SKRIPSI

PENERAPAN SOFTWARE DEFINED SECURITY UNTUK MITIGASI DDOS

Disusun dan diajukan oleh:

ANDI MUHAMMAD GHAZY AYMAN

D121171307

PROGRAM STUDI DEPARTEMEN INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2023

LEMBAR PENGESAHAN SKRIPSI

PENERAPAN SOFTWARE DEFINED SECURITY UNTUK MITIGASI DDOS

Disusun dan diajukan oleh

A. MUHAMMAD GHAZY AYMAN

D121171307

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 12 Juli 2023 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,	Pembimbing Pendamping,
Dr. Eng. Ir. Muhammad Niswar, S.T., M.InfoTech.	Dr.Eng. Ady Wahyudi Paundu, ST.,MT.
Nip. 197309221999031001	Nip. 197503132009121003

Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, ST, MT, M.Bus.Sys., IPM, ASEAN.Eng

Nip. 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;

Nama : Andi Muhammad Ghazy Ayman

NIM : D121171307

Program Studi : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

PENERAPAN SOFTWARE DEFINED SECURITY
UNTUK MITIGASI DDOS

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 12 JUN 2023

Yang Menyatakan

Andi Muhammad Ghazy Ayman

ABSTRAK

ANDI MUHAMMAD GHAZY AYMAN. *Penerapan Software Defined Security Untuk Mitigasi DDoS* (dibimbing oleh Dr. Eng. Muhammad Niswar, S.T., M.IT. Dan Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.)

Distributed Denial of Service (DDoS) merupakan *cyber threat* pada infrastruktur jaringan, yang mengalami lonjakan pada satu dekade terakhir dan menyebabkan kerugian yang besar tiap tahunnya. Begitupun, dengan serangan DDoS pada infrastruktur baik jaringan tradisional maupun Software Defined Network.

Untuk itu penelitian ini dimulai dengan menganalisis serangan DDoS terkhusus serangan volumetrik pada jaringan SDN, memberikan wawasan tentang dampaknya pada infrastruktur jaringan. Dilakukan studi komprehensif tentang prinsip-prinsip software-defined networking (SDN) untuk memahami potensinya dalam meningkatkan keamanan jaringan.

Kerangka kerja yang diusulkan dengan memanfaatkan fitur pemrograman dan kontrol terpusat dari SDN serta *third party tools* untuk mendeteksi dan merespons serangan DDoS secara dinamis secara *real time*. Melibatkan desain dan implementasi mekanisme baru, termasuk *monitor* dan analisis *traffic*, identifikasi serangan, dan mitigasi serangan.

Untuk mengevaluasi efektivitas dari kerangka keamanan berbasis perangkat lunak, serangkaian eksperimen dilakukan menggunakan *testbed* atau lingkungan percobaan. Beberapa skenario serangan DDoS disimulasikan, dan kinerja kerangka kerja dalam mendeteksi dan mitigasi serangan tersebut dinilai. Metrik seperti akurasi deteksi serangan, performansi, dan throughput jaringan diukur untuk mengukur efisiensi kerangka kerja.

Hasil dari penelitian ini adalah dengan menguji ketahanan jaringan serta efektifitas sistem dalam menganggulangi serangan DDoS. Pada

sistem ini tidak terdapat *background flow* atau *post-impact-mitigation* sehingga dampak pada kinerja tidak terlalu signifikan terlihat pada hasil throughput dari lima percobaan memiliki rata-rata keberhasilan 99.64%, latency 0.01127636 sec, atau 11.27636 ms, dan rata-rata packet loss 0.36% yang mengindikasikan kualitas yang baik pada sistem.

Kata Kunci: Distributed Denial of Service (DDoS), keamanan berbasis perangkat lunak, software-defined networking (SDN), mitigasi serangan, ketahanan jaringan.

ABSTRACT

ANDI MUHAMMAD GHAZY AYMAN. Software Defined Security Impelementation for Mitigating DDoS. (supervised by Dr. Eng. Muhammad Niswar, S.T., M.IT. & Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.)

Distributed Denial of Service (DDoS) is a cyber threat to network infrastructure that has seen a significant surge in the last decade, causing substantial annual losses. This applies to both traditional network infrastructure and Software Defined Network.

Therefore, this research begins by analyzing DDoS attacks, particularly volumetric attacks, on SDN networks, providing insights into their impact on network infrastructure. A comprehensive study of software-defined networking (SDN) principles is conducted to understand its potential in enhancing network security.

The proposed framework leverages the programming and centralized control features of SDN, along with third-party tools, to dynamically detect and respond to DDoS attacks in real time. This involves the design and implementation of new mechanisms, including traffic monitoring and analysis, attack identification, and attack mitigation.

To evaluate the effectiveness of the software-defined security framework, a series of experiments are conducted using a testbed environment. Several DDoS attack scenarios are simulated, and the framework's performance in detecting and mitigating these attacks is assessed. Metrics such as attack detection accuracy, performance, and network throughput are measured to evaluate the framework's efficiency.

The result of this research is to test network resilience and system effectiveness in countering DDoS attacks. In this system there is no background flow or post-impact-mitigation so that the impact on performance is not too significant as seen in the throughput results of the five trials which have an average success of 99.64%, latency of

0.01127636 sec, or 11.27636 ms, and an average packet loss of 0.36 % which indicates good quality of the system.
Keywords: Distributed Denial of Service (DDoS), software-defined security, software-defined networking (SDN), attack mitigation, network resilience.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
API	Application Programming Interface
CLI	Command Line Interface
DDoS	Distributed Denial of Services
DOS	Denial of Services
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
SDN	Software Defined Network
SDSec	Software Defined Security
SEP	Security Enforcement Point
SYN	Synchronize
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1. Latar Belakang

