Peningkatan Keamanan Protokol MQTT dengan Netpie sebagai Framework OAuth 1.0a

Witsqadianto Wicaksono[1], Vera Suryani[2]

[1,2]Fakultas Informatika, Universitas Telkom, Bandung
[1]witsqawicaksono@student.telkomuniversity.ac.id, [2]verasuryani@telkomuniversity.ac.id

Abstrak

Protokol MQTT merupakan protokol komunikasi yang sering digunakan untuk perangkat IoT. Protokol ini memiliki keunggulan, bandwidth yang kecil, transmisi yang ringan, dan penggunaan memori yang kecil. Tetapi disamping keunggulannya, protokol ini masih mudah terserang dengan serangan otentikasi. Penulis mengusulkan untuk meningkatkan keamanan pada protokol MQTT dengan menambahkan fitur otorisasi. Fitur otorisasi menggunakan platform Netpie yang menjadi Framework OAuth 1.0a. Skenario evaluasi yang dilakukan adalah pengujian serangan Man – in – The Middle attack dengan jenis ARP Spoofing, Eavesdropping, dan penggunaan memori pada device. Berdasarkan hasil dari pengujian, Netpie dapat menjadi solusi untuk meningkatkan keamanan MQTT pada bagian otentikasi dan otorisasi.

Kata kunci : MQTT, IoT, otentikasi, otorisasi, OAuth 1.0a

Abstract

The MQTT protocol is a communication protocol that is often used for IoT devices. This protocol has the advantages, small bandwidth, light transmission, and small memory usage. But despite its advantages, this protocol is still vulnerable to authentication attacks. The author proposes to increase the security of the MQTT protocol by adding an authorization feature. The authorization feature uses the Netpie platform which is the OAuth 1.0a Framework. The evaluation scenario carried out is testing the Man - in - The Middle attack with the type of ARP spoofing, eavesdropping, and memory usage on the device. Based on the results of testing, Netpie can be a solution to improve MQTT security in the authentication and authorization section.

Keywords : MQTT, IoT, authentication, authorization, OAuth 1.0a

1. Pendahuluan

1.1 Latar Belakang

Seiring perkembangan zaman, banyak inovasi teknologi yang telah berkembang, salah satunya Internet of Things (IoT). Internet of Things merupakan inovasi teknologi dimana sebuah benda yang berada di sekeliling kita memiliki komponen yang dapat menyimpan suatu data atau informasi mengenai lingkungannya dan berkomunikasi dengan benda lain atau manusia yang terhubung dengan internet [1]. IoT memberikan banyak manfaat dan kemudahan ketika digunakan, sehingga sekarang ini IoT telah digunakan di berbagai lingkungan. Untuk lingkungan konsumen, biasanya IoT memiliki smart sensor seperti smart watch, smart tv, health trackers, dan perangkat IoT yang lain. Sedangkan untuk lingkungan industri, biasanya IoT dikembangkan dan digunakan untuk pekerjaan yang dilakukan secara otomatis untuk diberbagai bidang seperti otomotif, medis, dan yang lainnya.

Perangkat IoT membutuhkan protokol agar dapat berkomunikasi, bertukar data dengan perangkat yang lainnya. Meskipun sekarang ini sudah ada banyak protokol yang diciptakan, tidak sembarang protokol dapat diterapkan dalam jaringan IoT. Hal ini disebabkan karena, perangkat IoT pada umumnya memiliki spesifikasi yang terbatas. Salah satu protokol yang dapat digunakan untuk jaringan IoT adalah MQTT. MQTT merupakan protokol popular yang mekanismenya berupa publish/subscribe pesan antar perangkatnya. Protokol ini memiliki

mekanisme publish/subscribe pesan antar perangkat dalam jaringannya. Kecilnya bandiwidth, transmsi yang ringan, dan penggunaan memori yang kecil merupakan alasan kenapa protokol ini sangat sering digunakan dalam jaringan IoT[2].

Penerapan IoT memang memudahkan dan membantu manusia dalam menyelesaikan suatu tugas. Tetapi ada hal yang perlu dipertimbangkan juga terkait keamanan pada jaringan IoT. Berdasarkan Book ISACA, terdapat tiga komponen mengenai keamanan informasi yaitu Data Confidentiality, Data Integrity, Data Availability. Selain itu, ada juga keamanan tingkat akses seperti authentication, authorization, dan access control [3]. Pada tahun ini banyak serangan siber yang ditargetkan ke perangkat IoT. Pada laporan RSA 2020, mesin smart cleaning menjadi target serangan remote attack, termasuk Denial of Service dan camera hacks. Selain itu, Bitdefender menemukan jenis botnet terbaru yang dijuluki Dark Nexus. Botnet ini terdiri lebih dari 1,372 bots yang tersebar di berbagai negara, seperti China, Korea Selatan, Thailand, Brasil, dan Rusia. Botnet ini mampu melakukan serangan Distributed Denial of Service (DDoS), Self-propagation, dan C&C communication.

