JURNAL PENELITIAN MULTIDISIPLIN BANGSA
e-ISSN : 3048-4251
Volume 1, No. 12, Tahun 2025
https://ejournal.amirulbangunbangsapublishing.com/index.php/jpnmb/index

Peningkatan Keamanan SSH Server Berbasis Linux melalui Implementasi Fail2Ban dan Uji Serangan *Brute Force*

M. Ridho[1], Ahmad Hafizh[2], Ilham Dani[3], Tamsir Ariyadi[4]
[1,2,3,4] *Universitas Bina Darma Palembang, Indonesia*

Corresponding Author
Nama Penulis: M. Ridho
E-mail: muhammadridho5229@gmail.com

Abstrak

Penelitian ini bertujuan untuk meningkatkan keamanan layanan SSH pada server berbasis Linux melalui penerapan Fail2Ban sebagai mekanisme pencegahan otomatis terhadap serangan brute force. Dengan pendekatan eksperimental, dilakukan simulasi serangan menggunakan tools seperti Hydra untuk membandingkan kondisi server sebelum dan sesudah pemasangan Fail2Ban. Evaluasi dilakukan dengan mengamati log sistem, statistik pemblokiran IP, dan waktu respon sistem terhadap aktivitas mencurigakan. Hasil pengujian menunjukkan bahwa Fail2Ban secara efektif membatasi percobaan login ilegal dengan memblokir sumber serangan dalam waktu singkat. Kesimpulan dari studi ini menunjukkan bahwa Fail2Ban dapat berfungsi sebagai alat mitigasi yang handal untuk meningkatkan perlindungan terhadap akses tidak sah melalui SSH pada server Linux.
Kata kunci *– SSH, Fail2Ban, brute force, keamanan server, Linux*

Abstract

This research focuses on enhancing the security of SSH services on Linux-based servers by implementing Fail2Ban as an automated defense mechanism against brute force attacks. A quantitative experimental method was applied, involving simulated brute force intrusions using tools such as Hydra to compare system behavior before and after Fail2Ban implementation. Data were collected through system log analysis, IP blocking records, and system response times. The findings demonstrate that Fail2Ban effectively mitigates unauthorized access attempts by promptly identifying and blocking suspicious IP addresses. Overall, the study confirms that Fail2Ban serves as a reliable and efficient solution for strengthening SSH security on Linux servers.
Keywords *– SSH, Fail2Ban, brute force, server security, Linux*

PENDAHULUAN

Di tengah perkembangan teknologi informasi yang pesat, keamanan sistem menjadi faktor yang tidak bisa diabaikan, terutama pada *server Linux* yang umum digunakan dalam berbagai infrastruktur TI. Salah satu protokol utama untuk mengakses *server* secara jarak jauh adalah *Secure Shell (SSH)*. Meskipun protokol ini dirancang untuk menyediakan komunikasi yang aman, faktanya *SSH* sering menjadi target serangan, khususnya serangan *brute force* yang mencoba masuk dengan menebak kombinasi *username* dan *password* secara otomatis. *SSH* menawarkan metode enkripsi yang handal untuk autentikasi dan transfer data melalui jaringan. Protokol ini dimanfaatkan untuk mengakses *shell* sistem pada *server* atau perangkat jaringan, serta untuk mengirimkan perintah dan mendapatkan balasan dari sistem yang terhubung secara jarak jauh. *SSH* menawarkan metode enkripsi yang handal untuk autentikasi dan transfer data melalui jaringan. Protokol ini dimanfaatkan untuk mengakses *shell* sistem pada *server* atau perangkat jaringan, serta untuk mengirimkan perintah dan mendapatkan balasan dari sistem yang terhubung secara jarak jauh.(Ariyadi et al., 2023)

Ancaman *brute force* pada *SSH* berpotensi menimbulkan risiko besar, seperti akses ilegal, kebocoran informasi penting, hingga penguasaan penuh atas *server*. Untuk menangani risiko ini, diperlukan mekanisme perlindungan yang dapat mengidentifikasi dan merespons ancaman secara otomatis. Salah satu solusi yang dapat diandalkan adalah penggunaan *Fail2Ban*, sebuah aplikasi sumber terbuka yang mampu memantau file *log* dan memblokir alamat *IP* yang dicurigai melakukan percobaan *login* yang berulang dalam jangka waktu tertentu.(Farhannullah & Hardjianto, 2022)