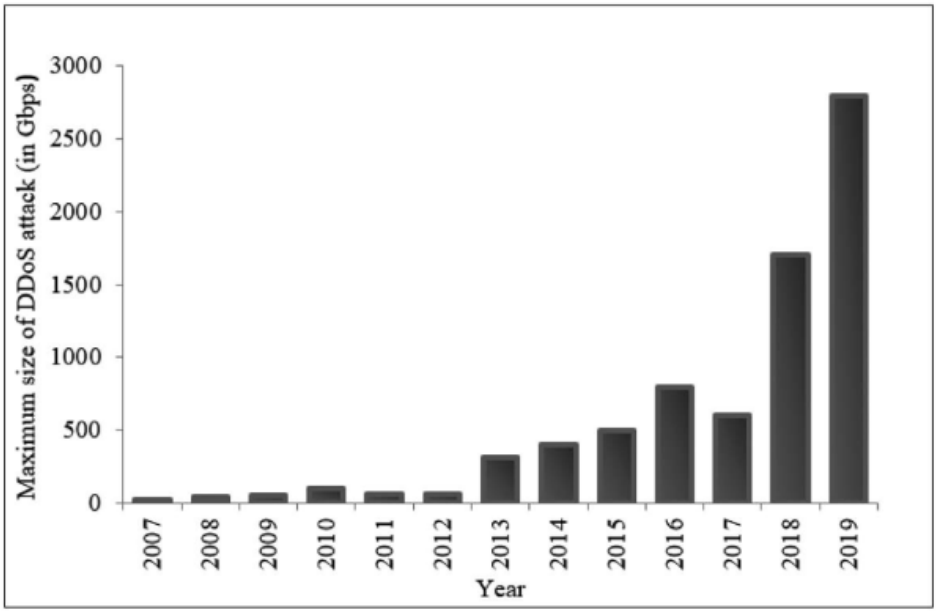
Software Defined Security (SDSec) adalah arsitektur keamanan jaringan dalam konsep jaringan terprogram yang memisahkan aturan kebijakan dan mekanisme keamanan. Pemisahan ini memungkinkan kebijakan keamanan untuk diubah dan dimodifikasi melalui perangkat lunak, tidak tersemat dalam mekanisme keamanan itu sendiri. SDSec memungkinkan kebijakan keamanan menjadi lebih fleksibel dan lebih mudah diperbarui dan dikelola.

Dalam arsitektur SDSec, kebijakan keamanan dikonfigurasi dalam server kebijakan terpusat, yang berkomunikasi dengan *Security Enforcement Point* (SEP) yang terletak di seluruh jaringan. SEP bertanggung jawab untuk menerapkan *policy* keamanan dengan memeriksa lalu lintas yang melewatinya dan menerapkan kontrol keamanan sesuai kebutuhan. Salah satu manfaat utama SDSec adalah memungkinkan administrator keamanan memperbarui dan mengelola kebijakan keamanan dengan mudah tanpa perlu mengubah mekanisme keamanan secara mendasar. untuk merespons perubahan ancaman dan persyaratan keamanan, dan dapat membantu meningkatkan postur keamanan organisasi secara keseluruhan. Karena SDN memusatkan kontrol jaringan terpusat, penting untuk memastikan bahwa controller terpusat dan komunikasi antar *controller* dan *switch* aman terhubung .

Menurut Azodolmolky, Siamak. *Software Defined Networking with OpenFlow*. Birmingham: Packt Publishing, 2013. SDN adalah sebuah konsep jaringan revolusioner yang berfungsi untuk menyederhanakan kontrol jaringan, manajemen, dan memungkinkan inovasi melalui programabilitas jaringan. Jaringan komputer tradisional dibangun dalam jumlah besar perangkat jaringan (seperti switch, router, firewall, dan sebagainya) dengan banyak protokol (perangkat lunak) yang kompleks, yang tersemat secara pabrikan dan

tidak dapat diubah. Teknisi jaringan yang bertugas untuk mengonfigurasi jaringan tradisional dengan berbagai lisensi dan skenario yang dikerjakan secara manual mulai dari *high-level policy* hingga *low-level* konfigurasi. Tugas yang sangat kompleks seperti ini sering diselesaikan dengan alat dan waktu yang terbatas. Dengan demikian, kontrol, kinerja, dan manajemen jaringan adalah tugas yang cukup rumit dan rawan akan kesalahan.

Dalam sisi keamanan atau *cybersecurity* kita dapat melihat bahwa baik di jaringan tradisional maupun jaringan terprogram (SDN) juga memiliki celah yang serupa, dalam hal ini sistem manapun memiliki kerentanan keamanannya masing-masing. Salah satu yang penulis coba untuk implementasikan adalah bagaimana jaringan SDN dapat berfungsi untuk memitigasi serangan yang sering terjadi dalam jaringan tradisional seperti DoS (Denial of Service) lalu kita dapat menspesifikkan metode serangan menjadi DDoS (Distributed Denial of Service) dengan mensimulasikan proyeksi serangan dan bagaimana API dari SDN yaitu Openflow dapat berfungsi untuk mencegah atau memitigasi serangan yang masuk dalam jaringan terprogram. Dapat dilihat pada Gambar 1 yang dikutip dari buku *Distributed Denial of Service (DDoS) Attacks Classification, Attacks, Challenges, and Countermeasures.* 2021. Menjelaskan bahwa tingkat serangan DDoS pada infrastruktur jaringan mengalami peningkatan yang pesat dari tahun ke tahun. Hal ini dikarenakan serangan DDoS digunakan untuk mengetes ketahanan sistem sebelum *hacker* melanjutkan tujuan utama mulai dari pencurian data hingga perusakan sistem.



Gambar 1: Serangan DDoS dari tahun ke tahun. Arbour Network Inc

Melihat dari latar belakang dan penjelasan ringkas terkait SDSec atau Software Defined Security dan bagaimana implementasinya dalam jaringan SDN maka dari itu penulis mengusulkan judul **"***Penerapan Software Defined Security Untuk Mitigasi DDoS.***"** Penelitian ini sekiranya dapat menjawab kebutuhan para teknisi jaringan terkhusus jaringan terprogram di bidang keamanan jaringan untuk dapat menjadi bahan referensi acuan dalam mengembangkan jaringan terprogram yang berorientasi pada keamanan internet dan jaringan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah pada penelitian ini antara lain:

1. Bagaimana merancang sistem jaringan SDN untuk penerapan keamanan jaringan?
2. Bagaimana membangun sistem pertahanan terhadap DDoS pada jaringan berbasis SDN.
3. Bagaimana mengukur keberhasilan sistem mitigasi pertahanan keamanan jaringan SDN?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini ialah membangun sistem jaringan Software-Defined Security dan mitigasi serangan pada jaringan Software Defined Network serta menganalisis performansi sistem SDSec dalam konteks akurasi atau ketepatan *forwarding packet* dalam jaringan SDN.

1.4. Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat dan menjadi referensi dan acuan untuk pengembangan lebih lanjut dalam bidang Software Defined Security.

1.5. Batasan Masalah

Batasan masalah yang ditentukan dalam sistem ini adalah:

1. Uji coba dilakukan dengan eksperimen simulasi model yang akan menghasilkan *virtual lab* jaringan,
2. Bentuk dari *virtual lab* adalah DDoS-script, SDN tools, dan OS,
3. Implementasi simulasi jaringan SDN menggunakan sistem operasi Ubuntu.
4. Pada Percobaan ini tidak terdapat background flow sehingga tidak ada *post-impact-mitigation* yang mempengaruhi laju jaringan.

