JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 14 No. 2, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v14i2.9327

IMPLEMENTASI SECURITY OFFLOADING PADA IOT GATEWAY MENGGUNAKAN MQTT OVER TLS DAN FIREWALL UNTUK MENGATASI KETERBATASAN KEAMANAN PERANGKAT IOT

Miftahul Hamdi[1*], Silfia Rifka[2], Ratna Dewi[3]

[1,2,3] Jurusan Teknik Elektro, Program Studi D4 Teknik Telekomunikasi, Politeknik Negeri Padang; Kampus Politeknik Negeri Padang Limau Manis Kecamatan Pauh Kota Padang 25164 Provinsi Sumatera Barat; Fax : 075172576

Keywords:
IoT, Security Offloading, MQTT over TLS, IoT Gateway, Firewall.

Corespondent Email:
miftahulhamdi67@gmail.com

Abstrak. Perangkat *Internet of Things* (IoT) seperti ESP32 memiliki keterbatasan sumber daya yang menyebabkan implementasi mekanisme keamanan secara langsung menjadi kurang optimal. Penelitian ini mengusulkan pendekatan *security offloading* dengan memanfaatkan IoT Gateway berbasis OpenWrt untuk menangani aspek keamanan komunikasi dan jaringan. Sistem dirancang menggunakan protokol MQTT dan MQTT *over* TLS (MQTTS) untuk komunikasi data antara perangkat IoT dan gateway, serta firewall berbasis iptables untuk pengendalian akses jaringan. Pengujian dilakukan dengan mengukur *latency* komunikasi, penggunaan memori pada perangkat IoT, serta analisis keamanan melalui *packet capture* dan simulasi serangan. Hasil penelitian menunjukkan bahwa penggunaan TLS mampu melindungi data dari penyadapan dengan mengenkripsi komunikasi, meskipun memberikan tambahan *overhead* pada *latency* dan penggunaan resource. Selain itu, implementasi *firewall* menunjukkan kemampuan yang signifikan dalam memblokir koneksi tidak terotorisasi serta menangkal potensi serangan pada broker MQTT. Oleh karena itu, pendekatan security offloading melalui IoT Gateway berpotensi menjadi alternatif yang andal guna memperkuat keamanan ekosistem IoT, khususnya pada perangkat yang memiliki keterbatasan kapasitas sumber daya.

Abstract. *Internet of Things (IoT) devices such as ESP32 have limited resources, making it challenging to implement comprehensive security mechanisms directly on the device. This study proposes a security offloading approach by utilizing an IoT Gateway based on OpenWrt to handle communication and network security. The system employs MQTT and MQTT over TLS (MQTTS) protocols for data transmission between IoT devices and the gateway, along with an iptables-based firewall for access control. The evaluation is conducted by measuring communication latency, memory usage on the IoT device, and security analysis through packet capture and attack simulation. The results show that TLS effectively secures data by encrypting communication, although it introduces additional overhead in latency and resource usage. Furthermore, the firewall implementation successfully restricts unauthorized access and mitigates potential attacks on the MQTT broker. Therefore, the security offloading approach on the IoT Gateway provides an effective solution to enhance IoT system security, especially for resource-constrained devices.*

1. PENDAHULUAN

Kemajuan teknologi *Internet of Things* (IoT) telah mendorong adopsi berbagai perangkat di sejumlah sektor, mulai dari industri, pertanian, hingga lingkungan cerdas. Teknologi ini memberi kemampuan bagi perangkat untuk berkomunikasi dan berbagi data secara langsung, sehingga turut mendukung peningkatan produktivitas dan otomatisasi sistem. Namun, meningkatnya jumlah perangkat yang terhubung juga diikuti oleh meningkatnya risiko keamanan, seperti penyadapan data, serangan *man-in-the-middle*, dan akses tidak sah. Hal ini disebabkan oleh keterbatasan mekanisme keamanan pada sebagian besar perangkat IoT yang digunakan secara luas [1].

Sebagian besar perangkat IoT, seperti mikrokontroler ESP32, memiliki keterbatasan sumber daya berupa kapasitas memori, daya komputasi, dan konsumsi energi. Keterbatasan ini menyebabkan perangkat IoT tidak mampu menjalankan mekanisme keamanan yang kompleks seperti *firewall* atau sistem deteksi intrusi secara langsung. Penelitian sebelumnya menunjukkan bahwa perangkat IoT dengan sumber daya terbatas lebih mudah terekspos terhadap berbagai jenis serangan jaringan [2]. Oleh karena itu, diperlukan pendekatan alternatif yang dapat meningkatkan keamanan tanpa membebani perangkat IoT secara langsung.