Menurut [3], terbatasnya sumber daya perangkat, banyaknya jumlah perangkat yang terkoneksi pada suatu jaringan, dan kurangnya kesadaran masyarakat merupakan faktor jaringan IoT mudah terkena serangan siber. Protokol MQTT merupakan protokol yang memiliki banyak keunggulan untuk perangkat IoT, tetapi hal itu tidak menutupi kekurangannya dalam segi keamanan. Protokol MQTT pada umumnya menyediakan keamanan hanya berupa username dan password yang berbentuk plain text. Keamanan yang disediakan tidaklah cukup dalam membuat sebuah jaringan aman. Dengan autentikasi dasar MQTT, sebuah jaringan dapat mudah diserang dengan sniffing attack. Setelah penyerang berhasil melakukan sniffing attack, penyerang dapat melakukan serangan yang lain seperti seperti mengganti topik paket dari publisher, denial of service, man – in – the middle attack [3], [4].

Untuk mengurangi ancaman serangan siber, keamanan pada protokol dapat ditingkatkan dengan menambahkan mekanisme pada otentikasinya, seperti pada riset [5] yang mengimplementasikan One – Time – Password pada MQTT. Selain autentikasi, keamanan juga dapat ditingkatkan dengan menambahkan fitur autorisasi. Authorization merupakan mekanisme keamanan yang mengikat suatu perangkat dengan izin yang telah ditetapkan. Salah satu framework autorisasi yang pernah populer yaitu Framework OAuth 1.0a. Framework ini merupakan framework yang digunakan untuk web servers. Pada riset [6], berhasil mengadaptasi framework OAuth1.0a agar dapat digunakan untuk jaringan IoT, yang kemudian dilanjutkan menjadi sebuah produk yang dinamakan Netpie. Dalam risetnya, menganalisis fungsionalitas fitur, penggunaan memori, dan time delay. Tetapi untuk analisis sekuritasnya masih berupa hipotesis terhadap beberapa serangan. Maka dari itu pada penelitian ini, penulis mengusulkan untuk meningkatkan keamanan protokol MQTT dengan menggunakan Netpie versi 2020 sebagai framework OAuth 1.0a sebagai fitur autorisasi. Kemudian, untuk menguji tingkat keamanannya dengan test case, man – in – the middle attack dan dilanjutkan dengan eavesdropping dan menguji memory consumption pada perangkat.

1.2 Topik dan Batasannya

Topik yang dibahas adalah peningkatan keamanan pada aspek otentikasi dan otorisasi menggunakan Netpie sebagai Framework OAuth 1.0a.

Batasan masalah pada tugas akhir ini adalah sebagai berikut :

1. Perangkat IoT yang akan digunakan pada penelitian ini adalah NodeMCU dengan sensor LM35.
2. Perkakas yang akan digunakan untuk pengujian adalah Wireshark, arpspoof, airodump.
3. Platform MQTT yang akan digunakan adalah MQTT yang disediakan oleh Netpie.

1.3 Tujuan

Tujuan yang ingin dicapai dalam pembuatan tugas akhir ini antara lain :

a) Mengimplementasi mekanisme autorisasi dengan menggunakan Netpie pada protokol MQTT.

b) Menganalisis hasil performa penggunaan memori, dan test case serangan *man – in – the middle attack* ARP Spoofing dan *eavesdropping* pada protokol MQTT dengan terimplementasinya Netpie.

1.4 Organisasi Tulisan

Pada bagian 2 akan dijelaskan penelitian sebelumnya serta landasan teori yang terkait dengan penelitian. Pada bagian 3 dijelaskan proses sistem yang dibangun, dan scenario pengujian serangan Man – in – The Middle Attack ARP Spoofing, Eavesdropping, dan penggunaan memori pada perangkat ESP8266 NodeMCU . Lalu, pada bab 4 dijelaskan mengenai pengujian scenario dan menganalisis hasilnya. Terakhir bab 5 terdapat kesimpulan dan saran pada penelitian ini.

2. Studi Terkait

Dalam jurnal [3], penulis menjelaskan terdapat dua jenis attack surface pada IoT yaitu local network dan public network. Untuk Public Network, penulis menggunakan Shodan search engine untuk mendapatkan jaringan IoT di seluruh dunia dengan menginputkan string “port : 1833” dan “MQTT”. Dari hasil pencarian, penulis mengakses jaringan MQTT dengan code otentikasi 0. Karena jaringan yang berkode 0 tidak menggunakan otentikasi pada user, sehingga mudah diserang. Selanjutnya, setelah masuk ke dalam jaringan, penulis dapat mencuri informasi jaringan dengan menggunakan topic wildcard “#”, mem-publish informasi yang tidak sesuai, atau membanjiri jaringan dengan DoS. Tetapi dalam paper, penulis tidak mendemokan skenario pertama ini. Untuk skenario kedua, yaitu skenario dimana penyerang berada di jaringan yang sama dengan IoT. Dengan sniffing, penulis mendapatkan paket MQTT yang berisi detail dari pesan yang akan di publish, seperti topik, isi pesan. Selain itu, penulis mendapatkan username, dan password pada paket “Connect”. Kemudian, integritas data pada MQTT sangatlah lemah karena mudah dimodifikasi. Dalam jurnal, penulis mengganti topic dari pesan yang di publish dengan menggunakan Etterfilter. Selain lemahnya autentikasi MQTT, tanpa adanya tambahan protokol keamanan seperti TLS, MQTT dapat terdeteksi pada Wireshark, sehingga mudah sekali untuk diserang.