1.6. Sistematika Penulisan

Laporan tugas akhir ini dibagi dalam lima bab yang tersusun secara sistematis, yaitu:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang pengambilan topik *Penerapan Software Defined Security untuk mitigasi DDoS*, mulai dari perumusan masalah, tujuan penelitian, batasan masalah, manfaat penulisan, hingga sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan konsep dasar dan landasan teori yang menunjang setiap perangkat pada penelitian ini. Teori-teori yang akan dijelaskan yaitu konsep dasar *software defined network*, *openflow API*, *software defined security*, *distributed denial of service*, *intrusion detection system*, dan *intrusion prevention system.*

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai metode yang digunakan dalam penelitian, mencakup tempat dan waktu penelitian, instrumen penelitian, prosedur penelitian, gambaran umum sistem, dan analisis mitigasi sistem.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil penerapan software defined security untuk mitigasi DDoS, serta hasil pengujian sistem dan pembahasan dari hasil pengujian yang dilakukan.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dan saran dari penelitian yang dilakukan dan saran-saran yang dapat dilakukan pada penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1. Konsep Dasar Software Defined Network

Dilansir dari Open Network Foundation (2022) Software-Defined Networking (SDN) adalah arsitektur jaringan baru yang dinamis, mudah dikelola, hemat biaya, dan mudah beradaptasi, menjadikannya ideal untuk bandwidth tinggi, sifat dinamis dari aplikasi saat ini. Arsitektur ini memisahkan kontrol jaringan dan fungsi penerusan yang memungkinkan kontrol jaringan menjadi dapat diprogram secara langsung dan infrastruktur dasar untuk diabstraksikan menjadi aplikasi dan layanan jaringan. Protokol OpenFlow adalah elemen dasar untuk membangun jaringan SDN.

Dalam arsitektur SDN, jaringan dibagi menjadi dua bagian: *control plane* dan *data plane*. *control plane* bertanggung jawab untuk mengelola dan mengonfigurasi jaringan, sedangkan *data plane* bertanggung jawab untuk meneruskan dan merutekan paket data. *control plane* biasanya diimplementasikan sebagai komponen perangkat lunak terpisah, yang dikenal sebagai *controller*, yang berkomunikasi dengan perangkat jaringan untuk mengonfigurasi dan mengelolanya. Hal ini memungkinkan pemisahan *controller* dan *data plane*, membuatnya lebih mudah untuk mengubah arsitektur jaringan dengan memodifikasi perangkat lunak controller, daripada harus mengonfigurasi ulang perangkat individual.

Salah satu manfaat utama SDN adalah memungkinkan jaringan yang dapat diprogram dan fleksibel, karena controller dapat diprogram untuk membuat keputusan berdasarkan berbagai faktor seperti lalu lintas jaringan, kebijakan pengguna, atau bahkan insiden. Ini memungkinkan manajemen jaringan otomatis dan kemampuan untuk merespons dengan cepat terhadap perubahan kondisi jaringan.

Manfaat lain dari SDN adalah memungkinkan tampilan jaringan yang lebih terpusat, membuatnya lebih mudah untuk memantau dan memecahkan masalah. Dengan SDN, administrator jaringan dapat dengan mudah melihat status semua perangkat dan koneksi di jaringan, serta dengan cepat mengidentifikasi dan memperbaiki masalah yang muncul.

Teknologi SDN masih relatif baru, sehingga masih terus ditingkatkan dan dikembangkan, namun memiliki potensi untuk merevolusi cara jaringan dikelola dan dioperasikan.

Berdasarkan definisi *software defined network* yang telah dijelaskan, SDN dapat didefinisikan sebagai sebuah arsitektur jaringan terprogram yang berguna untuk meminimalisir biaya infrastruktur jaringan serta mempermudah manajemen jaringan. Berdasarkan penelitian X. Zhu et al. (2016) dalam buku *Business Trends in the Digital Era*, terdapat beberapa jenis implementasi Software Defined Anything :

1. *Software Defined Storage* (SDS)

   Software Defined Storage (SDS) adalah jenis arsitektur penyimpanan di mana kontrol dan pengelolaan sumber daya penyimpanan dipisahkan dari perangkat keras penyimpanan fisik yang mendasarinya dan sebagai gantinya diimplementasikan melalui perangkat lunak. SDS memungkinkan fleksibilitas dan skalabilitas yang lebih besar dalam penyebaran dan pengelolaan sumber daya penyimpanan, dan memungkinkan penggunaan perangkat keras standar industri, daripada sistem penyimpanan *propietary*.

2. *Software Defined Data Center (SDDC)*

   Software Defined Data Center adalah merupakan pusat data kelas perusahaan yang menggunakan komputasi awan dan teknik virtualisasi. Biasanya, SDDC akan memiliki virtualisasi server, virtualisasi

penyimpanan, dan virtualisasi jaringan. Semua komponen ini akan dikelola oleh lapisan perangkat lunak yang akan menyediakan pengelolaan semua sumber daya secara terpusat dan terintegrasi.

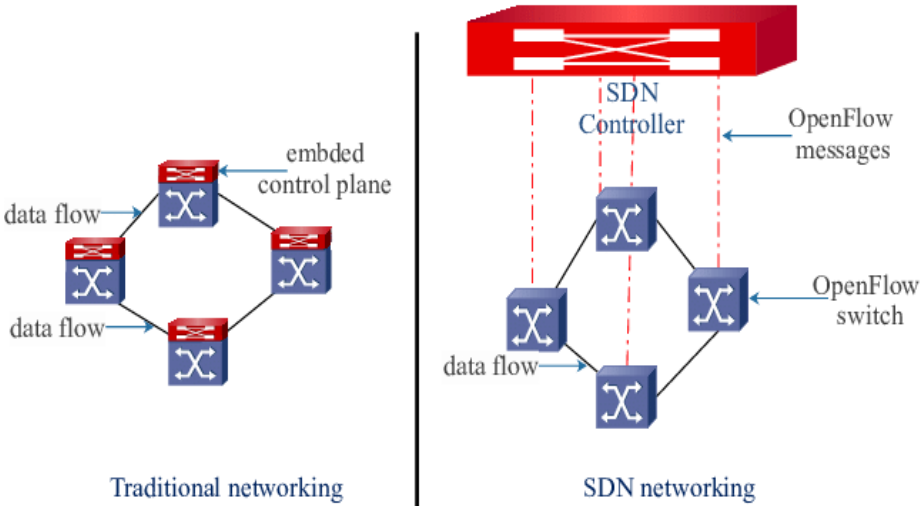
3. *Software Defined Infrastructure* (SDI)

   Software Defined Infrastructure (SDI) adalah pendekatan untuk mengelola dan menyediakan sumber daya teknologi informasi di mana kontrol dan pengelolaan infrastruktur diimplementasikan melalui perangkat lunak. SDI memungkinkan fleksibilitas, skalabilitas, dan otomatisasi yang lebih besar dalam penyebaran dan pengelolaan sumber daya infrastruktur, dan memungkinkan penggunaan perangkat keras standar industri, daripada *enterprise* sistem.