Message Queuing Telemetry Transport (MQTT) merupakan salah satu protokol komunikasi yang banyak diadopsi dalam ekosistem IoT karena sifatnya yang ringan dan efisien. Namun, MQTT tidak menyediakan mekanisme keamanan secara bawaan, sehingga data yang dikirimkan berpotensi disadap atau dimodifikasi oleh pihak tidak bertanggung jawab. Untuk mengatasi hal tersebut, penggunaan *Transport Layer Security* (TLS) dapat diterapkan untuk memberikan enkripsi pada komunikasi MQTT (MQTTS). Meskipun demikian, penerapan TLS pada perangkat IoT dengan sumber daya terbatas dapat menambah beban komputasi dan mempengaruhi performa sistem [3].

Penelitian ini mengusulkan pendekatan *security offloading* pada IoT gateway sebagai solusi untuk mengatasi keterbatasan kapabilitas keamanan pada perangkat IoT. Dalam penelitian ini, gateway berbasis OpenWRT digunakan sebagai pusat pengelolaan komunikasi dan keamanan, dengan menjalankan broker MQTT (Mosquitto) yang diamankan menggunakan *Transport Layer Security* (TLS) serta *firewall* iptables untuk pengendalian akses jaringan

Penelitian ini dirancang untuk membangun dan menerapkan sistem keamanan IoT dengan pendekatan security *offloading* yang mampu menyediakan komunikasi *end-to-end* yang aman tanpa membebani perangkat IoT. Di samping itu, penelitian ini turut mengkaji sejauh mana pendekatan yang diusulkan mampu memperkuat keamanan sistem IoT pada perangkat berkapasitas sumber daya terbatas.

2. TINJAUAN PUSTAKA

2.1 *Internet of Things* (IoT)

Internet of Things (IoT) adalah sebuah paradigma teknologi yang menghubungkan objek-objek fisik ke infrastruktur internet, memungkinkan pertukaran informasi secara otomatis tanpa keterlibatan manusia secara langsung. Perangkat dalam ekosistem IoT umumnya mencakup sensor, aktuator, dan mikrokontroler yang tersambung ke jaringan guna menjalankan fungsi pengambilan, pengiriman, serta pemrosesan data secara real-time. Dengan konektivitas tersebut, pengguna dapat memantau dan mengendalikan perangkat dari jarak jauh melalui infrastruktur internet [4].

Pada umumnya, sistem IoT dibangun atas tiga elemen pokok, yakni perangkat keras yang dilengkapi kemampuan konektivitas IoT, media transmisi jaringan seperti router maupun modem, serta platform penyimpanan seperti server atau layanan *cloud* yang bertugas mengelola dan memproses informasi. Ketiga elemen ini bekerja secara sinergis untuk mewujudkan sistem IoT yang mampu memberikan layanan secara otomatis dan efisien [5].

Selain itu, IoT juga mencakup berbagai teknologi pendukung seperti RFID, protokol

komunikasi TCP/IP, serta teknologi mobile yang memungkinkan identifikasi objek, pengumpulan data, pemrosesan, dan pertukaran informasi antara lingkungan fisik dan dunia digital. Integrasi berbagai teknologi ini menjadikan IoT sebagai solusi yang fleksibel dalam berbagai bidang, termasuk industri, pertanian, dan smart environment [6].

Meskipun memberikan banyak manfaat, perangkat IoT umumnya memiliki keterbatasan sumber daya seperti kapasitas memori, daya komputasi, dan konsumsi energi. Keterbatasan tersebut menjadi salah satu tantangan utama dalam penerapan sistem keamanan, sehingga perangkat IoT rentan terhadap berbagai ancaman seperti penyadapan data dan akses tidak sah.

2.2 *Message Queuing Telemetry Transport* (MQTT)

MQTT *(Message Queuing Telemetry Transport)* adalah protokol komunikasi berbobot ringan yang banyak diadopsi dalam implementasi IoT karena hemat dalam konsumsi bandwidth, memiliki kebutuhan daya yang rendah, serta mampu beroperasi dengan latensi minimal. Selain itu, MQTT dilengkapi dengan fitur *Quality of Service* (QoS) sebagai mekanisme pengaturan tingkat keandalan transmisi data sesuai kebutuhan sistem [7].

Dalam operasionalnya, MQTT menerapkan pola komunikasi publish-subscribe yang melibatkan tiga entitas utama, yaitu publisher, subscriber, dan broker. Publisher bertugas meneruskan data ke broker, yang selanjutnya mendistribusikan data tersebut kepada subscriber berdasarkan topik yang relevan, sehingga proses komunikasi menjadi lebih fleksibel dan efisien [8].