Pada paper [5], penulis berusaha untuk meningkatkan keamanan pada MQTT karena berdasarkan riset yang ia temukan, MQTT merupakan protokol yang sering digunakan dalam jaringan IoT yang memiliki keamanan yang sangat rentan. Penulis mencantumkan beberapa referensi paper yang menjelaskan beberapa serangan yang sering dilakukan terhadap keamanan pada MQTT. Beberapa diantaranya yaitu, DoS, brute force, sniffing attack. Dari ketiga serangan tersebut, brute force merupakan serangan yang dijadikan sebagai model case sebagai evaluasi terhadap sistem keamanan yang akan dibangun oleh penulis. Penulis mengimplementasikan One Time Password sebagai keamanan autentifikasi pada MQTT. Dalam skenarionya, penyerang telah mendapatkan IP Server MQTT. Setelah itu, penyerang berusaha untuk melakukan serangan brute force username dan password pada wordlist yang telah disiapkan. Hasil dari pengujiannya, sistem otentikasi OTP berhasil melindungi jaringan MQTT dari serangan brute force. Karena meskipun penyerang berhasil mendapatkan akses ke dalam jaringan, masih diperlukan kode OTP dari device yang telah terdaftar untuk men-publish message ataupun subscribe message.

Pada paper [8], penulis membahas dan menganalisis secara spesifik mengenai protokol MQTT dan serangan – serangan terhadap protokol tersebut. Penulis menjelaskan bahwa terdapat 4 jenis threat agents yang dapat menyerang jaringan pada IoT, yaitu malicious internal user, curious user, bad manufacture, dan external user. Setelah itu, jenis – jenis dan skenario serangan pada protokol MQTT dijelaskan secara mendetail. Terdapat 5 jenis serangan yang umum pada MQTT, yaitu Denial of Service (DoS), spoofing, information disclosure, elevation of privileges, tampering data. Dari setiap jenis serangan, penulis menyampaikan beberapa skenario yang dapat terjadi untuk setiap serangan. Tetapi dari ke 5 jenis itu, untuk pengujian simulasi, penulis hanya menguji DoS pada protokol MQTT, karena DoS merupakan salah satu serangan yang paling sering terjadi, dan ancaman yang paling berbahaya. Dalam skenario pengujiannnya, penulis berasumsi penyerang sudah mendapatkan akses ke dalam jaringan IoT, dan mencoba untuk mengirim pesan ke broker dengan payload yang sangat besar dan menghabiskan bandwidth server menggunakan TCP-based attack SYN flood. Device yang berperan sebagai penyerang mengirimkan 2000 pesan publish dengan 4mb payload. Setelah simulasi, penulis melakukan test bed untuk melihat performansi server dan kondisi server ketika terkena serangan DoS.

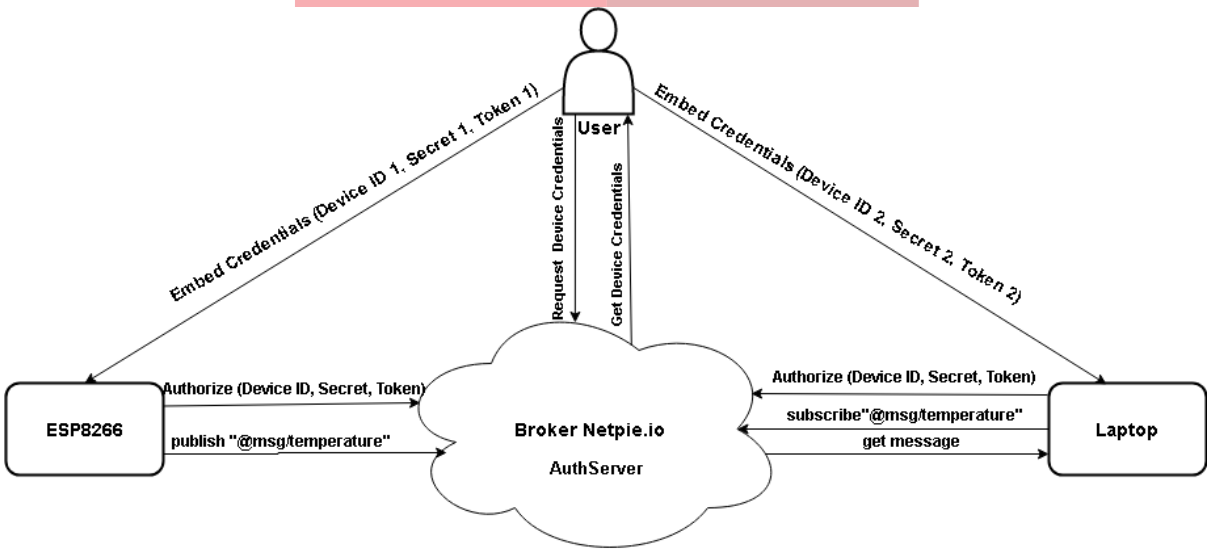
Penelitian yang dilakukan oleh Aimaschana et al [6], MQTT memiliki tingkat keamanan yang rendah, yang dimana keamanannya adalah otentikasi yang berupa sepasang user dan password yang digunakan untuk semua device yang terhubung ke broker. Penulis meningkatkan keamanan dengan keamanan selain otentikasi, yaitu otorisasi yang berguna untuk memanajemen dan kontrol akses perangkat yang terperinci. Fitur otorisasi yang

