Jurnal TIKomSiN, Vol. 11, No. 1, April 2023 ISSN Cetak : 2338-4018
DOI: https://doi.org/10.30646/tikomsin.v11i1.641 ISSN Online : 2620-7532

Cloud Storage untuk Embedded *Intrusion Detection System*

Agus Reza Aristiadi Nurwa[1)]; Dimas Febriyan Priambodo[2*)]; Fahdel Achmad[3)]
[1,3)]Rekayasa Perangkat Keras Kriptografi, Politeknik Siber dan Sandi Negara
[2)]Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara
[1)]agus.reza@poltekssn.ac.id; [2)]fahdel.achmad@student.poltekssn.ac.id; [3)]dimas.febriyan@poltekssn.ac.id

ABSTRACT

The Corona Virus (COVID-19) pandemic has had a major social and economic impact on the world. Along with the potential challenges of sharing domains, brings with it many cybersecurity challenges that need to be addressed in a timely manner for critical infrastructure. The increase in the use of internet technology during this pandemic is directly proportional to the increase in the development of Information and Communication Technology (ICT) and cybercrime. Therefore, it is necessary to elaborate the existing ICTs to reduce the impact caused by attacks on the network according to the needs and capabilities of the users. This study applies a Network Intrusion Detection System (NIDS) based on the Raspberry Pi 4 Model B using Snort IDS with log storage media on cloud storage by visualizing the alerts generated to facilitate the analysis of anomalies that occur on the network. The result of this research is that there are attack signatures that are not available in the default rules so that further configuration is needed on Snort. The performance of the IDS sensor does not reduce the capability of the IDS sensor which acts as a hotspot when an attack occurs.

***Keywords** : IDS, cloud storage, Raspberry Pi 4 Model B, VPN, Snort*

I. PENDAHULUAN

Pandemi Virus Corona (COVID-19) telah menyebabkan dampak sosial dan ekonomi yang cukup besar dan bertahan lama di dunia. Seiring dengan tantangan potensial lainnya dari berbagi ranah, membawa banyak tantangan keamanan siber yang perlu ditangani tepat waktu untuk melindungi korban dan infrastruktur penting [1]. Adanya pandemi ini juga memaksa pemerintahan dari berbagai negara untuk menerapkan berbagai sistem kerja, salah satunya adalah *Work From Home* (WFH). Menurut Kementerian Komunikasi dan Informatika Republik Indonesia (KOMINFO), terjadi peningkatan penggunaan teknologi internet baik di daerah perkantoran maupun pemukiman antara 30 hingga 40 persen [2]. Peningkatan penggunaan teknologi ini memiliki andil yang besar dengan adanya peningkatan kejahatan dunia maya. *Federal Bureau of Investigation* (FBI) dewasa ini melaporkan perkiraan bahwa serangan siber meningkat sebanyak 4000 kejadian setiap hari sejak awal pandemi COVID-19 [3]. Peningkatan serangan siber tersebut dapat terjadi karena banyaknya serangan siber yang mengarah pada jaringan rumah, dimana sangat jarang diterapkan sistem pengamanan jaringan. Merujuk pada laporan [4] tercatat sebanyak 8.5 juta percobaan serangan siber yang menargetkan jaringan rumah. Oleh karena itu, penerapan *Intrusion Detection System* (IDS) pada jaringan rumah dapat menjadi salah satu rekomendasi solusi untuk mendeteksi adanya serangan atau ancaman pada jaringan. Hal ini bertujuan untuk menghindari ataupun mengurangi dampak dapat diakibatkan oleh serangan pada jaringan.

IDS merupakan sistem berupa perangkat keras ataupun lunak yang dapat mendeteksi aktivitas ataupun serangan berbahaya menggunakan aturan atau skrip konfigurasi tertentu [5]. IDS secara dinamis memantau tindakan tertentu, seperti lalu lintas data, catatan *syslog*, atau panggilan sistem dari sistem operasi tertentu, untuk menentukan apakah tindakan tersebut merupakan penggunaan yang sah atau gejala yang terkait dengan serangan yang diberikan [6]. Penerapan IDS perlu mempertimbangkan kemudahan instalasi dan biaya yang murah berdasarkan skala jaringan yang diawasi. Salah satu contoh dari IDS tersebut adalah Snort. Snort merupakan perangkat lunak pendeteksi intrusi jaringan yang paling banyak digunakan secara luas yang dirancang untuk menyediakan analisis waktu nyata dan mendeteksi paket data pada jaringan komunikasi [7]. Snort memiliki kemampuan pencatatan log file hingga 747 paket data perdetik [5] sehingga setidaknya diperlukan media penyimpanan sebesar satu TeraByte [8].

IDS dapat diimplementasikan menggunakan *single-board* komputer seperti

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

Raspberry Pi. Perangkat berbasis prosesor *Advanced RISC Machines* (ARM) multiguna dan murah yang dapat digunakan di lingkungan jaringan komputer seperti rumah, kantor kecil, ataupun institusi pendidikan [5]. Raspberry Pi dapat digunakan secara headless yang dikendalikan dari jarak jauh dan diprogram untuk menjalankan skrip secara mandiri. Raspberry Pi berbeda dari mikrokontroler, seperti Arduino yang hanya dapat diprogram untuk menjalankan program yang ditulis pengguna tunggal dan berkomunikasi dengan sensor dan elektronik lainnya [9]. Selain itu, pengembangan perangkat Raspberry Pi saat ini sudah sampai pada model Raspberry Pi 4 Model B dengan kemampuan modul 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, penyimpanan Micro-SD untuk penyimpanan data dan memuat sistem operasi, dan prosesor Broadcom BCM2711, Quad-core Cortex-A72 [10]. Ketahanan atau reliability dari Raspberry Pi untuk melakukan pemrosesan secara terus-menerus dapat bertahan hingga lebih dari 300 hari dengan selalu terhubung pada sumber daya dan dapat bertahan dilingkungan dengan suhu lebih dari 43 derajat celcius [11].

