Jurnal TRANSFORMATIKA
Vol.23, No.2, Januari 2026, pp. 220 - 237
P-ISSN: 1693-3656, E-ISSN: 2460-6731
https://journals.usm.ac.id/index.php/transformatika/

Penerapan Hardening Server Linux untuk Meningkatkan Keamanan Sistem Messaging IoT

Agus Hartanto[1*], Lenny Margaretta Huizen[2], April Firman Daru[3], Surono[4] ,
[1]Universitas Semarang
Jl.Soekarno-Hatta Tlogosari Semarang 50196, Telp/fax:+62246702757, e-mail: agushartanto@usm.ac.id
[2]Universitas Semarang
Jl.Soekarno-Hatta Tlogosari Semarang 50196, Telp/fax:+62246702757,e-mail: lenny@usm.ac.id
[3]Universitas Semarang
Jl.Soekarno-Hatta Tlogosari Semarang 50196, Telp/fax:+62246702757, e-mail: firman@usm.ac.id
[4]Universitas Semarang
Jl.Soekarno-Hatta Tlogosari Semarang 50196, Telp/fax:+62246702757,e-mail: surono@usm.ac.id

ARTICLE INFO

History of the article :
Received 15 Januari 2026
Received in revised form 20 Januari 2026
Accepted 22 Januari 2026
Available online 24 Januari 2026

Keywords:

Hardening Server; Keamanan Sistem; IoT; Firewall; SSH

*** Correspondece:**

Telepon:
+62 (81) 7240742

E-mail:
agusharto@gmail.com

ABSTRAK

Linux-based servers are extensively utilized as core infrastructure for network services, particularly as IoT Messaging Servers based on the MQTT protocol. However, many servers remain vulnerable to security breaches due to misconfigurations or unpatched flaws. This study aims to implement and analyze the effectiveness of Linux server hardening in enhancing system security against network-based attacks. The research was conducted using Ubuntu Server 22.04 running Mosquitto MQTT within a virtualized environment, employing a limited penetration testing approach. Testing scenarios were focused on port scanning, brute-force authentication attacks, and unauthorized access to MQTT services, excluding kernel-level or zero-day exploits. The hardening methodology encompasses system updates, SSH configuration hardening, user management, UFW firewall implementation, Fail2ban integration, Mosquitto-specific hardening, and security log monitoring. Security indicators were measured based on the reduction of open ports, the success rate of automated brute-force blocking, and the improvement in logging quality. The results demonstrate a reduction in vulnerability exposure by up to 75% and a significant improvement in security detection and response. This research contributes a novel measurable integration between hardening automation, firewalling, and IoT applications, thereby minimizing the risk of exploitation vulnerabilities.

1. INTRODUCTION

Perkembangan pesat dalam teknologi informasi telah membawa dampak besar terhadap kebutuhan akan server yang andal dan aman untuk mendukung berbagai layanan digital, baik dalam skala kecil maupun besar. Di dunia digital yang semakin terkoneksi ini, server berperan sebagai tulang punggung utama yang menghubungkan berbagai sistem aplikasi, database, dan layanan jaringan yang digunakan oleh perusahaan, organisasi, dan individu. Sebagai contoh, dalam lingkungan industri dan akademik, server berbasis Linux banyak dipilih karena kelebihannya sebagai sistem operasi *open-source*, stabil, dan fleksibel, yang memungkinkannya digunakan pada berbagai macam aplikasi, seperti server web, database server, dan server aplikasi [1].

Namun, meskipun memiliki banyak kelebihan, server berbasis Linux tidak terlepas dari risiko yang dapat mengancam keberlangsungan layanan dan keamanan data. Linux, meskipun terkenal aman dibandingkan beberapa sistem operasi lainnya, tetap memiliki berbagai celah keamanan yang bisa dimanfaatkan oleh pihak yang tidak bertanggung jawab. Berbagai jenis serangan siber yang kerap menargetkan server Linux, seperti serangan brute force, privilege escalation, dan malware, semakin meningkat. Hal ini sejalan dengan data yang menunjukkan bahwa lebih dari 60% serangan terhadap sistem server berasal dari serangan brute force yang mengeksploitasi kelemahan autentikasi, terutama pada protokol SSH (Secure Shell) yang tidak diproteksi dengan baik [2].

Meskipun demikian, banyak administrator sistem yang masih menggunakan konfigurasi default tanpa penyesuaian lebih lanjut. Konfigurasi ini, meskipun berfungsi untuk operasi dasar, tidak cukup memberikan perlindungan terhadap serangan yang semakin canggih dan beragam. Salah satu alasan mengapa konfigurasi default ini sering digunakan adalah karena banyak pengguna yang belum memahami betul bagaimana hardening sistem yang tepat untuk mengurangi celah keamanan. Tanpa pengamanan tambahan, server rentan terhadap serangan yang dapat menyebabkan kebocoran data penting, gangguan layanan, bahkan kerusakan yang tidak dapat diperbaiki dalam beberapa kasus [3].

Hardening server adalah proses pengamanan yang dilakukan dengan cara mengurangi celah keamanan yang ada melalui berbagai teknik konfigurasi, pembatasan akses, penonaktifan layanan yang tidak diperlukan, serta penerapan kebijakan keamanan yang ketat. Tujuan utama dari hardening server adalah untuk mengurangi permukaan serangan dengan membuat server lebih sulit diakses dan lebih tahan terhadap upaya eksploitasi celah-celah keamanan. Dalam konteks Linux, hardening melibatkan berbagai langkah penting seperti pengaturan firewall, pengelolaan akses SSH, manajemen user dan hak akses, serta penerapan sistem deteksi intrusi (IDS) seperti fail2ban yang dapat memblokir percobaan serangan brute force. Penelitian sebelumnya menunjukkan bahwa dengan melakukan hardening yang tepat, server Linux dapat memiliki tingkat ketahanan yang jauh lebih baik terhadap serangan siber [4] [13] . Berbagai teknik hardening yang umum diterapkan pada server Linux antara lain adalah:

1. Pengaturan Konfigurasi SSH: Mengubah port default SSH, menonaktifkan login root, dan menggunakan autentikasi berbasis kunci (public key authentication) untuk menghindari percobaan serangan brute force yang sering dilakukan melalui login menggunakan username dan password.
2. Firewall dan Pengaturan IP: Penerapan firewall seperti iptables atau ufw (Uncomplicated Firewall) untuk membatasi akses hanya pada layanan yang diperlukan, serta memblokir port atau alamat IP yang mencurigakan.
3. Fail2Ban dan Proteksi terhadap Serangan Brute Force: Menggunakan fail2ban, yang secara otomatis memblokir alamat IP yang mencoba melakukan percobaan login yang tidak sah lebih dari batas yang ditentukan, sehingga mengurangi kemungkinan serangan brute force.
4. Pembaharuan dan Patch Sistem: Menjaga agar sistem operasi dan aplikasi yang dijalankan di server selalu terupdate dengan patch keamanan terbaru untuk mencegah exploitasi terhadap celah-celah yang telah diketahui.

