SERANGAN DDOS PADA SOFTWARE DEFINED NETWORK

Tugas Akhir Mata Kuliah Keamanan Informasi dan Jaringan

EL5241

Demby Pratama

ABSTRAK

Serangan DoS atau DDoS merupakan bentuk serangan yang dilakukan dengan mengirim paket secara terus menerus kepada mesin bahkan jaringan komputer. Serangan ini akan mengakibatkan sumber daya mesin ataupun jaringan tidak bisa diakses atau digunakan oleh pengguna. Serangan DDoS biasanya berasal dari beberapa mesin yang dioperasikan oleh pengguna ataupun oleh bot, sedangkan serangan Dos dilakukan oleh satu orang atau satu sistem. Dalam makalah ini, istilah yang akan digunakan adalah istilah DDoS untuk mewakili serangan DoS ataupun DDoS. Pada dunia jaringan, *Software Defined Network* (SDN) merupakan paradigma yang cukup menjanjikan. SDN memisahkan *control plane* dengan *forwarding plane* untuk meningkatkan *network programmibility* dan manajemen jaringan. Sebagai bagian dari jaringan, maka SDN tidak luput dari serangan DDoS. Maka makalah ini secara umum akan membahas bentuk serangan DDoS yang ditujukan khusus pada SDN.

Keyword: DoS, DDoS, Software Defined Network, Network

DAFTAR ISI

1. PENDAHULUAN

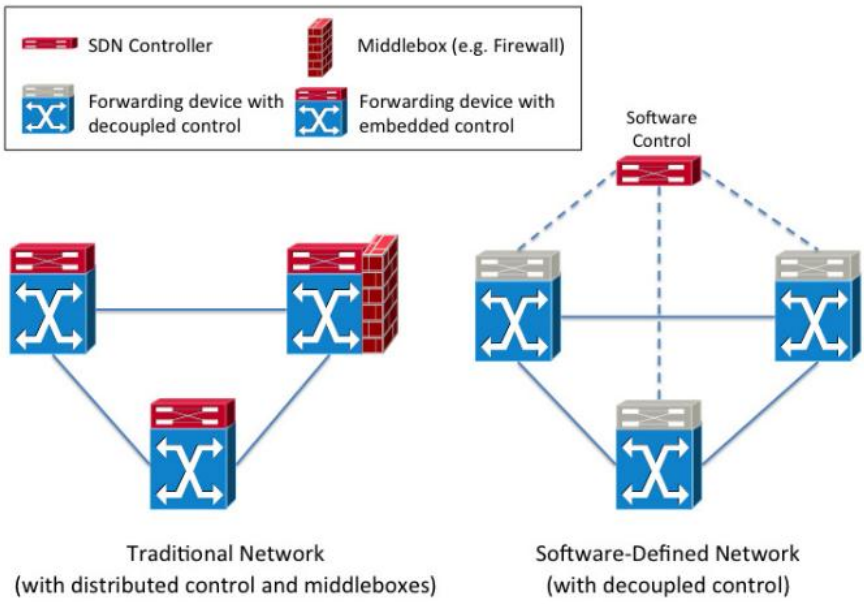
Sejak kemunculannya beberapa tahun lalu, Software Defined Network (SDN) menjadi salah satu isu yang menarik dalam dunia jaringan baik akademisi maupun praktisi. SDN dimunculkan untuk menggantikan jaringan yang sudah ada saat ini. Jaringan saat ini dianggap kaku dan sulit untuk dikembangkan. Pada jaringan saat ini, perangkat-perangkat seperti *switch*, router dan perangkat jaringa lainnya, bagian kontrol dan data tergabung secara fisik sehingga tidak fleksibel. Dengan adanya SDN, maka kedua bagian dapat dipisahkan, sehingga secara fisik perangkat yang ada di jaringan adalah bagian data atau *data plane*.

