Keamanan Jaringan *Software Defined Networking* Pada Sisi Aplikasi dan Pengolahan Paket

Oleh :

232 14 348

Wahid Miftahul Ashari

Sekolah Teknik Eleketro Informatika

Institut Teknologi Bandung

2015

Daftar Isi

Daftar Gambar

Abstrak

Banyak teknologi baru yang muncul dalam dunia jaringan, hal ini disebabkan adanya teknologi yang mampu memisahkan data plane dan control plane pada sistem jaringan komputer, teknologi tersebut bernama SDN （software Defined Networking）, salah satunya yaitu menerapkan Northbound API untuk memungkinkan aplikasi pihak ketiga untuk mengakses sumber daya jaringan. Namun, hampir semua implementasi Northbound API tidak mempertimbangkan aspek keamanan. Serta pengiriman paket pada jaringan SDN juga belum terjamin keamananya. Oleh karena itu, kami merancang skema yang aman untuk implementasi jaringan SDN khususnya mekanisme keamanan pada Northbound API dan mekanisme kemanan pada proses pengiriman paket yang berjalan pada jaringan SDN ini. Desain ini terdiri dari otentikasi ganda untuk aplikasi dan user, yang bertanggungjawab untuk mengontrol dan menggunakan aplikasi dalam jaringan dengan menggunakan Oauth 2,0 protokol. Serta untuk mekanisme keamanan pada pengiriman paket, mekanisme keamanan paket pada jaringan yang berjalan akan mengguanakan enkripsi paket yang dipilih dan dikirim melalui jaringan SDN menggunakan algoritma HASH.

Kata kunci : SDN, Northbound API, Oauth 2.0, Algoritma HASH.

1. Latar Belakang

SDN merupakan teknologi baru yang dapat dimanfaatkan oleh aplikasi pihak ketiga melalui NorthBound API (NBI). Saat ini sudah ada beberapa *controller* SDN yang sudah dikembangkan dan dipakai dalam dunia jaringan, yaitu : REST API, Java API, dan sebagainya [1].

Karena banyaknya macam NBI yang ada munculah beberapa kesulitan dalam dunia jaringan, yaitu keanekaragaman yang menyebabkan kendala apabila ada dua jaringan yang menggunakan NBI yang berbeda dan ingin saling terhubung. Oleh karena itu ONF (Open Network Federeation) membentuk sebuah kelompok kerja untuk menstandarkan NBI yang ada, sehingga semua NBI dapat berinteraksi dan dapat diterapkan pada semua platform jaringan yang ada. Salah satu standar yang dipilih untuk NBI yaitu REST API, REST API dipilih karena memiliki keunggulan daripada NBI yang lain yaitu mampu menyediakan integrasi yang sangat sederhana dan memungkinkan interaksi dibagian overhead yaitu interaksi antara client dan server. Karena alasan tersebut beberapa perusahaan besar seperti facebook, google, dan yahoo menggunakan REST API untuk sistem jaringan komputer mereka. Alasan yang sama juga digunakan oleh para pengembang SDN, sebagian besar kontroler SDN saat ini menggunakan REST API untuk kontrolernya, yang sering disebut dengan SISA Northbound API (SISA NBI), SISA NBI ini juga digunakan untuk memberikan informasi jaringan kepada pihak ketiga[2]. Sayangnya, implementasi REST NBI di sebgaian besar kontroler SDN belum dilengkapi dengan keamanan, seperti dalam tabel 1 sebagian besar dari contoh REST NBI tidak memiliki fitur keamanan.

Hal ini menimbulkan masalah kemananan pada SISA NBI, satu contoh kelemahan NBI jika tidak memiliki kemanan yaitu setiap client yang ada bebas mengakses dan memanfaatkan SISA NBI untuk kepentinganya sendiri selama mereka tahu URL endpointnya. Gregory Picket menunjukan kelemahan ini pada DEFCON[3], dimana seorang user mampu mendapatkan informasi dan memiliki akses kontrol terhadap jaringan yang dia gunakan untuk kepentinganya sendiri, dengan demikian desain kemananan REST NBI harus benar-benar dipertimbangkan. Dalam paper ini REST NBI yang akan dibahas yaitu Floodlight dan ODL, dua NBI tersebut digunakan oleh sebagian besar pengembang jaringan SDN kareana dua NBI tersebut lebih fleksibel dalam penerapan fitur-fitur tambahan termasuk fitur kemananan. Dalam paper ini penulis berencana menerangkan beberapa contoh mekanisme keamanan REST NBI khususnya membahas otentikasi dan otorisasi dalam jaringan SDN (SISA NBI). Dalam konfigurasi default, ODL memberikan otentikasi dasar untuk mengamankan REST NBI, yaitu dengan memberikan permintaan username dan password kepada pihak pengguna. Cara ini dianggap masih kurang baik dan kurang efektif untuk mengamankan REST NBI.