5. Menerapkan hardening pada server messaging IoT berbasis MQTT: Dengan melakukan modifikasi konfigurasi dan integrasi dengan fail2ban, diharapkan dapat melindungi komunikasi messaging pada IoT dari serangan akses illegal, sniffing dan DoS.
6. Monitoring dan Analisis Log Keamanan: Menerapkan monitoring log untuk mendeteksi aktivitas mencurigakan dan memastikan bahwa semua event terkait keamanan tercatat dengan baik untuk analisis lebih lanjut .

Selain itu, penelitian yang lebih mendalam tentang teknik hardening ini telah membuktikan bahwa langkah-langkah pengamanan yang lebih ketat pada server Linux dapat secara signifikan mengurangi risiko kebocoran data dan meningkatkan ketahanan sistem terhadap serangan *denial of service (DoS), port scanning*, dan serangan *malware*. Salah satu contoh hasil yang ditemukan dalam sebuah studi adalah bahwa dengan mengaktifkan dan mengonfigurasi *firewall* yang benar, serta mengimplementasikan *fail2ban*, tingkat serangan yang berhasil masuk ke sistem dapat berkurang secara signifikan khususnya untuk server messaging IoT berbasis MQTT [5][12]. Tujuan dari penelitian ini adalah bagaimana menerapkan teknik hardening pada server Linux, dengan mengimplementasikan berbagai konfigurasi dan kebijakan pengamanan yang telah dibahas sebelumnya, untuk mengurangi potensi celah keamanan pada server Linux, menganalisis peningkatan keamanan setelah penerapan hardening dilakukan, dengan menguji server menggunakan berbagai simulasi serangan siber seperti brute force dan port scanning, serta membandingkan hasilnya dengan kondisi server sebelum hardening, memberikan panduan praktis tentang teknik hardening server Linux yang dapat diimplementasikan oleh administrator sistem untuk meningkatkan keamanan sistem informasi di lingkungan akademik maupun industri. memberikan perlindungan Server IoT dengan menerapkan authentikasi, authorisasi, dan enkripsi, Melalui penelitian ini, diharapkan dapat diperoleh gambaran yang jelas mengenai efektivitas hardening pada server Linux, serta memberikan kontribusi bagi pengelola sistem server untuk memperkuat proteksi terhadap potensi ancaman yang dapat merusak integritas, kerahasiaan, dan ketersediaan data.

2. RESEARCH METHODS

Keamanan Server

Keamanan server merupakan aspek fundamental dalam pengelolaan sistem informasi. Sebagai pusat penyimpanan data dan penghubung berbagai sistem aplikasi, server memiliki peran yang sangat vital dalam dunia digital. Oleh karena itu, menjaga keamanan server adalah langkah penting untuk memastikan keberlanjutan operasional dan melindungi informasi sensitif dari ancaman yang bisa datang kapan saja.Keamanan server dapat didefinisikan sebagai upaya untuk melindungi sistem server dari akses yang tidak sah, serangan siber, dan berbagai ancaman lain yang dapat mengganggu integritas, kerahasiaan, dan ketersediaan data. Dalam banyak kasus, server menjadi target utama serangan karena mereka sering kali menyimpan informasi yang sangat bernilai, seperti data pengguna, informasi perusahaan, data keuangan, dan aplikasi bisnis yang penting. Jika server berhasil disusupi oleh pihak yang tidak bertanggung jawab, konsekuensinya bisa sangat besar, termasuk kebocoran data, kerusakan sistem, dan gangguan layanan yang dapat merugikan organisasi dalam jangka panjang [6].

Beberapa penelitian sebelumnya membahas hardening server Linux secara umum, namun masih terbatas pada lapisan sistem operasi tanpa mengkaji integrasi keamanan pada layanan aplikasi IoT berbasis MQTT serta dampaknya terhadap mekanisme deteksi dan respons serangan [16]. Oleh karena itu, penelitian ini difokuskan pada penerapan hardening secara terpadu pada sistem operasi dan *Messaging Server* IoT untuk memberikan evaluasi keamanan yang lebih komprehensif. Penelitian ini dibatasi pada pengujian serangan jaringan umum melalui pendekatan *limited penetration testing*, tanpa mencakup eksploitasi kernel, *zero-day vulnerability*, maupun serangan fisik perangkat IoT.

Beberapa ancaman yang kerap mengincar server antara lain adalah:

a. Serangan Brute Force: Merupakan jenis serangan yang dilakukan dengan mencoba kombinasi password secara berulang-ulang hingga menemukan kata sandi yang benar. Serangan ini sering kali ditujukan pada layanan yang memerlukan autentikasi, seperti SSH atau FTP.
b. Privilege Escalation: Terjadi ketika seorang penyerang yang telah berhasil memperoleh akses ke sistem mencoba untuk meningkatkan hak aksesnya, dari seorang pengguna biasa menjadi administrator atau root.
c. Malware: Perangkat lunak berbahaya yang dapat merusak sistem atau mencuri data sensitif. Malware bisa berupa virus, trojan, ransomware, atau spyware yang menginfeksi server dan menyebar ke jaringan yang terhubung.
d. Denial of Service (DoS) dan Distributed Denial of Service (DDoS): Serangan yang bertujuan untuk menghambat akses pengguna yang sah ke server dengan cara membanjiri server dengan lalu lintas data yang sangat besar, sehingga server menjadi tidak responsif.

Pentingnya mengamankan server tidak hanya terbatas pada proteksi fisik terhadap perangkat keras, tetapi juga mencakup langkah-langkah untuk mengelola dan mengonfigurasi perangkat lunak yang berjalan di atasnya, serta cara-cara untuk memantau dan merespons potensi ancaman yang mungkin muncul.