Perkembangan teknologi komputasi juga memberikan banyak pilihan layanan bagi pengguna berupa perangkat lunak baik secara *online* ataupun *offline* yang dapat disesuaikan dengan kebutuhan dan kemampuan pengguna. Penggunaan *software as a service dalam cloud computing dapat membantu* menghemat biaya tambahan pada perangkat keras [12]. Penyimpanan *cloud* yang merupakan bagian cloud computing merupakan salah satu solusi sekaligus memperluas akses karena dapat diakses menggunakan perangkat apa pun [13].

Kondisi penyimpanan fisik perlu proses penanganan yang dirancang untuk melindungi penyimpanan dari akses yang tidak sah, kehilangan, atau kehancuran, dan dari pencurian ataupun bencana [14]. *Cloud* dapat meningkatkan aspek kolaborasi, kemudahan, penskalaan, dan ketersediaan, serta memberikan potensi pengurangan biaya melalui komputasi yang dioptimalkan dan efisien [15]. Penggunaan penyimpanan *cloud* menjadi mudah bagi organisasi untuk memelihara dan menyimpan data mereka ke pihak ketiga [13].

Google mengembangkan layanan berbayar bernama *Cloud* IDS yang berbasis *cloud-native* dengan biaya sewa $1.5 setiap jam pengawasan dari setiap *endpoint* yang diawasi dan $0.07 untuk setiap Gigabyte data yang dianalisis dan diproses [16]. Integrasi antara sensor IDS dengan *cloud storage* melalui *Virtual Private Network* (VPN) dapat menjadi solusi untuk permasalahan keamanan jaringan dan retensi data dari penyimpanan perangkat sensor. Selain melakukan integrasi dan implementasi sistem, hasil penelitian ini dilengkapi dengan visualisasi menggunakan Elasticsearch, Logstash, dan Kibana (ELK) stack.

Penggunaan Raspberry Pi 4 Model B pada penelitian ini juga merupakan pengembangan dari penelitian yang dilakukan oleh Kyaw et al [5] yang mengalami kendala menggunakan Raspberry Pi 2 yang tidak dapat melakukan *forwarding* untuk seluruh *traffic* data yang masuk karena keterbatasan RAM, kemampuan pemrosesan dan kecepatan pembacaan atau penulisan data.

II. TINJAUAN PUSTAKA

2.1. Intrusion Detection System (IDS)

*M*erupakan teknologi pemantauan proaktif dan mekanisme pertahanan untuk melindungi infrastruktur teknologi informasi penting dari tindak kejahatan, yang dapat membahayakan data sensitif dan aplikasi penting melalui serangan siber [16]. Terdapat dua jenis IDS [7], yaitu:

- *Host-based* IDS (HIDS): IDS yang digunakan pada mesin host dan memindai lalu lintas mesin host untuk menangkap paket berbahaya dan mencatat serangan.
- *Network-based* IDS (NIDS): IDS yang digunakan pada jaringan fisik dan intersepsi semua lalu lintas yang dikirim ke jaringan untuk mendeteksi paket abnormal dan serangan log.

Pendeteksian aktivitas jahat pada IDS dapat menggunakan dua metode, yaitu *Behaviour-based* mendeteksi berbasis anomali atau berbasis perilaku dengan memantau jaringan, *host*, dan pengguna dari waktu ke waktu untuk menghasilkan data dan kemudian menghasilkan peringatan ketika lalu lintas yang tidak biasa atau tidak normal [7]. *Misuse-based* atau *Signature-based* mendeteksi serangan yang diketahui yang telah ditentukan sebelumnya. Setiap serangan yang diketahui telah ditentukan sebelumnya dalam bentuk signature dan disimpan, data yang masuk dicocokkan dengan *signature* untuk menentukan serangan [17].

2.2. Snort

Salah satu perangkat lunak pendeteksi intrusi jaringan bersifat open-source yang paling banyak digunakan secara luas yang dirancang untuk menyediakan analisis waktu nyata dan mendeteksi paket data pada jaringan komunikasi [7]. Snort muncul pada tahun 1998 dan mengalami perkembangan serta penyempurnaan konstan selama lebih dari satu dekade. Snort telah menjadi perangkat lunak pendeteksi dan pencegahan intrusi jaringan di seluruh dunia. Snort dapat dengan kuat menganalisis aliran data dan protokol secara real time [18]. Snort melihat anomali dengan menyamakan prototipe anomali sebelumnya dan kemudian akan mengambil tindakan yang sesuai untuk menghindari anomali tersebut [17]. Alur kerja Snort terdiri dari enam bagian, yaitu menangkap paket data, menganalisis kode data, *pre-processing* paket, menguraikan *rule*, *detection engine*, dan *logging* [18].

2.3. Cloud Storage

Solusi penyimpanan dan manajemen data dalam jumlah besar, sistem *cloud storage* dapat mengonfigurasi banyak perangkat penyimpanan sekaligus tergantung penggunaan. Penerapan melalui fungsi *cluster*, *grid* atau sistem file terdistribusi, jaringan berbagai jenis perangkat penyimpanan menggunakan perangkat lunak untuk bekerja sama, penyimpanan data eksternal dan fungsi akses dari sistem bisnis [19]. Perluasan skala data yang berkelanjutan, arsitektur sistem *cloud storage* menjadi semakin kompleks. Para peneliti sedang mengembangkan cara untuk meningkatkan kinerja dari arsitektur yang berkembang, termasuk menggunakan distribusi data dan metode migrasi data untuk menyeimbangkan beban, menggunakan pemilihan node layanan dan metode alokasi sumber daya untuk meningkatkan ketersediaan sistem [20].

2.4. Virtual Private Network (VPN)

Jaringan pribadi yang membentang di jaringan publik atau internet. Ini memungkinkan pengguna untuk mengirim dan menerima data melalui jaringan bersama atau publik seolah-olah perangkat komputasi mereka terhubung langsung ke jaringan pribadi [21]. VPN dapat dianggap sebagai *tunnel* yang diautentikasi dan dienkripsi untuk berfungsi sebagai tunnel yang terhubung secara virtual melalui infrastruktur publik bersama [22]. VPN dapat dibuat dengan menghubungkan kantor dan pengguna tunggal (termasuk pengguna seluler) ke penyedia layanan terdekat POP (*Point of Presence*) dan menggunakan jaringan *backbone* penyedia layanan tersebut, atau bahkan internet, sebagai saluran antar kantor [21].