Oleh karena itu, proyek AAA [9] dan HP [10] datang dengan ide otentikasi berbasis token. Paper ini juga membahas tentang penerapan beberapa fitur keamanan aplikasi pada SDN, paper ini nantinya akan membandingkan beberapa model keamanan aplikasi pada SDN untuk memperoleh sebuah mekanisme kemanan terbaik pada aplikasi (REST NBI) yang berjalan pada SDN. Selain token pada AAA dan HP pada paper ini akan dibahas sebuah mekanisme yang berbasis token yang dikembangkan oleh penelitian, yang mengembangkan token berbasis ID

user atau sering disebut ID-Based Cryptogtaphy (IBC) [5], mekanisme ini diharapkan dapat menggantikan peran dari TLS yang diperkirakan dapat diretas dalam beberapa tahun kemudian. Disamping token paper ini juga akan menerangkan tentang keamanan tambahan pada SISA NBI dengan menggunakan JSON Web. Dan dari beberapa model keamanan tersebut akan dipilih salah satu atau perpaduan dari beberapa mekanisme kemanan untuk membangun sebuah infrastruktur SISA NBI yang lebih aman. Software Defined Network (SDN) adalah istilah yang merujuk pada konsep/paradigma baru dalam mendesain, mengelola dan mengimplementasikan jaringan[10], terutama untuk mendukung kebutuhan dan inovasi di bidang ini yg semakin lama semakin kompleks.

2. Survey Literatur

2.1. SDN (Software Defined Networking)

Konsep dasar SDN adalah dengan melakukan pemisahan eksplisit antara control dan forwarding plane, serta kemudian melakukan abstraksi sistem dan meng-isolasi kompleksitas yg ada pada komponen atau sub-sistem dengan mendefinisikan antar-muka (interface) yg standard. Beberapa aspek penting dari SDN adalah : 1. Adanya pemisahan secara fisik/eksplisit antara forwarding/data-plane dan control-plane 2. Antarmuka standard (vendor-agnosic) untuk memprogram perangkat jaringan 3. Control-plane yang terpusat (secara logika) atau adanya sistem operasi jaringan yang mampu membentuk peta logika (logical map) dari seluruh jaringan dan kemudian memrepresentasikannya melalui (sejenis) API (Application Programming Interface) 4. Virtualisasi dimana beberapa sistem operasi jaringan dapat mengkontrol bagian-bagian (slices atau substrates) dari perangkat yang sama.

Di dalam arsitektur SDN, control plane dipisahkan dari data plane, network intelligence dan network state disentralisasi secara logika, dan infrastruktur jaringan dipisahkan dari aplikasi. Hasilnya, perusahaan dan operator bisa memperoleh programabilitas, otomatisasi, dan kendali pada jaringan yang belum pernah ada sebelumnya sehingga memungkinkan mereka untuk membuat jaringan yang bisa diperluas dengan mudah dan fleksibel yang siap untuk beradaptasi sesuai dengan kebutuhan bisnis.

Tujuan dari SDN adalah untuk menyediakan antarmuka terbuka yang dapat mendukung pengembangan software yang dapat mengendalikan konektivitas suatu network resources dan aliran dari trafik jaringan yang melaluinya, bersamaan dengan inspeksi dan modifikasi trafik yang mungkin dapat diterapkan pada jaringan. Pada layer terbawah, data plane terdiri dari network element kapabilitasnya diekspos oleh SDN Datapath melalui Control-Data-Plane Interface (CDPI) aget. Pada layer teratas, application plane mencakup berbagai macam SDN application yang saling bertukar informasi tentang requirement masing-masing melalui NorthBound interface (NBI) Drivers. Pada layer di tengah, SDN Controller menerjemahkan requirement dari layer atasnya dan mengirimkan low-level control melalui SDN Datapaht sambil memberikan informasi yang relevan kepada SDN Applications. Management dan Admin Plane bertanggung jawab untuk menyiapkan network elements, menetukan SDN Datapaths pada SDN Controller, dan melakukan konfigurasi policy yang mendefinisikan ruang lingkup kendali yang diberikan kepada SDN Controller atau SDN Application. Arsitektur jaringan SDN ini dapat dijalankan berdampingan dengan jaringan non-SDN, khususnya untuk tujuan migrasi ke jaringan SDN penuh.