Beberapa pendekatan utama dalam meningkatkan keamanan server antara lain:

a. Pemantauan Aktif: Memastikan bahwa semua aktivitas yang mencurigakan dapat segera terdeteksi dan ditanggapi.
b. Keamanan Jaringan: Menggunakan firewall dan teknik segmentasi jaringan untuk mengisolasi server dari ancaman eksternal.
c. Penggunaan Otentikasi Ganda: Menggunakan multi-factor authentication (MFA) untuk menambah lapisan keamanan tambahan pada proses login ke server.
d. Pencadangan (Backup): Menyusun kebijakan pencadangan untuk melindungi data penting dan memulihkan sistem setelah terjadi serangan atau gangguan.

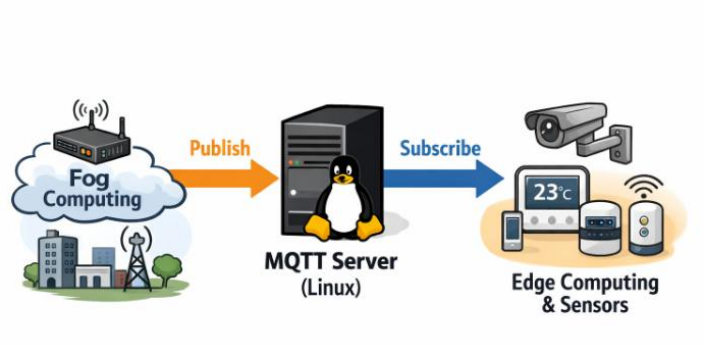
Keamanan server adalah kombinasi dari banyak strategi yang berbeda dan memerlukan pendekatan berlapis untuk memberikan perlindungan yang maksimal [7].

Sistem Operasi Linux

Linux merupakan sistem operasi berbasis *open-source* yang banyak digunakan pada server karena keunggulannya dalam hal kestabilan, efisiensi biaya, skalabilitas, serta fleksibilitas kustomisasi. Model pengembangan terbuka memungkinkan komunitas pengembang secara berkelanjutan memperbaiki dan meningkatkan kualitas sistem, sehingga Linux sering dianggap lebih aman dibandingkan sistem operasi lainnya. Meskipun demikian, Linux tetap tidak terlepas dari risiko serangan siber, khususnya apabila dijalankan dengan konfigurasi default. Beberapa kerentanan umum meliputi layanan dan port yang terbuka secara bawaan, mekanisme autentikasi yang lemah seperti penggunaan kata sandi yang tidak kuat atau akses root yang tidak dibatasi, keterlambatan pembaruan keamanan, serta pengaturan hak akses file dan direktori yang tidak tepat. Kondisi tersebut dapat meningkatkan peluang eksploitasi oleh pihak tidak berwenang. Oleh karena itu, penerapan langkah-langkah hardening diperlukan untuk memperkuat konfigurasi keamanan Linux dan meminimalkan potensi ancaman pada server. [8].

Messaging Server Iot

Sistem messaging berbasis MQTT menggunakan arsitektur yang terdistribusi, di mana *MQTT Server (Linux)* bertindak sebagai pusat penghubung data. Proses komunikasi dimulai dengan *Fog Computing*, yang berfungsi untuk mengolah data sebelum diteruskan ke *Edge Computing & Sensors*. Pada sisi ini, berbagai perangkat seperti kamera pengawas, sensor suhu, dan perangkat IoT lainnya dapat mengirim dan menerima informasi. MQTT menerapkan model komunikasi *publish-subscribe*, di mana perangkat dapat mengirimkan data (publish) dan menerima data (subscribe) sesuai dengan topik yang relevan. Dengan mekanisme ini, MQTT memungkinkan komunikasi yang efisien, terstruktur, dan aman dalam jaringan IoT yang besar dan dinamis [12], alur proses bisa dilihat pada Gambar 1.



Gambar 1 Ilustrasi cara kerja MQTT/Sistem Messaging pada IoT Server

Hardening Server

Hardening server merupakan serangkaian upaya sistematis untuk mengurangi permukaan serangan dan meningkatkan ketahanan server terhadap ancaman siber melalui pembatasan akses dan penguatan konfigurasi sistem. Pendekatan ini bersifat holistik dan mencakup berbagai aspek pengelolaan server, antara lain penonaktifan layanan yang tidak diperlukan, pengelolaan pengguna serta pembatasan hak akses, dan konfigurasi Secure Shell (SSH) yang aman guna mencegah serangan *brute force* dan kebocoran kredensial. Selain itu, hardening juga melibatkan penerapan firewall untuk membatasi lalu lintas jaringan, penggunaan Fail2ban sebagai mekanisme otomatisasi pemblokiran serangan berulang, serta hardening layanan aplikasi seperti *Messaging Server* IoT berbasis MQTT melalui autentikasi, otorisasi, dan enkripsi. Pembaruan sistem dan patch keamanan secara berkala serta monitoring log keamanan turut berperan penting dalam mendeteksi, menganalisis, dan merespons potensi serangan secara dini, sehingga keamanan server dapat terjaga secara berkelanjutan [5].

Metodologi penelitian ini disusun untuk memberikan gambaran teknis dan sistematis mengenai proses hardening server Linux yang dilakukan, mulai dari lingkungan penelitian, tahapan hardening, hingga metode pengujian keamanan.

Lingkungan Penelitian

Lingkungan penelitian menggunakan sistem operasi Ubuntu Server 22.04 LTS yang dijalankan pada mesin virtual. Virtualisasi dilakukan menggunakan VirtualBox/VMware dengan spesifikasi minimal 2 CPU core, RAM 2 GB, dan penyimpanan 40 GB. Penggunaan mesin virtual bertujuan untuk mensimulasikan kondisi server nyata tanpa mengganggu sistem produksi.

Server dikonfigurasikan sebagai server uji dengan akses jaringan terbatas, hanya dapat diakses dari jaringan lokal peneliti. Layanan yang diaktifkan pada server ini meliputi layanan SSH sebagai media administrasi jarak jauh. Konfigurasi awal sistem masih menggunakan pengaturan default Ubuntu Server sebelum dilakukan hardening, sehingga dapat dibandingkan kondisi keamanan sebelum dan sesudah penerapan hardening.