diimplementasi pada jaringannya menggunakan Framework OAuth. Penulis menggunakan Framework OAuth 1.0a daripada OAuth 2.0 karena memiliki keunggulan yang lebih dalam segi keamanan dan ringan seperti, Client harus meng-generate signature baru setiap adanya request baru, sedangkan OAuth 2.0 menggunakan satu pasang kredensial yang rentan untuk dicuri, serangan replay, dan serangan spoofing. Agar dapat digunakan dalam lingkungan IoT, penulis mengadaptasi alur otorisasi Framework OAuth. Hasil dari evaluasi penelitiannya adalah fitur otorisasi Framework OAuth berfungsi dengan baik, dan memiliki time delay yang cukup tinggi karena device yang terhubung banyak, dan penggunaan memori yang ringan sehingga dapat digunakan di semua papan Arduino dengan spesifikasi minimal 32 KB Flash Memory dan 2 KB RAM. Tetapi untuk analisis sekuritasnya masih berupa hipotesis terhadap beberapa serangan.

3. Sistem yang Dibangun

3.1 Gambaran Umum Sistem

Pada tugas akhir ini dirancang dan dibangun system monitoring suhu untuk menjadi lingkungan yang akan diamati untuk diuji keamanannya. Berikut gambaran umum system yang dirancang :

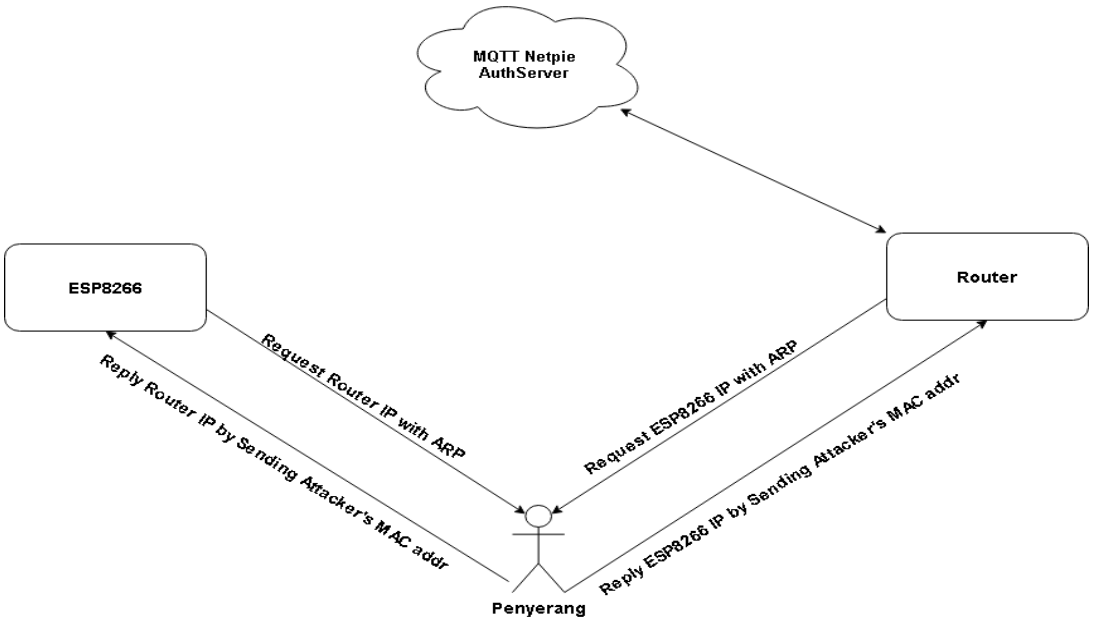


Gambar 1 Gambaran Umum Sistem

Pada rancangan ini terdapat 1 NodeMCU dengan menggunakan 1 sensor LM35 yang terhubung ke broker MQTT pada Cloud Netpie. ESP8266 NodeMCU berperan sebagai publisher yang mengirimkan informasi suhu ke suatu topic yang akan diteruskan oleh broker. Sedangkan Laptop berperan sebagai subscriber yang menerima informasi suhu yang dikirimkan oleh ESP8266 NodeMCU. Broker MQTT pada Netpie sudah terintegrasi dengan keamanan OAuth 1.0a. Maka, seperti yang sudah dijelaskan pada bab sebelumnya, User atau Resource Owner perlu mendapatkan device credentials yang kemudian harus disisipkan ke perangkat agar bisa terhubung ke broker.