Meskipun penggunakan VPN membebani lalu lintas data termasuk lapisan lain, menyebabkan efek pada *throughput, latency*, *frame loss rate*, di antara parameter lainnya, mempengaruhi *Quality of Service* (QoS) jaringan. Besaran konsumsi CPU jaringan nirkabel oleh VPN tergantung pada protokol yang digunakan dan algoritma enkripsi [23]. Penggunaan VPN pada penelitian ini sebagai sarana pengamanan transmisi data dan koneksi antara administrator, *cloud storage* dan sensor IDS serta pembatasan akses terhadap *cloud storage*. Aplikasi yang digunakan sebagai penyedia layanan adalah OpenVPN dengan *cloud storage* dan sensor IDS sebagai klien VPN.

2.5. Throughput

Throughput merupakan kecepatan pengiriman data pada periode waktu tertentu. Hal ini dipengaruhi oleh faktor seperti hilangnya paket atau transmisi ulang, penggunaan *shared media, transport layer protocol,* keterbatasan perangkat keras, sinyal ke radio dan sebagainya [23]. *Throughput* dapat dihitung dengan membagi jumlah data yang dikirim dibagi lamanya seluruh data yang dikirimkan dengan standar kualitas *throughput* pada Tabel 1 [24].

Tabel 1. Standar Kualitas *Throughput*

Kategori	Nilai
Buruk	0 – 338 Kbps
Jelek	338 – 700 Kbps
Sedang	700 – 1200 Kbps
Bagus	1200 Kbps -2.1 Mbps
Sangat Bagus	>2.1 Mbps

2.6. Latency

Didefinisikan sebagai interval waktu bit terakhir dari *frame* yang masuk mencapai *port input* di awal ketika bit pertama dari *frame* yang sama terlihat di *port output* di akhir. *Latency* dianggap sebagai penundaan (delay) antara dikirim dari informasi, dari pengirim, dan dekripsi di penerima [23]. *Latency* dihitung dengan membagi jumlah data yang dikirim dibagi lamanya seluruh data yang dikirimkan dengan standar kualitas *latency* pada Tabel 2 [25].

Tabel 2. Standar Kualitas *Latency*

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

Kategori	Nilai
Sangat Bagus	< 150 ms
Bagus	150 - 300 ms
Sedang	300 - 450 ms
Buruk	>450 ms
Kategori	Nilai

2.7. Frame Loss Rate (FLR)

Persentase *frame* yang hilang antara *interface* pengirim dan penerima. Kerugian ini disebabkan oleh kemacetan jaringan. Persentase FLR yang lebih tinggi menyebabkan dampak negatif pada *throughput* dan *latency*, yang menyebabkan kecepatan koneksi yang lebih rendah [23]. FLR dapat dihitung dengan membagi jumlah data yang dikirim dibagi lamanya seluruh data yang dikirimkan dengan standar kualitas FLR pada Tabel 3 Standar Kualitas *Frame Loss Rate* [25].

Tabel 3. Standar Kualitas Frame Loss Rate

Kategori	Nilai
Sangat Bagus	0%
Bagus	3%
Sedang	15%
Buruk	25%

2.8. Serangan keamanan jaringan

Serangan keamanan jaringan dapat diklasifikasikan sebagai serangan pasif dan serangan aktif [26], yaitu:

- Serangan pasif berupa mempelajari atau menggunakan informasi dari sistem tetapi tidak memengaruhi sumber daya sistem. Serangan pasif bersifat menguping, atau memantau pada proses transmisi. Tujuan penyerang adalah untuk mendapatkan informasi yang sedang ditransmisikan [26].
- Serangan aktif melibatkan beberapa modifikasi pada aliran data atau pembuatan aliran palsu. Terdapat beberapa contoh dari serangan aktif, yaitu:

SYN *Flooding* merupakan salah satu bentuk serangan *Denial-of-Service* (DoS) di mana penyerang mengirimkan permintaan SYN berturut-turut ke sistem target. Serangan logika yang mengeksploitasi kelemahan perangkat lunak yang ada untuk menyebabkan *remote* server berhenti, dan serangan *flooding* yang menyerang CPU, memori, atau sumber daya jaringan korban dengan mengirimkan sejumlah besar permintaan palsu. Serangan logika dapat dicegah dengan memperbarui perangkat lunak atau memfilter urutan paket tertentu [27]. Terdapat dua hal penting yang harus diperhatikan dalam melakukan serangan SYN-*flood* yaitu *Barrage Size* dan *Barrage Frequency*. Ukuran rentetan (*Barrage Size*) berarti berapa banyak paket SYN yang akan dikirimkan ke korban. Biasanya ukuran atau jumlah SYN yang akan dikirim diukur menurut antrian *backlog* (ruang memori yang dialokasikan untuk koneksi masuk). Jadi, jumlah paket SYN harus lebih besar atau sama dengan ukuran antrian *back-log* untuk menggunakan semua memori yang dicadangkan untuk koneksi masuk. Frekuensi rentetan (*Barrage Frequency*) frekuensi request yang dikirimkan sebagai serangan. Serangan SYN-*flood* bertujuan untuk menghabiskan sumber daya host akhir tetapi tidak memenuhi *bandwidth* [28].

ARP *spoofing* merupakan serangan yang mengeksploitasi kerentanan protokol ARP yang menerjemahkan alamat logis ke alamat fisik perangkat. Serangan ini memalsukan permintaan ARP atau balasan ARP palsu. Biasanya, penyerang memalsukan alamat MAC *gateway*. Penyerang meyakinkan korban untuk mengirim frame yang ditujukan untuk gateway ke alamat lain sebagai gantinya [29]. Pada penelitian ARP *spoofing* bertujuan untuk mengelabui target untuk mengirimkan *frame* yang ditujukan ke *gateway* menuju *Media Access Control* (MAC) address sesuai dengan yang diinginkan oleh penyerang. Uji coba serangan ini dilakukan menggunakan *tool* Bettercap. Hal ini dilakukan dengan berpura-pura menjadi target dalam hal ini perangkat *wireless* untuk mendapatkan paket data yang seharusnya dikirimkan oleh *gateway* ke perangkat *wireless*.