2.2. Controller.

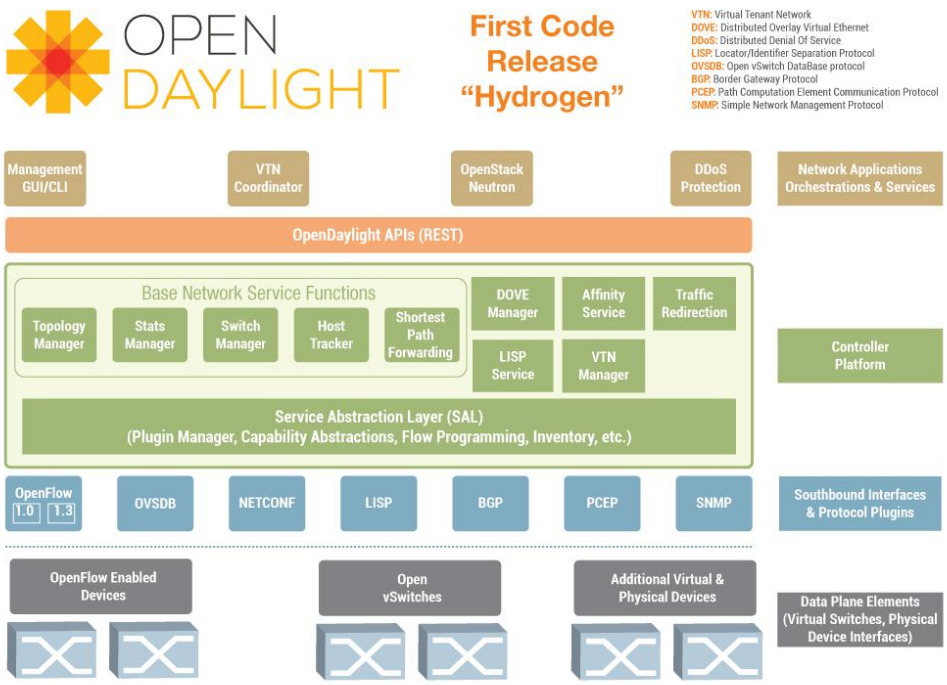
Controller SDN biasanya berisi kumpulan "pluggable" modul yang melakukan tugas jaringan yang berbeda. Controller memiliki beberapa tugas pokok, termasuk tugas pokok yaitu mengontrol dan menyimpan protocol pada semua device yang berada pada jaringan komputer yang ada. dalam jaringan SDN, controller dapat ditambahkan algoritma-algoritma untuk meningkatkan performa dari jaringan, seperti penambahan algoritma analisis dan algoritma protocol bagi seluruh jaringan.

2.3. ODL (Open Day Light)

8 April 2013, yayasan open-source, mengumumkan telah merilis sebuah controller baru untuk jaringan SDN yaitu OpenDaylight, yang dikembangkan oleh anak perusahaan dari linux foundation. Controller ini berbasis Java, dan berasala dari desain Beacon. Tidak hanya untuk Northound namun OpenDayLight juga merilis beberapa sistem yang berada pada Southbound seperti Cisco Oplex. Controller OpenDaylight dapat dioperasikan dan disimpan dalam JVM (Java Virtual Machine), controller ini juga dapat digunakan pada beberapa perangkat dari berbagai vendor jaringan. OpenDayLight merilis kode pertamanya yaitu Hidrogen, Hidrogen menawarkan tiga edisi yang berbeda untuk pengguna. Pada September 2014, OpenDaylight meluncurkan kode keduanya yaitu Helium. Kode kedua ini lebih terbuka, sehingga jaringan memiliki Programability yang tinggi dan dapat menyesuaikan kebutuhan terutama untuk perusahaan. Helium juga digunakan berbagai vendor besar seperti Cisco dan Brocade untuk mendevelop jaringan SDN mereka.

Sebagai tantangan baru untuk OpenDaylight Controller, ternyata perusahaan pesaing seperti AT&T, Microsoft, Hp, Ericsson, NTT, Ciena,

dan Extreme Networks juga mengembangkan controller untuk SDN. Gambar dibawah ini menjelaskan tentang arsitektur Helium OpenDaylight sebagai controller SDN.