MQTT dirancang dengan kesederhanaan yang memudahkan penerapannya pada perangkat berkapasitas terbatas, sekaligus mampu melayani banyak klien dalam satu server. Akan tetapi, protokol ini tidak memiliki lapisan perlindungan bawaan, sehingga diperlukan mekanisme keamanan tambahan seperti Transport Layer Security (TLS) guna menjaga kerahasiaan dan integritas data yang ditransmisikan [9].

2.3 *Transport Layer Security* (TLS)

Transport Layer Security (TLS) adalah protokol keamanan berbasis kriptografi yang dirancang untuk melindungi pertukaran informasi melalui jaringan komunikasi dengan memanfaatkan mekanisme enkripsi, sehingga informasi yang dikirimkan antar perangkat tidak dapat dibaca atau dieksploitasi oleh pihak yang tidak memiliki otorisasi. TLS bekerja pada lapisan transport dan menyediakan keamanan *end-to-end* dalam proses pertukaran data antar sistem [10].

TLS beroperasi di atas protokol *Transmission Control Protocol* (TCP) dan menggunakan mekanisme handshake untuk membangun koneksi yang aman. Pada proses ini, kedua pihak yang berkomunikasi melakukan autentikasi serta menyepakati metode enkripsi yang digunakan sebelum data dikirimkan [11]. Dengan mekanisme ini, TLS mampu menjaga kerahasiaan dan integritas data selama proses transmisi.

Dalam penerapannya pada *Internet of Things* (IoT), TLS digunakan untuk mengamankan komunikasi pada protokol MQTT sehingga menjadi MQTT over TLS (MQTTS). Penggunaan TLS memungkinkan data dienkripsi pada lapisan transport antara dua perangkat yang berkomunikasi secara langsung, sehingga dapat mencegah serangan seperti penyadapan data dan manipulasi informasi [12]. Namun, implementasi TLS pada perangkat dengan sumber daya terbatas dapat meningkatkan beban komputasi, sehingga diperlukan pendekatan seperti *security offloading* untuk memindahkan proses keamanan ke perangkat yang lebih mampu.

2.4 IoT Gateway

IoT gateway merupakan perangkat yang berperan sebagai jembatan komunikasi antara perangkat IoT dengan infrastruktur jaringan yang lebih luas, seperti server atau *cloud*. Gateway menerima data dari perangkat IoT, kemudian melakukan pemrosesan awal sebelum meneruskan data tersebut ke sistem lain. Selain itu, gateway juga dapat mengurangi latensi dan meningkatkan efisiensi jaringan dengan melakukan pengolahan data secara lokal sebelum dikirimkan ke server [13].

Dibandingkan dengan perangkat IoT seperti mikrokontroler, IoT gateway umumnya dibekali kapasitas sumber daya yang jauh lebih besar sehingga dapat menjalankan fungsi tambahan seperti pengolahan data, manajemen komunikasi, serta mekanisme keamanan. Oleh karena itu, gateway dapat digunakan sebagai titik sentral untuk meningkatkan keamanan

sistem IoT tanpa membebani perangkat IoT secara langsung.

2.3 *Firewall* (iptables)

Firewall adalah komponen keamanan jaringan yang bertugas memantau dan mengendalikan aliran paket data yang masuk maupun keluar sesuai dengan kebijakan yang telah ditetapkan. Mekanisme ini hanya mengizinkan lalu lintas yang terverifikasi aman untuk melewati jaringan, serta mampu memblokir upaya akses tidak terotorisasi terhadap sistem [14].

Pada sistem berbasis Linux, *firewall* dapat diimplementasikan menggunakan iptables yang berfungsi untuk memfilter paket data berdasarkan alamat IP, *port*, maupun protokol yang digunakan. Iptables bekerja dengan cara mengatur aturan untuk mengizinkan atau menolak paket data yang melewati jaringan, sehingga dapat digunakan untuk melindungi jaringan privat dari akses yang tidak diinginkan [15].

Dalam sistem *Internet of Things* (IoT), penerapan firewall pada IoT gateway menjadi penting untuk meningkatkan keamanan jaringan. Dengan memanfaatkan iptables, gateway dapat mengontrol komunikasi antara perangkat IoT dan jaringan, sehingga hanya koneksi yang valid yang diperbolehkan. Pendekatan ini mendukung konsep *security offloading* dengan memindahkan mekanisme pengamanan jaringan ke gateway, sehingga perangkat IoT tidak perlu menjalankan fungsi keamanan yang kompleks.

