Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893
p-ISSN: 2089-3787

Implementasi *Traefik* sebagai *Reverse Proxy* dengan Prinsip *Zero Trust*

Arnanda Nuryasa[1*], Imam Suharjo[2]
Informatika, Universitas Mercu Buana Yogyakarta, Sleman, Indonesia
*e-mail *Corresponding Author:* nandaar504@gmail.com

Abstract

Technology and application development are experiencing massive development, without realizing it, cyber crime is also increasing rapidly. The increase in cyber attacks is a concern and makes concerns about important data owned. The impact can make applications or services lose and important data can be stolen and misused. With the application of Treafik as one of the reverse proxies, it aims to be a solution in an effort to increase application security by integrating it with the development of an IP whitelist system that has Zero Trust Security principles. The method used in the development contains several stages of research, namely, system analysis, system design, configuration, and test scenarios. The results of testing and research show that the IP whitelist system runs well according to the system design and the application of traefic can improve the application security system by denying unrecognized access to the IP whitelist system with a "forbidden" message.
Keywords: *Traefik; Reverse Proxy; Whitelist IP; Zero Trust Security*

Abstrak

Teknologi dan pengembangan aplikasi mengalami perkembangan yang sangat masif, tanpa disadari ternyata kejahatan *cyber* juga meningkat pesat. Meningkatnya serangan *cyber* ini menjadi keresahan dan membuat kekhawatiran akan data penting yang dimiliki. Dampaknya dapat membuat aplikasi atau *service* menjadi *loss* dan data-data penting dapat dicuri dan disalahgunakan. Dengan penerapan *Traefik* sebagai salah satu *reverse proxy* bertujuan untuk dapat menjadi solusi dalam upaya meningkatkan keamanan aplikasi dengan di integrasikan pada pengembangan sistem *whitelist IP* yang berprinpip *Zero Trust Security.* Metode yang digunakan dalam pengembangan memuat beberapa tahapan penelitian yaitu, analisis sistem, perancangan sistem, konfigurasi, dan skenario pengujian. Hasil pengujian dan penelitian menunjukkan sistem *whitelistIP* berjalan dengan baik sesuai rancangan sistem dan penerapan *traefik* dapat meningkatkan sistem keamanan aplikasi dengan menolak akses masuk yang tidak dikenali pada sistem *whitelist IP* dengan pesan "*forbidden*".
Kata Kunci: *Traefik; Reverse Proxy; Whitelist IP; Zero Trust Security*

1. Pendahuluan

Pada proses pengembangan aplikasi sangat penting untuk menerapkan dan memperhatikan aspek keamanan aplikasi baik proses, teknologi, maupun *people*. Ancaman serangan *cyber* selalu muncul dari segala arah dan dari segala kemungkinan kerentanan. Terlebih lagi akses yang mudah didapat yang menjadikan sistem atau aplikasi mendapat banyak permintaan akses-akses yang illegal dan akses masuk yang tidak sah ke dalam aplikasi untuk mendapatkan dan mencuri data-data penting.

Saat ini, banyak aplikasi atau sistem mudah di *hacked* oleh *hacker* karena akses masuknya yang mudah di akses dan kerap kali tidak ada pembatasan ke aplikasi. Pada umumnya banyak kasus yang sebenarnya aplikasi atau sistem tersebut memiliki akses terbatas dan tidak secara menyeluruh dapat di akses oleh publik, dalam pengembangan berbasis *microservice* pun tidak semua *service* memiliki akses publik. Hal inilah yang menjadi titik awal seorang *hacker* dapat menyerang aplikasi karena akses yang begitu bebas. Diperlukan *Policy Firewall* untuk penggunaan *access control list* yang dapat melakukan pembatasan akses[1] berdasarkan IP *address* sehingga meningkatkan lapisan keamanan aplikasi karena adanya *filtrasi* permintaan masuk yang menyebabkan seorang penyerang pun tidak langsung bisa mengakses aplikasi. Namun, dalam tekonologi *firewall* yang tradisional banyak proses *filtering* yang masih manual, ini

sangat tidak efektif karena harus manual menambhakan daftar yang dapat mengakses sistem secara manual satu persatu. Maka diperlukan pengembangan sistem yang dapat membuat proses manual tersebut menjadi otomatis dengan membuat sistem *Whitelist IP*.

Dalam hal ini, salah satu solusi meningkatkan keamanan dan pembatasan akses aplikasi seperti *policy control access* dapat diimplementasi pada penerapan *reverse proxy* yang mana data yang datang dari *request client* akan ditangani terlebih dahulu sebelum diarahkan ke server tujuan[2]. Semua proses dan request akan di distribusikan dan pendistribusian ini merupakan fungsi *load balancer*[3].Salah satu alat untuk penerapan *reverse proxy* adalah *Traefik*. *Traefik* merupakan *tools open sources* yang dapat berbepan sebagai *reverse proxy* dan memiliki fungsi yang berbeda jika dibandingkan *dengan tools reverse proxy* lainnya. *Traefik* memiliki fungsi *whitelistIP* yang mana Fungsi ini akan menolak *request* masuk sebelum di teruskan ke server kecuali *request* dari *ip address* yang sudah masuk daftar *whitelist*.[4]

Dalam implementasinya, *whitelist ip address* ini membaca sebuah file yang berisi daftar *ip address*, untuk mempermudah *update list ip address* maka *traefik* dapat menggunakan *ETCD* sebagai tempat untuk menyimpan key pair atau dalam penelitian ini list *ip address*. Untuk menghindari manual *update list*, dikembangkan sebuah sistem *whitelist* ip berbasis web yang akan mengelola list ip address tersebut dan dintegrasikan dengan ETCD sebagai tempat menyimpan key pair atau list ip address dan ETCD memiliki API yang dapat digunakan sebagai endpoint untuk memudahkan komunikasi dalam memperbaharui *list ip address*[5]. Sistem *whitelist ip address* akan menangani autentikasi dan *autorisasi user* dengan penerapan prinsip *zero trust*.[6] Dengan begitu, lapisan keamanan akan di tingkatkan karena sebelum masuk ke dalam *whitelist, user* harus di authentikasi terlebih dahulu, sehingga ketika *user* tidak dapat di authentikasi dan mendapatkan authorize maka user tersebut gagal dalam mendapatkan *trust* untuk dapat mengakses ke *service* yang di kelola oleh traefik.