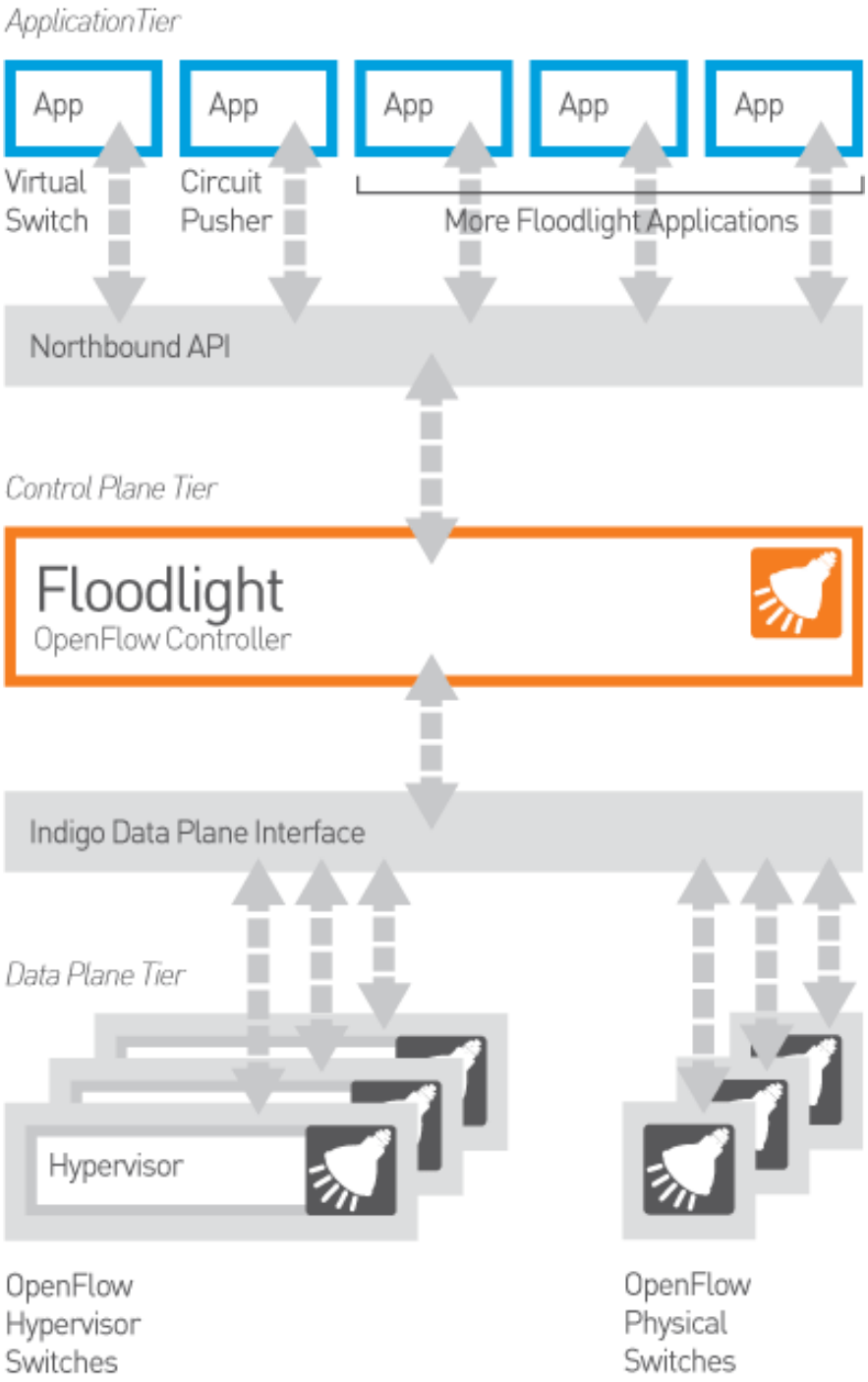
Gambar 1 : Arsitektur ODL (OpenDayLight)

2.4. Floodlight

Floodlight adalah sebuah controller OpenFlow yang enterprise, berlisensi Apache dan berbasis Java. Hal ini didukung oleh komunitas pengembang termasuk sejumlah teknisi yang ikut dalam pengembangan floodlight.

OpenFlow adalah standar terbuka yang dikelola oleh ONF, OpenFlow menentukan protokolnya melaui switch control yang dapat memodifikasi semua protokol dalam jaringan yang didefinisikan dengan baik "set instruksi fowarding". Floodlight dirancang untuk mengatasi kompleksitas jaringan yang semakin hari semakin bertambah rumit,

terutama jumlah switch dan router yang semakin banyak. Berikut ini adalah gambar dari arsitektur controller floodligth.



Gambar 2 : Arsitektur Floodlight

2.5. Algoritma Hash

Fungsi Hash merupakan sebuah algoritma yang mengubah text atau message menjadi sederetan karakter acak yang memiliki jumlah karakter yang sama. Hash juga termasuk salah satu bentuk teknik kriptografi dan dikategorikan sebagai kriptografi tanpa key (unkeyed cryptosystem). Selain itu hash memiliki nama lain yang juga dikenal luas yaitu "one-way function". Dalam dunia kemanan digital sering sekali menjumpai hash di website-website yang menyediakan layanan untuk download file ataupun program secara resmi. Hash memang umumnya digunakan untuk mengecek integritas dari sebuah pesan atau file. File atau pesan yang sudah berubah akan memiliki nilai hash yang berbeda. Sebagai contoh, dengan sebuah algoritma hash, pesan'hello' akan memberikan nilai hash 12345 sedangkan pesan 'hallo' memiliki nilai hash83746. Dengan kata lain output hash dari kata 'hello' tidak akan sama dengan 'hallo'. Bahkan sekalipun dalam kacamata kita kedua pesan tersebut terlihat hanya memiliki perbedaan sedikit saja, namun nilai hash yang dimiliki oleh kedua pesan tersebut sangat jauh berbeda.