3. METODE PENELITIAN

3.1 Rancangan Penelitian

Studi ini menerapkan pendekatan implementasi sistem melalui mekanisme *security offloading* pada arsitektur IoT. Pendekatan tersebut dimaksudkan untuk mengalihkan proses keamanan dari perangkat IoT yang memiliki keterbatasan sumber daya ke IoT gateway yang memiliki kapasitas lebih besar..

Fokus penelitian ini adalah pada komunikasi antara perangkat IoT (ESP32) dan IoT gateway, dengan membandingkan komunikasi tanpa keamanan (MQTT) dan komunikasi aman menggunakan MQTT over TLS (MQTTS), serta penerapan firewall pada gateway untuk meningkatkan keamanan jaringan.

3.2 Arsitektur Sistem

Sistem ini dibangun atas dua elemen inti, yakni perangkat IoT dan IoT Gateway.

1. Perangkat IoT (ESP32).
   ESP32 berfungsi sebagai *publisher* yang mengirimkan data ke gateway menggunakan protokol MQTT atau MQTT over TLS (MQTTS).
2. IoT Gateway
   IoT gateway menggunakan perangkat STB HG680P dengan sistem operasi OpenWRT yang berfungsi sebagai broker MQTT (Mosquitto). Gateway juga berfungsi sebagai pusat keamanan dengan menerapkan enkripsi TLS dan firewall iptables untuk mengontrol akses jaringan.

Pada arsitektur ini, mekanisme keamanan seperti enkripsi dan pengendalian akses dipusatkan pada gateway, sehingga perangkat IoT tidak terbebani oleh proses keamanan yang kompleks.

3.3 Metode Pengambilan Data

Data dikumpulkan melalui metode pengukuran langsung terhadap parameter performa komunikasi antara perangkat IoT dan gateway. Parameter yang diamati meliputi:

1. Latensi komunikasi
   Latensi diukur menggunakan metode *round-trip time* (RTT), yaitu dengan mencatat waktu saat data dikirim dari perangkat IoT dan waktu saat respon diterima kembali dari gateway.
2. Penggunaan sumber daya perangkat IoT
   Penggunaan sumber daya diukur melalui parameter *Free heap* memori dan kualitas sinyal (RSSI)
3. Kemanan komunikasi
   Analisis keamanan dilakukan dengan mengamati perbedaan karakteristik komunikasi antara MQTT tanpa enkripsi dan MQTT dengan TLS.

3.4 Skenario Pengujian

Pengujian dilakukan pada sistem IoT yang terdiri dari perangkat ESP32 sebagai node sensor dan IoT Gateway berbasis OpenWRT yang berfungsi sebagai broker MQTT. Pengujian difokuskan pada komunikasi antara ESP32 dan gateway dalam dua skenario, yaitu tanpa enkripsi (MQTT) dan dengan enkripsi (MQTT over TLS).

ESP32 mengirimkan data berupa timestamp, nilai heap memory, dan RSSI secara periodik ke

broker. Pengukuran latensi dilakukan menggunakan metode *round-trip time* dengan mekanisme echo, di mana timestamp yang dikirim oleh ESP32 dikembalikan oleh gateway, kemudian dihitung selisih waktunya. Selain itu, dilakukan juga pengukuran penggunaan resource pada ESP32 dan analisis keamanan menggunakan *packet capture* seperti Wireshark serta *firewall* berbasis iptables pada gateway.

4. HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Latensi

Hasil pengujian latency komunikasi antara ESP32 dan IoT Gateway ditunjukkan pada

Tabel 1. Perbandingan Latensi MQTT dan MQTTS

No	MQTT (ms)	MQTTS (ms)
1	13	141
2	15	9
3	11	11
4	9	13
5	18	16
6	18	15
7	18	13
8	15	13
9	11	31
10	17	15

Hasil menunjukkan bahwa komunikasi MQTT tanpa TLS memiliki latensi yang relatif stabil pada rentang 9–18 ms. Sementara itu, komunikasi MQTT over TLS menunjukkan latency awal yang tinggi sebesar 141 ms, dan selanjutnya berada pada rentang 9–31 ms.

Latency awal yang tinggi pada komunikasi TLS disebabkan oleh proses TLS *handshake* yang melibatkan negosiasi parameter keamanan, autentikasi sertifikat, serta pertukaran kunci kriptografi. Setelah koneksi terbentuk, latensi komunikasi TLS cenderung stabil dan mendekati komunikasi tanpa enkripsi. Hal ini menunjukkan bahwa *overhead* TLS lebih dominan pada fase inisialisasi koneksi dibandingkan pada fase pertukaran data.