Penelitian ini betujuan untuk mengembangkan sistem yang mengimplementasikan *traefik* sebagai *reverse proxy* untuk meningkatkan keamanan dengan berprinsip *zero trust*. Riset ini di harapkan dapat menjadi solusi keamanan yang memadai dan meningkatkan *security awareness* di lingkungan pengembang aplikasi dalam menjaga sistem atau aplikasi dari serangan *cyber*. Dengan begitu, sistem ini dapat memberi manfaat yang signifikan untuk aplikasi dalam hal efektifitas dan otomatisasi dalam mementingkan faktor keamanan aplikasi.

2. Tinjauan Pustaka

Penelitian mengenai *Zero Trust Architecture: Trend and Impact on Information Security* menyimpulkan bahwa kerangka Tradisional Security memiliki kelemahan dengan masih masif terjadinya *ransomware attach, data theft, intrusion, dan malware*. Dengan Model *Zero Trust* dapat meningkatkan keamananan dan dapat dikombinasikan dengna model security lain dalam upaya meningkatkan keamanan aplikasi. Dengan slogan *ZTA (Zero Trust Architecture)* bahwa *"never trust"* membuat lingkungan sistem yang lebih aman dan preventif [7].

Penelitian *Experimental Studies of the Features of using WAF to Protect Internal Services in the Zero Trust Structure* dengan meningkatnya popularitas pengembangan apikasi web membuat kebutuhan akan perlindungan terhadap aplikasi meningkat. Disebutkan bahwa ternyata 75% ancaman serangan *cyber* terjadi kepada aplikasi yang dimiliki perusahaan. Dilakukan WAF untuk melindungi sistem internal dengan pengujian pada struktur *Zero Trust*, pada kerentanan tingkat rendah dan tingkat menengah mengalami penurunan yang signifikan dan hasilnya ketika aplikasi di proteksi pada keamanan ini response yang di berikan adalah *"403 forbidden"* [8].

Penelitian *Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall* menyimpulkan bahwa pada penelitian yang telah dilakukan dalam mengimplementasikan WAF pada aplikasi berbasis Web menggunakan *ModSecurity* dan *Reverse proxy* dapat menambah fungsi kemanan dari apliaski itu sendiri. Ancaman *XSS*, *SQL Injection, dan Unauthorized Vulnerability Web Scanning* pada penelitian tersebut sukses melakukan pencegahan atau tindakan *prevention* terhadap ancaman tersebut [9].

Penetilian *Design and Implementation of a Vulnerability-Tolerant Reverse Proxy Based on Moving Target Government Application* menyimpulkan dalam implementasi *Reverse Proxy base on Moving Target Defense* pada aplikasi pemerintah yang dapat mengubah kata kunci dari URL statis ke dinamis secara acak. Telah disajikan prototipe untuk mengevaluasi efektivitas dan kelayakan dari implementasi tersebut, hasilnya skema yang diajukan pada prototipe penelitian efektif dan layak [10].

Penelitian Analisis Implementasi *Modsecurity* dan *Reverse Proxy* Untuk Pencegahan Serangan Keamanan *DDoS* pada *Web Server* menyimpulkan bahwa dalam hasil pengujian serangan ketiga tidak bisa ditolak jika hanya implementasi *Reverse Proxy*, namun jarak ke server menjadi lebih jauh dan memakan waktu lebih Panjang 53,12509 ms. Serangan ketiga dapat di tolak dengan implementasi WAF *Modsecurity*, dan untuk serangan ketiga dapat ditolak dengan implementasi *Reverse Proxy* dan *ModSecurity.*[11]

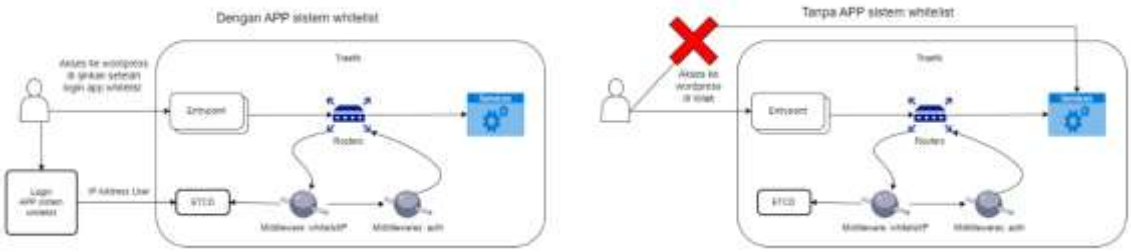
State of the Art pada penelitian ini adalah integrasi model sistem *whitelist IP* yang berprinsip *zero trust* dengan penerapan *traefik* yang membuat otomatisasi dalam proses *whitelist IP* yang dilakukan oleh *traefik*. Konsep ini mengkombinasikan antara implementasi *reverse proxy* dengan sebuah prinsip *zero trust security* yang dikembangkan dalam model sistem berbasis Web. Sistem ini akan digunakan sebagai validasi dengan proses autentikasi dan autorisasi dan dikembangkan menggunakan bahasa pemrograman Python menggunakan *framework Django* yang mendukung pengembangan aplikasi berbasis web.

3. Metodologi

Pada metode penelitian ini, peneliti akan menguraikan Langkah-langkah dalam melaksanakan penelitian hingga tercapai hasil dan tujaun dari penelitian. Tahapan penilitan yang dilakukan terbagi menjadi beberapa bagian, yaitu: Analisis Sistem, Perancangan Sistem, Konfigurasi, dan Skenario Pengujian [12].

3.1 Analisis Sistem

Pada penelitian ini akan mengimplementasikan *Traefik sebagai reverse proxy* dan berperan juga sebagai *load balancer*. *Traefik* memiliki fungsi yang berbeda jika dibandingkan dengan beberapa *tools* serupa seperti *nginx* dan *squid* yang memiliki fungis filter dan keamanan jaringan.[13] Pada konsep ini, traefik sebagai *reverse proxy* akan menjadi perantara user ketika ingin mengakses *service* dalam kasus ini (wordpress). Traefik akan menjadi pelayan yang akan membagi traffic dan juga fungsi lainnya seperti menggunakan *middleware* yang memiliki salah satu fungsi yang akan di bahas yaitu *Whitelist IP. Whitelist IP* yang dimiliki oleh traefik ini akan di integrasikan dengan konsep *Zero Trust* yang akan menagani manajemen user itu mengakses atau dapat dikatakan berhasil masuk di *whitelist IP*, maka user tersebut harus melalui *authentication* terlebih dahulu.[14][15] Berikut gambaran gambar1. alur sistem bekerja.