4. *Software Defined Security* (SDSec)

   Software Defined Security adalah jenis model keamanan di mana keamanan informasi dalam lingkungan komputasi diimplementasikan, dikendalikan, dan dikelola oleh perangkat lunak keamanan.

   Ini adalah keamanan yang dikelola perangkat lunak, digerakkan oleh kebijakan, dan diatur di mana sebagian besar kontrol keamanan seperti deteksi intrusi, segmentasi jaringan, dan kontrol akses diotomatisasi dan dipantau melalui perangkat lunak.



Gambar 2: Arsitektur Jaringan Tradisional & SDN. Maaloul (2018).

Menurut Maaloul 2018 Maaloul, et al (2018). Dalam jurnal *"Energy-Aware Routing in Carrier-Grade Ethernet using SDN Approach."* Menjelaskan bahwa Software Defined Networking (SDN) dan jaringan tradisional sangat berbeda dalam prinsip desain dan manajemen sistem. Jaringan tradisional mengandalkan desentralisasi, di mana setiap perangkat, seperti router atau switch, menangani lalu lintas jaringan secara independen berdasarkan tabel routingnya sendiri, dan sangat bergantung pada perangkat keras dengan fungsi yang tetap.

Sebaliknya, SDN memusatkan kontrol ke dalam satu entitas yang memiliki kontrol menyeluruh tentang jaringan, mengabstraksi logika pengambilan keputusan ke dalam perangkat lunak, meningkatkan fleksibilitas dan adaptabilitas, dan menawarkan pemrograman yang dapat menciptakan jaringan yang lebih otomatis dan fleksibel seperti yang dijabarkan pada Gambar 2.

2.2. Openflow

Gambar 3: Logo Openflow

OpenFlow adalah protokol jaringan *Software-Defined Network* (SDN) yang memungkinkan pemisahan control plane dan data plane dalam perangkat jaringan, seperti switch dan router. Openflow menyediakan antarmuka antara control plane dan data plane, memungkinkan control plane untuk mengelola konfigurasi jaringan dan aliran lalu lintas, sedangkan data plane menangani penerusan paket.

Protokol OpenFlow mendefinisikan sekumpulan pesan dan tindakan yang dapat digunakan untuk mengontrol aliran lalu lintas jaringan, dan memungkinkan pembuatan, modifikasi, dan penghapusan aliran jaringan secara dinamis. Openflow memberikan tingkat fleksibilitas dan kemampuan program yang tidak mungkin dilakukan dengan perangkat jaringan tradisional, yang biasanya terprogram dan sulit diubah.

OpenFlow didukung oleh berbagai vendor jaringan dan digunakan dalam berbagai kasus penggunaan, termasuk jaringan pusat data, jaringan kampus, dan jaringan penyedia layanan. Protokol OpenFlow banyak digunakan untuk kasus penggunaan seperti rekayasa lalu lintas, *load balancing*, dan virtualisasi jaringan.

OpenFlow umumnya digunakan bersama dengan kontroller SDN, yang merupakan aplikasi perangkat lunak yang mengelola konfigurasi jaringan dan arus lalu lintas sesuai dengan kebijakan yang

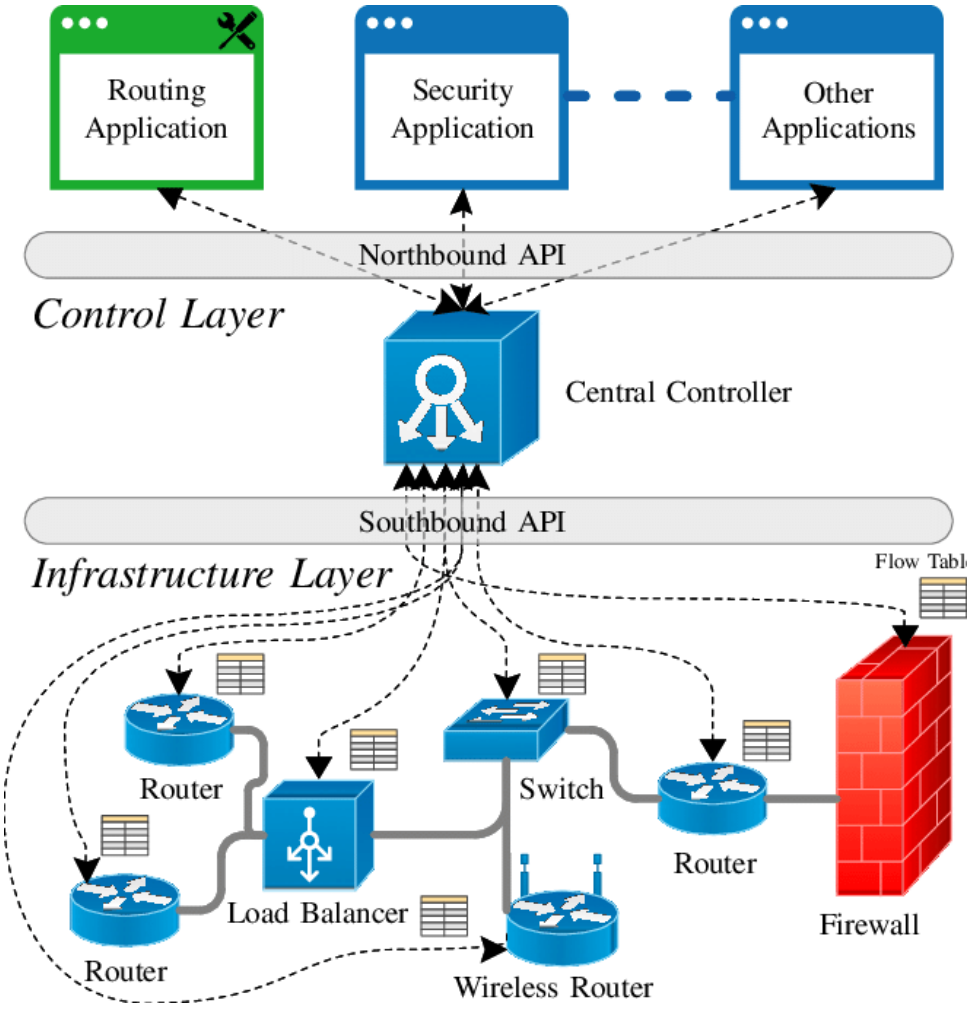
telah ditentukan. controller dapat berkomunikasi dengan perangkat jaringan menggunakan protokol OpenFlow untuk mengonfigurasi dan mengontrol aliran data dalam jaringan.