Berbeda dengan teknik enkripsi dalam kriptografi, tujuan hash memang mengubah sebuah pesan yang dapat dibaca (readable text) menjadi pesan acak (unreadable text) sama seperti enkripsi, namun hal mendasar yang menjadi perbedaan dari hash adalah pesan yang telah acak tadi tidak dapat diubah kembali menjadi pesan yang seharusnya. Inilah mengapa hash disebut juga sebagai "one-way function".

3. Implementasi Mekanisme Keamanan SDN

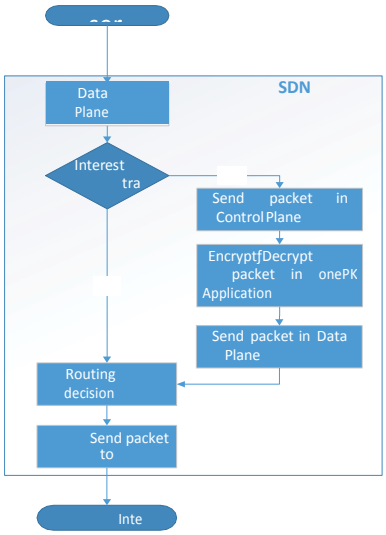
Solusi terbaik untuk kemanan jaringan adalah dengan menyediakan hardware kriptografi, tetapi solusi ini sangatlah mahal untuk implementasinya dan administrasinya tidak layak untuk kebanyakan kasus. Solusi lain adalah dengan menggunakan IPSec (Internet Protocol Security) pada VPN (Virtual Private Network). IPSec adalah protokol standar untuk menjaga kerahasiaan setiap lapisan internet dan otentikasi lalu lintas internet. Standar mendefinisikan dua format yaitu header otentikasi disebut (AH) yang berfungsi untuk menyediakan integritas data, untuk format satunya yaitu Encapsulating Secure Packet (ESP) untuk menyediakan kerahasiaan dan integritas data [6]. Standar ini memperkenalkan lapisan perantara antara layer 3 dan layer 4, sehingga sangat mudah dideteksi adanya serangan.

Penggunaan algoritma publik justru membuat IPSec rentan akan adanya serangan termasuk serangan BruteForce [6]. Solusi lain menggunakan SSL (Secure Socket Layer) atau dengan SSH (Secure Shell Hub). Namun, Solusi ini tidak dapat diimplementasikan pada layer 3 dan solusi ini masih terdapat delay penanganan yang cukup lama karena penanganan masih berada pada level aplikasi. Solusi lain adalah dengan adanya mekanisme keamanan berbasis SDN.

3.1. Arsitektur Keamanan Mengguankan Aplikasi ONE PK

Teknologi seperti IPSec, SSH, SSL, dan lain lain, dapat digunakan untuk melindungi komunikasi yang bergerak melauli jaringan tidak aman (internet), tetapi semua ini menggunakan algoritma kriptografi standar. Namun, pemerintah, militer, dan lembaga tinggi perlu menggunakan algoritma yang tidak standar, karena lembaga-lembaga tersebut

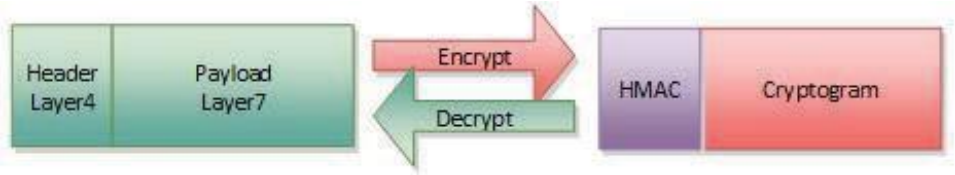
memerlukan algoritma yang benar-benar dapat menjaga informasi yang mereka punya. Namun jika menggunakan algoritma standar maka semua jaringan akan mendapatkan prioritas yang sama meskipun tingkat kemanan yang masih rendah, sehingga model kemanan seperti ini justru membuat kinerja jaringan menurun karena tingkat keamanan yang diterapakan secara kuantitas tinggi dan secara kualitas rendah. Maka harus diterapkan sebuah mekanisme algoritma yang hanya melindungi lapisan atau sisi jaringan yang memang membutuhkan prioritas keamanan. Salah satunya yaitu Aplikasi ONE PK, aplikasi ini memungkinkan untuk menyesuaikan enskripsi lalu-lintas dengan memodifikasi algoritma kriptografi yang ada dan memungkinkan untuk memilih salah satu traffic atau jenis tertentu dari lalu-lintas jaringan untuk dijamin keamananya. Gambar dibawah ini menunjukan bagaimana arsitektur ONE PK dalam proses mengamankan lalu lintas jaringan.