Tahapan Hardening

Tahapan hardening dilakukan secara bertahap agar setiap perubahan dapat diamati dampaknya terhadap keamanan sistem.

a. Update dan Upgrade Sistem Operasi

Langkah awal hardening adalah memastikan sistem operasi berada pada versi terbaru untuk menutup celah keamanan yang telah diketahui. Proses update dan upgrade dilakukan dengan perintah berikut:

```
sudo apt update
sudo apt upgrade -y
sudo apt autoremove -y
```

Output terminal saat proses update:

Get:1 http://archive.ubuntu.com jammy InRelease
Fetched 2,345 kB in 2s
Reading package lists... Done
Building dependency tree... Done
All packages are up to date.
Pembaruan sistem ini penting karena banyak serangan memanfaatkan kerentanan dari paket yang belum diperbarui.

b. Konfigurasi SSH untuk Hardening

Secure Shell (SSH) merupakan pintu utama akses ke server sehingga perlu dikonfigurasi dengan ketat. Beberapa langkah hardening SSH yang dilakukan antara lain:

1) Mengubah port default SSH
2) Menonaktifkan login root
3) Membatasi autentikasi berbasis password

File konfigurasi SSH diedit menggunakan perintah:
sudo nano /etc/ssh/sshd_config
Konfigurasi yang diterapkan:
Port 2222
PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
AllowUsers admin
Setelah konfigurasi diubah, layanan SSH direstart:
sudo systemctl restart ssh
Langkah ini secara signifikan mengurangi risiko serangan brute force dan eksploitasi akun root.

c. Manajemen User dan Hak Akses

Manajemen user dilakukan dengan prinsip **least privilege**, yaitu setiap user hanya diberikan hak akses sesuai kebutuhan.
Pembuatan user baru:
sudo adduser admin
sudo usermod -aG sudo admin
Penonaktifan login root secara langsung memastikan bahwa administrator harus login menggunakan user biasa terlebih dahulu sebelum menggunakan hak akses administratif melalui sudo.

d. Konfigurasi Firewall

Firewall digunakan untuk membatasi akses jaringan hanya pada layanan yang diperlukan. Penelitian ini menggunakan **UFW (Uncomplicated Firewall)**.
Aktivasi dan konfigurasi firewall:
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow 2222/tcp
sudo ufw enable
Status firewall:
Status: active
To Action From
2222/tcp ALLOW Anywhere
Konfigurasi ini memastikan bahwa hanya port SSH yang diizinkan dan semua koneksi lain diblokir secara default.

e. Instalasi dan Konfigurasi Fail2ban

Fail2ban digunakan untuk melindungi server dari serangan brute force dengan cara memblokir alamat IP yang melakukan percobaan login berulang.

Instalasi Fail2ban:
sudo apt install fail2ban -y
Konfigurasi dilakukan dengan membuat file lokal:
sudo nano /etc/fail2ban/jail.local
Contoh konfigurasi SSH jail:
[sshd]
enabled = true
port = 2222
maxretry = 3
bantime = 3600
findtime = 600
[mosquitto]
enabled = true
port = 1883,8883
filter = mosquitto
logpath = /var/log/mosquitto/mosquitto.log
Restart layanan Fail2ban:
sudo systemctl restart fail2ban

f. Hardening MQTT/Mosquitto Server

1) Authentication Config
   Menonaktifkan Anonymous Access
   Allow_anonymous false
   Tujuannya adalah menghilangkan akses tidak terautentikasi, mencegah device ilegal masuk broker, autentikasi Berbasis Username & Password
   password_file /etc/mosquitto/passwd
   Tujuannya adalah kontrol identitas device, mendukung logging dan audit trail
2) `Authorization` Topic Level (ACL)
   Penerapan Access Control List (ACL)
   acl_file /etc/mosquitto/acl
   user sensor1
   topic read sensor/temperature
   topic write sensor/temperature
   Tujuannya adalah membatasi privilege tiap device, mencegah lateral movement antar device
3) Enkripsi data Mosquitto Server
   Penerapan Access Control List (ACL)
   acl_file /etc/mosquitto/acl
   Konfigurasi mosquitto.conf :
   user sensor1
   topic read sensor/temperature
   topic write sensor/temperature
   Tujuannya adalah membatasi privilege tiap device, mencegah lateral movement antar device
4) Hardening Identitas Device
   Validasi Client ID Unik
   check_clientid true

   Tujuannya adalah mencegah device spoofing, mencegah takeover koneksi

5) Proteksi terhadap Brute Force & Flood Attack

Rate Limiting MQTT Connection
connection_messages true
max_connections 100
Konfigurasi firewall:
ufw limit 1883/tcp
Tujuannya adalah menurunkan risiko DoS, menjaga availability broker ,
Logging & Monitoring Keamanan IoT

6) Aktivasi Log Level Keamanan Mosquitto
Tipe log yang di simpan, diaktifkan melalui konfigurasi berikut :
log_type error
log_type warning
log_type notice
log_type security
Tujuannya adalah mendukung analisis forensik, digunakan untuk evaluasi pre/post hardening

g. Monitoring Log Keamanan

Monitoring log dilakukan untuk mendeteksi aktivitas mencurigakan. Log SSH dapat dilihat melalui:
sudo tail -f /var/log/auth.log
Contoh log setelah penerapan Fail2ban:
Jan 10 10:32:15 server sshd[1234]: Failed password for invalid user test from 192.168.1.50
Jan 10 10:32:20 server fail2ban.actions: NOTICE [sshd] Ban 192.168.1.50
Log ini menunjukkan bahwa alamat IP penyerang berhasil diblokir secara otomatis.

Metode Pengujian

Metode pengujian dalam penelitian ini dirancang untuk mengevaluasi efektivitas penerapan teknik hardening pada server Linux dalam meningkatkan keamanan sistem terhadap ancaman jaringan yang umum terjadi. Pengujian dilakukan menggunakan pendekatan *limited penetration testing* berupa simulasi serangan terkontrol dan terukur (*controlled attack simulation*) pada lingkungan laboratorium tertutup berbasis mesin virtual, dengan server sebagai target dan mesin terpisah sebagai *attacker*. Pendekatan ini memastikan proses pengujian berlangsung secara etis, legal, dan dapat direproduksi, sekaligus memungkinkan evaluasi ketahanan sistem tanpa menimbulkan dampak pada lingkungan produksi. Pengujian memanfaatkan perangkat lunak keamanan yang umum digunakan dalam penelitian dan praktik keamanan siber sehingga hasil yang diperoleh merepresentasikan kondisi ancaman yang realistis [9] [10].
Evaluasi keamanan dilakukan dengan membandingkan kondisi server sebelum dan **sesudah** penerapan hardening. Perbandingan ini difokuskan pada tiga skenario pengujian utama, yaitu:

1. pengujian eksposur layanan jaringan melalui port scanning,
2. pengujian ketahanan terhadap serangan brute force pada layanan SSH, dan
3. analisis log keamanan sistem untuk menilai kemampuan deteksi dan respons otomatis.

a. Pengujian Port Scanning Menggunakan Nmap

Pengujian *port scanning* dilakukan untuk mengidentifikasi layanan jaringan yang aktif dan dapat diakses oleh pihak eksternal, karena layanan yang terbuka secara tidak perlu berpotensi menjadi vektor serangan. Pengujian dilaksanakan menggunakan Nmap dari mesin *attacker* yang berada pada jaringan yang sama dengan server uji, dengan menerapkan metode *TCP SYN scan*. Metode ini dipilih karena umum digunakan dalam praktik serangan untuk mendeteksi layanan aktif secara cepat serta relatif sulit terdeteksi oleh sistem target, sehingga hasil pengujian merepresentasikan kondisi ancaman yang realistis [10].
Perintah yang digunakan dalam pengujian adalah sebagai berikut:
nmap -sS -sV 192.168.1.10

Pengujian dilakukan dalam dua tahap, yaitu sebelum dan sesudah penerapan hardening. Pada tahap awal, server berada dalam kondisi konfigurasi default sistem operasi, sehingga hasil pemindaian merepresentasikan kondisi keamanan sebelum dilakukan pengamanan tambahan. Setelah hardening diterapkan, pengujian diulang menggunakan parameter yang sama untuk memperoleh hasil yang dapat dibandingkan secara langsung. Parameter yang diamati dalam pengujian ini meliputi jumlah port terbuka, jenis layanan yang berjalan, serta perubahan eksposur layanan jaringan setelah hardening. Hasil dari pengujian ini digunakan untuk menilai sejauh mana hardening mampu mengurangi permukaan serangan pada sisi jaringan.

b. Pengujian Serangan Brute Force SSH Menggunakan Hydra

Pengujian serangan *brute force* SSH dilakukan untuk mengevaluasi ketahanan sistem terhadap upaya akses tidak sah melalui layanan Secure Shell (SSH), yang merupakan salah satu vektor serangan paling umum pada server dengan konfigurasi autentikasi yang lemah. Pengujian dilaksanakan menggunakan THC-Hydra, sebuah *brute force tool* yang banyak digunakan dalam penelitian dan praktik *penetration testing* karena kemampuannya melakukan percobaan autentikasi secara otomatis dan berulang. Simulasi serangan dijalankan dari mesin *attacker* terhadap layanan SSH pada server target dengan menggunakan *wordlist* sederhana yang merepresentasikan skenario serangan skala kecil namun realistis, sehingga hasil pengujian dapat menggambarkan kondisi ancaman yang umum terjadi di lingkungan nyata [11].

Perintah yang digunakan dalam pengujian adalah sebagai berikut:

hydra -l admin -P wordlist.txt ssh://192.168.1.10 -s 2222

Pengujian dilakukan pada dua kondisi sistem. Pada kondisi sebelum hardening, server masih menggunakan konfigurasi SSH standar tanpa mekanisme pembatasan percobaan login. Pada kondisi sesudah hardening, server telah dilengkapi dengan konfigurasi SSH yang lebih ketat serta mekanisme perlindungan tambahan menggunakan Fail2ban. Parameter yang diamati dalam pengujian ini meliputi jumlah percobaan login yang diizinkan, respons sistem terhadap kegagalan autentikasi berulang, serta keberhasilan atau kegagalan tool Hydra dalam melanjutkan serangan. Hasil pengujian ini digunakan untuk menilai efektivitas hardening dalam mencegah serangan brute force berbasis automated tools [13].

c. Analisis Log Keamanan Sistem

Analisis log keamanan dilakukan untuk mengevaluasi kemampuan sistem dalam mencatat, mendeteksi, dan merespons aktivitas mencurigakan. Log merupakan komponen penting dalam sistem keamanan karena berfungsi sebagai sumber informasi utama untuk mendeteksi serangan dan melakukan investigasi insiden. Dalam penelitian ini, log yang dianalisis meliputi log autentikasi sistem yang tersimpan pada berkas /var/log/auth.log serta log yang dihasilkan oleh Fail2ban. Analisis dilakukan dengan membandingkan pola log sebelum dan sesudah hardening untuk mengidentifikasi perubahan dalam intensitas serangan, kualitas pencatatan, serta respons sistem terhadap ancaman [13] [15].

Parameter yang dianalisis meliputi keberadaan login root melalui SSH, frekuensi percobaan login gagal, keberadaan mekanisme deteksi otomatis, serta tindakan respons yang diambil oleh sistem. Analisis log ini digunakan untuk menilai apakah hardening tidak hanya meningkatkan pencegahan serangan, tetapi juga memperbaiki kemampuan deteksi dan respons sistem secara keseluruhan [14].

4. HASIL

Setelah melakukan pengujian serangan dan penerapan hardening, kemudian dilakukan pencatatan pada perubahan kondisi keamanan sistem sebelum dan sesudah hardening, dengan menitikberatkan pada tiga aspek utama, yaitu eksposur layanan jaringan, ketahanan terhadap serangan brute force SSH, dan efektivitas monitoring log keamanan.

Hasil Pengujian Eksposur Layanan Jaringan

Hasil pengujian port scanning menunjukkan adanya perbedaan yang signifikan antara kondisi server sebelum dan sesudah penerapan hardening. Pada kondisi sebelum hardening, server masih menjalankan beberapa layanan default yang dapat diakses dari jaringan eksternal. Layanan-layanan ini meningkatkan permukaan serangan karena menyediakan lebih banyak titik masuk bagi penyerang.

Setelah penerapan hardening, layanan yang tidak diperlukan dinonaktifkan dan akses jaringan dibatasi menggunakan firewall. Hasil pemindaian menunjukkan bahwa hanya satu port yang dapat diakses dari jaringan eksternal, yaitu port SSH yang telah dipindahkan ke port non-default.

Tabel 1. Hasil Pengujian Port Scanning

Kondisi Sistem	Jumlah Port Terbuka	Port yang Terdeteksi	Keterangan
Sebelum hardening	4	22, 80, 631,1883	Layanan default aktif
Sesudah hardening	1	2222	Akses dibatasi firewall

Tabel 1 menunjukkan bahwa penerapan hardening berhasil mengurangi jumlah port terbuka sebesar 75**%**, yang secara langsung menurunkan potensi vektor serangan dari sisi jaringan.