3.2 Skenario Pengujian Serangan MiTM ARP Spoofing

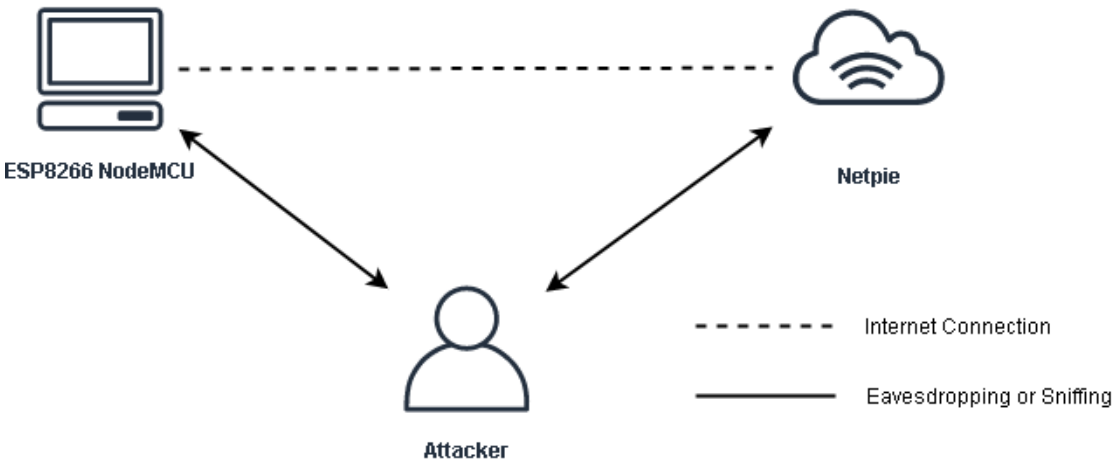
Berikut ini merupakan gambar dari scenario penyerangan MiTM ARP Spoofing :



Gambar 2 Skenario Penyerangan MiTM ARP Spoofing

Penyerang menggunakan perangkat laptop dengan sistem operasi Linux Ubuntu 20.04. Untuk memonitoring IP pada jaringan WiFi, penyerang menggunakan USB Wireless TP-Link WN722N dan tools airodump-ng. Ketika device ESP8266 dan Router saling menanyakan ARP nya untuk mengetahui IP masing – masing device, penyerang akan me-*reply* ARP kedua device tersebut dengan MAC Address device penyerang. Sehingga dalam ARP tabel ESP8266 NodeMCU, IP address router akan tergantikan MAC addressnya dengan penyerang. Sedangkan dalam ARP tabel router, IP address ESP8266 NodeMCU akan tergantikan MAC addressnya dengan penyerang. Dalam kondisi ini, penyerang telah berhasil melakukan serangan ARPSpoofing dan berada di tengah komunikasi ESP8266 NodeMCU dan router.

3.3 Skenario Pengujian Serangan Eavesdropping



Gambar 3 Skenario Pengujian Serangan Eavesdropping

Setelah penyerang berhasil melakukan serangan ARP Spoofing, paket yang keluar dan masuk dari ESP8266 NodeMCU ke broker MQTT Netpie akan melewati penyerang terlebih dahulu. Pada tahap ini, penyerang melakukan *eavesdropping* menggunakan tools Wireshark untuk mengamati paket yang lewat dan menangkap paket yang dikirimkan oleh ESP8266 NodeMCU ke broker MQTT Netpie. Paket yang ditangkap akan diperiksa informasinya dari kedua pihak tersebut.

3.4 Skenario Pengujian Penggunaan Memori

Skenario memory consumption adalah untuk mengetahui penggunaan memori pada microcontroller ketika berperan sebagai client dalam jaringan IoT. Dalam scenario Memory Consumption *tools avr-gcc compiler* digunakan untuk melihat pengunaan memory pada microcontroller. Tools avr mensegmentasi ukuran menjadi 3 bagian, yaitu *.text*, *.data*, dan *.bss*. *.Text* adalah segmen file objek atau bagian yang sesuai dari ruang alamat *virtual program* yang berisi *executable instructions* disimpan dan umumnya *read-only* dan memiliki ukuran yang tetap [13]. .Data adalah segmen dari variable global atau statis yang nilainya sudah ditentukan dan dapat dimodifikasi. Biasanya nilai pada variable ini disimpan pada memori *read-only* atau di *.text* kemudian disalin ke segmen data selama rutinitas *start-up program* [13]. .Bss adalah segmen yang berisi semua variable global dan statis yang diinisialisasi ke - nol atau tidak memiliki inisialisasi eksplisit dalam *source code* [13].