Gambar 3 : Arsitektur ONE PK

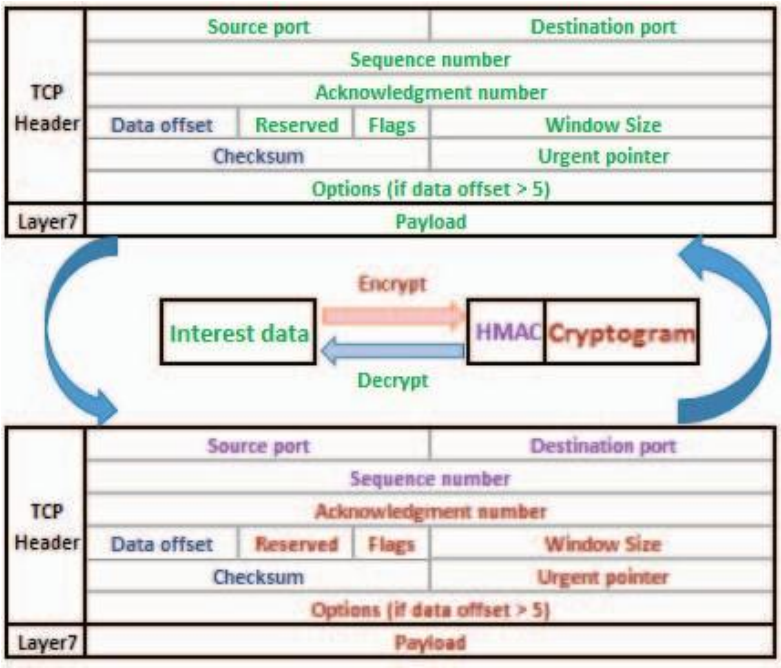
Pertama paket tiba , jika mengandung lalu-lintas yang mencurigakan maka paket akan diarahkan menuju aplikasi ONE PK, didalam ONE PK paket akan dienkripsi atau dideskripsi sesuai aturan dari ONE PK, setelah itu paket akan dikembalikan kepada data plane.

Aplikasi ini juga menawarkan kemanan untuk lalu-lintas pada bagian TCP dan UDP, kemanan dapat diatur sesuai keinginan admin jaringan namun tetap sesuai dengan protokol yang disediakan oleh aplikasi ONE PK, seperti cantoh admin dapat meilih mekanisme kemanan pada header layer 4 dan payload pada layer 7 atau hanya pada payload layer 7. Gambar 4 menunjukan bagaimana data unit pada layer 4 dan layer 7 diamankan.



Gambar 4 : Enkripsi dan Deskripsi Paket

Selanjutnya, untuk lebih manjamin kemanan jaringan ditambahkan sbuah protokol kemanan yaitu dengan tidak memasukan data kedalam paket yang digunakan untuk determinate jika paket berisi informasi yang dienkripsi atau tidak. Gambar 3 menunjukan pengamanan jaringan pada layer 4 dan layer 7 dalam jaringan TCP. Mekanisme ini menawarkan kemungkinan untuk memilih bidang layer 4 yang akan dijamin keamananya. Namun ada beberapa bidang yang tidak mungkin untuk diamankan karena bidang inin berisi tentang paket yang secara otomatis diubah seperti cheksum dan offset.



Gambar 5 : Proses Enkripsi pada jaringan TCP

3.2. Implementasi Mekanisme Kemanan ONE PK

Untuk impelemntasinya penulis menuliskan sebuah penelitian yang dirilis pada IEEE pada tahun 2013 [8]. Dalam penelitian ini SDN dirancang menggunakan 2 router, 2 controller dan 2 pengguna yang ditunjukan gambar 7.

Melalui controller SDN, admin jaringan akan mengkonfigurasi router 1 untuk mengenskripsi paket yang dialihkan dengan sember user 1 dan tujuan user 2, lalu lintas yang lain tidak berubah seperti konfigurasi default. Desain Topologi Untuk menguji SDN berbasis keamanan ONE PK, ONE PK berpusat pada SDK dan API yang digunakan untuk mengembangkan aplikasi di ONE PK Cisci IOS. Untuk menguji aplikasi yang digunakan kita harus memiliki ntopologi yang terdiri dari beberapa alat, dipenelitian ini peneliti menggunakan virtual machine emulator untuk mengelola perangkat-perangkat yang digunakan untuk penelitian.