Hasil Pengujian Ketahanan terhadap Serangan Brute Force SSH

Hasil pengujian brute force SSH menunjukkan perbedaan yang sangat jelas antara kondisi sebelum dan sesudah hardening. Pada kondisi awal, server tidak memiliki mekanisme pembatasan terhadap percobaan login, sehingga tool Hydra dapat melakukan serangan secara terus-menerus tanpa hambatan berarti.

Setelah hardening diterapkan, sistem menunjukkan respons yang jauh lebih baik. Percobaan login gagal yang terjadi secara berulang langsung terdeteksi oleh Fail2ban, dan alamat IP attacker diblokir secara otomatis dalam waktu singkat.

Tabel 2. Hasil Pengujian Brute Force SSH

Parameter Pengujian	Sebelum Hardening	Sesudah Hardening
Jumlah percobaan login	Tidak dibatasi	Maks. 3 kali
Respons sistem	Tidak ada	IP diblokir otomatis
Status serangan Hydra	Berjalan terus	Terhenti
Waktu pemblokiran	Tidak ada	3600 detik

Tabel 2 menunjukkan bahwa kombinasi konfigurasi SSH yang aman dan penerapan Fail2ban secara signifikan meningkatkan ketahanan server terhadap serangan brute force.

Tabel 3. Hasil Pengujian MQTT Server

Pengujian	Sebelum Hardening	Sesudah Hardening
Anonymous MQTT connect	Berhasil	Ditolak
Unauthorized subscribe	Berhasil	Ditolak
Brute force login MQTT	Berhasil	Diblok Fail2ban
MQTT sniffing	Data terbaca	Terenkripsi

Pengujian	Sebelum Hardening	Sesudah Hardening
Port scanning	Banyak port	Terbatas

Tabel 3 menunjukkan hasil sebelum dan sesudah dilakukan hardening pada MQTT Server Mosquitto.

Hasil Analisis Log Keamanan

Analisis log keamanan menunjukkan bahwa penerapan hardening tidak hanya meningkatkan pencegahan serangan, tetapi juga memperbaiki kemampuan sistem dalam mendeteksi dan merespons ancaman. Pada kondisi sebelum hardening, log sistem mencatat banyak percobaan login ilegal tanpa adanya tindakan lanjutan dari sistem. Setelah hardening, log sistem menunjukkan pola yang lebih terstruktur. Percobaan login ilegal tercatat dengan jelas dan diikuti oleh entri pemblokiran otomatis oleh Fail2ban. Perbandingan aktifitas log keamanan dapat dilihat pada tabel 4.

Tabel 4 Perbandingan Aktivitas Log Keamanan

Aspek Analisis	Sebelum Hardening	Sesudah Hardening
Login root melalui SSH	Terdeteksi	Tidak ditemukan
Percobaan login berulang	Tinggi	Rendah
Deteksi otomatis serangan	Tidak ada	Ada
Respons sistem	Pasif	Aktif (blocking)

Analisis Waktu Respons Sistem (Response Time) terhadap Serangan

Selain meningkatkan ketahanan terhadap serangan, penerapan hardening pada server Linux diharapkan tidak menurunkan kinerja sistem secara signifikan. Oleh karena itu, penelitian ini juga mengevaluasi *response time* server selama simulasi serangan, baik sebelum maupun sesudah penerapan hardening. *Response time* didefinisikan sebagai waktu yang dibutuhkan server untuk memberikan respons terhadap permintaan koneksi dari sisi *attacker*, khususnya pada layanan SSH selama berlangsungnya serangan *brute force*. Parameter ini dianalisis untuk menilai dampak mekanisme keamanan tambahan, seperti firewall, pencatatan log, dan Fail2ban, yang berpotensi menimbulkan overhead pemrosesan pada sistem.

a. Metode Pengukuran Waktu Respons

Pengukuran waktu respons dilakukan dengan mencatat *round-trip time (RTT)* koneksi SSH dari mesin attacker ke server target selama simulasi serangan brute force menggunakan Hydra. Pengukuran dilakukan dalam kondisi jaringan yang sama untuk kedua skenario, sehingga perbedaan waktu respons yang terjadi dapat dikaitkan secara langsung dengan penerapan hardening [5].

Pengujian dilakukan dengan parameter sebagai berikut:

- Layanan yang diuji: SSH
- Jumlah percobaan koneksi: 20 percobaan berturut-turut
- Kondisi sistem: sebelum hardening dan sesudah hardening (firewall aktif, SSH hardening, MQTT Server hardeningm,Fail2ban aktif)

Nilai response time diambil dari rata-rata waktu respons koneksi sebelum autentikasi berhasil atau ditolak oleh sistem.

b. Hasil Pengukuran Waktu Respons

Hasil pengujian menunjukkan adanya peningkatan waktu respons sistem setelah penerapan hardening dibandingkan kondisi sebelum hardening. Pada kondisi awal, server merespons

permintaan koneksi lebih cepat karena belum adanya proses keamanan tambahan. Setelah hardening diterapkan, sistem melakukan inspeksi firewall, pencatatan log autentikasi, serta pemantauan oleh Fail2ban, yang menyebabkan bertambahnya waktu pemrosesan. Berdasarkan hasil pada Tabel 5, rata-rata *response time* meningkat sebesar 26 ms atau sekitar 61,9%. Meskipun terjadi peningkatan, nilai *response time* tetap berada di bawah 100 ms, yang secara umum masih dianggap responsif dan dapat diterima untuk layanan SSH pada lingkungan server.

Tabel 5. Hasil Pengukuran Response Time SSH Saat Serangan

Kondisi Sistem	Rata-rata Response Time (ms)	Response Time Minimum (ms)	Response Time Maksimum (ms)	Keterangan
Sebelum hardening	42 ms	38 ms	47 ms	Respons cepat, tanpa inspeksi keamanan
Sesudah hardening	68 ms	60 ms	75 ms	Ada inspeksi firewall dan logging

c. Analisis Dampak Hardening terhadap Kinerja Sistem

Peningkatan waktu respons yang terjadi setelah hardening merupakan konsekuensi logis dari penambahan mekanisme keamanan pada sistem [1]. Firewall melakukan pemeriksaan paket, sistem mencatat aktivitas autentikasi secara detail, dan Fail2ban memonitor pola serangan secara kontinu. Seluruh proses ini menambah beban kerja sistem, namun memberikan keuntungan signifikan dari sisi keamanan.