Gambar 1. Analisis sistem topologi traefik

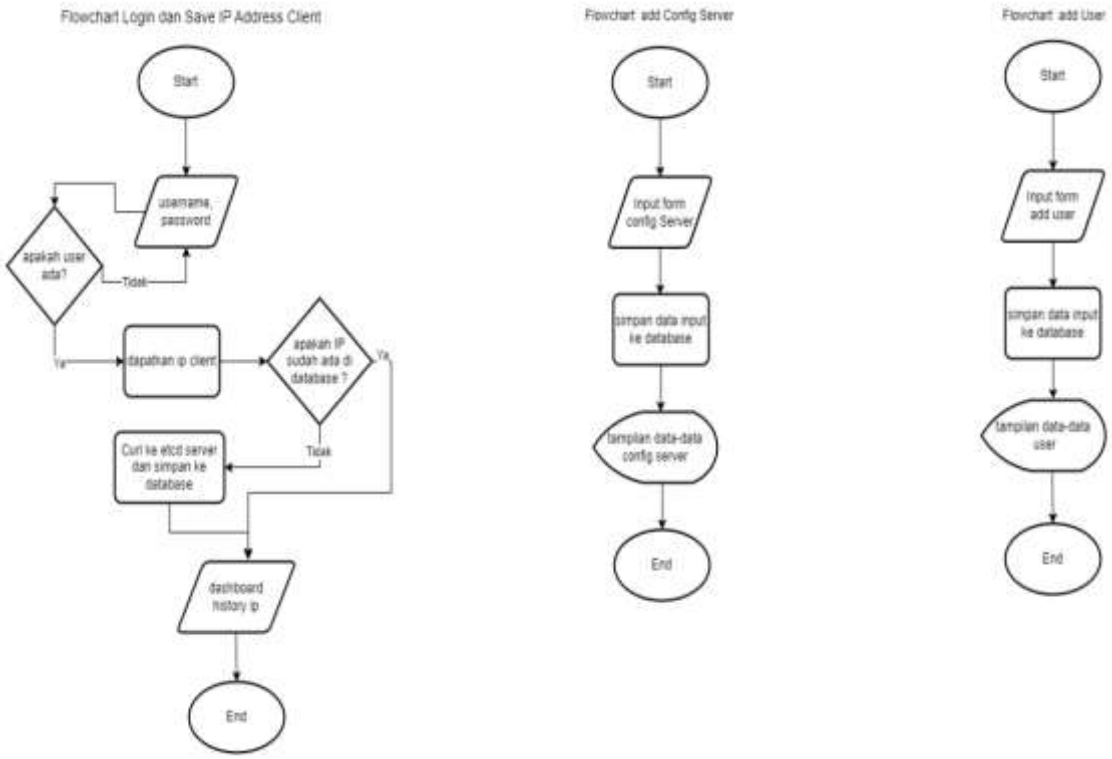
Pada gambaran diatas, Sistem *Whitelist IP* akan menangani *authentication* dan *authorization* user dan akan mengambil ip address user dan menyimpannya ke dalam etcd yang akan di simpan sebagai daftar list ip. Sistem *whitelist ip* yang akan memberikan hak akses kepada user bahwa memang user tersebut berhak mengakses *service* (dalam penelitian ini *wordpress)*.

3.2 Perancangan Sistem

Sistem *Whitelist IP* merupakan sistem yang akan di integrasikan dengan implementasi traefik dan berkomunikasi dengan server ETCD sebagai tempat menyimpan *key value* atau *ip adddress*. Sistem ini dibangun ber basis web dengan menggunakan *framework Django.*

3.2.1 *Flowchart*

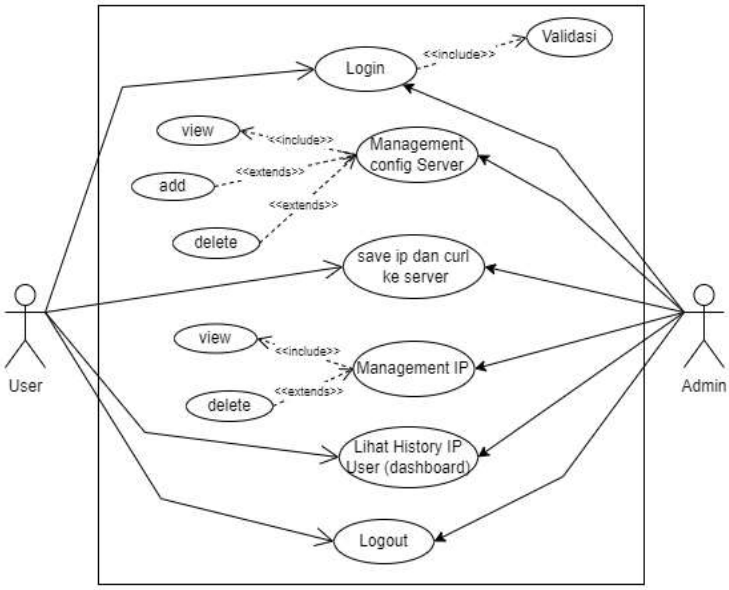
Flowchart menjelaskan alur bagaimana sistem dapat bekerja dengan baik mulai dari awal hingga akhir. Pada rancangan ini dijelaskan bagaimana pengembangan sistem *whitelist IP* ini bekerja, mulai dari bagaimana integrasi dengan traefik sebagai *reverse proxy* yang akan meng-handle sistem whitelist dalam meningkatkan keamanan.