Pada bidang keamanan jaringan. Dalam jurnal *OpenFlow: A Security Analysis* (2013) OpenFlow dapat digunakan untuk membuat aturan aliran yang memblokir jenis lalu lintas tertentu (misalnya, lalu lintas dari alamat IP tertentu atau lalu lintas yang ditujukan untuk port tertentu) atau meneruskan lalu lintas ke perangkat keamanan untuk pemeriksaan lebih lanjut. Administrator jaringan memiliki lebih banyak kontrol atas lalu lintas apa yang diizinkan di jaringan, dan dapat membantu mencegah serangan dari malware atau hacker.

Cara lain OpenFlow dapat meningkatkan keamanan adalah dengan mengizinkan pembuatan jaringan virtual. Jaringan virtual ini dapat digunakan untuk membagi jaringan dan membatasi jumlah kerusakan yang dapat disebabkan oleh pelanggaran keamanan. Misalnya, jaringan virtual dapat dibuat untuk departemen atau aplikasi tertentu, dan lalu lintas dapat dibatasi sehingga hanya dapat mengalir antar perangkat di dalam jaringan virtual. Ini dapat membantu mencegah data sensitif dan membatasi penyebaran malware.

Fitur lain dari Openflow yang dapat menambahkan lapisan keamanan tambahan adalah memungkinkan untuk mengarahkan lalu lintas melalui peralatan keamanan seperti firewall, IDS atau IPS dengan mencocokkan aliran dan mengarahkannya sesuai dengan itu.

Openflow digunakan sebagai komponen kunci dalam strategi keamanan SDN, membantu meningkatkan visibilitas, kontrol, dan otomatisasi keamanan jaringan.



Gambar 4: Arsitektur Software Defined Network (Awad, 2017)

2.3. Mininet

Definisi *mininet* menurut Rodríguez P., *et al.* (2014) dalam Jurnal *Using Mininet for emulation and prototyping Software-Defined Networks* Mininet adalah emulator jaringan *open source* memungkinkan pengguna membuat jaringan virtual menggunakan konsep jaringan Software Defined Network (SDN). Mininet memungkinkan pengguna untuk meniru perilaku topologi jaringan yang kompleks pada satu mesin, dan biasanya digunakan untuk menguji, men-debug, dan membuat prototipe berbagai skenario jaringan. Mininet mendukung berbagai elemen jaringan, termasuk host,

switch, controller, dan link, dan dapat digunakan untuk meniru jaringan tradisional dan SDN.

Mininet juga menyediakan *Command Line Interface* (CLI) dan API Python untuk berinteraksi dengan jaringan virtual dan menjalankan eksperimen jaringan. Dengan API, kita dapat membuat dan mengonfigurasi elemen jaringan, memulai dan menghentikan layanan jaringan, serta mengumpulkan statistik dan data tentang perilaku jaringan. Mininet juga menyediakan utilitas untuk membuat controller dan swithc khusus yang dapat digunakan dengan jaringan virtual.

Mininet dapat digunakan untuk penelitian dan pengujian keamanan dengan memungkinkan pengguna membuat dan bereksperimen dengan jaringan virtual dalam lingkungan yang terkontrol dan terisolasi. Dengan membuat jaringan virtual, peneliti keamanan dapat menguji dan mengembangkan teknik dan alat keamanan baru tanpa membahayakan stabilitas jaringan. Selain itu, Mininet dapat digunakan untuk mensimulasikan berbagai jenis serangan keamanan, seperti serangan denial *of service*, serangan *man-in-the-middle*, dan network *sniffing*, dan untuk mempelajari efek dari serangan ini terhadap performa dan keamanan jaringan.

Mininet dapat digunakan bersama dengan berbagai jenis controller *Software Defined Network* (SDN), seperti OpenFlow, untuk memungkinkan peneliti bereksperimen dengan konfigurasi jaringan yang berbeda dan untuk menguji keefektifan kebijakan keamanan yang berbeda. Berguna untuk mengidentifikasi kerentanan dan kelemahan dalam jaringan, serta mengembangkan strategi keamanan baru untuk melindungi dari potensi ancaman.

Mininet sering digunakan dalam ekosistem akademik dan penelitian, memungkinkan siswa dan peneliti untuk belajar tentang cara kerja bagian dalam jaringan komputer, dan bereksperimen dengan

konfigurasi jaringan yang berbeda dan teknik keamanan dalam lingkungan yang aman dan terkendali.

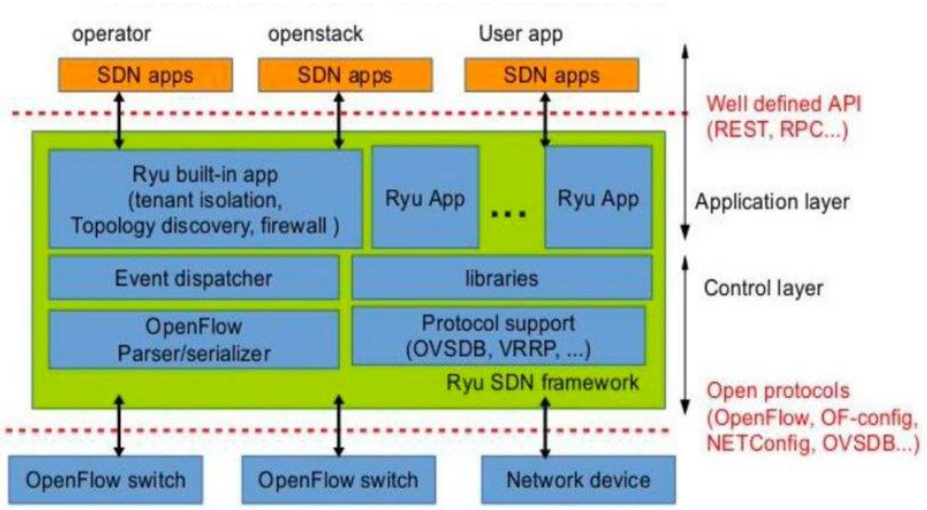
2.4. Controller

Dalam *Software Defined Network* (SDN) kontroler adalah komponen kunci yang mengelola aliran data melalui jaringan. Kontroler bertanggung jawab untuk memelihara topologi jaringan dan membuat keputusan tentang ke mana harus mengirim paket berdasarkan keadaan jaringan saat ini. controller berada di antara *forwarding plane* (switch dan router yang secara fisik meneruskan paket) dan *management plane* (di mana administrator mengonfigurasi jaringan), dan berkomunikasi dengan kedua bidang untuk membuat keputusan cerdas tentang cara merutekan lalu lintas. Controller biasanya menggunakan antarmuka terbuka, seperti protokol OpenFlow, untuk berkomunikasi dengan bidang penerusan, yang memungkinkannya memprogram perilaku perangkat jaringan.