Karakteristik fundamental dari SDN adalah pemisahan antara *control plane* dari *forwarding plane.* SDN secara fungsional terbagi menjadi tiga lapisan yaitu lapisan infrastruktur, lapisan *control* dan lapisan aplikasi seperti yang ditampilkan pada gambar 3. Maka ketiga lapisan tersebut memiliki potensi untuk diserang dengan serangan DDoS. Karena kemungkinan diatas, maka serangan DDoS dibagi menjadi tiga kategori yaitu serangan DDoS pada lapisan aplikasi, serangan DDoS pada lapisan *control* dan serangan DDoS pada lapisan infrastruktur.

Pada makalah ini akan dibahas serangan DDoS pada ketiga lapisan diatas dan beberapa solusi yang pernah diusulkan oleh peneliti untuk mengatasinya. Makalah ini dibagi atas enam bagian yaitu, Pendahuluan, Overview SDN, Tinjauan Keamanan Pada SDN, Serangan DDoS, Serangan DDoS pada SDN dan Simpulan.

2. OVERVIEW SDN

Software Defined Network (SDN) menjadi salah satu agenda pada dunia jaringan yang paling menarik sejak kemunculannya beberapa tahun yang lalu. Karakteristik fundamental dari SDN adalah pemisahan antara *control plane* dari *forwarding plane* berbeda dengan jaringan sebelumnya yang menggabungkan keduanya di semua perangkat, seperti terlihat pada gambar 1. Pada SDN, fungsi dari *control plane* secara logika adalah menjaga keadaan di jaringan dan memberikan instruksi ke *data plane* [1].

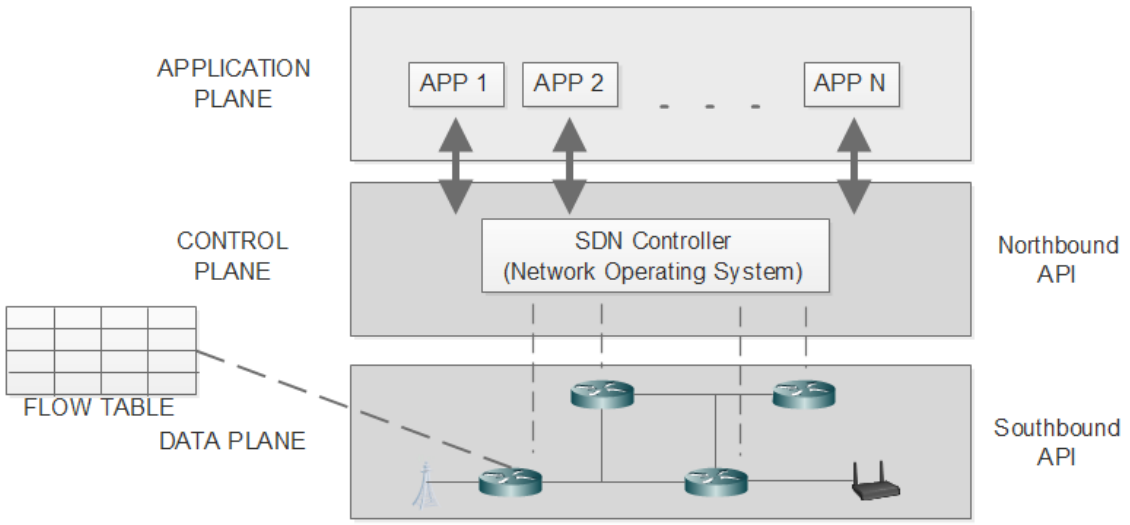


Gambar 1. Jaringan Tradisional dan *Software-Defined Network*[2]

Arsitektur SDN yang terlihat pada gambar 1 memisahkan logika kontrol dari perangkat keras penerusan, dan memungkinkan konsolidasi middlebox, manajemen kebijakan yang lebih sederhana, dan fungsionalitas baru. Garis-garis yang solid menentukan hubungan data-*plane* dan garis putus-putus link-*plane* kontrol. Seiring dengan lahirnya teknologi baru yang diperuntukkan untuk pusat data dan jaringan, yaitu teknologi cloud computing dan virtualisasi, diikuti semakin meningkatkan ketertarikan dunia akademis dan praktisi pada perkembangan SDN.