Gambar 2. *Flowchart* Aplikasi *Whitelist IP*

3.2.2 *UseCase* Diagram

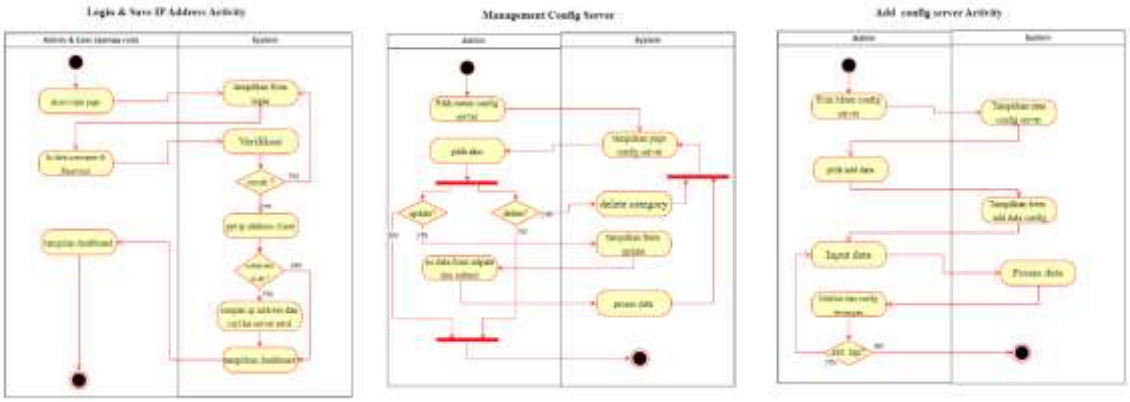
UseCase Diagram memberikan paparan bagaimana interkasi antara aktor atau bisa disebut user yang ada di sistem tersebut dengan aplikasi. Interaksi apa saja yang bisa dilakukan oleh setiap user gambarkan pada diagram ini.



Gambar 3. *Usecase Diagram* Aplikasi *Whitelist IP*

3.2.3 *Activity* Diagram

Diagram ini menjelaskan alur kerja apa saja yang ada di dalam sistem. Kemudian di gambarkan juga user yang ada di sistem melakukan aktivitas apa saja dan semua proses yang ada di sistem di jelaskan di dalam diagram *activity.*

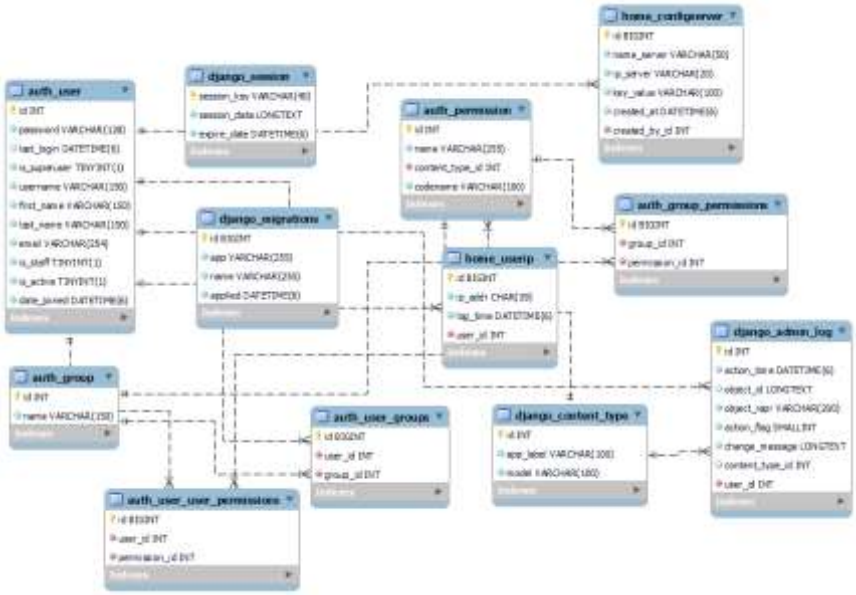


Gambar 4. *Activity Diagram* Aplikasi *Whitelist IP*

Gambar 5. *Activity Diagram* aplikasi *whitelist ip*

3.2.4 *Class* Diagram

Relasi *database* yang akan digunakan dalam pembuatan sistem di jelaskan dalam diagram ini. Pada Gambar dibawah ini merupakan ERD yang akan digunakan dalam mengembangkan sistem *Whitelist IP* Address.



Gambar 6. Rancangan *ERD* Aplikasi *Whitelist IP*

3.3 Konfigurasi

3.3.1 *Service*

Dalam penelitian ini, untuk implementasi dan sebagai bahan pengujian digunakan *wordpress* yang di install di lokal untuk contoh *service* yang akan menjadi target sistem di dalam implementasi traefik ini. Service ini lah yang akan menjadi bahan pengujian dan test bahwa apakah user dengan *ip public* yang tidak ada di *whitelist* masih bisa akses ke *service* ini atau tidak. Pada penelitian ini, peneliti menggunakan *wordpress* yang hanya bisa diakses lokal dengan detail berikut.

Tabel 1 *Detail Service (Wordpress)*

Nama *service*	*Wordpress*
Url	http://172.21.0.2
Port	80
Domain (untuk penelitian)	wordpress.yasa.my.id

3.3.2 Installasi *ETCD*

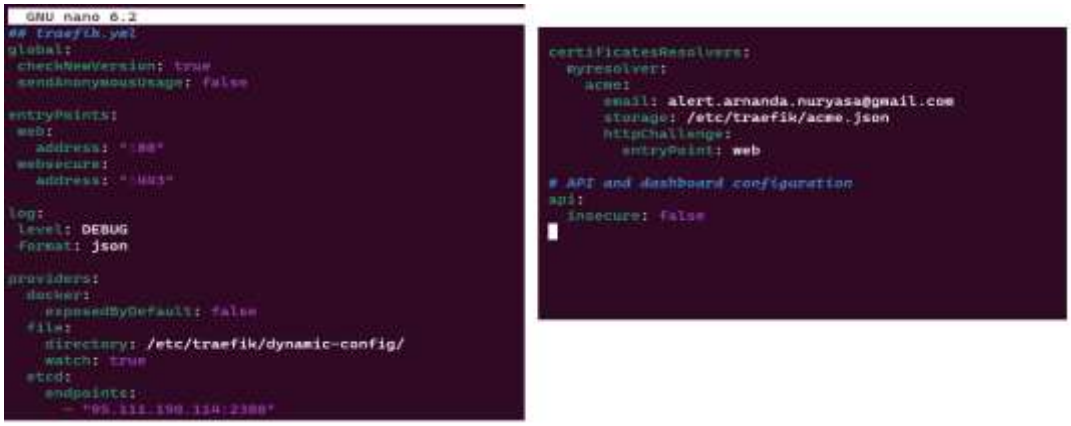
ETCD akan digunakan sebagai providers didalam traefik untuk menyimpan *key pairs* atau tepatnya pada penelitian ini akan digunakan untuk menyimpan *list IP Address* yang masuk ke dalam *whitelist ip*. Untuk *install etcd* dapat menggunakan perintah berikut.