Menurut Ahmad, I., *et al* (2015). *Security in Software Defined Networks: A Survey*. Controller juga mampu memanajemen jaringan yang dapat diprogram dan digunakan untuk mengelola dan mengontrol perilaku perangkat jaringan di SDN. Salah satu cara controller SDN dapat digunakan untuk keamanan jaringan adalah dengan menyediakan titik pusat kontrol dan visibilitas untuk jaringan, yang membuatnya lebih mudah untuk mendeteksi dan merespons ancaman keamanan. Selain itu, controller SDN dapat diprogram untuk menerapkan kebijakan keamanan, seperti firewall dan pemantauan lalu lintas, di seluruh jaringan. Kontroler berfungsi untuk mempersulit penyerang untuk melewati langkah-langkah keamanan, karena semua lalu lintas harus melewati controller. Cara lain adalah dengan memberikan segmentasi jaringan dan microsegmentation dengan menggunakan konsep virtualisasi jaringan metode ini memberikan isolasi lalu lintas yang berbeda untuk meminimalkan permukaan serangan.

Ada beberapa controller SDN yang populer dan sering digunakan oleh para peneliti dan insinyur jaringan antara lain :

1. OpenDaylight: Controller SDN yang sangat modular dan dapat diperluas yang mendukung berbagai perangkat dan protokol jaringan.
2. ONOS: controller SDN open-source yang berfokus pada kinerja tinggi dan skalabilitas.
3. OpenContrail: controller SDN yang dikembangkan oleh Juniper Networks yang dirancang untuk digunakan dalam jaringan multi-enterprise yang besar.
4. RYU: controller SDN open source yang ditulis dengan Python dan mendukung berbagai protokol dan perangkat.
5. Cisco APIC-EM: controller SDN Cisco untuk jaringan perusahaan, berfokus pada otomatisasi dan manajemen jaringan yang disederhanakan
6. NSX-T: Solusi Software-Defined Networking dari VMware yang menyediakan kemampuan virtualisasi jaringan dan segmentasi mikro.
7. Open Networking Operating System (ONOS): controller SDN open-source yang berfokus pada jaringan tingkat operator.
8. controller SDN HPE: controller SDN HPE untuk mengelola jaringan fisik dan virtual.



Gambar 5: Arsitektur RYU controller

2.5. Wireshark

Gambar 6: Logo Wireshark

Wireshark adalah tools analisa paket *open source* yang digunakan untuk pemecahan masalah jaringan, analisis, pengembangan perangkat lunak dan protokol komunikasi, dan penelitian. Wireshark mampu menganalisis berbagai protokol, termasuk Ethernet, DNS, HTTP. Wireshark tersedia untuk Windows, macOS, Linux, dan platform lainnya.

Wireshark dapat digunakan untuk menganalisis dan memecahkan masalah lalu lintas di jarinan SDN. SDN adalah jenis arsitektur jaringan di mana control plane dipisahkan dari data plane, memungkinkan lebih banyak fleksibilitas dan kemampuan pemrograman dalam manajemen jaringan. Wireshark dapat digunakan untuk menangkap dan menganalisis lalu lintas yang mengalir melalui jaringan, termasuk lalu lintas control plane dan data plane.

Menurut E.V Josy *et al* dalam *Information and Communication Technology for Competitive Strategies* (ICTCS 2020) kita dapat dengan mudah mengintegrasikan wireshark dan SDN. Di ekosistem SDN, Wireshark dapat digunakan untuk memeriksa protokol OpenFlow,

yang biasanya digunakan untuk berkomunikasi antara *control plane* dan *data plane* di SDN. Wireshark dapat berguna untuk mengidentifikasi dan memecahkan masalah terkait dengan tabel aliran pada switch, atau masalah komunikasi antara controller dan switch.

Wireshark juga dapat digunakan untuk menganalisa trafik pada protokol lain yang digunakan di SDN, seperti BGP-LS, PCEP, PCE dan lainnya. Wireshark dapat membantu mengidentifikasi masalah perutean dan rekayasa lalu lintas, serta membantu mengoptimalkan kinerja jaringan.

Penting juga untuk menyebutkan bahwa untuk jaringan skala besar dan untuk data besar, lebih baik menggunakan alat lain seperti Tshark, CIFS, dan lainnya yang lebih dirancang untuk pengambilan dan analisis data skala besar.

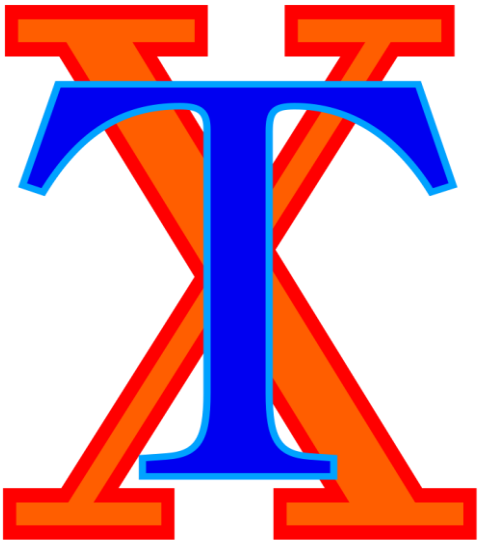
Secara umum ada beberapa cara untuk mengoperasikan wireshark, antara lain :

1. Instal Wireshark di komputer kita. Ini tersedia untuk Windows, macOS, dan Linux.
2. Mulai Wireshark. Di Windows, kita dapat melakukannya dengan mencari "Wireshark" di menu Mulai. Di macOS dan Linux, kita dapat memulai Wireshark dengan menjalankan perintah "wireshark" di terminal.
3. Pilih antarmuka jaringan untuk menangkap paket. Wireshark akan menampilkan daftar antarmuka jaringan yang tersedia di komputer kita
4. Klik tombol "Start" untuk mulai menangkap paket. Semua paket yang ditangkap akan ditampilkan di jendela Wireshark.
5. Gunakan filter untuk menampilkan hanya paket yang kita minati. Wireshark menggunakan sintaks Berkeley Packet Filter (BPF) untuk memfilter. Misalnya, kita dapat memfilter semua lalu lintas HTTP dengan mengetik "http" di kotak filter.