2.1 Arsitektur SDN

Framework Software-Defined Network memfasilitasi kemampuan program jaringan dan memberikan kemampuan untuk mengelola, mengubah dan mengendalikan perilaku jaringan secara dinamis melalui *open interface*. SDN memungkinkan fitur tambahan seperti alokasi sumber daya sesuai permintaan, penyediaan layanan mandiri, dan jaringan yang benar-benar tervirtualisasi melalui perangkat cerdas dan *provisioning system*. Arsitektur SDN yang dilahirkan oleh Open Networking Foundation (ONF) ditunjukkan pada Gambar 2. Arsitektur tersebut terdiri dari tiga lapisan utama yaitu, Data Plane, Control Plane dan Application Plane. Pada arsitektur tersebut, setiap lapisan memiliki fungsi dan komponen spesifiknya masing-masing pada setiap SDN *deployment* termasuk Southbound API, SDN Controller (atau *Network Operating System/NOS*), Northbound API dan aplikasi jaringan [3].



Gambar 2. Arsitektur SDN

2.1.1 Data Plane

Data *plane* terdiri dari perangkat jaringan seperti router dan *switch* yang memiliki kemampuan untuk meneruskan paket. Perangkat ini hanya melakukan *forwarding* secara sederhana tanpa kemampuan lain seperti melakukan *autonomous decision*. Perangkat-perangkat ini berkomunikasi dengan *controller* melalui standar *interface* OpenFlow. Interface ini memastikan konfigurasi dan kompatibilitas komunikasi serta interoprability antar perangkat.

2.1.2 Southbound API

Southbound API merupakan bagian yang menyebabkan *controller* mampu mengendalikan perilaku jaringan dengan menjaga alur masukan semua perangkat yang terhubung ke *switch*. Oleh karena itu Southbound API menjadi salah satu komponen kritikal pada sistem SDN. Kemampuan ini menjembatani antara perangkat *forwarding* dengan *control plane*. Untuk itu Southbound API menyediakan *interface* yang umum untuk lapisan atas. Sehingga *controller* dapat terhubung menggunakan southbound API yang berbeda-beda misalnya OpenFLow, OpFLex dan OpenState. Southbound API juga memiliki kemampuan untuk menerima plugin bermascam protokol yang berfungsi untuk menngatur perangkat fisik atau virtual yang baru seperti BGP, SNMP dan NetConf.

2.1.3 Northbound API

Northbound API merupakan software ekosistem yang menyediakan *interface* umum untuk membuat aplikasi. Oleh karena itu, bersama dengan southbound API, northbound API menjadi kunci dari abstrasksi SDN. Interface yang disediakan menjadi penghubung atau penerjemah antara instruksi low-level yang digunakan oleh *interface* southbound ke program pada perangkat *forwarding*, termasuk didalamnya otomasi global, manajemen data aplikasi juga routing dan keamanan.

2.1.4 SDN Controller

Komponen ini menjadi otak dari jaringan dengan men-generate konfigurasi jaringan berdasarkan aturan atau *policy* yang didefinisikan oleh operator jaringan. Komponen ini menerjemahkan aturan pada *lower level* sehingga tersedia untuk application *plane* melalui service utama dan API untuk developer.

2.1.5 Application Plane

Komponen ini bertanggung jawab untuk memaksakan aturan atau *policy* yang dimasukkan ke *control plane*. Pelaksanaanya dengan mengimplementasikannya pada perangkat *forwarding* jaringan. Aplikasi SDN yang terpasang pada *controller* terdiri atas SDN App Logic dan A-CPI Driver.

2.2 Perkembangan Terkini Teknologi SDN

OpenFlow secara standard terbatas dan terlalu kaku sehingga ada beberapa peneliti mengusulkan tambahan untuk menambah flexibility. Secara umum ada tiga kategori dari usulan tersebut.