4.2 Hasil Pengujian *Resource* Perangkat IoT

Evaluasi *resource* dilakukan guna mengidentifikasi dampak penggunaan TLS terhadap perangkat IoT yang memiliki keterbatasan sumber daya. Hasil pengujian ditunjukkan pada**Tabel 2. Perbandingan *Resource* Esp32**

Parameter	MQTT	MQTTS
Heap Memory (byte)	237124	192840
RSSI (dBm)	-42 s/d -46	-57 s/d -59

Hasil menunjukkan bahwa penggunaan TLS menyebabkan penurunan *heap memory* sebesar sekitar 44 KB atau sekitar 18% dari total memori yang tersedia. Hal ini menunjukkan bahwa implementasi mekanisme keamanan berbasis TLS memberikan beban tambahan yang signifikan pada perangkat IoT.

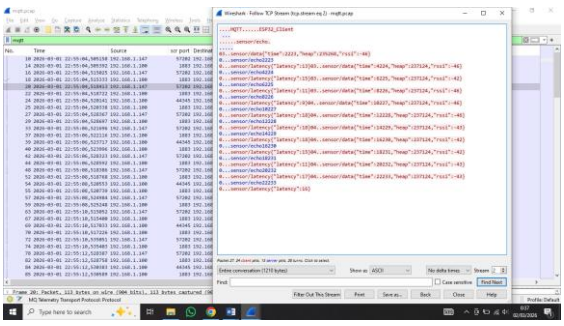
Selain itu, nilai RSSI pada pengujian TLS cenderung lebih rendah dibandingkan dengan pengujian tanpa TLS. Hal ini menunjukkan bahwa kondisi jaringan juga berpengaruh terhadap performa komunikasi, terutama terhadap variasi latensi yang terjadi.

4.3 Analisis Keamanan Komunikasi

Analisis keamanan dilakukan dengan melakukan *packet capture* pada IoT Gateway menggunakan tcpdump dan dianalisis menggunakan Wireshark.

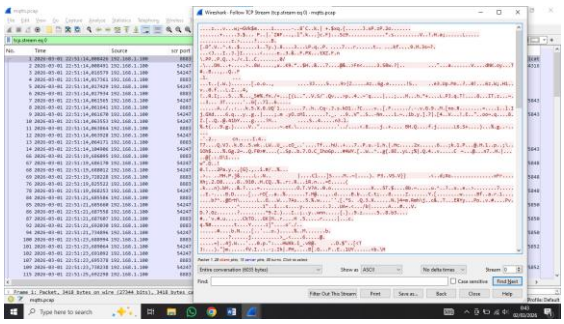
Pada komunikasi MQTT tanpa TLS, *payload* data dapat terlihat secara langsung dalam paket jaringan. Hal ini menunjukkan bahwa data yang dikirimkan tidak terlindungi dan berpotensi untuk disadap oleh pihak yang tidak berwenang.

Sebaliknya, pada komunikasi MQTT over TLS, payload data tidak dapat dibaca karena telah terenkripsi. Hal ini menunjukkan bahwa TLS mampu memberikan mekanisme keamanan berupa enkripsi data *end-to-end* antara perangkat IoT dan gateway, sehingga meningkatkan keamanan komunikasi.



Gambar 1 Hasil capture Wireshark pada MQTT

Pada Gambar 1 Hasil *capture* Wireshark Pada MQTT, terlihat bahwa data yang dikirimkan antara ESP32 dan IoT Gateway dapat terbaca secara jelas dalam bentuk teks (*plaintext*). Informasi seperti *client ID*, topik komunikasi seperti *sensor/data*, *sensor/echo*, dan *sensor/latency*, serta isi payload dapat diamati secara langsung. Payload yang dikirimkan oleh ESP32 berupa data JSON yang berisi nilai waktu (*time*), penggunaan memori (*heap*), dan kekuatan sinyal (*RSSI*), serta data latency yang dihasilkan dari proses komunikasi. Hal ini menunjukkan bahwa tidak terdapat mekanisme pengamanan pada lapisan transport, sehingga seluruh data yang dikirimkan dapat dengan mudah disadap oleh pihak lain yang berada pada jaringan yang sama.