Port scanning adalah fase dalam pembuatan *footprinting* dan pemindaian. Pemindaian *port* bertujuan untuk menemukan *port* yang terbuka dalam suatu sistem. *Port* terbuka ini dimanfaatkan oleh penyerang untuk melakukan serangan dan eksploitasi. Pemindai *port* umumnya mengirim sejumlah besar paket dengan *flag* tertentu yang diatur ke sejumlah besar *port* pada mesin target sehinga semakin membedakan aktivitas *online* umum dari pemindaian *port* [30]. Pada penelitian ini, uji coba serangan dilakukan menggunakan *tool* Nmap untuk melakukan *fingerprinting* pada perangkat *wireless*.

Malware adalah perangkat lunak berbahaya, perangkat lunak ini bisa digunakan untuk mengganggu pengoperasian komputer, mengumpulkan informasi sensitif, atau mendapatkan akses ke sistem komputer. Bentuk *Malware* ini dapat muncul dalam

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

bentuk kode dieksekusi (exe), script, konten aktif, dan perangkat lunak lainnya. *'Malware'* adalah istilah umum yang digunakan untuk merujuk kepada berbagai bentuk perangkat lunak yang bersikap bermusuhan atau mengganggu [31]. Salah satu jenis dari malware adalah *Trojan Backdoor*. *Malware* memungkinkan *hacker* untuk mengakses secara *remote* terhadap komputer yang telah terinfeksi. Penyerang setelah itu dapat melakukan berbagai tindakan pada komputer yang terkena, dari mulai mencuri informasi sampai menggunakan komputer untuk mengirimkan SPAM [31]. Pada penelitian ini akan dilakukan pengujian serangan *malware-backdoor* menggunakan tool Metasploit Framework dan Msfvenom. Hal ini dilakukan dengan mengeksekusi *malware* tersebut yang telah ditanamkan pada perangkat *wireless* dengan tujuan untuk membuat koneksi dengan *server command and control* dari *malware* tersebut.

2.9. Raspberry Pi

Perangkat komputer yang berfungsi penuh dalam paket kecil dan murah. Raspberry Pi dikenal sebagai komputer *single-board*, yang artinya persis seperti desktop, laptop, atau *smartphone*, tetapi dibangun di atas single printed circuit board. System-on-chip Raspberry Pi 4 Model B berada dalam Broadcom BCM2711B0 yang memiliki empat inti 64-bit ARM Cortex-A72 central processing unit (CPU), masing-masing berjalan pada 1,5GHz atau 1,8GHz dengan unit pemrosesan grafis (GPU) Broadcom VideoCore VI. System-on-chip terhubung 4GB LPDDR4 (Low-Power Double-Data-Rate 4) RAM yang berjalan pada 3200MHz. Memori ini dibagi antara prosesor pusat dan prosesor grafis. Slot kartu microSD mendukung penyimpanan hingga 512GB. Port Ethernet mendukung koneksi gigabit (1000Mbps, 1000-Base-T), sedangkan radio mendukung jaringan WiFi 802.11ac yang berjalan pada pita frekuensi 2.4GHz dan 5GHz, Bluetooth 5.0, dan koneksi Bluetooth Low Energy (BLE) [32]. Pada penelitian ini, Raspberry Pi 4 Model B digunakan sebagai media implementasi sensor NIDS berupa Snort. Konektivitas perangkat tersebut terhubung langsung menggunakan kabel LAN dengan router dan klien VPN menggunakan sertifikat VPN yang di-generate oleh *cloud storage* selaku VPN server.

2.10. Penelitian Terkait

a. *Integration of end-user Cloud storage for CMS analysis*

Penelitian dari Riahi dkk [33] melakukan integrasi CERNBox dengan CERN IT dengan CERN AuthN dan AuthZ service dan keseluruha sistem diintegrasikan dalam grid computing. Konsep prototyping yang diterapkan sejalan dengan penelitian yang dilakukan dengan objjek dan fungsi yang berbeda.

b. *A Secure IoT Data Integration in Cloud Storage Systems using ABAC Access Control Policy*

Peneltian dari ismail chahid dkk [34] melekukan pendekata integrasi perangkat IOT dengan cloud menggunaka policy ABAC. Pengusulan arsitektur ini mampu memberikan gambaran tentang manfaat lebih dari teknologi cloud dengan tidak lupa menambahkan fitur keamanannya.

c. *ViewBox: Integrating Local File Systems with Cloud Storage Services*

Penelitian dari yupu zhang dkk [35] melakukan eksplorasi lebih berkenaan dengan failover dengan menambahkan cloud sebagai penyimpanan backup dengan melakukan sinkronisasi secara terintegrasi dengan penyimpanan file lokal

d. *Integration of Heterogeneous Cloud storages through an Intermediate WCF Service*

Penelitian dari Parvider Kaur dan Manish Mahajan [36] melakukan integrasi beberapa layanan cloud untuk mendapatkan state of the art dan komparasi sehingga memberikan gambaran berkenaan dengan dinamika dalam integrasi dengan sistem cloud.

Keempat penelitian terkait diatas mempunyai kesamaan dalam memberikan gambaran manfaat layanan cloud. Penelitian ini mempunyai novelty pada penerapan dalam Intrution Detection System yang belum pernah diterapkan sebelumnya dan divalidasi dengan pengujian akurasi.