Gambar 7. Implementasi Installasi ETCD

Ketika berhasil terinstall dapat dilihat versi etcd yang digunakan dan secara default setelah berhasil terinstall, etcd berjalan pada port 2379. Dengan port ini sistem whitelist akan bisa berkomunikasi dengan *ETCD API* dan menyimpan *ip address client* melalui *method POST*

3.3.2 Implementasi *Traefik*

Penerapan Traefik akan sebagai fungsi reverse proxy untuk meng*handle load balancer* dan juga keamanan tambahan. Konfigurasi *traefik* dapat berupa file ber type, yaml atau toml, perbedaan dari keduanya tidak banyak, sedikit perbedaan terdapat pada format penulisan saja. Berikut ini file configurasi dengan nama *traefik.yml* yang berada pada */etc/traefik/* untuk implementasi sebagai *reverse proxy* [16].



Gambar 8. Implementasi Konfirgurasi *Treafik*

Dalam konfigurasi diatas, Traefik akan membuka 2 port 433 dan 80 yang mana ini adalah protokol HTTP dan HTTPS. Untuk penelitian ini, peneliti menggunakan 3 *providers* pada traefik seperti yang ada pada konfigurasi di atas yaitu *provider docker*, file, dan etcd. Pada Providers file, traefik akan meng-include kan semua file konfigurasi yang ada pada *directory* diatas. Kemudian karena menggunakan juga prokolol https, perlu konfigurasi untuk *certificate*, pada konfigurasi diatas *certificate* di konfigurasi pada baris *certificateResolvers.*

Kemudian buat file .yml untuk konfigurasi pada folder yang telah di atur pada konfigurasi sebelumnya pada bagian *providers file.* Pada file konfigurasi ini akan memuat aturan untuk konfigurasi yang dibutuhkan dalam implementasi traefik yakni *router, servic*e, dan *middleware*. Berikut ini file konfigurasi config.yml (peneliti memberi nama file config.yml)

Gambar 9. Implementasi Konfigurasi *Provider File Traefik*

Pada file konfigurasi Gambar 9 di atas, telah diatur protokol yang akan digunakan adalah http kemudian pada protokol http di deklarasikan pembuatan *routers*. *Routers* ini berisi konfigurasi entrypoint apa yang akan dipakai, rules untuk domain, *service* yang akan digunakan, dan *middleware* yang akan diterapkan. Peneliti membuat *routers* untuk *dashboard* dan *wordpress*. Keduanya memiliki *rules*, *entrypoint*, *service*, dan *middleware*. Selanjutnya, peneliti juga membuat *service* pada konfigurasi diatas. *Service* ini menggunakan *loadBalancer* dan *load balancer* akan mengarahkan permintaan user ke url yang telah di tetapkan pada file konfigurasi diatas. Kemudian peneliti juga membuat *middleware* pada baris konfigurasi diatas. *Middleware* ini yang akan memanipulasi permintaan *client* sebelum di teruskan ke server.

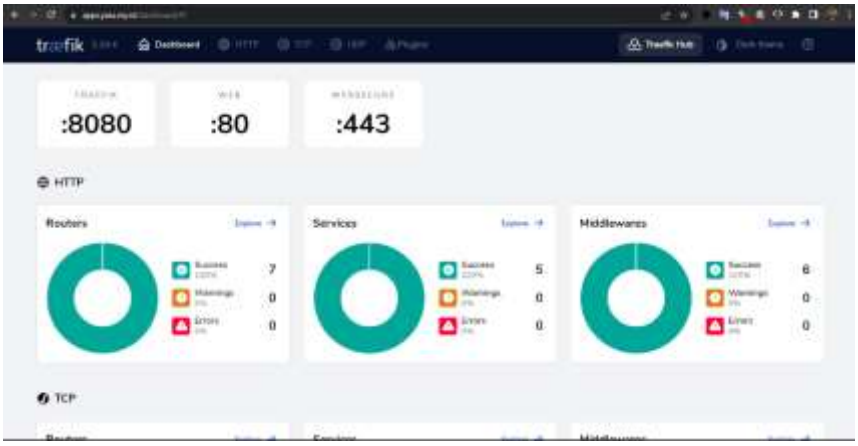
Perhatikan pada konfigurasi baris wordpress-433, pada konfigurasi ini di maksudkan bahwa *router* wordpress-433 akan di ekspose ke *public* melalui port 433 (mengambil dari *entrypoint*) kemudian rule mendefinisikan bahwa akan menggunakan domain tertera, dan service menggunakan wordpress adalah service yang telah di definisikan pada konfigurasi diatas. Maka dengan domain yang di definisikan pada rule akan diarahkan kepada *service wordpress*, dan selanjutnya *middleware* menggunakan Middleware11@etcd artinya ini adalah *providers etcd*, secara otomatis akan menggunakan *key* yang telah disimpan pada *provider etcd*.

3.4 Skenario Pengujian

Pengujian ini dilakukan untuk mengetahui hasil dari iplementasi traefik dan juga pengujian terhadap sistem *Whitelist IP* yang akan menjadi perantara untuk secara otomatis mengintegrasikan *Ip client* ke daftar *whitelist ip* yang ada pada ETCD. Sistem *Whitelist IP* ini lah yang akan menerapkan konsep *Zero Trust*, jadi hasil yang diharapkan adalah bagaimana ketika *user* atau *client* akan mengakses *service* dengan domain ketika user tidak melakukan autentikasi dan di authorize maka otomatis IP user tidak ada di *whitelist* dan akses akan ditolak atau dengan *message "403 forbidden".* Selain itu akan di coba *test scanning* dengan menggunakan *tools scanning* seperti *dirsearch* dan NMAP.