A) Menambah *multiple flow tables* pada perangkat *forwarding*.
B) Meningkatkan fleksibilitas *match rule*
C) *Stateful data planes*.

3. TINJAUAN KEAMANAN PADA SDN

Arsitektur SDN yang memisahkan definisi dan penyimpanan *policy* jaringan dari pelaksanaan dan implementasinya maka peneliti [4], mengkategorikan serangan terhadapat kelima komponen utama sesuai dengan dampaknya pada *policy*, *enforcement* dan *implementation*.

3.1 Implementation Attacks

Tiga jenis serangan yang ditujukan pada *data plane* diantaranya adalah *Device Attack, Protocol Attack* dan *Side Channel Attack. Device Attack* merupakan serangan yang mengeksploit vulnerabilites pada software maupun hardware *switch* yang memiliki kemampuan SDN untuk menyusupi *data plane*. Penyerang menargetkan bugs pada software ataupun hardware pada perangkat *forwarding*. *Protocol Attack* merupakan serangan yang mengexploit kelemahan pada protokol jaringan pada perangkat *forwarding*. Contoh serangan ini adalah BGP *attack*. *Side Channel Attack* dilakukan dengan menganalisa performa dari perangkat *forwarding*.

Apabila pada *Data Plane* ada tiga jenis serangan, maka pada Southbound API ada empat jenis serangan yang ditargetkan terhadapnya. Serangan tersebut adalah *Interaction, Eavesdrop, Avail- ability* dan *TCP attacks*. *Eavesdrop Attack* merupakan serangan yang bertujuan mempelajari informasi yang terjadi antara *control plane* dengan *data plane* untuk menargetkan serangan yang lebih besar. *Interception Attack* bertujuan untuk merusak kondisi jaringan dengan mengubah informasi atau pesan yang dkirim diantara *control plane* dengan *data plane*. *Availability Attack* sama seperti serangan *Denial of Service (DoS)*. Pada *Availibility attack, Southbound API* di banjiri dengan paket permintaan yang menyebabkan kegagalan implementasi *policy* atau aturan.

3.2 Enforcement Attacks

Enforcement attack merupakan serangan yang bertujuan untuk menghalangi SDN dalam menerapkan instruksi secara benar. Serangan ini dapat dengan mengubah waktu, kapan dan bagaimana *policy* seharusnya dijalankan pada jaringan. Target serangan ini adalah pada *Control Plane, Southbound API dan Northbound API*.

3.3Policy Attacks

Policy Attack merupakan serangan yang biasanya ditujukan karena kemampuan SDN untuk mendefinisikan dan menyimpan *policy* jaringan secara benar. Penyerang biasanya menargetkan level *policy* untuk mengganggu atau menyusupi *control plane* dan application *plane* SDN. Dengan menyusupi *controller*, maka penyerangan dapat mengubah informasi yang dishare dengan application *plane* terutama tentang jaringan dan keputusan yang akan dibuat. Biasanya serangan ini merupakan bagian dari serangan yang lebih besar untuk menyusupi atau mengganggu jaringan. Serangan ini juga dilakukan untuk menghindari deteksi yang dilakukan oleh instruction

detection system (IDS), sehingga penyerang dapat memiliki akses jaringan secara menyeluruh.

4. SERANGAN DDOS

Serangan DDoS merupakan serangan yang mudah dilakukan namun sulit untuk ditanggulangi. Serangan DDoS biasanya ditujukan pada organisasi atau perusahaan yang terhubung ke Internet. Dalam laporan yang dibuat pada tahun 2017, Akamai menyatakan bahwa 72 persen perusahaan yang menyatakan bahwa perusahaan mereka kurang efektif dalam mencegah serangan DOS[5]. Hal tersebut disebabkan salah satunya karena kurangan sumber daya manusia yang memiliki kualifikasi dalam mencegah serangan tersebut. Dalam beberapa tahun terakhir, paling tidak terjadi sedikitnya lima kali serangan DDoS yang mengakibatkan terjadinya *downtime* pada jaringan mereka rata-rata 8,2 jam. Sedangkan waktu untuk memitigasi serangan tersebut dibutuhkan waktu lebih kurang satu jam.

