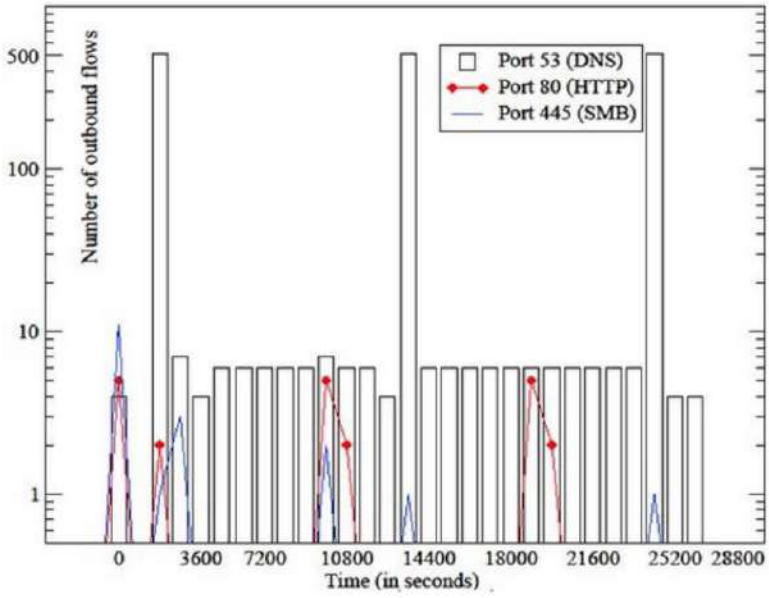
12:00:31.131 TCP 172.25.88.31:50797 -> 59.104.110.104:445 ...S. 2 96 1

3.2 Deteksi dan Analisis *Worm* menggunakan NBA (Network Behavior Analysis)

AlienVault OSSIM [11], merupakan *sofware Open Source* dari Security Information and Event Management (SIEM). Software ini digunakan untuk menganalisis *network behavioral* dengan data *event collection*, melakukan normalisasi dan korelasi. Tujuan dari penggunaan AlienVault OSSIM adalah untuk merancang dan menerapkan *Intrusion Detection System* (IDS) pada jaringan kampus. Sistem ini mengamati *traffic* jaringan yang telah menggunakan FlowMon, mendeteksi anomali di jaringan dan menggambarkan *traffic* anomali di jaringan.

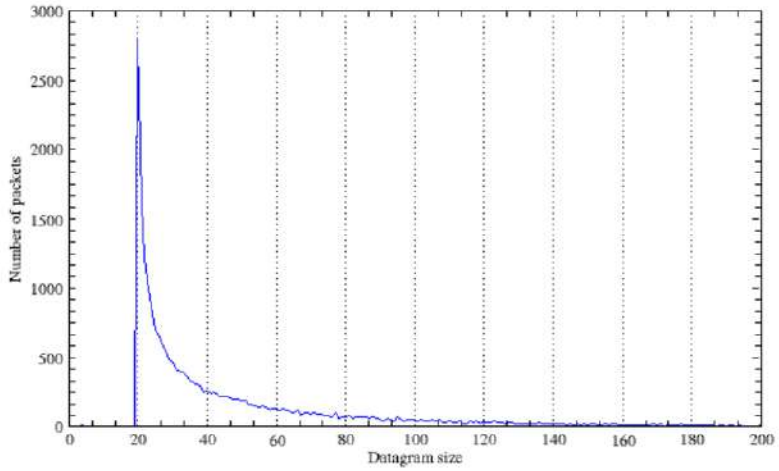
AlienVault OSSIM mengurutkan aliran berdasarkan klasifikasi *traffic* dan menempatkan potensi *traffic* anomali yang dapat dilihat pada gambar 8. Titik puncak grafik sebagai penanda *traffic* anomali dengan mudah ditemukan dan kemudian dianalisis dalam tool analisis *traffic*, seperti yang ditampilkan pada gambar 9.

Gambar 8: Prosentase aliran data saat aktivitas penyebaran Conficker.



Gambar 9: Analisis yang menggambarkan distribusi traffic Conficker berdasarkan port tujuan

Pada Gambar 10 menunjukkan distribusi plot ukuran datagram dari paket UDP PING dengan nilai minimum 20 byte. Dalam beberapa kasus penyebaran dimulai terlebih dulu dengan aktivitas UDP yang kemudian diikuti aktivitas TCP. Hal ini karena saluran UDP yang menyediakan negosiasi seperti *key exchange* sebelum pertukaran data TCP.



Gambar 10: Distribusi datagram size untuk paket PING UDP

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Pemanfaatan Kolektor Netflow dapat digunakan oleh CERT (Computer Emergency Response Teams) atau

Network Administrator dalam mendeteksi insiden keamanan komputer secara lebih efisien. Target masalahnya adalah karena dibutuhkannya pengetahuan dan pengalaman yang tinggi pada seorang Network Security Engineer dan terbatasnya kemampuan seorang operator yang mampu menangani semua *traffic* jaringan secara *real time*. Sehingga pemakaian Netflow ini dapat mereduksi tingkat kebutuhan dan pengetahuan yang tinggi bagi pengguna dan membantu penanganan *traffic* secara *real time*.