Penelitian ini difokuskan pada penerapan *Fail2Ban* dalam meningkatkan keamanan layanan *SSH* pada *server* berbasis *Linux*, serta melakukan simulasi serangan *brute force* untuk menguji efektivitasnya. Melalui pendekatan ini, diharapkan dapat diperoleh pemahaman yang lebih baik mengenai efektivitas *Fail2Ban* sebagai sistem mitigasi otomatis, serta memberikan panduan praktis bagi administrator *server* dalam memperkuat keamanan layanan *remote* akses mereka.(Al Amien, 2020)

TINJAUAN PUSTAKA

Perlindungan Sistem dan Protokol SSH

Perlindungan terhadap sistem informasi menjadi hal esensial, khususnya dalam pengelolaan *server* yang menyediakan layanan akses jarak jauh. *SSH (Secure Shell)* merupakan salah satu protokol komunikasi terenkripsi yang umum digunakan untuk mengakses dan mengelola *server* secara aman. Protokol ini hadir sebagai pengganti metode koneksi yang tidak terenkripsi seperti *Telnet*, dan menyediakan autentikasi melalui password maupun kunci publik. Meski memiliki fitur keamanan yang cukup baik, layanan *SSH* tetap rentan terhadap percobaan akses tidak sah melalui metode *brute force* yang menebak kredensial secara sistematis.(Ikhsan & Handaga, 2021)

Karakteristik Serangan Brute Force

Metode *brute force* adalah salah satu teknik serangan yang sering digunakan untuk mengakses layanan *SSH* tanpa izin. Teknik ini dijalankan dengan mencoba berbagai kombinasi nama pengguna dan kata sandi dalam waktu yang singkat. Tanpa adanya sistem pencegahan, serangan ini bisa mengakibatkan kompromi keamanan serta membebani sumber daya *server* secara signifikan. Oleh karena itu, dibutuhkan sistem deteksi dini dan tindakan otomatis yang dapat menghentikan upaya tersebut sebelum berhasil. Serangan *Brute Force* terhadap *Secure Shell (SSH*) merupakan usaha untuk memperoleh akses yang tidak sah ke sistem yang dilindungi oleh protokol *SSH* dengan cara mencoba berulang kali berbagai kombinasi kata sandi hingga yang tepat ditemukan. Taktik ini sering digunakan oleh penyerang untuk menguasai akun *SSH* yang memiliki kata sandi lemah atau yang mudah ditebak.(Utomo et al., 2024)

Pemanfaatan Fail2Ban untuk Menangkal Serangan

Fail2Ban merupakan aplikasi *open-source* yang berfungsi sebagai pelindung sistem dengan memantau log aktivitas dan secara otomatis memblokir *IP* yang terindikasi melakukan percobaan *login*

mencurigakan. *Fail2ban* beroperasi dengan cara mengubah pengaturan *firewall* dan berusaha untuk menciptakan aturan berdasarkan catatan dari protokol *SSH*. *Fail2ban* bertugas untuk memantau frekuensi kegagalan *login* pada protokol *SSH* di sistem serta memblokir alamat IP yang dianggap melakukan serangan *brute force*. (Batistuta et al., 2024)

METODE

Penelitian ini menggunakan pendekatan eksperimen untuk mengevaluasi efektivitas *Fail2Ban* dalam melindungi layanan *SSH* pada *server* berbasis *Linux* terhadap ancaman serangan *brute force*. Proses penelitian ini terbagi dalam empat tahapan utama: persiapan sistem, implementasi *Fail2Ban*, simulasi serangan, dan analisis hasil.