Serangan DDoS dilakukan terhadap target setidaknya dalam dua bentuk serangan, yaitu:

- Penyerangan menghabiskan semua *bandwidth* atau *resource* dari sistem yang dimiliki oleh target
- Penyerangan menemukan bug atau kelemahan pada implementasi software yang dapat mengganggu layanan.

Sebelum melakukan DdoS, penyerang biasanya akan menyiapkan mesin zombie. Mesin zombie adalah host atau mesin yang berada dalam suatu jaringan yang digunakan sebagai agen untuk melakukan DDoS. Mesin zombie ini didapatkan dari hasil scanning terhadap suatu jaringan, apabila mesin tersebut memiliki *vulnerability*, maka si penyerang akan memasang software didalamnya tanpa diketahui pemilik, sehingga mesin tersebut dapat dikuasai oleh penyerang.

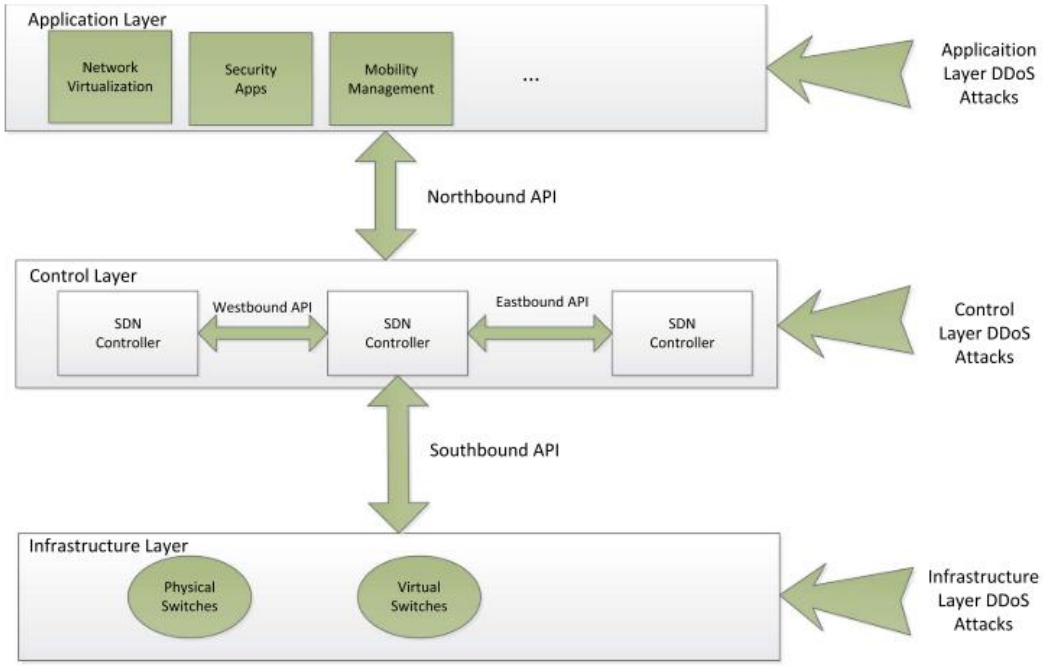
Saat melakukan serangan DDoS, mesin zombie akan dipasang ip spoofing sehingga serangan sulit untuk dilacak. Semakin banyak mesin zombie maka serangan akan semakin berbahaya dan semakin sulit dilacak. Target utama dari serangan DDoS adalah sumber daya seperti *bandwidth*, CPU dan sebagainya. Biasanya sumber daya ini terbatas di dalam jaringan. Meskipun sumber daya tersebut ditingkatkan untuk mengurangi dampak serangan namun tetapi saja akan ada dampak kerugian pada finansial.