4. Hasil dan Pembahasan

Traefik memiliki halaman tampilan web yang dapat diakses sesuai dengan konfigurasi yang telah di buat. Tampilan web traefik hanya memiliki fungsi *read* saja, tidak dapat menambah ataupun menghapus data, jadi konsepnya adalah konfigurasi dan implementasi traefik yang telah di lakukan melalui server di visualisasi dalam bentuk web yang lebih mudah untuk di pahami dan dibaca oleh user.

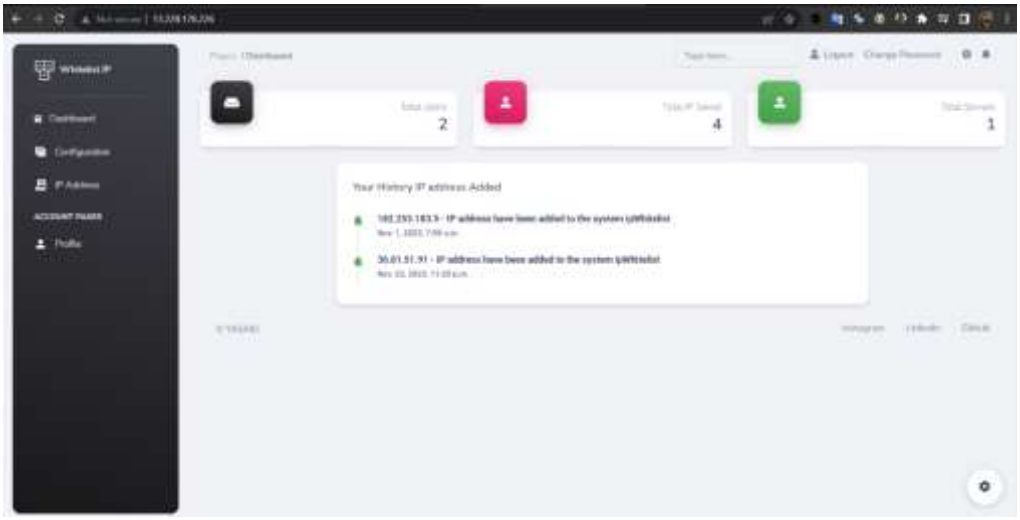


Gambar 10. Hasil implementasi *Traefik*

Gambar 10 diatas merupakan tampilan traefik melaui web dan data-data yang di tampilkan merupakan apa yang telah di konfigurasikan sebelumnya. Terdapat *Router, Service*, dan *middleware* yang telah di konfigurasikan pada file config.yml, dan informasi port pada tampilan diatas merupakan konfigurasi dari *entrypoint* yang telah di definisikan pada file traefik.yml.

4.1 Tampilan Sistem *Whitelist*

Pada halaman ini terdapat informasi dan *history* bahwa *IP address user* tersebut telah berhasil ditambkan ke dalam daftar *whitelist ip* yang berhak mengakses ke sistem atau service (dalam penelitian ini *Wordpress*). Dapat dilihat pada gambar dibawah ini, user ketika berhasil login dan di autorisasi, diarahkan ke *dashboard* dan terdapat informasi bahwa ip address user tersebut berhasil ditambahkan kedalama whitelist ip yang disimpan di server etcd.



Gambar 11. Hasil tampilan Aplikasi *whitelist IP*

Tampilan Gambar 11 merupakan tampilan untuk input detail server yang digunakan untuk menyimpan daftar *ip address user*, dalam penelitian ini merupakan server ETCD. Ketika

user login nanti maka ip address user secara otomatis akan disimpan pada server yang telah di konfigurasikan pada halaman diatas.

4.2 Pengujian Sistem

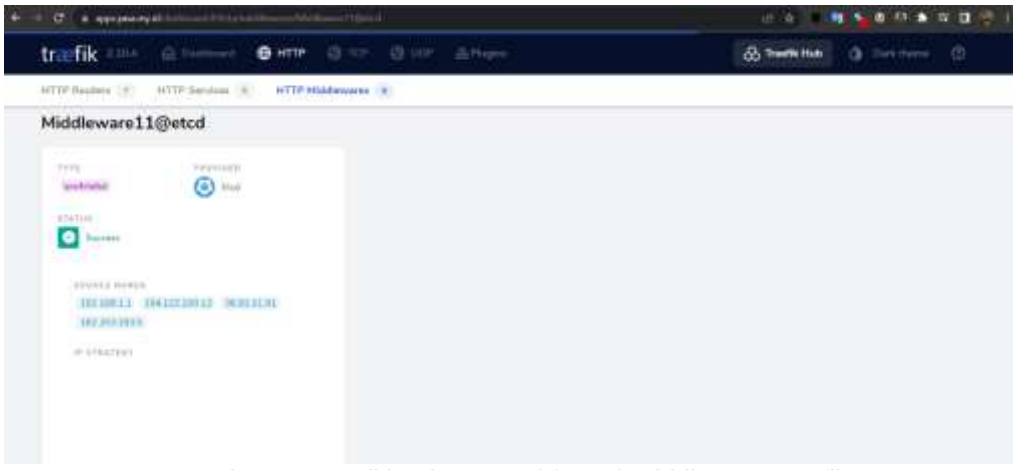
Berikut adalah table hasil pengujian sistem, pengujian ini meliputi pengujian fitur dan juga paling utama hasil pengujian untuk beberapa user Ketika berhasil login maka *ip address user* akan di *save* ke dalam *server etcd* dan *ip address* akan di simpan ke dalam database.

Tabel 2 Hasil pengujian sistem Aplikasi *Whitelist IP*

No	User	Fungsi	Hasil yang diharapkan	Status
1	Admin	Configuration	Dapat menambahkan, mengedit, dan menghapus detail dan informasi server	Valid
2	Admin	Management IP Address	Dapat melihat dan menghapus ip address yang tersimpan di database	Valid
3	All user	Profile	Dapat melihat informasi profile user tersebut	Valid
4	All user	Dashboard	Dapat melihat informasi history ip address dan pesan ip address berhasil di tambkan ke server etcd	Valid
5	Admin	Management User	Dapat menambahkan, mengedit, dan menghapus user yang ada di sistem	Valid
6	All User	Login	Dapat Login ke Ssitem dan secara otomatis Menyimpan IP Address user ke database dan server ETCD	Valid

4.3 Pengujian implementasi Traefik

Pada pengujian ini, telah dilakukan pengujian terhadap implementasi traefik dengan integrasi dengan sistem *whitelist* yang telah di uji pada sub bab sebelumnya. Hasil tujuan utama pengujian yang berhasil di dapat terdapat 2 poin yang menjadi fokus pembahasan, yakni pengujian *whitelist ip* dan juga pengujian terhadap target atau *service* yang menjadi bahan pengujian.