Tahap Persiapan dan Instalasi Sistem

Tahap pertama penelitian adalah menyiapkan *server* berbasis *Linux (Ubuntu 20.04 LTS*) yang akan digunakan untuk pengujian penerapan *Fail2Ban*. Layanan *SSH* diinstal dan dikonfigurasi sesuai pengaturan *default* untuk memastikan kondisi sistem yang standar. Setelah itu, *Fail2Ban* diinstal menggunakan paket resmi, dan konfigurasi dasar dilakukan untuk memantau aktivitas *SSH*. Pengaturan pada *Fail2Ban* meliputi pengaturan durasi pemblokiran *IP* yang gagal *login*, batasan jumlah percobaan *login*, dan konfigurasi *log*. (Pratama et al., 2020)

Tahap Implementasi Fail2Ban dan Pengaturan Keamanan

Setelah *server* siap dan *Fail2Ban* terinstal, tahap berikutnya adalah melakukan konfigurasi *Fail2Ban* untuk memantau *log* aktivitas *SSH* dan mencegah serangan *brute force*. Pada tahap ini, parameter seperti "*maxretry*" (jumlah percobaan *login* gagal) dan "*bantime*" (durasi pemblokiran *IP* penyerang) ditentukan. *Fail2Ban* kemudian dikonfigurasi untuk menggunakan *iptables* sebagai metode pemblokiran alamat *IP* yang terindikasi melakukan serangan. Selain itu, filter *Fail2Ban* disesuaikan agar dapat memonitor *log* dengan efektif untuk mendeteksi serangan *brute force*. (Tambunan & Neyman, 2024)

Tahap Simulasi Serangan Brute Force

Untuk menguji efektivitas *Fail2Ban*, simulasi serangan *brute force* dilakukan menggunakan alat seperti *Hydra* atau *Medusa*. Alat ini memungkinkan penyerang untuk mencoba berbagai kombinasi nama pengguna dan kata sandi dalam waktu singkat secara otomatis. Pengujian dilakukan dalam dua skenario: pertama tanpa *Fail2Ban* dan kedua setelah *Fail2Ban* diterapkan. Simulasi serangan dilakukan dengan mengubah variasi jumlah percobaan *login* gagal dan periode waktu percobaan. Hal ini bertujuan untuk mengukur bagaimana *Fail2Ban* merespons serangan dengan intensitas yang berbeda-beda.(Mulyanto et al., 2022)

Tahap Pengukuran Kinerja dan Analisis Data

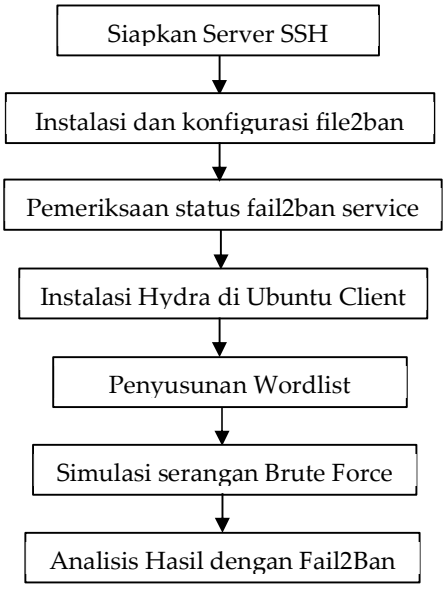
Setelah simulasi serangan, dilakukan pengukuran efektivitas sistem dengan menganalisis beberapa indikator, seperti jumlah percobaan *login* yang berhasil, waktu deteksi dan pemblokiran *IP* penyerang, serta pengaruh terhadap kinerja *server*. Waktu yang dibutuhkan *Fail2Ban* untuk memblokir *IP* yang berusaha melakukan serangan juga dihitung untuk mengetahui efisiensi sistem. Data yang diperoleh dari pengujian sebelum dan sesudah penerapan *Fail2Ban* dibandingkan untuk melihat sejauh mana *Fail2Ban* dapat mengurangi percobaan *login* yang berhasil serta meningkatkan keamanan *SSH*.(Likmi, 2020)