Berdasarkan level protokol target, serangan DDoS diklasifikasikan menjadi dua kategori, sebagai berikut [6]:

1) Serangan DDoS pada level network/transport
   Serangan ini biasanya dilakukan menggunakan paket protokol TCP, UDP, ICMP dan DNS. Tujuan utama dari serangan ini adalah mengganggu konektivitas pengguna dengan menghabiskan *bandwidth* target.
2) Serangan DDoS pada level *application*
   Serangan ini bertujuan utama untuk mengganggu layanan pengguna dengan menghabiskan sumber daya server, seperti CPU, memory, *bandwidth disk*, *bandwidth database* dan *bandwidth* I/O.

5. SERANGAN DDOS PADA SDN

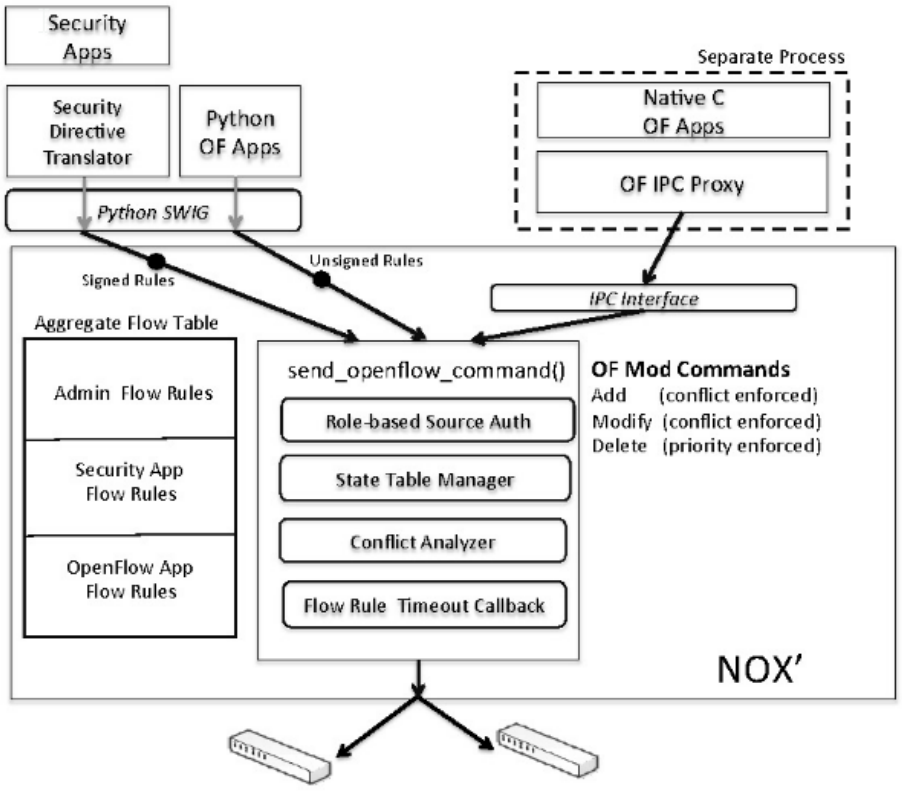
Seperti pada jaringan pada umumnya, SDN juga menjadi target serangan DDoS. SDN secara fungsional terbagi menjadi tiga lapisan yaitu lapisan infrastruktur, lapiran *control* dan lapisan aplikasi seperti yang ditampilkan pada gambar 3. Maka ketiga lapisan tersebut memiliki potensi untuk diserang dengan serangan DDoS. Karena kemungkinan diatas, maka serangan DDoS dibagi menjadi tiga kategori yaitu serangan DDoS pada lapisan aplikasi, serangan DDoS pada lapisan *control* dan serangan DDoS pada lapisan infrastruktur, seperti ditampilkan pada gambar 3.



Gambar 3. Serangan DDoS Pada SDN [7]

1. Serangan DDoS pada lapisan aplikasi.
   Metode yang digunakan untuk menjalankan serangan DDoS pada lapisan aplikasi ada dua jenis. Yang pertama adalah menyerang langsung aplikasi dan yang kedua adalah dengan menyerang northbound API. Penyerangan terhadap satu aplikasi pada SDN akan berdampak terhadap aplikasi-aplikasi lainnnya. Karena isolasi antar aplikasi atau resources pada SDN belum dapat diatasi dengan baik.