Topologi A menggambarkan bagaimana Unsur-unsur tersebut diatur dan terhubung dalam jaringan komputer. Pada topologi ini semua platform menggunakan Cisco. Desain Keamanan Mekanisme keamanan yang akan dilaksanakan bertujuan untuk memberikan kerahasiaan dan integritas dengan cara mengenskripsi data yang telah disesuaikan antara dua pengguna yang menggunakan SDN. Cara ini dikembangkan dengan bahasa pemrograman C, karena C merupakan satu-satunya bahasa yang mendukung untuk Datapath Service Set (DPSS). DPSS memungkinkan aplikasi untuk berinteraksi dalam tiga mode (copy, sampel, dan pengalihan) dengan paket aliran pada router tersebut paket dapat dikelola dan dipilih yang akan dienkripsi dan dideskripsi sesuai protokol pemilihan paket yang telah dipasang. Sebelum mekanisme keamanan dijalankan, controller harus dikonfigurasi dengan benar, disini peneliti menggunakan layanan ONEP pada controllernya. Setelah semua dikonfigurasi, mekanisme keamanan dibangun dengan cara-cara seperti ini :

1. Menentuakan di mana interface paket yang dikirim atau diterima.
2. Menentukan lalu-lintas jaringan yang dipilih dan menentukan fungsi pengolahan paket.

Accsess Control Element (ACE) dibuat untuk memberikan informasi yang spesifik untuk kepentingan lalu-lintas. Maka tindakan pengalihan DPSS dibuat, dan fungsi pengolahan diinformasikan untuk memperkenalkan kembali paket ke aliran fungsi dengan cara memberikan informasi seperti berikut pada paket (onep_policy_action_set_stateful). Fungsi pengolahan memiliki dua peran yaitu :

a. Menampilkan informasi tentang paket mencurigakan yang diterima.

b. Mengenskripsi atau mendeskripsi paket yang mencurigakan.

Pada saat mekanisme berjalan controller mendisplay tentang keadaan paket seperti aplikasi wireshark. Algoritma kriptografi akan diterapkan pada layer 4 dan layer 7. Dan berikut susunan kinerja dari algoritma kriptografi yang diterapkan

a. Langkah-langkah enkripsi paket data.

1) Mengestrak informasi mencurigakan dari paket.
2) Membuat HMAC dengan algoritma Hash dan kunci enkripsi simetris untuk informasi tersebut.
3) Menambahkan padding yang diperlukan (standar PKCS7)
4) Mengenkripsi informasi dan padding menggunakan algoritma simetris dan kunci enkripsi simetris.
5) Bentuk kriptogram yagn harus mengirim dengan menggabungkan HMAC dengan kriptogram dari langkah sebelumnya.
6) Informasi mencurigakan diganti dengan kriptogram dari langkah sebelumnya.

b. Langkah-lengkah deskripsi paket data

1) Mengestrak informasi mencurigakan dari paket data yang dikirim.
2) Ekstrak HMAC dan Kriptogram.
3) Mendeskripsi Kriptogram menggunakan algoritma Hash dan kunci enkripsi serta menghapus padding yang telah ditambahkan sewaktu enkripsi.

4) Membuat HMAC dengan algoritma Hash dan kunci enkripsi simetris untuk informasi.

5) Verifikasi integritas informasi dengan membandingkan HMAC dihitung dengan satu paket, jika hasilnya benar, langkah berikutnya dilakukan, jika salah paket akan dibuang.

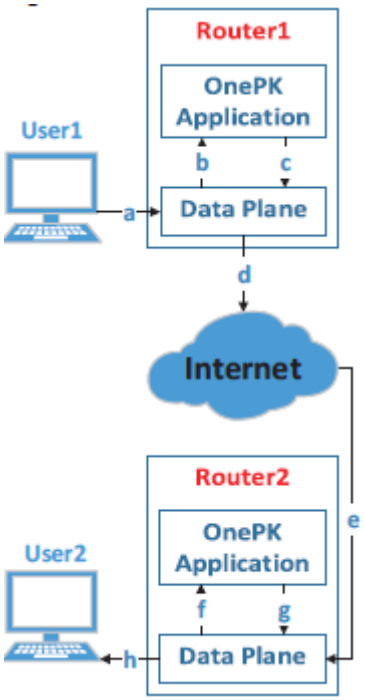
6) Informasi yang mencurigakan pada paket diganti dengan kriptogram dari langkah sebelumnya.

7) Akhirnya paket akan dikirim ulang untuk data plane dan mengikuti prosedur yang telah ditentukan.

Setelah konfigurasi dibuat pada router, sebuah paket dari user 1 menuju user 2 akan memiliki perjalanan seperti berikut, dan akan ditunjukan pada gambar 1.6.

a. Router menerima paket
b. Paket yang cocok dengan ACL dikonfigurasi dan dikirim dari data plane aplikasi ONE PK, dan kemudian diharapkan balasanya.
c. Aplikasi ONE PK mengenkripsi isi paket dan mengirimkanya kembali ke data plane.
d. Data diterima dari router dan diteruskan ke router 2.
e. Router 2 menerima paket.
f. Paket yang cocok dengan ACL dikonfigurasi dan dikirim ke pesawat data ke aplikasi ONE PK, dan diharapkan hasilnya.
g. Aplikasi ONE PK mendeskripsi isi paket dan mengirimkanya kembali ke data plane.

h. Dari data plane masuk ke user 2.