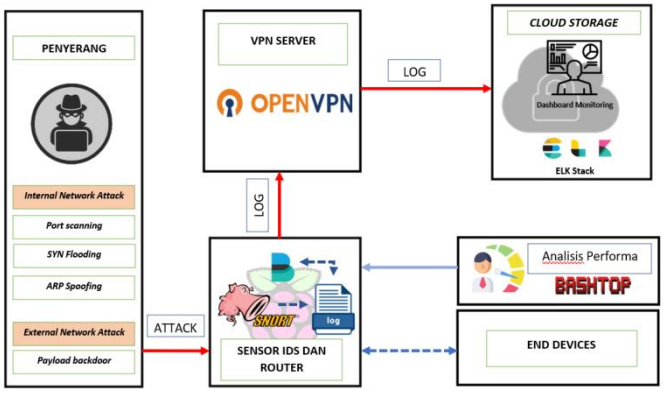
III. METODE PENELITIAN

Objek yang akan diteliti adalah perangkat keamanan jaringan berbasis Raspberry Pi 4 Model B yang diintegrasikan dengan Snort IDS untuk identifikasi serangan dan VPN untuk melakukan koneksi pengiriman log yang aman ke *cloud storage*. *Cloud storage* yang digunakan berupa Virtual Private Server (VPS) yang diimplementasikan pada server Proxmox dengan spesifikasi sistem operasi Linux Ubuntu 20.04, Random Access Memory

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

(RAM) 4 GB, 6 Core CPU, dan 100 GB memori penyimpanan. Data didapatkan dari hasil analisis akurasi perangkat IDS dan performa perangkat pada saat dijalankan, dan visualisasi menggunakan ELK stack pada *cloud storage*. Pengukuran akurasi perangkat IDS dilakukan dengan pengujian serangan dan analisis statik berdasarkan *alert* yang dimunculkan oleh sensor IDS dan *alert* yang ditampilkan pada *cloud storage*. Pengukuran performa perangkat dilakukan dengan pengukuran penggunaan memori dan CPU.

Sistem yang hendak dibangun berdasarkan hasil analisis kebutuhan direpresentasikan pada gambaran umum sistem pada Gambar 1. Terdapat tiga machine yang memiliki perannya masing – masing yaitu, Raspberry Pi sebagai sensor IDS, *Cloud storage*, dan penyerang. Dilakukan juga empat skenario serangan untuk mengetahui kemampuan dari perangkat IDS berupa *Port scanning*, SYN *flooding*, *Payload backdoor* [37], dan ARP *Spoofing* [38]. Serangan tersebut mengarah pada *end user device*. Sensor IDS akan melakukan sniffing pada jaringan data untuk mendeteksi adanya penyimpangan kondisi paket data atau anomali yang melewati jaringan berdasarkan *signature* yang telah ditetapkan sebelumnya. Hasil dari pendeteksian hal tersebut berupa *log* (IP address penyerang dan target, jenis serangan, dan waktu). Tingkat akurasi deteksi anomali dinilai berdasarkan banyaknya *packet loss* yang terjadi pada proses transmisi data *log alert* dari perangkat sensor IDS menuju *cloud storage* yang bertindak sebagai ELK *server* untuk melakukan visualisasi *log alert*. Pengukuran perfoma dilakukan dengan analisis menggunakan *tool* 'bashtop' pada perangkat IDS ketika dilakukan uji coba serangan SYN *Flooding*.



Gambar 1. Gambaran Umum Sistem

IV. HASIL DAN PEMBAHASAN

Berdasarkan statistik pengiriman data dari pengujian empat metode serangan yang berbeda dapat dianalisis nilai *throughtput, latency,* dan *frame rate loss* dari sensor IDS menuju ELK server melalui jalur VPN seperti yang dapat dilihat pada Tabel 4, 5, dan 6.

Tabel 4. *Throughtput* Pengiriman *Alert*

Serangan	Throughput (*Mbps*)	Kualitas
Port Scanning	11.449	Sangat Bagus
ARP *Spoofing*	1.97535	Bagus
SYN *Flooding*	4.13516	Sangat Bagus
Malware Backdoor	2.63636	Sangat Bagus
Rata - Rata	5.0489675	Sangat Bagus

Nilai *throughtput* bergantung pada lamanya pengamatan paket yang masuk dan jumlah data yang diterima. Sehingga, semakin lama waktu pengamatan paket dan jumlah data yang melewati jaringan sedikit maka nilai throughput dari pengiriman data tersebut kecil. Hal ini terjadi pada serangan ARP *spoofing* dengan *alert* yang dihasilkan berdasarkan inkonsistensi dari alamat MAC dan IP yang telah dideklarasi pada konfigurasi Snort.

Tabel 5. *Latency* Pengiriman *Alert*

Serangan	Latency (*ms*)	Kualitas
Port Scanning	0.705225	Sangat Bagus
ARP *Spoofing*	165.2175	Bagus
SYN *Flooding*	1.99557	Sangat Bagus
Malware Backdoor	3.50466	Sangat Bagus
Rata - Rata	42.855738	Sangat Bagus

Nilai *latency* dapat dihasilkan berdasarkan pembagian dari durasi waktu dengan banyaknya paket yang tercatat pada proses sniffing. Sebaliknya, nilai latency paling kecil didapatkan dari hasil pengujian serangan ARP spoofing. Hal ini disebabkan sedikitnya *alert* yang dihasilkan bergantung interval dari paket ARP request interval berdasarkan pengaturan host yang digunakan. Selain itu, data *alert* log ditransmisikan memiliki rentang waktu antar paketnya lebih lama dibanding dengan pengujian serangan port scanning.

Tabel 6. *Frame Rate Loss* Pengiriman *Alert*

Serangan	FLR (%)	Kualitas
Port Scanning	0.00330	Bagus
ARP *Spoofing*	0.00114	Bagus
SYN *Flooding*	0.00154	Bagus
Malware Backdoor	0.04060	Bagus
Rata - Rata	0.011645	Bagus

Nilai *frame rate loss* dapat dibandingkan dengan standar yang digunakan sehingga dapat disimpulkan bahwa kualitas dari *frame loss rate* pengiriman data memiliki kualitas *"*Bagus*"*. Berdasarkan formula perhitungan nilai, *frame loss rate* bergantung pada selisih

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

dari paket yang dikirimkan dan diterima antara dua *host* serta jumlah total paket yang dikirimkan.

Berdasarkan hasil pengujian dari empat serangan yang berbeda dapat dianalisis data peningkatan penggunaan CPU dan *memory* dari sensor IDS dengan membandingkannya dengan kondisi *idle* pada Tabel 7.