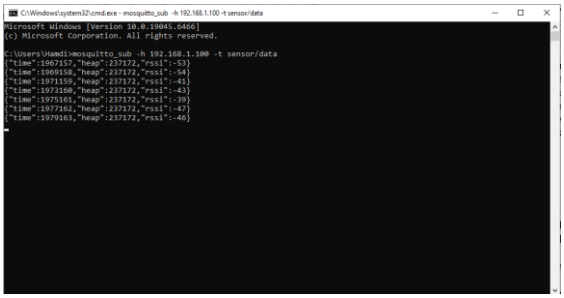
Gambar 2 Hasil Capture Wireshark pada MQTTS

Pada Gambar 2 hasil *capture* Wireshark untuk komunikasi MQTTS, terlihat bahwa data yang ditransmisikan tidak dapat dibaca secara langsung dan ditampilkan dalam bentuk karakter acak. Kondisi ini terjadi karena data telah dienkripsi menggunakan protokol Transport Layer Security (TLS). Seluruh informasi yang sebelumnya dapat terlihat secara jelas pada komunikasi MQTT, seperti topik dan payload data, menjadi tidak dapat diinterpretasikan tanpa proses dekripsi yang sesuai. Hal ini menunjukkan bahwa mekanisme TLS berhasil melindungi kerahasiaan data selama proses transmisi, sehingga pihak yang tidak memiliki sertifikat atau kunci yang valid tidak dapat mengetahui isi komunikasi.

Berdasarkan kedua hasil pengujian tersebut, dapat dianalisis bahwa penggunaan MQTT tanpa TLS memiliki tingkat kerentanan yang tinggi terhadap serangan *packet sniffing*, karena seluruh data dikirimkan dalam bentuk terbuka. Sementara itu, penggunaan MQTTS memberikan peningkatan keamanan yang signifikan dengan mengenkripsi data, sehingga mampu menjaga kerahasiaan informasi yang dikirimkan. Meskipun hasil pengukuran latency menunjukkan bahwa perbedaan antara MQTT dan MQTTS tidak terlalu signifikan dalam kondisi jaringan lokal, penerapan TLS tetap sangat penting terutama untuk sistem IoT yang beroperasi pada jaringan publik atau tidak aman. Dengan demikian, penggunaan MQTTS direkomendasikan sebagai solusi untuk meningkatkan keamanan komunikasi antara ESP32 dan IoT Gateway.

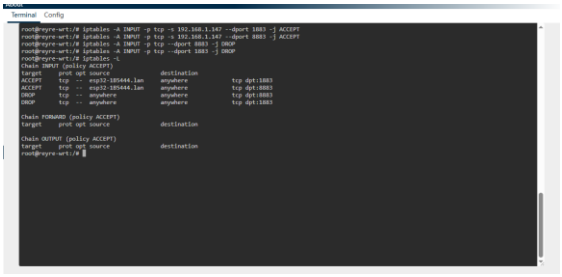
4.5 Pengujian Firewall

Pengujian keamanan pada penelitian ini dilakukan untuk mengevaluasi kemampuan firewall dalam melindungi IoT Gateway dari akses yang tidak sah serta serangan yang dapat mengganggu ketersediaan layanan. Firewall yang digunakan adalah iptables yang berjalan pada sistem operasi OpenWRT di perangkat IoT Gateway. Pengujian dilakukan dengan mengamati perubahan kondisi jaringan sebelum dan sesudah penerapan aturan *firewall*.. Parameter yang diamati adalah kemampuan akses ke broker MQTT serta stabilitas layanan ketika terjadi aktivitas komunikasi.



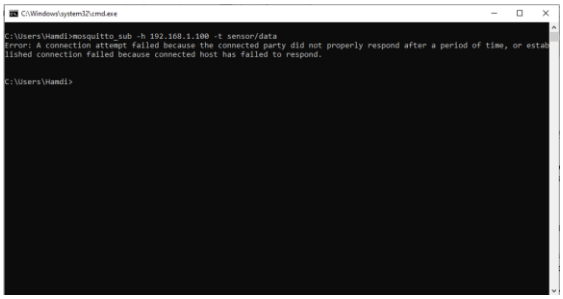
Gambar 3 Percobaan Subscribe Ke Broker Tanpa Adanya Firewall

Pada Gambar 3 Percobaan Subscribe Ke Broker Tanpa Adanya Firewall menunjukkan hasil percobaan subscribe ke broker MQTT tanpa adanya firewall, dimana terlihat bahwa data sensor dikirim secara real-time dan dapat diterima oleh client tanpa hambatan. Kondisi ini menunjukkan bahwa sistem masih bersifat terbuka dan rentan terhadap akses tidak sah maupun potensi serangan seperti flooding.