Memori yang dihitung pada skenario ini adalah penggunaan memory *FLASH* dan *SRAM*. Berikut rumus untuk menghitung penggunaan memory *FLASH* [13] :

$$\textbf{\textit{Flash Memory Usage}} = \textbf{\textit{.text}} + \textbf{\textit{.data}} \quad (1)$$

Sedangkan untuk rumus untuk menghitung penggunaan RAM-nya sebagai berikut [13]:

$$\textbf{\textit{RAM Memory Usage}} = \textbf{\textit{.data}} + \textbf{\textit{.bss}} \quad (2)$$

4. Evaluasi

Penggunaan software yang digunakan pada penilitian ini terdapat pada tabel 1. Program untuk publish/subscribe dituliskan dalam software Arduino 1.8.12. Kemudian, program akan dijalankan pada perangkat ESP8266 NodeMCU. Pada penelitian ini Netpie menjadi broker MQTT sekaligus keamanan AuthServernya yang berjalan pada port 1883.

No.	Nama	Deskripsi
1.	Arduino 1.8.12	Publish/subscribe untuk perangkat mikrokontroler.
2.	Netpie	MQTT Broker Server dan AuthServer.
3.	Linux Ubuntu 20.04	OS for the attacker
4.	arpspoof	Tools untuk ARP spoofing Man – in – the middle attack.
5.	airodump-ng	Tools untuk mendeteksi access point dan device yang terhubung pada access point tersebut.
6.	Wireshark	Tools untuk memonitor dan menangkap paket data pada traffic interface jaringan.

Penyerang menggunakan Linux Ubuntu 20.04 sebagai sistem operasi yang digunakan pada perangkatnya untuk melakukan simulasi serangan man – in – the middle attack ARP spoofing. Airodump-ng merupakan tools yang penting untuk mendeteksi dan mendapatkan IP dan MAC address access point dan device yang terhubung pada access point tersebut, yang akan digunakan untuk ARP spoofing. Arpspoof salah satu tools yang dapat digunakan untuk ARP spoofing, untuk menyerang dengan tools ini, penyerang membutuhkan 2 IP yang telah

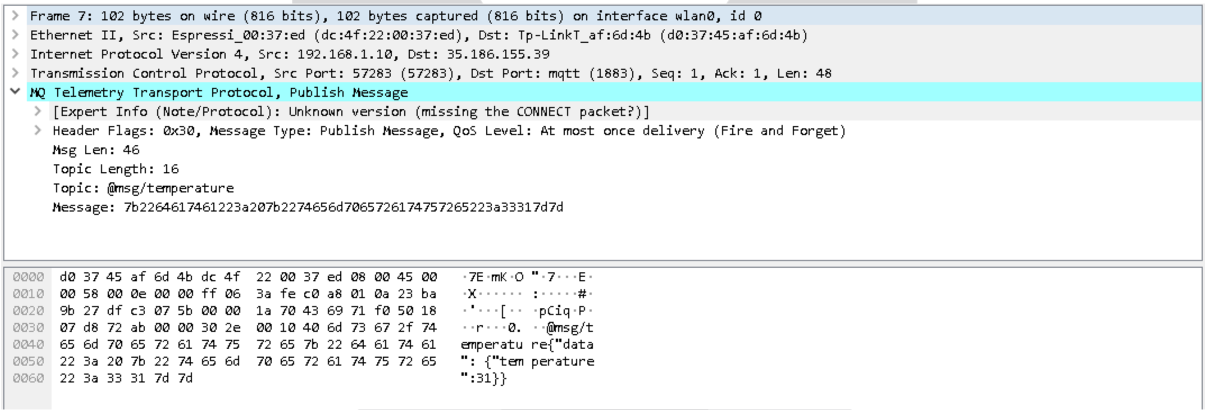
didapat dari monitoring sebelumnya. Setelah itu, Wireshark digunakan untuk menangkap paket data dari publisher yang terhubung ke broker MQTT Netpie.

4.1 Hasil Pengujian Serangan MiTM ARP Spoofing

Gambar 4 ARP Spoof pada access point dan ESP8266 NodeMCU

Pada gambar 6, menunjukkan bahwa serangan ARP Spoof berhasil dilakukan. Ketika ARP Spoof berhasil dilakukan, maka komunikasi ESP8266 NodeMCU yang seharusnya ke router kemudian terhubung ke internet untuk ke Netpie, sekarang harus melewati perangkat penyerang terlebih dahulu sebelum ke router. Serangan ARP Spoofing berhasil diindikasikan ketika alamat MAC router pada tabel ARP perangkat target berubah atau penyerang dapat melihat paket dalam wireshark dimana perangkat penyerang membalas ARP bahwa IP router adalah milik penyerang dan sebaliknya.

4.2 Hasil Pengujian Serangan Eavesdropping



Gambar 5 Wireshark packet header

Ketika menggunakan wireshark, untuk menspesifikkan penangkapan data dapat dengan memfilter ip ESP8266 NodeMCU yaitu “ip.src == IP NodeMCU” atau langsung dengan “mqtt”, karena device terhubung dengan port 1883. Dengan begitu, seluruh paket data yang telah terfilter hanya pada IP ESP8266 NodeMCU. Gambar diatas merupakan salah satu paket ESP8266 NodeMCU yang ditangkap pada interface Wireshark penyerang. Paket data berisi detail dari paket yang dikirimkan. Dari hasil testing, dapat dilihat bahwa penyerang mendapatkan informasi Topic dari pesan, dan isi dari pesan berupa suhu.