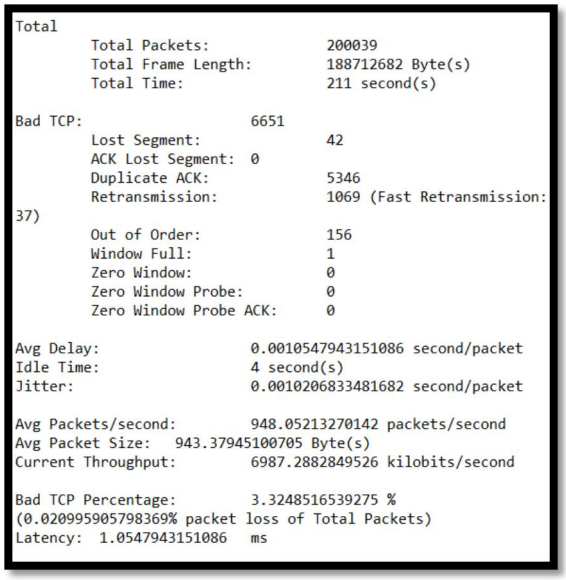
Tabel 7. Perbandingan Kinerja Sensor IDS

Kondisi	CPU	*Memory*
Idle	9%	574 Mb
Port Scanning	38%	907 Mb
ARP *Spoofing*	22%	809 Mb
SYN *Flooding*	52%	890 Mb
Malware Backdoor	18%	834 Mb

Tabel 7 menjelaskan bahwa sensor IDS mengalami peningkatan penggunaan CPU tertinggi pada uji coba serangan SYN *flooding*. Peningkatan penggunaan memori tertinggi pada uji coba serangan *port scanning*. Dalam kedua kondisi tersebut sensor masih dapat berfungsi dengan baik sebagai *hotspot* dan IDS dengan tetap mendeteksi adanya uji coba serangan serta mengirimkan *alert* pada ELK *server*.

SYN *flooding* merupakan serangan yang paling banyak menggunakan sumber daya dari perangkat sensor IDS. Hal ini dikarenakan perangkat tersebut bertugas sebagai *hotspot* untuk melakukan *packet forwarding* perlu meneruskan paket data dengan frekuensi yang tinggi. Sehingga, *alert* yang dihasilkan berbanding lurus dengan frekuensi anomali yang dideteksi sebagai serangan SYN *flooding* dan ukuran *file log alert*. Perangkat sensor juga mengirimnya menuju *cloud storage* dan ELK *server*.

Cloud storage yang diterapkan menggunakan Next*cloud* dengan tujuan untuk memudahkan *file sharing* berupa *file log.* Pengirimkan data *file log* yang tersimpan pada *cloud storage* dengan ukuran sebesar 188.712.682 *bytes* dengan durasi pengiriman 211 detik dengan jumlah paket sebanyak 200.039 dapat dilihat pada Gambar 2.



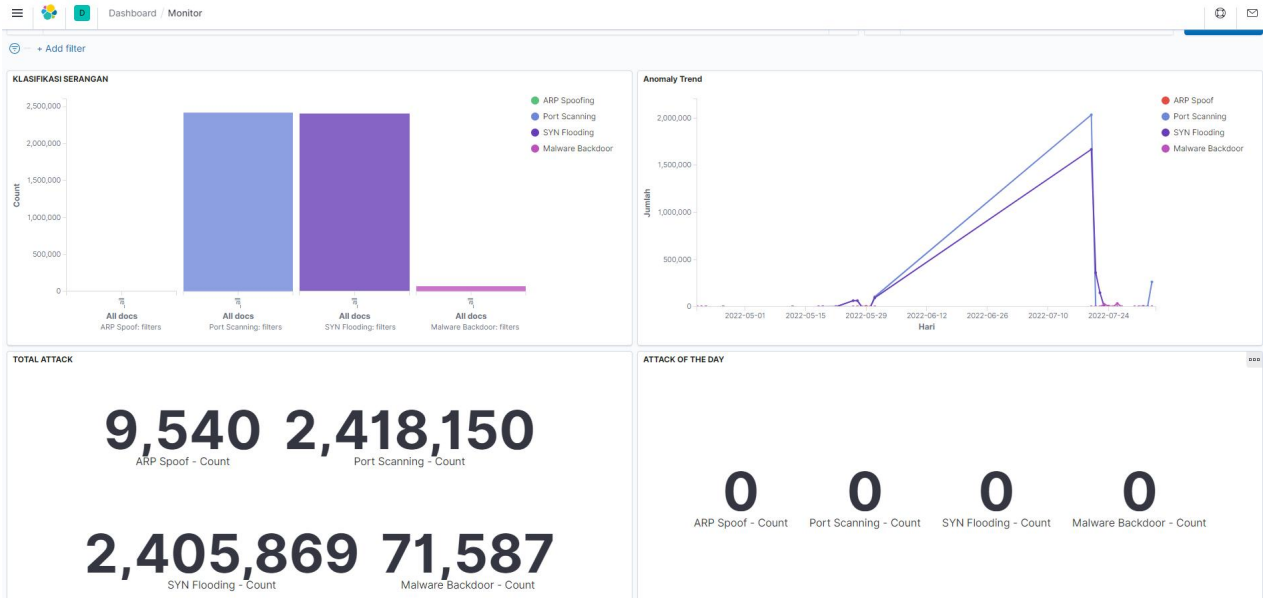
Gambar 2. Statistik Transfer File

Hasil yang didapatkan dari proses pengunduhan *file log* tersebut memiliki nilai *throughput* 6.8235 yang diklasifikasikan "Sangat Bagus", *latency* sebesar 1.054 ms yang diklasifikasikan "Sangat Bagus", dan FLR sebesar 0.02099% yang diklasifikan "Bagus". Nilai tersebut berbanding lurus dari hasil pengujian pengiriman data *file log alert* dan *log alert* dengan melakukan uji coba serangan pada jaringan. Dengan adanya FLR pada transmisi data maka diperlu penyocokan nilai *hash* dari *file log alert* yang tersimpan pada *cloud storage* dengan yang diunduh. Maka didapatkan nilai *hash* yang sama yaitu c378f5c77ee14d92bd4621dd32e4d846 hingga dapat dikatakan tidak ada data yang hilang selama pengiriman *file log alert* yang ditransmisikan.