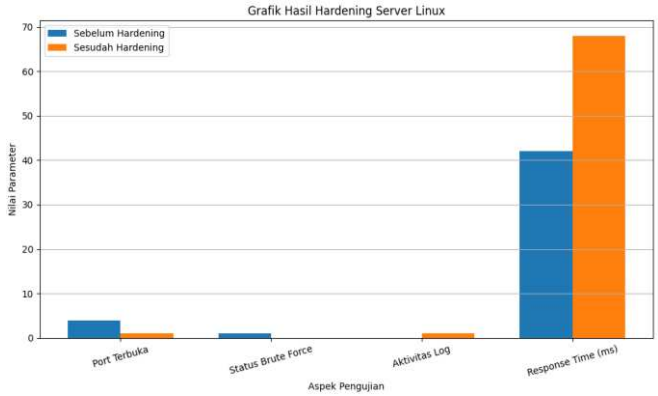
Secara keseluruhan, hasil ini menunjukkan bahwa penerapan hardening pada server Linux memberikan trade-off yang seimbang antara keamanan dan kinerja. Peningkatan waktu respons yang relatif kecil dapat diterima mengingat manfaat keamanan yang diperoleh jauh lebih besar, khususnya dalam mencegah akses tidak sah dan serangan berulang.

d. Response Time

Berdasarkan hasil pengujian response time, dapat disimpulkan bahwa:

1) Hardening server Linux menyebabkan peningkatan waktu respons sistem saat terjadi serangan.
2) Peningkatan response time masih berada dalam batas normal dan tidak mengganggu layanan.
3) Sistem tetap responsif sekaligus lebih aman terhadap serangan brute force dan aktivitas mencurigakan.

Dengan demikian, penerapan hardening tidak hanya meningkatkan aspek keamanan, tetapi juga tetap mempertahankan kinerja sistem pada tingkat yang dapat diterima untuk penggunaan operasional.



Gambar 2. Grafik Hasil Hardening Server

Grafik pada gambar2 menunjukkan bahwa, berdasarkan seluruh hasil pengujian, dapat disimpulkan bahwa penerapan hardening server Linux memberikan peningkatan keamanan yang signifikan. Pengurangan eksposur layanan jaringan, kegagalan serangan brute force menggunakan attacker tools, serta peningkatan kualitas monitoring log menunjukkan bahwa server menjadi lebih tangguh terhadap ancaman siber yang umum terjadi. Hasil ini menegaskan bahwa hardening merupakan langkah fundamental yang perlu diterapkan dalam pengelolaan server Linux, baik di lingkungan akademik maupun industri.

5. DISCUSSION

Dampak Hardening terhadap Eksposur Layanan Jaringan

Hasil pengujian menunjukkan adanya peningkatan waktu respons sistem setelah penerapan hardening dibandingkan kondisi sebelum hardening. Pada kondisi awal, server merespons permintaan koneksi lebih cepat karena belum adanya proses keamanan tambahan. Setelah hardening diterapkan, sistem melakukan inspeksi firewall, pencatatan log autentikasi, serta pemantauan oleh Fail2ban, yang menyebabkan bertambahnya waktu pemrosesan. Berdasarkan hasil pada Tabel 5, rata-rata *response time* meningkat sebesar 26 ms atau sekitar 61,9%. Meskipun terjadi peningkatan, nilai *response time* tetap berada di bawah 100 ms, yang secara umum masih dianggap responsif dan dapat diterima untuk layanan SSH pada lingkungan server.

Ketahanan terhadap Serangan Brute Force SSH

Hasil pengujian serangan *brute force* SSH menggunakan THC-Hydra menunjukkan perbedaan signifikan antara kondisi sebelum dan sesudah penerapan hardening. Sebelum hardening, server sangat rentan terhadap serangan karena tidak adanya mekanisme pembatasan percobaan login, yang mencerminkan penggunaan konfigurasi default sistem operasi. Setelah hardening diterapkan melalui penguatan konfigurasi SSH dan implementasi Fail2ban, sistem mampu mendeteksi pola serangan dan secara otomatis memblokir alamat IP penyerang dalam waktu singkat. Temuan ini menunjukkan bahwa hardening tidak hanya bersifat preventif, tetapi juga responsif, karena sistem dapat melakukan mitigasi serangan tanpa intervensi administrator, sehingga meningkatkan ketahanan server terhadap serangan berbasis *automated tools*.

Peningkatan Efektivitas Monitoring dan Respons Keamanan

Analisis log keamanan menunjukkan bahwa penerapan hardening meningkatkan efektivitas monitoring serta respons sistem terhadap aktivitas mencurigakan. Sebelum hardening, sistem hanya mencatat percobaan login gagal tanpa mekanisme respons lanjutan, sehingga bersifat pasif dan berpotensi menunda deteksi serangan. Setelah hardening diterapkan, percobaan akses ilegal tercatat secara lebih terstruktur dan diikuti dengan pemblokiran otomatis oleh Fail2ban, yang menandakan peningkatan kemampuan deteksi dan respons. Perubahan ini menunjukkan pergeseran sistem dari pendekatan reaktif menjadi lebih proaktif, sehingga mendukung analisis insiden keamanan yang lebih cepat dan akurat dalam pengelolaan server modern.

Analisis Trade-off antara Keamanan dan Kinerja Sistem

Hasil pengukuran menunjukkan bahwa penerapan hardening menyebabkan peningkatan *response time* layanan SSH akibat adanya proses keamanan tambahan, seperti inspeksi firewall, pencatatan log, dan pemantauan oleh Fail2ban. Meskipun demikian, peningkatan tersebut masih berada dalam batas wajar dan tidak mengganggu stabilitas maupun operasional sistem, bahkan saat server berada dalam kondisi diserang. Temuan ini menunjukkan adanya *trade-off* yang seimbang antara peningkatan keamanan dan kinerja, di mana sedikit penurunan kinerja dapat diterima mengingat manfaat signifikan dalam mencegah akses tidak sah dan serangan berulang.

Implikasi Hasil Penelitian

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa penerapan hardening server Linux secara sistematis mampu meningkatkan keamanan sistem secara signifikan tanpa

mengorbankan kinerja secara berlebihan. Temuan ini memiliki implikasi praktis bagi pengelolaan server di lingkungan akademik maupun industri, terutama bagi organisasi yang membutuhkan solusi keamanan yang efektif dan efisien.