   Solusi yang dapat digunakan untuk mengatasi masalah ini adalah dengan menggunakan metode FortNOX. FortNOX [8] adalah *policy enforcement* keamanan baru yang dapat dipasang sebagai ekstension atau modul pada NOX OpenFlow *controller*. Metodenya adalah dengan memediasi semua *rule* OpenFlow yang berisi permintaan untuk melakukan penyisipan. FortNOX mengimplementasikan otentikasi berbasis *role* untuk menentukan otorisasi keamanan pada setiap aplikasi yang ada pada OpenFlow. FortNOX juga memberlakukan prinsip *least privilege* untuk memastikan integritas pada saat proses mediasi sehinngga meskipun penyerang mencoba menyisipkan aturan secara strategis, aturan atau *rule* tetap akan melewati proses analisis dan mediasi.



Gambar Implementasi FortNOX [8]

2. Serangan DDoS pada lapisan *control*

   Controller merupakan target yang menarik bagi serangan DDoS pada arsitektur SDN karena memiliki fungsi yang vital di dalam jaringan. Serangan terhadap *control plane* dapat dilakukan dengan menyerang langsung *controller* atau melalui northboundAPI, southboundAPI, westboundAPI serta eastbound API. Sebagai contoh, *flow rules* yang bertentangan pada aplikasi yang berbeda dapat menyebabkan terjadinya serangan DDoS pada lapisan kontrol. *Data plane* pada saat SDN beroperasi, akan bertanya kepada *control plane* ketika adanya paket baru yang tidak dapat diatasi. Apabila ada *flow* baru ketika *flow* tersebut tidak sesuai dengan *flow* tabel, maka untuk mengatasinya ada dua pilihan, yang pertama adalah paket dianggap lengkap atau sebagian header paket dikirimkan ke *controller* untuk menyelesaikan query-nya. Dengan besarnya volume trafik jaringan, pengiriman paket yang lengkap ke *controller* akan memakan *bandwidth* yang besar.

   Solusi yang diusulkan untuk mengatasi ini adalah dengan melakukan deteksi DDoS secara ringan dan cepat berbasiskan *entropy* [9]. Mekanisme ini dapat melindungi *controller* dengan memperhitungkan kemampuan *controller*. Dengan mendeteksi paket diawal antara 250 sampai dengan 500 paket, maka penambahan kode pada *controller* tidak akan meningkatkan beban CPU baik pada saat jaringan normal maupun ketika terjadi serangan. Usulan tersebut diimplementasikan oleh peneliti menggunakan Mininet dan POX Controller.

3. Serangan DDoS pada lapisan infastruktur

   Serangan pada lapirsan infrastruktur dapat dilakukan dengan dua cara, pertama dengan melakukan serangan langsung terhadap *switch-switch* atau menyerang southbound API. Sebagai contoh, jika hanya header informasi yang dikirimkan ke *controller* maka paket tersebut harus disimpan pada node memory sampai *flow table entry* dikembalikan. Hal tersebut menjadi ide bagi penyerang dapat mengirimkan sejumlah *flow* baru dan tidak dikenali sebagai serangan DDoS. Serangan ini akan mnenyebabkan elemen memori pada node mengalami *bottleneck* akibat beban yang tinggi. Akibatnya penyerang mampu membuat memori *switch* menjadi kelebihan beban.