Gambar 4 Konfigurasi Firewall

Gambar 4 Konfigurasi Firewall menunjukkan konfigurasi firewall yang diterapkan pada IoT Gateway. Pada konfigurasi tersebut dilakukan pembatasan akses terhadap port MQTT sehingga tidak semua perangkat dapat melakukan koneksi secara bebas. Selain itu, firewall juga dapat digunakan untuk melakukan rate limiting guna mencegah terjadinya serangan flooding yang dapat menyebabkan penurunan performa sistem.



Gambar 5 Percobaan Subscribe Ke Broker Dengan Adanya Firewall

Gambar 5 Percobaan Subscribe ke Broker Dengan Adanya Firewall menunjukkan hasil percobaan subscribe ke broker MQTT setelah firewall diaktifkan. Terlihat bahwa data tidak lagi dapat diterima seperti pada kondisi sebelumnya, yang menunjukkan bahwa mekanisme pembatasan akses telah berfungsi sebagaimana mestinya, sehingga membuktikan bahwa firewall mampu mengontrol komunikasi yang terjadi pada jaringan dan mencegah akses yang tidak sah..

Berdasarkan hasil pengujian, dapat disimpulkan bahwa penerapan firewall pada IoT Gateway memiliki peran penting dalam meningkatkan keamanan sistem, khususnya dalam aspek pengendalian akses dan perlindungan terhadap serangan berbasis jaringan. Pada kondisi tanpa firewall, sistem cenderung terbuka dan rentan terhadap berbagai jenis serangan, sedangkan dengan penerapan firewall, akses dapat dibatasi sehingga hanya perangkat yang terotorisasi yang dapat berkomunikasi dengan broker MQTT. Selain itu, firewall juga dapat digunakan untuk mengurangi dampak serangan flooding dengan cara membatasi jumlah koneksi atau paket yang diterima. Dengan demikian, kombinasi penggunaan MQTT over TLS dan firewall memberikan mekanisme keamanan yang lebih komprehensif, dimana TLS berperan dalam menjaga kerahasiaan data, sementara firewall berfungsi untuk melindungi ketersediaan layanan dari akses yang tidak sah dan serangan jaringan.

5. KESIMPULAN

a. Implementasi konsep security offloading pada IoT Gateway terbukti dapat mengatasi keterbatasan kapabilitas keamanan pada perangkat IoT seperti ESP32, yang memiliki keterbatasan resource dalam menjalankan mekanisme keamanan secara mandiri.
b. Penggunaan MQTT *over* TLS (MQTTS) berhasil meningkatkan aspek keamanan komunikasi data dengan menyediakan enkripsi *end-to-end*, sehingga informasi yang ditransmisikan tidak dapat diinterpretasikan pihak yang tidak memiliki otorisasi, sebagaimana dibuktikan melalui analisis packet capture menggunakan Wireshark.
c. Hasil pengujian performa menunjukkan bahwa penggunaan TLS memberikan tambahan overhead pada latency dan penggunaan memori (heap) pada ESP32, namun peningkatan tersebut masih dalam batas yang dapat diterima untuk implementasi pada jaringan lokal.
d. Penerapan firewall berbasis iptables pada IoT Gateway mampu membatasi akses terhadap broker MQTT dan mencegah koneksi yang tidak sah, sehingga meningkatkan aspek keamanan pada sisi jaringan.
e. Pengujian keamanan menunjukkan bahwa tanpa firewall, broker MQTT dapat diakses secara bebas oleh perangkat lain, sedangkan dengan firewall, akses dapat dikontrol sesuai dengan aturan yang telah ditentukan, termasuk pembatasan terhadap potensi serangan seperti flooding.

f. Kombinasi antara MQTT over TLS dan firewall pada IoT Gateway memberikan mekanisme keamanan yang lebih komprehensif, dimana TLS menjaga kerahasiaan data, sedangkan firewall menjaga kontrol akses dan ketersediaan layanan.
g. Keterbatasan penelitian ini terletak pada pengujian yang hanya dilakukan pada jaringan lokal serta skenario serangan yang masih sederhana, sehingga belum mencerminkan kondisi jaringan yang lebih kompleks atau skala besar.
h. Untuk pengembangan selanjutnya, penelitian dapat diperluas dengan menguji performa pada jaringan internet yang lebih luas, menerapkan mekanisme keamanan tambahan seperti authentication berbasis sertifikat, serta mengimplementasikan sistem deteksi serangan (IDS) pada IoT Gateway.

UCAPAN TERIMA KASIH

Apresiasi setinggi-tingginya disampaikan kepada seluruh pihak yang telah memberikan kontribusi dan dukungan dalam penyelesaian penelitian ini..