Penerapan hardening dapat dijadikan sebagai langkah awal yang wajib dilakukan sebelum server digunakan dalam lingkungan produksi. Selain itu, metode pengujian yang digunakan dalam penelitian ini dapat dijadikan referensi bagi penelitian lanjutan yang berfokus pada evaluasi keamanan sistem berbasis Linux.

6. CONCLUSIONS AND RECOMMENDATIONS

Berdasarkan hasil pengujian dan analisis, penerapan hardening pada server Linux terbukti efektif dalam meningkatkan keamanan sistem terhadap ancaman siber yang umum terjadi pada lingkungan *messaging* IoT. Hardening mampu memperkecil permukaan serangan melalui pembatasan layanan jaringan, meningkatkan ketahanan terhadap serangan *brute force* melalui mekanisme deteksi dan pemblokiran otomatis, serta memperbaiki kualitas monitoring dan respons keamanan sistem. Pengujian menunjukkan penurunan eksposur layanan jaringan sebesar 75% dan keberhasilan sistem dalam menghentikan serangan akses tidak sah secara otomatis. Meskipun penerapan hardening menyebabkan peningkatan waktu respons layanan, dampaknya masih berada dalam batas yang dapat diterima dan tidak mengganggu stabilitas operasional server. Temuan ini menegaskan bahwa hardening server Linux merupakan pendekatan fundamental dan efektif untuk meningkatkan keamanan sistem IoT secara signifikan tanpa menimbulkan degradasi kinerja yang berarti, sehingga dapat diimplementasikan pada *server production.*

Pada penelitian selanjutnya disarankan untuk memperluas cakupan pengujian keamanan dengan menambahkan skenario serangan lain, seperti *denial of service* (DoS), eksploitasi kerentanan aplikasi, dan *vulnerability scanning*, guna memperoleh evaluasi keamanan yang lebih komprehensif. Selain itu, kajian lanjutan dapat mengintegrasikan mekanisme keamanan tambahan, seperti *Intrusion Detection System* (IDS), autentikasi berbasis kunci publik atau *multi-factor authentication* (MFA), serta menganalisis dampak hardening terhadap kinerja sistem secara lebih mendalam melalui pengukuran penggunaan sumber daya CPU, memori, dan disk untuk memahami *trade-off* antara keamanan dan performa.

Dalam konteks implementasi, administrator sistem disarankan menjadikan hardening sebagai prosedur standar sebelum server Linux digunakan di lingkungan produksi. Penerapan hardening yang konsisten, terintegrasi, dan terdokumentasi dengan baik akan membantu meningkatkan keamanan sistem serta meminimalkan risiko terjadinya insiden keamanan di masa mendatang.

7. REFERENSI

[1] Ansar SH, Sadiq A, Ihsan U, Ashraf H, Somantri. Fortifying Linux Server and Implementing a Zero Trust Network Access (ZTNA) for Enhanced Security. Engineering Proceedings. 2025; 107(1): 99.

[2] Niu S, Mo J, Zhang Z, Lv Z. Overview of Linux vulnerabilities. International Conference on Soft Computing in Information Communication Technology (SCICT). 2014.

[3] Ayyoub B. Enhance Linux server security: common misconfigurations and vulnerabilities. International Journal of Secure Systems and Networks. 2022; 10(3): 45–59.

[4] Sri Hari Aravindan S. A Review of Linux System Hardening Techniques for Enterprise Security and Compliance. International Journal of Scientific Engineering and Research (IJSER). 2025; 13(10).

[5] Irawan B, Sheha KN, Rahaman M, Erzed N, Herwanto A. Evaluating the Effectiveness of Center for Internet Security (CIS) Benchmark for Hardening Ubuntu Server 22.04

Against Cyber Threats. International Journal of Scientific Research (IJSR). 2025; 14(6): doi: 10.55324/josr.v4i6.2544.

[6] Chen H, Han X, Zhang Y. Endogenous Security Formal Definition, Innovation Mechanisms, and Experiment Research in Industrial Internet. Tsinghua Science and Technology. 2023; 29(2): 492–505. doi: 10.26599/TST.2023.9010034.

[7] Almaiah MA, et al. Classification of Cybersecurity Threats, Vulnerabilities and Countermeasures in Database Systems. Computers, Materials & Continua. 2024; 77(3): 6845–6869. doi: 10.32604/cmc.2024.057673.

[8] Bhurtel S. Unveiling the Landscape of Operating System Vulnerabilities. Future Internet. 2023; 15(7): 248.

[9] Alhamed M, Albahrani I, et al. A Systematic Literature Review on Penetration Testing in Networks. Applied Sciences. 2023; 13(12): 6986. doi: 10.3390/app13126986.

[10] Abu Bakar R, Kijsirikul B. Enhancing Network Visibility and Security with Advanced Port Scanning Techniques. Sensors. 2023; 23(17): art. 7541. doi: 10.3390/s23177541.

[11] Park J, Yim K, Choi S, Kim G, Lee Y. Network Log-Based SSH Brute-Force Attack Detection Model. Journal of Network and Computer Applications. 2021; 186: 103063.

[12] Ruambo FA, Ruambo MM, Dandalo DT, Makwembere K. Brute-force attack mitigation on remote access services via a zero-trust-aligned software-defined perimeter. Scientific Reports. 2025; 15: art. no. 10805. doi: 10.1038/s41598-025-01080-5.

[13] Landauer M, Onder S, Skopik F, Wurzenberger M. Deep learning for anomaly detection in log data: A survey. Machine Learning with Applications. 2023; 13: 100470. doi: 10.1016/j.mlwa.2023.100470.

[14] Park J, Kim JS, Gupta BB, Park N. Network Log-Based SSH Brute-Force Attack Detection Model. Computers, Materials & Continua. 2021; 68(1): 888–901. doi: 10.32604/cmc.2021.015172.6

[15] Bangare PS, Patil KP. Enhancing MQTT security for Internet of Things: Lightweight two-way authorization and authentication with advanced security measures. Measurement: Sensors. 2024; 78: 100250. doi: 10.1016/j.measurements.2023.100250.

[16] Lazzaro S, De Angelis V, Mandalari AM, Buccafurri F. Hiding identities of MQTT devices against a global network adversary. EURASIP Journal on Information Security. 2025;2025:8. doi: 10.1186/s13635-025-00194-7.