Gambar 12. Hasil implementasi fungsi *middleware traefik*

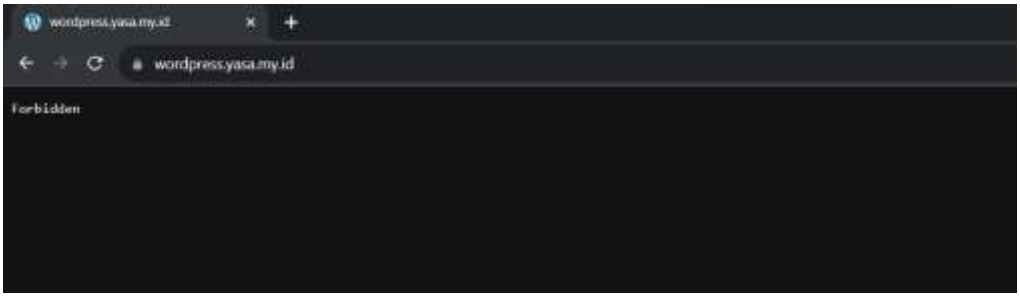
Halaman web traefik diatas, merupakan hasil pengujian di traefik pada fungsi *middlewares*. Pengujian *middleware* ini sangat penting dalam tujuan penelitian ini, bisa dilihat

pada gambar 12 terdapat *middleware* dengan nama *Middleware11@etcd* dengan *type ipwhitelist* dan *provider etcd* sesuai dengan konfigurasi dan perencanan yang dibuat pada penelitian ini.

Hasil pengujian menunjukkan *middleware* ini memiliki *source range* atau *list ip address* yang merupakan *ip address* user yang akan diijinkan mengakses ke *service (wordpress) dan ip address* ini sama dengan *ip address* yang ada di sistem *whitelist* yang telah di bahas pada sub bab sebelumnya dan *source range* ini akan selalu update jika ada perubahan atau penambahan user yang telah berhasil di authentikasi pada *sistem whitelist ip* yang telah dibuat pada sub bab sebelumnya.

Gambar 13. Hasil pengujian *user authenticated request ke service*

Pada gambar diatas user arnanda yang memiliki ip public 36.81.xx.xx telah terdaftar pada *source range* di *middleware traefik* berhasil mengakses *service wordpress* seperti gambar 13. *User* arnanda telah terdaftar pada sistem whitelist dan telah di *authentication* dan di *authorizes* sehingga dapat mengakses *service Wordpress* diatas.



Gambar 14. Hasil pengujian implementasi *whitelist ip traefik*

Pengujian diatas merupakan hasil kondisi ketika *user* dengan *IP public* yang belum terdaftar pada whitelist middleware di traefik, hasilnya adalah ketika *user* atau *client* tersebut melakukan *request* ke *service Wordpress* maka secara langsung akses tersebut ditolak karena *user* tersebut belum masuk daftar *whitelist* dan belum autentikasi dan di autorise pada sistem *whitelist ip* yang sudah dibuat pada sub bab sebelumnya. Maka jika *user* ini ingin akses nya di terima maka harus authentikasi terlebih dahulu ke sistem whitelist yang telah dibuat pada penelitian ini.

Dari hasi pengujian sistem dan hasil pengujian implementasi traefik yang telah dilakukan, konsep ini berjalan dengan baik dan berhasil meningkatkan lapisan keamanan aplikasi dan menyelesaikan masalah pembatasan akses dan *awareness* dilingkungan *developer*. Dan sistem *Whitelist IP Address* mampu menjadi solusi terhadap peran *Zero Trust* dan otomatiasi terhadap pengelolaan *IP address client*, autentikasi *user*, dan autorisasi *user* lebih efektif dan efisien agar dapat memiliki akses yang sah ke aplikasi.

Gambar 15. Gambar *scanning* dan *enumeration aplikasi*

Pada Gambar 15 menunjukkan bahwa, dengan menggunakna *tools scanning* dan *enumeration* yang biasa di pakai oleh seorang *hacker* untuk mendapatkan informasi aplikasi, hasilnya tidak ditemukan informasi yang berkaitan dengan aplikasi dan informasi yang didapat tidak yang seharusnya informasi penting tentang aplikasi karena dari awal akses masuk ke aplikasi sudah di tolak oleh traefik karena permintaan dari *tools scanning* tidak sah dan tidak di autorisasi oleh sistem *whitelist* dan di anggap permintaan *client* tersebut merupakan akses yang tidak sah untuk diijinkan, hasil ini juga menunjukkan bahwa konsep dan model sistem ini mampu mengurangi resiko dari serangan *cyber* secara langsung ke aplikasi oleh *hacker*.

Penelitian ini menemukan bahwa hasil dari pengujian sistem, pengujian implementasi traefik, dan percobaan *tools scanning* hasilnya mampu meningkatkan keamanan aplikasi dengan *filtering* yang sukses teradap permintaan yang masuk dan penolakan akses tidak sah. Temuan ini sejalan dengan penelitian terdahulu yang menjadi tinjuan pustaka dan terdapat temuan baru yang berbeda dari penelitian-penelitian terdaulu. Pertama, *traefik* mampu melakukan *whitelist IP* dengan menolak akses dengan response http *"403 forbidden"* dan membuka akses secara langsung dengan di integrasikan dengan sistem yang di kembangkan untuk melakukan otomatisasi dan mengelola *IP Address client.* Kedua, selain menolak akses untuk keamanan aplikasi, hasil scanning menunjukkan aplikasi tidak bisa secara langsung dapat di *scan* oleh *tools* untuk *information gathering* karena informasi yang didapat tidak *valid* sama sekali. Dengan berhasilnya hasil penelitian ini, maka mampu menjadi solusi dalam permasalahan keamanan aplikasi baik dari tekonologi infrastruktur dengan *traefik* sebagai *reverse proxy* nya, proses pengembangan dengan integrasi sistem *whitelist IP* membuat automatisasi dan pengelolaan efektif dan efisien dan *people* dengan *mindset security awareness* yang terbentuk secara alami berdasarkan *knowledge* pada implementasi ini.