System pendeteksi berbasis *flow* ini mampu mengidentifikasi *untrustfull traffic* dan kejadian-kejadian yang bersembunyi dibalik *traffic* normal. Banyaknya analisis aliran data di dalam data statistik dan keseluruhan nilai agregasi dari seluruh kondisi jaringan dapat dimanfaatkan oleh operator dalam menginvestigasi permasalahan. Selain itu, operator dapat memilih laporan-laporan yang hanya diperlukan dengan lebih efisien.

4.2 Saran

Dalam penelitian ini dibutuhkan *link* jaringan dengan kecepatan tinggi terkait kebutuhan *bandwidth*. Untuk itu, infrastruktur jaringan dengan *bandwidth* tinggi ini membuka peluang untuk bisa diteliti lebih mendalam karena pengawasan *traffic* dengan volume yang tinggi hampir tidak mungkin dilakukan secara manual.

Pengukuran *traffic* data yang akurat adalah bagian penting dari deteksi anomali jaringan. Pada analisis *behavior* jaringan yang dilakukan secara *on-line* berdampak pada kemungkinan kesalahan klasifikasi terkait pengambilan sampel. Pengambilan sampel dapat berdampak negatif terhadap proses deteksi, semisal mengarah pada penemuan infeksi pada tahap awal, sehingga membuat reaksi menjadi kurang efisien. Untuk itu diperlukan metode untuk mengurangi kesalahan klasifikasi dalam pengambilan sampel.

Mendeteksi serangan dalam sebuah *data set traffic* biasanya hanya berdasarkan pada paradigma deteksi anomali. Untuk meningkatkan akurasi dari paradigma ini, diperlukan metode pendeteksian dengan membangun sebuah model *traffic* yang membandingkan model hasil prediksi dengan *traffic* aktual yang diamati.

DAFTAR PUSTAKA

[1] Filipe Moreira, Joel Luz. The Role and Responsibilities of the System's Administrator. https://www.igi-global.com, 2014.
[2] Shikha Agrawal, Jitendra Agrawal. Survey on Anomaly Detection using Data Mining Techniques. in Procedia Engineering (2015) 708-713. www.elsevier.com/locate/procedia, 2015.
[3] Liberouter Project. FlowMon Probe. https://www.liberouter.org/flowmon-networks/, 2017.
[4] Peter Haag. NFDUMP - NetFlow processing tools. https://github.com/phaag/nfdump/, 2017.
[5] Peter Haag. NfSen - NetFlow Sensor. http://nfsen.sourceforge.net/, 2017.
[6] Martin Žádník. Network Monitoring Based on IP Data Flows. Best Practice Documents, http://www.terena.org/activities/campus-bp/pdf/gn3-na3-t4-cbpd131.pdf., 2010.
[7] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosis Network-Wide Traffic Anomalies. In *ACM SIGCOMM'04*, pages 219 - 230, New York, NY, USA, 2004. ACM Press.
[8] Liming Zheng, Peng Zou, Yan Jia, and Weihong Han. Traffic Anomaly Detection and Containment Using Filter-Ary-Sketch. in Procedia Engineering 29 (2012) 4297–4306. www.elsevier.com/locate/procedia, 2012.
[9] Phillip Porras, Hassen Saidi, and Vinod Yegneswaran. A Foray into Conficker's Logic and Rendezvous Points, http://www.csl.sri.com/users/vinod/papers/cfkr_leet.pdf, 2016.
[10] F-Secure. Preemptive Blocklist and More Downadup Numbers, http://www.f-secure.com/weblog/archives/00001582.html, 2017.
[11] AlienVault OSSIM, https://www.alienvault.com/, 2017

BIODATA PENULIS

Slamet, S.T., M.T., Lulus S1 di Jurusan Teknik Elektro Universitas Muhammadiyah Surabaya pada tahun 2000. Lulus S2 di Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2013. Saat ini adalah dosen tetap Program Studi Sistem Informasi, Institut Bisnis dan Informatika Stikom Surabaya. Penulis mengampu mata kuliah – mata kuliah yang terkait dengan Keamanan Sistem Informasi dan Jaringan Komputer. Mendapatkan Sertifikasi Internasional dari Cisco, CCNA (Cisco Certified Network Associate). Selama kurang lebih 10 tahun menjabat sebagai Kepala Seksi Pengembangan Infrastruktur Jaringan di Stikom Surabaya. Tugasnya adalah mendesain,

mengimplementasikan dan merawat sistem jaringan dan sistem komputer yang digunakan Institusi. Penulis dapat dihubungi melalui email slamet@stikom.edu atau HP/WA 08123287190.

Model Monitoring Keamanan

ORIGINALITY REPORT

4%	3%	0%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur Student Paper	4%

Exclude quotes	On	Exclude matches	< 3%
Exclude bibliography	On		