Visualisasi hasil pencatatan *log alert* berbentuk *dashboard monitor* dengan tujuan untuk mengetahui banyaknya anomali yang dapat dideteksi dan memudahkan analisis terhadap tren serangan yang terjadi pada jaringan. Sehingga pengguna dapat meminimalisir dampak serangan siber dengan melakukan mitigasi berdasarkan tren serangan yang sedang ataupun pernah terjadi sebagai bentuk pengembangan keamanan jaringan. Pada penelitian ini, *dashboard monitor* dibentuk menjadi tiga visualisasi berupa perbandingan jumlah anomali dari tiap serangan yang diuji cobakan kedalam diagram seperti yang ditampilkan pada Gambar 4.

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

Gambar 3. Perbandingan Nilai *Hash File Log Alert* pada *Cloud* dan Diunduh



Gambar 4. Visualilasi *Dashboard Monitor* dari *Alert* IDS

V. PENUTUP

Hasil implementasi *cloud storage* sebagai media penyimpanan *log* sensor IDS berupa pengidentifikasian *file log* yang unik berdasarkan tanggal ataupun sesi dapat membantu mengurangi penggunaan *resource* berupa memori penyimpanan dari perangkat sensor IDS. Penghapusan *log* yang tersimpan pada sensor IDS dapat dilakukan secara praktis dan otomatis menggunakan sistem penjadwalan. pengukuran akurasi dari sistem bernilai "Baik" berdasarkan rata – rata nilai *frame rate loss* dari seluruh pengujian serangan yang dilakukan yaitu 0.01164%.

DAFTAR PUSTAKA

[1] M. Hijji and G. Alam, "A Multivocal Literature Review on Growing Social Engineering Based Cyber-Attacks/Threats during the COVID-19 Pandemic: Challenges and Prospective Solutions," *IEEE Access*, vol. 9, pp. 7152–7169, 2021, doi: 10.1109/ACCESS.2020.3048839.

[2] Kominfo, "Penggunaan Internet Naik 40% Akibat Physical Distancing," *Kementerian Komunikasi dan Informatika Republik Indonesia*. pp. 1–2, 2020, [Online]. Available: file:///D:/Fahdel/Kuliah/Bismillah Tugas Akhir/Latar belakang/Penggunaan Internet Naik 40%25 akibat Physical Distancing – Ditjen Aptika.html.

[3] A. B. M. Kamrul Riad, H. Shahriar, M. Valero, and M. Hossain, "Cybersecurity risks and mitigation techniques during covid-19," *Proc. - 2021 IEEE 45th Annu. Comput. Software, Appl. Conf. COMPSAC 2021*, pp. 1351–1356, 2021, doi: 10.1109/COMPSAC51774.2021.00190.

[4] TrendMicro, "A Look Into the Most Noteworthy Home Network Security Threats of 2017," 2017. https://www.trendmicro.com/vinfo/us/secu

rity/research-and-analysis/threat-reports/roundup/a-look-into-the-most-noteworthy-home-network-security-threats-of-2017.

[5] A. K. Kyaw, Y. Chen, and J. Joseph, "Pi-IDS: Evaluation of open-source intrusion detection systems on Raspberry Pi 2," *2015 2nd Int. Conf. Inf. Secur. Cyber Forensics, InfoSec 2015*, pp. 165–170, 2016, doi: 10.1109/InfoSec.2015.7435523.

[6] N. Oliveira, I. Praça, E. Maia, and O. Sousa, "Intelligent cyber attack detection and classification for network-based intrusion detection systems," *Appl. Sci.*, vol. 11, no. 4, pp. 1–21, 2021, doi: 10.3390/app11041674.

[7] S. Tripathi and R. Kumar, "Raspberry Pi as an Intrusion Detection System, a Honeypot and a Packet Analyzer," *Proc. Int. Conf. Comput. Tech. Electron. Mech. Syst. CTEMS 2018*, pp. 80–85, 2018, doi: 10.1109/CTEMS.2018.8769135.

[8] "How to Install Snort NIDS on Ubuntu Linux." [Online]. Available: https://blog.rapid7.com/2017/01/11/how-to-install-snort-nids-on-ubuntu-linux/.

[9] J. W. Jolles, "Broad-scale applications of the Raspberry Pi: A review and guide for biologists," *Methods Ecol. Evol.*, vol. 12, no. 9, pp. 1562–1579, 2021, doi: 10.1111/2041-210X.13652.

[10] R. Pi, "Raspberry Pi 4 Model B specifications – Raspberry Pi," *Raspberry Pi Foundation*. p. 5, 2020, [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/.

[11] "Raspberry Pi Industrial Reliability for 24/7," 2018. https://forums.raspberrypi.com/viewtopic.php?t=211764.

[12] M. S. Kavitha and P. Damodharan, "Software as a Service in Cloud Computing," *Int. J. Recent Adv. Eng. Technol.*, vol. 08, no. 04, pp. 1–4, 2020, doi: 10.46564/ijraet.2020.v08i04.001.

[13] F. Khalil-Ur-Rehman, "Raspberry Pi Personal Cloud Storage," pp. 1–67, 2015, [Online]. Available: https://www.researchgate.net/publication/331975048_Raspberry_Pi_Personal_Cloud_Storage.

[14] A. C. Territory, "Guideline for Records Management Number 7 – Physical Storage of Records," no. 7, [Online]. Available: file:///D:/Fahdel/Kuliah/Bismillah Tugas Akhir/Latar belakang/Guideline-No-7-Physical-Storage-August-2008.pdf.

[15] L. O. Akingbade, "Cloud Storage problems , benefits and solutions provided by Data De-duplication," vol. 5, no. 6, pp. 70–77, 2016, [Online]. Available: file:///D:/Fahdel/Kuliah/Bismillah Tugas Akhir/Referens/Cloud Storage problems, benefits and solutions provided by Data De-duplication..pdf.

[16] N. A. Premathilaka, A. C. Aponso, and N. Krishnarajah, "Review on state of art intrusion detection systems designed for the cloud computing paradigm," *Proc. - Int. Carnahan Conf. Secur. Technol.*, pp. 1–6, 2013, doi: 10.1109/CCST.2013.6922049.