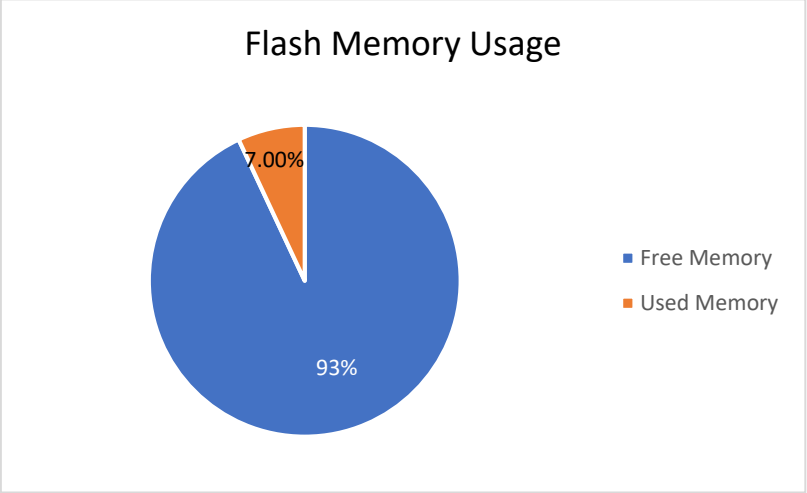
4.3 Hasil dan Analisis Pengujian Penggunaan Memori

Pada Gambar [6] merupakan hasil dari penggunaan compiler avr-gcc untuk melihat penggunaan memori pada perangkat ESP8266 NodeMCU.

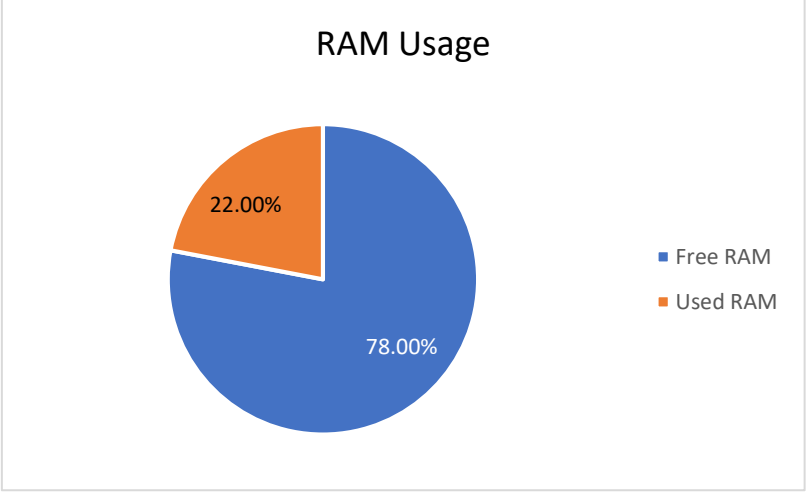
```
C:\Users\McHanzo\AppData\Local\Temp\arduino_build_934038>avr-size Projekt_Netpie2020.ino.elf
   text    data     bss     dec     hex filename
 274956    2432   26512  303900   4a31c Projekt_Netpie2020.ino.elf
```

Gambar 6 Hasil Pengujian Memory Consumption

Hasil dari compiler avr-gcc masih dijabarkan dalam bentuk segment – segment, maka dari itu perlu diproses lagi untuk mendapatkan penggunaan memori FLASH dan RAM menggunakan persamaan (1) dan (2).



Gambar 7 Penggunaan Memori Flash



Gambar 8 Penggunaan RAM

Pada pengujian penggunaan memory, device yang digunakan adalah ESP8266 NodeMCU yang memiliki 4 MB Flash Memory dan 128 KB RAM. Gambar diatas merupakan hasil penghitungan penggunaan memori dari avr-gcc. Program menghabiskan sebanyak 270,8 KB dari 4 MB untuk FLASH Memory. sedangkan untuk penggunaan RAM, program menggunakan 28,2 KB dari 128 KB. Dari hasil pengujian penggunaan memori dapat disimpulkan bahwa, program tidak memberatkan perangakat karena menggunakan sedikit memori dan perangkat yang terhubung ke MQTT dengan keamanan Netpie setidaknya memiliki spesifikasi dengan 320 KB Flash Memory dengan 32 KB RAM.