6. Menganalisis paket. Klik pada paket dalam daftar untuk melihat detailnya di panel bawah. kita juga dapat menggunakan opsi "Ikuti TCP Stream" untuk melihat seluruh percakapan antara dua alamat IP.

7. Simpan paket yang diambil ke file. Setelah kita menangkap paket yang kita butuhkan, kita dapat menyimpannya ke file dengan mengklik menu "File", lalu memilih "Save As."

2.6. XTerm



Gambar 7: Logo XTerm

Xterm adalah emulator terminal untuk Sistem X Window. Xterm berfungsi mengemulasi fitur dan fungsionalitas terminal DEC VT100, dan banyak digunakan pada sistem berbasis Unix seperti Linux dan macOS. xterm mendukung berbagai opsi dan fitur, termasuk dukungan untuk warna, font yang dapat disesuaikan, dan pemetaan kunci, dan dukungan untuk berbagai protokol terminal seperti VT100, VT220, dan VT320.

Menurut Lorenzo Piccioni (2004) dalam jurnal *Xterm : A Flexible Stkitard-Compliant XML-Based Termbase System.* XTerm dirancang sebagai “ Proyek Bahasa dan Kegiatan Produktif” tetapi

tidak terbatas pada itu. Bahkan dapat dengan mudah disesuaikan dengan proyek terminologis yang berbeda karena bukan hanya termbase, tapi dapat digunakan sebagai Term Base Management System (TBMS).

Seluruh sistem terdiri dari 4 komponen utama:

1. Mesin basis data (mySQL, Oracle, Access) yang menangani penanganan data mentah tingkat rendah;

2. Xterm.NET, lingkungan grafis untuk penyisipan data, manajemen termbase, permintaan dan visualisasi;

3. satu atau lebih file konfigurasi XML yang mendefinisikan data struktur termbase (jumlah yang hampir tidak terbatas termbase dengan struktur berbeda dapat dihosting mesin yang sama);

4. XTerm.portal, aplikasi web yang menyediakan layanan umum akses ke termbase(s) melalui mesin queri komprehensif.

2.7. DDoS

Serangan Distributed Denial of Services (DDoS) adalah jenis *cyber attack* di mana banyak komputer, sering kali disusupi oleh malware, digunakan untuk membanjiri situs web atau server yang ditargetkan dengan lalu lintas dalam jumlah besar, sehingga situs web atau server yang ditargetkan menjadi tidak tersedia kepada pengguna. Tujuan serangan DDoS adalah membuat situs web atau layanan tidak tersedia bagi pengguna yang sah dengan membanjiri lalu lintas dari berbagai arah.

Ada beberapa contoh model serangan DdoS antara lain :

1. Serangan berbasis volume: Serangan ini membanjiri target dengan volume lalu lintas yang tinggi, seperti membanjirnya paket UDP atau ICMP *request*.

2. Serangan protokol: Serangan ini mengeksploitasi kerentanan dalam protokol yang digunakan oleh target, seperti *SYN Flood* atau serangan paket terfragmentasi.

3. Serangan lapisan aplikasi: Serangan ini menargetkan kelemahan spesifik pada lapisan aplikasi situs web atau layanan, seperti serangan slowloris atau eksploitasi zero-day.
4. Serangan amplifikasi: Ini menyerang penyalahgunaan penyelesai DNS terbuka dan perangkat jaringan lain yang salah konfigurasi untuk memperkuat lalu lintas serangan.

5. Serangan botnet: Botnet adalah jaringan komputer yang dikendalikan oleh penyerang, yang dapat digunakan untuk meluncurkan serangan DdoS.
6. Serangan hybrid: Serangan ini merupakan kombinasi dari beberapa jenis serangan yang disebutkan di atas.

Penting untuk diperhatikan bahwa jenis serangan DDoS terus dikembangkan oleh penyerang, jadi penting untuk selalu mengetahui tren terbaru dan praktik terbaik untuk mempertahankan diri dari serangan DDoS.

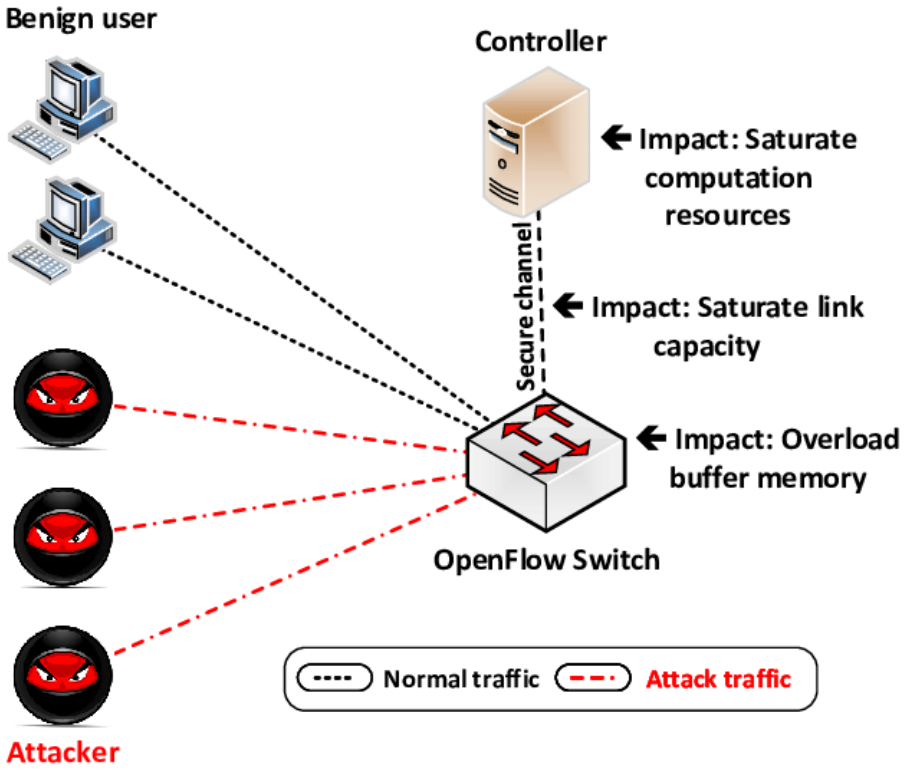
Dalam jurnal *Flooding DDoS mitigation and traffic management with software defined networking* ditulis oleh Kalliola, A., *et al* (2015). Serangan *distributed denial of service* (DDoS) pada jaringan *software defined network* (SDN) dapat berdampak signifikan pada ketersediaan dan kinerja jaringan.