[17] A. Garg and P. Maheshwari, "Performance analysis of Snort-based Intrusion Detection System," *ICACCS 2016 - 3rd Int. Conf. Adv. Comput. Commun. Syst. Bringing to Table, Futur. Technol. from Arround Globe*, pp. 0–4, 2016, doi: 10.1109/ICACCS.2016.7586351.

[18] Z. Zhou, Z. Chen, T. Zhou, and X. Guan, "The study on network intrusion detection system of snort," *2010 Int. Conf. Netw. Digit. Soc. ICNDS 2010*, vol. 2, pp. 194–196, 2010, doi: 10.1109/ICNDS.2010.5479341.

[19] Q. He, Z. Li, and X. Zhang, "Analysis of the key technology on cloud storage," *2010 Int. Conf. Futur. Inf. Technol. Manag. Eng. FITME 2010*, vol. 1, pp. 426–429, 2010, doi: 10.1109/FITME.2010.5656540.

[20] W. Ke, Y. Wang, and M. Ye, "GRSA: Service-aware flow scheduling for cloud storage datacenter networks," *China Commun.*, vol. 17, no. 6, pp. 164–179, 2020, doi: 10.23919/JCC.2020.06.014.

[21] C. M. M. T. Fancy, "An Evaluation of Alternative Protocols-Based Virtual Private LAN Service (VPLS)," *MPLS-Enabled Appl.*, pp. 373–420, 2011, doi: 10.1002/9780470976173.ch13.

[22] A. Alshalan, S. Pisharody, and D. Huang, "A Survey of Mobile VPN Technologies," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1177–1196, 2016, doi:

10.1109/COMST.2015.2496624.

[23] L. Caldas-Calle, J. Jara, M. Huerta, and P. Gallegos, “QoS evaluation of VPN in a Raspberry Pi devices over wireless network,” *2017 Int. Caribb. Conf. Devices, Circuits Syst. ICCDCS 2017*, pp. 125–128, 2017, doi: 10.1109/ICCDCS.2017.7959718.

[24] M. Ridwan, E. Safrianti, and L. Oktaviana, “Perancangan dan Analisis Performansi Private Cloud Computing untuk Penyimpanan Data di SMPN 1 Karimun,” *Jom FTEKNIK Vol.*, vol. 6, 2019, [Online]. Available: file:///D:/Fahdel/Kuliah/Bismillah Tugas Akhir/Referens/Perancangan dan Analisis Performansi Private Cloud Computing untuk Penyimpanan.pdf.

[25] A. Charisma, A. D. Setiawan, G. Megiyanto Rahmatullah, and M. R. Hidayat, “Analysis Quality of Service (QoS) on 4G Telkomsel Networks in Soreang,” *TSSA 2019 - 13th Int. Conf. Telecommun. Syst. Serv. Appl. Proc.*, no. October, pp. 145–148, 2019, doi: 10.1109/TSSA48701.2019.8985489.

[26] W. Stallings, M. Bauer, and E. M. Hirsch, *COMPUTER SECURITY Second Edition*. 2013.

[27] L. Arshadi and A. H. Jahangir, “Entropy based SYN flooding detection,” *Proc. - Conf. Local Comput. Networks, LCN*, no. April, pp. 139–142, 2011, doi: 10.1109/LCN.2011.6115171.

[28] R. M. Bani-hani and Z. Al-ali, “SYN Flooding Attacks and Countermeasures : A Survey SYN Flooding Attacks and Countermeasures : A Survey,” no. April 2013, 2017, [Online]. Available: file:///D:/Fahdel/Kuliah/Bismillah Tugas Akhir/Referens/SYNFloodingAttacksandCountermeasures.pdf.

[29] M. Data, “The Defense Against ARP Spoofing Attack Using Semi-Static ARP Cache Table,” *3rd Int. Conf. Sustain. Inf. Eng. Technol. SIET 2018 - Proc.*, pp. 206–210, 2018, doi: 10.1109/SIET.2018.8693155.

[30] J. Gadge and A. A. Patil, “Port scan detection,” *Proc. 2008 16th Int. Conf. Networks, ICON 2008*, 2008, doi: 10.1109/ICON.2008.4772622.

[31] BPPT, “Panduan Penanganan Insiden Malware,” *Insid. malware*, pp. 1–39, 2018, [Online]. Available: file:///D:/Fahdel/Kuliah/Bismillah Tugas Akhir/Referens/Panduan-malware.pdf.

[32] G. Halfacree, *Raspberry Pi Beginners Guide*. 2020.

[33] H. Riahi *et al.*, “Integration of end-user Cloud storage for CMS analysis,” *Futur. Gener. Comput. Syst.*, vol. 78, pp. 1079–1082, 2018, doi: https://doi.org/10.1016/j.future.2017.04.021.

[34] I. Chahid and A. Marzouk, “A Secure IoT Data Integration in Cloud Storage Systems using ABAC Access Control Policy,” *Int. J. Adv. Eng. Res. Sci.*, vol. 4, no. 8, pp. 34–37, 2017, doi: 10.22161/ijaers.4.8.6.

[35] Y. Zhang, C. Dragga, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, “Viewbox: Integrating local file systems with cloud storage services,” *Proc. 12th USENIX Conf. File Storage Technol. FAST 2014*, pp. 119–132, 2014.

[36] P. Kaur and M. Mahajan, “Integration of Heterogeneous Cloud Storages through an Intermediate WCF Service,” *Int. J. Inf. Eng. Electron. Bus.*, vol. 7, no. 3, pp. 45–51, 2015, doi: 10.5815/ijieeb.2015.03.07.

[37] A. Hafiz, T. Kurniawan, N. A. Sivi, F. K. Ikhsan, and P. Andhika, “Analisis Celah Keamanan Jaringan Dan Server Menggunakan Snort Intrusion Detection System,” *J. Inf. dan Komput.*, vol. 8, no. 2, pp. 59–66, 2020, doi: 10.35959/jik.v8i2.185.

[38] C. Y. and J. J. K. Kyaw, “Pi-IDS: Evaluation of Open-Source Intrusion Detection Systems on Raspberry Pi 2,” *Ieee*, pp. 165–170, 2015.