4.4 Analisis Hasil Pengujian Serangan ARP Spoofing dan Eavesdropping

Dari hasil pengujian dapat dilihat bahwa penyerang berhasil menyadap komunikasi antara devices dengan server broker MQTT Netpie dengan ARP Spoofing. Untuk serangan eavesdropping, penyerang berhasil mendapatkan informasi pesan yang dikirimkan oleh NodeMCU yang berupa topik dan isi pesan.Meskipun isi pesan data dapat dibaca oleh penyerang, keamanan Netpie berhasil dalam melindungi kredensial device. Hal ini dapat diketahui, karena dalam serangan eavesdropping, penyerang tidak mendapatkan kredensial device yang digunakan untuk terotorisasi dengan Netpie dan terhubung ke MQTT. Sedangkan keamanan dasar otentikasi yang disediakan oleh MQTT, pesan tidak terenkripsi dan selain isi pesan dan topik, penyerang mendapatkan username dan password agar terhubung MQTT. Kemudian dengan adanya fitur otorisasi yang disediakan oleh Netpie, ketika penyerang mendapatkan device credentials suatu client, penyerang tidak dapat menggunakan topik wildcard '#' untuk mencuri semua informasi pada berbagai topik yang bertukar pada broker MQTT. Karena alur mekanisme otorisasi Netpie, resource owner harus menentukan scope masing – masing device terlebih dahulu dalam bentuk group sebelum client dapat membuat koneksi ke broker MQTT Netpie. Suatu client tidak dapat berkomunikasi dengan device lain meskipun terhubung pada topik yang sama, ketika tidak dalam group yang sama. Selain itu, resource owner juga dapat menggantikan device credentials yang lama dengan yang baru dan menghapus yang lama, jika terjadi kasus terburuk, penyerang berhasil masuk kedalam jaringan. Sehingga Netpie dapat menjadi salah satu solusi dalam mengatasi kelemahan pada protocol MQTT pada aspek otentikasinya, dan dapat meningkatkan keamanan dengan fitur otorisasi yang disediakan.

5. Kesimpulan

Dalam penelitian ini, penulis telah diimplementasikan sistem MQTT dengan menggunakan Netpie sebagai framework OAuth 1.0a. Kemudian, sistem diujikan dengan penggunaan memori pada device dan serangan berbasis otentikasi, man – in – the – middle attack, dan eavesdropping. Hasil pengujian menunjukkan bahwa, Netpie dapat menjadi salah satu solusi dalam mengatasi permasalahan otentikasi pada protokol MQTT dan dapat meningkatkan keamanan dengan tambahan fitur otorisasi yang disediakan. Tetapi disamping fitur keamanannya, tidak semua perangkatdapat terhubung kedalam jaringan, karena dari hasil pengujian penggunaan memori, setidaknya spesifikasi minimum perangkat harus memiliki 320 KB Flash Memory dan 32 KB RAM.

Untuk penelitian selanjutnya, dapat dilakukan peningkatan keamanan dengan aspek yang lain,yaitu integrity, dengan cara memberikan enkripsi pada pesan dan topik yang dikirimkan.

Reference

[1] R. Minerva, A. Biru, and D. Rotondi, "Toward a definition of the Internet of Things," IEEE Initiat., 2015.

[2] D. Soni and A. Makwana, "A survey on mqtt: a protocol of internet of things(IoT)," Int. Conf. Telecommun. Power Anal. Comput. Tech. (Ictpact - 2017), no. April, pp. 0–5, 2017.

[3] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of mqtt communication protocol in iot system," in International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2017.

[4] A. M. Gamundani, A. Phillips, and H. N. Muyingi, "An Overview of Potential Authentication Threats and Attacks on Internet of Things(IoT): A Focus on Smart Home Applications," Proc. - IEEE 2018 Int. Congr. Cybermatics 2018 IEEE Conf. Internet Things, Green Comput. Commun. Cyber, Phys. Soc. Comput. Smart Data, Blockchain, Comput. Inf. Technol. iThings/Gree, no. November, pp. 50–57, 2018.

[5] F. Fauzi, P. Sukarno, and A. A. Wardana, "Otentikasi pada Internet-of-Things berbasis MQTT Menggunakan One-Time-Password pada Kasus IoT Home Gateway," vol. 6, no. 2, pp. 9219–9231, 2019.

[6] Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for MQTT-based Internet of Things," in 2016 IEEE International Conference on Communications Workshops, ICC 2016, 2016.

[7] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," Int. Conf. Electr. Eng. Comput. Sci. Informatics, vol. 2017-Decem, no. September 2017, 2017.

[8] S. N. Firdous, Z. Baig, C. Valli, and A. Ibrahim, "Modelling and evaluation of malicious attacks against the IoT MQTT protocol," Proc. - 2017 IEEE Int. Conf. Internet Things, IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, vol. 2018-Janua, pp. 748–755, 2018.

[9] "RFC 5849 - The OAuth 1," 2010. [Online]. Available: https://tools.ietf.org/html/rfc5849. [Accessed: 28-Oct-2019].

[10] Shapeways, "Shapeways Documentation," 2014. [Online]. Available: http://developers.shapeways.com/learn/understanding-authorization#authentication. [Accessed: 25-Nov-2020].

[11] Netpie, "Netpie 2020 Features vs Netpie 2015 Features," Netpie, 2018. [Online]. Available: https://netpie.io/compare. [Accessed: 5 Nov 2020].

[12] A. Efe, G. Kalkanci, M. Donk, S. Cİhangİr, and Z. Uysal, "A Hidden Hazard : Man-in-The-Middle Attack in Networks," no. 2, pp. 96–116, 2019.

[13] NonGNU, "Memory Sections," NonGNU, August 2014. [Online]. Available: https://www.nongnu.org/avr-libc/user-manual/mem_sections.html. [Accessed 3 December 2020].