Dalam serangan DDoS pada SDN, penyerang dapat menargetkan *controller* SDN, yang berfungsi untuk mengelola dan mengarahkan lalu lintas jaringan. Jika controller terbebani dengan lalu lintas, mungkin layanan menjadi tidak tersedia atau tidak dapat memproses lalu lintas yang sah, yang menyebabkan kemacetan jaringan dan gangguan layanan.

Target potensial lain untuk serangan DDoS pada SDN adalah switch virtual jaringan atau perangkat jaringan lainnya. Jika perangkat ini kewalahan dengan lalu lintas, perangkat tersebut mungkin menjadi

tidak tersedia atau tidak dapat memproses lalu lintas yang sah, yang menyebabkan gangguan jaringan lebih lanjut.

SDN menggunakan konsep programabilitas jaringan, sehingga serangan yang digunakan lebih mutakhir dan ditargetkan pada kerentanan tertentu, membuatnya lebih sulit untuk dideteksi dan dimitigasi. Penting untuk memiliki strategi pertahanan DDoS yang kuat yang mencakup kemampuan pemantauan, deteksi, dan mitigasi untuk melindungi dari serangan DDoS di SDN. Seperti yang terlihat pada Gambar 8 dimana serangan DDoS berfokus pada sisi data plane yang menyebabkan terjadinya *bottleneck* pada transmisi data sehingga jaringan akan *outservice* pada rentang waktu tertentu.



Gambar 8: Serangan DDoS di jaringan SDN

2.8. Sflow

Gambar 9: Logo sFlow

sFlow adalah teknologi pemantauan lalu lintas jaringan yang menyediakan statistik *real time* tentang penggunaan jaringan dan pola lalu lintas. sFlow beroperasi dengan mengambil sampel paket data di berbagai titik di jaringan dan melaporkan informasi tentang isinya ke sistem manajemen pusat. sFlow dapat digunakan untuk analisis kinerja jaringan, pemantauan keamanan, dan perencanaan kapasitas.

Menurut Ujjan, R. M. A., *et al* (2020). dalam jurnal *Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN*. sFlow dapat digunakan untuk mananggulangi serangan Distributed Denial of Service (DDoS) di lingkungan Software-Defined Networking (SDN). Dengan memantau lalu lintas jaringan dan menganalisis pola lalu lintas, sFlow dapat mengidentifikasi perilaku jaringan yang tidak normal yang mungkin mengindikasikan serangan DDoS. Informasi ini kemudian dapat digunakan untuk memicu tindakan mitigasi otomatis seperti pembatasan laju, pemfilteran lalu lintas, atau pengalihan lalu lintas ke pusat scrubbing. Namun, penting untuk dicatat bahwa sFlow hanyalah salah satu komponen dari strategi mitigasi DDoS yang komprehensif,

dan harus digunakan bersamaan dengan tindakan keamanan lainnya seperti firewall, sistem deteksi intrusi, dan layanan anti-DDoS.

Dalam kasus ini, kontroller Ryu adalah pengontrol OpenFlow berbasis Python yang dapat digunakan dengan sFlow untuk mengurangi serangan DDoS. Teknologi sFlow dapat digunakan untuk memantau lalu lintas jaringan dan mengidentifikasi potensi serangan DDoS, sedangkan kontroller Ryu dapat digunakan untuk mengimplementasikan tindakan mitigasi secara otomatis. Misalnya, kontroller Ryu dapat menerima data sFlow yang menunjukkan volume lalu lintas yang tinggi dari alamat IP sumber tertentu dan merespons dengan membatasi lalu lintas dari alamat IP tersebut.

Untuk menggunakan sFlow dengan pengontrol Ryu untuk mitigasi DDoS, sFlow harus diterapkan pada switch jaringan dan dikonfigurasikan untuk mengirim data sFlow ke pengontrol Ryu. Pengontrol Ryu kemudian harus diprogram untuk memproses data sFlow dan mengambil tindakan yang tepat untuk mengurangi serangan DDoS.

2.9. Iperf

Menurut “M. Abdullah, N. Al-awad, and F. W. Hussein, *"Performance Comparison and Evaluation of Different Software Defined Networks Controllers," International Journal of Computing and Network Technology, vol. 6, no. 2, 2018*” Iperf adalah alat diagnostik berbasis command line yang berfungsi untuk menganalisis dan mengukur kinerja jaringan. Dengan menggunakan Iperf, throughput jaringan maksimum yang dapat ditangani oleh suatu server dapat diukur, memungkinkan deteksi dan pemecahan masalah terkait performa jaringan. Iperf menciptakan aliran data antara klien dan server Iperf, dan mengevaluasi throughput yang dapat dicapai melalui aliran ini. Evaluasi ini memberikan pemahaman mendalam tentang kemampuan suatu tautan jaringan

dalam menangani trafik, atau bagaimana suatu jaringan berperforma dalam skenario tertentu .

2.10. Metode Keamanan

Menurut Valdovinos, I. A., *et al* (2021) dalam jurnal *Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions* menjelaskan ada beberapa cara untuk mengurangi serangan Distributed Denial of Service (DDOS) pada controller di Software-Defined Network (SDN):

1. Gunakan firewall untuk memfilter lalu lintas dan memblokir paket *malware*.
2. Terapkan pembatasan traffic untuk membatasi jumlah lalu lintas yang dapat mencapai controller.
3. Gunakan *load balancer* untuk mendistribusikan lalu lintas ke beberapa controller, sehingga lebih sulit bagi penyerang untuk membebani satu controller secara berlebihan.
4. Gunakan Intrusion Detection Prevention System (IDPS) untuk mendeteksi dan memblokir lalu lintas berbahaya.
5. Terapkan protokol komunikasi aman seperti SSL/TLS untuk komunikasi controller-ke-switch.
6. Gunakan layanan perlindungan DDoS khusus, seperti layanan berbasis cloud, untuk melindungi controller.

Metode keamanan yang digunakan pada pengujian kali ini adalah penggabungan antara metode keamanan Intrusion Detection System (IDS) dan Intrusion Prevention System (IPS). Intrusion Detection System berfungsi untuk mendeteksi anomali pada jaringan SDN sedangkan Intrusion Prevention System berfungsi untuk memitigasi serangan yang masuk pada jaringan SDN dan mengembalikan kondisi jaringan pada situasi semula.

2.11. Metode Analisis

Metode analisis yang digunakan berupa skenario pengujian tiap-tiap variabel *Quality of Service* yang mengacu pada standar ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) dan IETF (Internet Engineering Task Force) dengan menganalisa jaringan SDN serta memitigasi serangan DDOS dengan mempertimbangkan *availability* atau keadaan jaringan setelah mitigasi, dan *performance* atau akurasi dan kecepatan jaringan pada saat dalam kondisi terserang.