Gambar 6 Lalu-lintas Pengamanan Paket Dengan Aplikasi ONE PK

Untuk semua operasi kriptografi yang digunakan yaitu kriptografi Open SSL (toilet version), kriptografi sudah bekerja dengan baik. Aplikasi yang dikembangkan tidak menuntut sebuah algoritma kriptografi apapun, sehingga admin bebas menentukan algoritma kriptografi yagn akan dipakai. SDN juga dikonfigurasi menggunakan web interface, pengguna dikonfigurasi menggunakan alamat IP router atau nama host yang akan meminta username dan password pada setiap user yang akan mengakses.

4. Kesimpulan

Penggunaan yang sudah terkonfirmasi memiliki akses ke penerbangan atau halaman manajemen. Halaman login menampilkan informasi tentang suatu keadaan saat ini dan hasil dari perintah konfigurasi. Melalui halaman manajemen, administrator dapat mengatur

router untuk mengenkripsi atau mendeskripsi lalu-lintas yang ditentukan (gambar 7). Parameter yang dapat dikonfigurasi adalah

a. Jaringan lokal

b. Jaringan remote

c. Layer 4 protokol

Mekanisme keamanan berbasis SDN diuji dalam lingkungan virtual "Cisco PK All-in-One", dengan infrastruktur 5 router dan 2 controller. Dari sudut pandang persyaratan kemanan, mekanisme yang diusulkan menawarkan kerahasiaan tingkat tinggi pada paket yang dipertukarkan antara router satu dan router yang lain.

Dalam uji coba serangan penyerang tidak tahu paket mana yang berisi informasi yang penting, sehingga tingkat probabilitas kebenaran paket untuk diserang sangatlah kecil, mekanisme kemanan juga tidak memperlambat kinerja jaringan bahkan untuk transfer file berukuran besar FTP. Seperti yang ditunjukan oleh gambar 8.Mekanisme ini dapat disimpulkan mengalahkan mekanisme IPSec. Dari tingkat keamanan sedikit lebih unggul, namun pada IPSec proses pengamanan memerlukan waktu yang cukup panjang, sehingga menurunkan kinerja jaringan.

Mekanisme yang dirancang merupakan sebuah solusi keamanan untuk jaringan SDN. Mekanisme ini memberikan integritas dan kerahasiaan untuk paket yang diperlukan saja. Mekanisme ini tidak membutuhkan waktu yang panjang dalam proses kerjanya sehingga tidak menurunkan kualitas dari jaringan. Dibanding dengan IPSec, mekanisme ini menawarkan keamanan tambahan yaitu dengan kebebasan dalam memilih algoritma kriptografi yang dibutuhkan. Dari hasil percobaan yang ada dapat disimpulkan bahwa mekanisme keamanan ini layak untuk diterapkan

pada jaringan SDN. Dan dapat diusulkan sebagai perbaikan keamanan untuk jaringan komputer yang berbasis SDN.

5. Referensi

[1] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic," SIGPLAN Not., vol. 46, no. 9, p. 279, Sep. 2011.

[2] ONF, "Network Interfaces Working Group," Oct. 2013

[3] G. Pickett, "Abusing Software Defined Networking," In DEFCON 22, Las Vegas, 2014.

[4] IETF, "The OAuth 2.0 Authorization Framework," RFC 6749, 2012.

[5] OpenDaylight, "OpenDaylight Developer Guide," Oct. 2014.

[6] Michael Jarschel, Simon Oechsner, Daniel, Schlosser, Rastin Pries, Sebastian Goll, Phuoc, Tran Gia, " Modeling and Performance Evaluation of an OpenFlow Architecture ", *Proceedings of ITC,* 2012.

[7] Sachin Sharma, Dimitri Staesan, Didier Colle, Mario Pickavet and Piet Demeester , " Enabling Fast Failure Recovery in OpenFlow Networks ", *8th International Workshop on the Design of Reliable Communication Network*, 2011.

[8] Luciano Jerez Chaves, Vitor Marge Eichemberger, Islene Calciolari Garcia, and Edmundo R. Mauro Madeira Conference, " Integrating OpenFlow to LTE: some

issues toward Software-Defined Mobile Networks ", APNOMS, 2015.

[9] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey, *"Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76,* Jan 2015.

[10] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN Security: A Survey," inFuture Networks and Services (SDN4FNS), 2013 IEEE SDN for, Nov 2013, pp. 1–7.

[11] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures," *in Proceedings of 2015 Annual Network and Distributed System SecuritySymposium (NDSS'15)*, February 2015.