   Solusi yang dapat digunakan untuk mengatasi serangan ini adalah AVANT-GUARD. AVANT-GUARD[10] bertujuan untuk meningkatkan keamanan aplikasi SDN sehingga lebih responsif dan skalabel terhadap ancaman jaringan yang dinamis. AVANT-GUARD terdiri dari dua metode. Metode pertama adalah dengan dengan

menginspeksi sesi TCP pada bagian *forwarding* sebelum melakukan notifikasi pada kontroler sehingga memungkinkan peningkatan ketahanan jaringan SDN. Bagian kedua bertujuan untuk meningkatkan respon sehingga keamanan aplikasi dapat mengakses statistik jaringan secara efisien dalam menanggapi ancaman. Metodenya adalah dengan membuat *actuating trigger* yang mememungkinkan kontroler untuk mendeteksi dan merespon ancaman. Bagian ini direalisasikan melalui pengumpulan statistik jaringan secara efisien yang secara otomatis akan mengatur *flow rules* sesuai statistik jaringan.

Solusi lainnya adalah menggunakan FLOODGUARD. FLOODGUARD [11] menggunakan dua teknik / modul baru, yaitu proactive *flow rule* analyzer dan packet migration. Untuk mempertahankan *policy* enforcement jaringan, proactive *flow rule* analyzer secara dinamis mengambil *rule* aliran proaktif dengan logika runtime *controller* SDN / OpenFlow dan aplikasinya. Teknik yang kedua, yaitu packet migration, digunakan untuk melindungi pengontrol dari kelebihan beban, dengan membuat cache paket flooding secara temporer dan mengirimkannya ke *controller* OpenFlow menggunakan rate limit dan penjadwalan round-robin.

6. SIMPULAN

SDN menjadi topik yang menarik dibidang jaringan, sehingga terus dikembangkan baik oleh praktisi maupun akademisi. Sifatnya yang memisahkan antara control plane dengan data plane menjadi salah satu kelemahan bagi SDN. Salah satu variasi serangan yang dilakukan adalah serangan DDoS. Serangan DDoS dilakukan terhadap tiga lapisan fungsional yang dimiliki oleh SDN. Serangan-serangan tersebut bervariasi pada tiap lapisan, dan dapat dilakukan terhadap lima komponen yang terdapat pada SDN. Serangan tersebut dapat dilakukan secara langsung pada perangkat maupun melalui API yang dimiliki oleh SDN.

Untuk mengatasi serangan DDos tersebut, peneliti mengusulkan beberapa ide. Ide tersebut dapat diimplementasikan pada perangkat secara langsung terutama pada *controller*. Solusi tersebut bisa dilakukan pada saat deteksi awal ketika serangan akan terjadi maupun pada saat serangan telah terjadi.

Daftar Pustaka

[1] S. . A. Scott-Hayward, S.a , Natarajan, S.b , Sezer, "A Survey of Security in Software Defined Networks," *A Surv. Secur. Softw. Defin. Networks*, vol. 18, no. 1, 2016.

[2] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking : Past , Present , and Future of Programmable Networks," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[3] A. Shaghaghi and S. Jha, "Software-Defined Network (SDN) Data Plane Security: Issues, Solutions and Future Directions," 2018.

[4] S. Gao, Z. Li, B. Xiao, and G. Wei, "Security Threats in the Data Plane of Software-Defined Networks," *IEEE Netw.*, vol. 32, no. 4, pp. 108–113, 2018.

[5] A. Technologies, "Cost of Web Application & Denial of Service Attacks," no. October, 2018.

[6] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *IEEE Commun. Surv. TUTORIALS*, pp. 1–24, 2013.

[7] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDOS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys and Tutorials*. 2016.

[8] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A Security Enforcement Kernel for OpenFlow Networks," *Proc. first Work. Hot Top. Softw. Defin. networks*, pp. 121–126, 2012.

[9] S. M. Mousavi and P. Affairs, "Early Detection of DDoS Attacks in Software Defined Networks Controller Early Detection of DDoS Attacks in Software Defined Networks Controller," Carleton Univ, 2014.

[10] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD : Scalable and Vigilant Switch Flow Management in Software-Defined Networks Categories and Subject Descriptors," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS '13*, 2013, pp. 413–424.

[11] H. Wang, "FloodGuard : A DoS Attack Prevention Extension in Software-Defined Networks," 2015.