DAFTAR PUSTAKA

[1] A. F. Gentile, D. Macrì, D. L. Carnì, E. Greco, and F. Lamonaca, "A Performance Analysis of Security Protocols for Distributed Measurement Systems Based on Internet of Things with Constrained Hardware and Open Source Infrastructures," *Sensors*, vol. 24, no. 9, pp. 1–22, 2024, doi: 10.3390/s24092781.

[2] D. Canavese, L. Mannella, L. Regano, and C. Basile, "Security at the Edge for Resource-Limited IoT Devices," *Sensors*, vol. 24, no. 2, pp. 1–16, 2024, doi: 10.3390/s24020590.

[3] N. O. Gavriilidis, S. T. Halkidis, and S. Petridou, "Empirical Evaluation of TLS-Enhanced MQTT on IoT Devices for V2X Use Cases," *Appl. Sci.*, vol. 15, no. 15, 2025, doi: 10.3390/app15158398.

[4] A. Maruf, "SISTEM PENDETEKSI KEKOSONGAN AIR MINUM `DI KANDANG AYAM MENGGUNAKAN INTERNET OF THINGS (IoT)," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, pp. 404–410, 2025, doi: 10.23960/jitet.v13i1.5619.

[5] Efendi dan Yoyon, "Internet of Things (IoT) Sistem Pengendalian Lampu Menggunakan Raspberrry PI Berbasis Mobile," *J. Ilm. Ilmu Komput.*, vol. 4, no. 1, pp. 19–26, 2018.

[6] C. Bersani, C. Ruggiero, R. Sacile, A. Soussi, and E. Zero, "Internet of Things Approaches for Monitoring and Control of Smart Greenhouses in Industry 4.0," *Energies*, vol. 15, no. 10, 2022, doi: 10.3390/en15103834.

[7] B. M. Susanto, E. S. J. Atmadji, and W. L. Brenkman, "Implementasi Mqtt Protocol Pada Smart Home Security Berbasis Web," *J. Inform. Polinema*, vol. 4, no. 3, pp. 201–205, 2018, doi: 10.33795/jip.v4i3.207.

[8] R. F. Pratama, R. S. R. Wicaksono, and A. N. Pramudhita, "Perancangan Dan Implementasi Protokol Mqtt Pada Sistem Parkir Cerdas Berbasis Iot," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 475–483, 2023, doi: 10.23960/jitet.v11i3.3191.

[9] G. Y. Saputra, A. D. Afrizal, F. K. R. Mahfud, F. A. Pribadi, and F. J. Pamungkas, "Penerapan Protokol MQTT Pada Teknologi Wan (Studi Kasus Sistem Parkir Univeristas Brawijaya)," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 12, no. 2, p. 69, 2017, doi: 10.30872/jim.v12i2.653.

[10] R. M. Wahyudi, "Mengimplementasikan SSL/TLS pada Web Server Apache di dalam Jaringan Internal Praktikum untuk Pengembangan Web Server," *J. Majemuk*, vol. 3, no. 1, pp. 13–31, 2024, [Online]. Available: https://jurnalilmiah.org/journal/index.php/majemuk/article/view/655

[11] C. Lesjak *et al.*, "Securing smart maintenance services: Hardware-security and TLS for MQTT," *Proceeding - 2015 IEEE Int. Conf. Ind. Informatics, INDIN 2015*, pp. 1243–1250, 2015, doi: 10.1109/INDIN.2015.7281913.

[12] A. R. Alkhafajee, A. M. Ali Al-Muqarm, A. H. Alwan, and Z. R. Mohammed, "Security and Performance Analysis of MQTT Protocol with TLS in IoT Networks," *4th Int. Iraqi Conf. Eng. Technol. Their Appl. IICETA 2021*, pp. 206–211, 2021, doi: 10.1109/IICETA51758.2021.9717495.

[13] M. K. Alami, W. Yahya, and A. Basuki, "Pengembangan IoT Gateway Terdistribusi Berbasis Kubernetes," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 2, pp. 2548–964, 2025, [Online]. Available: http://j-ptiik.ub.ac.id

[14] U. Dirgantara and M. Suryadarma, "FIREWALL DAN IPTABLES PADA JARINGAN KOMPUTER Peniarsih 1 , Iswandir 2 1," 2024.

[15] M. S. Hawari, "Penerapan Iptables Firewall Pada Linux Dengan Menggunakan Fedora," *J. Manaj. Inform.*, vol. 6, pp. 198–207, 2017.