5. Simpulan

Penerapan *traefik* sebagai *reverse proxy* dalam pengembangan aplikasi memberikan hasil peningkatan keamanan pada aplikasi itu sendiri. Fungsi *middleware* yang terdapat pada traefik memberikan keamanan pada aplikasi dengan *control access* berdasarkan ip dan mem-*filter request* yang masuk. Dengan membuat aplikasi *whitelist IP* dan diintegrasikan dengan implementasi traefik mampu menambah lapisan keamanan dan automasi yang efisian dalam lingkup keamanan aplikasi. User terlebih dahulu harus melakukan *authentication* dan di *authorize* pada aplikasi *whitelist ip* baru kemudian ip address akan ditambahkan ke dalam daftar whitelist dan permintaan ke service akan diijinkan, terbukti *request* masuk yang berasal dari *user* yang belum di ketahui dan belum dipercaya langsung ditolak dengan pesan *response forbiden*.

Daftar Referensi

[1] M. Wahyudi, "Analisis Performa Access Control List menggunakan Metode Firewall Policy Base Performance Analysis of the Access Control List Using the Firewall Policy-Based Method Article Info ABSTRAK," *Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, vol. 20, no. 2, pp. 283–292, 2021, doi: 10.30812/matrik.v20i1.1068.

[2] B. S. Renuka and G. T. Prafulla Shashikiran, "Model of Load Distribution Between Web Proxy Servers Using Network Traffic Analysis," *SN Comput Sci*, vol. 1, no. 2, pp. 1-8, Mar. 2020, doi: 10.1007/s42979-020-0108-7.

[3] S. M. Hosseini, A. H. Jahangir, and S. Daraby, "Session-persistent Load Balancing for Clustered Web Servers without Acting as a Reverse-proxy," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 360–364. doi: 10.23919/CNSM52442.2021.9615592.

[4] W. Ramadhani, M. Arif, and F. Ridha, “Jurnal Politeknik Caltex Riau Perbandingan Kinerja Ingress Controller Pada Kubernetes Menggunakan Traefik Dan Nginx,” vol. 8, no. 2, pp. 289-295 2022. [Online]. Available: https://jurnal.pcr.ac.id/index.php/jkt/

[5] A. Jeffery, H. Howard, and R. Mortier, “Rearchitecting Kubernetes for the Edge,” in *EdgeSys 2021 - Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2021*, Association for Computing Machinery, Inc, Apr. 2021, pp. 7–12. doi: 10.1145/3434770.3459730.

[6] S. Mandal, D. A. Khan, and S. Jain, “Cloud-Based Zero Trust Access Control Policy: An Approach to Support Work-From-Home Driven by COVID-19 Pandemic,” *New Gener Comput*, vol. 39, no. 3–4, pp. 599–622, Nov. 2021, doi: 10.1007/s00354-021-00130-6.

[7] O. C. Edo, T. Tenebe, E. Etu, A. Ayuwu, J. Emakhu, and S. Adebiyi, “Zero Trust Architecture: Trend and Impacton Information Security,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 7, pp. 140–147, Jul. 2022, doi: 10.46338/ijetae0722_15.

[8] Lakhno V *et al.*, “Experimental Studies Of The Features Of Using Waf To Protect Internal Services In The Zero Trust Structure,” *J Theor Appl Inf Technol*, vol. 15, no. 3, pp. 705–721, Feb. 2022, [Online]. Available: www.jatit.org

[9] R. A. Muzaki, O. C. Briliyant, M. A. Hasditama, and H. Ritchi, “Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall,” in *2020 International Workshop on Big Data and Information Security (IWBIS)*, 2020, pp. 85–90. doi: 10.1109/IWBIS50925.2020.9255601.

[10] Z. Chen, L. Han, Y. Xu, and Z. Yu, “Design and Implementation of A Vulnerability-Tolerant Reverse Proxy Based on Moving Target Defense for E-Government Application,” in *2021 2nd Information Communication Technologies Conference (ICTC)*, 2021, pp. 270–273. doi: 10.1109/ICTC51749.2021.9441622.

[11] A. Rosyida Zain, I. Muhamad, M. Matin, and D. K. Kautsar, “Analisis Implementasi Modsecurity dan Reverse Proxy Untuk Pencegahan Serangan Keamanan DDoS pada Web Server,” in *SNIV: Seminar Nasional Inovasi Vokasi*, 2023, pp. 118–127.

[12] I. Yuliarto and E. Kaburuan, “Implementasi Reverse Proxy Server Sebagai Load Balancing Dan Https Proxy Menggunakan Nginx Pada Google Cloud Platform,” Skripsi, Universitas Mercu Buana Jakarta, Jakarta, 2021. [Online]. Available: https://lib.mercubuana.ac.id/

[13] R. Maulana, M. Hatta, and I. Syafrinal, “Analisa Penerapan Filtering Proxy Server Pada Keamanan Jaringan Komputer Untuk Meminimalisr Penyebaran Malware (Studi Kasus Cakrabuana Cruiseship & School Cirebon),” vol. 12, no. 1, pp. 64–73, Sep. 2021, [Online]. Available: https://jurnal.umj.ac.id/index.php/just-it/index

[14] M. Xu, J. Guo, H. Yuan, and X. Yang, “Zero-Trust Security Authentication Based on SPA and Endogenous Security Architecture,” *Electronics (Switzerland)*, vol. 12, no. 4, pp. 1–21, Feb. 2023, doi: 10.3390/electronics12040782.

[15] Jasson Casey, “The rise of Zero Trust authentication,” February 7, 2023. Accessed: Nov. 28, 2023. [Online]. Available: https://aijourn.com/the-rise-of-zero-trust-authentication/

[16] A. Raina, “How To Use Traefik v2 as a Reverse Proxy for Docker,” Collabnix. Accessed: Nov. 28, 2023. [Online]. Available: https://collabnix.com/how-to-use-traefik-v2-as-a-reverse-proxy-for-docker/