Tahap Evaluasi Hasil dan Kesimpulan

Hasil dari pengujian dievaluasi berdasarkan indikator keberhasilan dalam menanggulangi serangan dan peningkatan keamanan yang dicapai. Evaluasi juga mencakup analisis dampak terhadap performa *server* saat melawan serangan *brute force*, dengan membandingkan kondisi sistem sebelum dan sesudah penerapan *Fail2Ban*. Berdasarkan analisis ini, kesimpulan akan diambil mengenai sejauh mana *Fail2Ban* dapat meningkatkan perlindungan terhadap *server SSH* dan memberikan rekomendasi untuk pengelolaan sistem keamanan pada *server* berbasis *Linux*. Keamanan sistem dapat dipahami

sebagai langkah-langkah untuk mencegah dan mengidentifikasi pengguna yang tidak berwenang pada suatu sistem atau jaringan komputer. Tujuannya adalah untuk mengantisipasi risiko yang dapat mengancam jaringan atau sistem komputer, baik dari segi fisik maupun logis.(Wiryadinata et al., 2022)

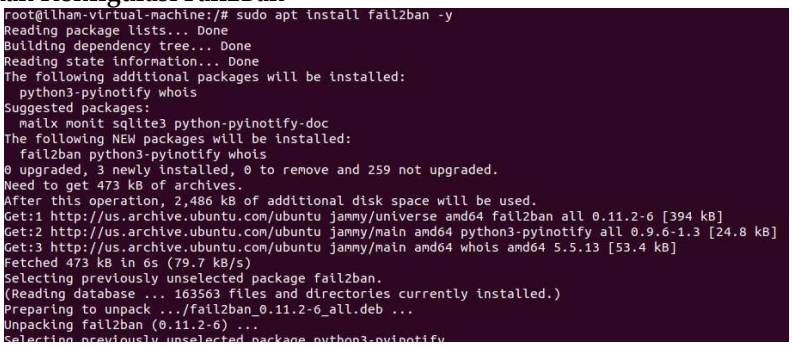
PEMBAHASAN



Gambar 1.
Kerangka Kerja

1. **Persiapan dan Pengaturan SSH Server**

   Langkah pertama dalam eksperimen ini adalah menyiapkan *server Ubuntu* sebagai target utama. Layanan *SSH Server* diinstal terlebih dahulu untuk memungkinkan koneksi jarak jauh yang aman ke *server*. *SSH (Secure Shell)* berfungsi sebagai jalur aman untuk komunikasi dan menjadi salah satu titik serangan yang disimulasikan. Setelah *SSH Server* berhasil diinstal, konfigurasi dilakukan untuk memastikan port 22 tetap aktif dan dapat menerima koneksi. Pengujian awal dilakukan dengan menggunakan perangkat lain dalam jaringan untuk memastikan bahwa koneksi *SSH* berfungsi dengan baik. *Username* dan *password* untuk akun administrator juga dipersiapkan sebagai referensi untuk validasi *login* pada percobaan berikutnya.(Likmi, 2020)
2. **Instalasi dan Konfigurasi Fail2Ban**



Gambar 2.
Instalasi *Fail2ban*

Setelah *SSH Server* aktif, langkah selanjutnya adalah memasang *Fail2Ban* untuk memberikan perlindungan terhadap *server* dari serangan *brute force* atau *login* yang berulang. *Fail2Ban* bekerja dengan memantau *log* autentikasi dan secara otomatis memblokir alamat *IP* yang melakukan percobaan *login* gagal berulang kali.

```
root@ilham-virtual-machine:/etc/fail2ban# sudo cp jail.conf jail.local
root@ilham-virtual-machine:/etc/fail2ban# sudo nano jail.local
```

Gambar 3.
Konfigurasi *Jail.local*

Konfigurasi *Fail2Ban* difokuskan pada pengaturan untuk *SSH*, yang dapat dilakukan melalui *file* konfigurasi seperti *jail.local*. Beberapa pengaturan yang dikonfigurasi termasuk:

1. Batas jumlah percobaan *login* yang gagal, misalnya hingga tiga kali.
2. Durasi pemblokiran, yang dapat bervariasi antara satu hingga tiga hari.
3. File *log* yang dimonitor, yaitu *log* autentikasi yang mencatat percakapan *login*.

Setelah konfigurasi selesai, *Fail2Ban* diaktifkan dan siap beroperasi sebagai sistem perlindungan otomatis.

```
[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
enabled = true
port    = ssh
logpath = /var/log/auth.log
maxretry = 3
findtime = 3m
bantime = 1h
backend = %(sshd_backend)s
```

Gambar 4.
Konfigurasi *SSHD*

3. Pemeriksaan Status Fail2ban Service

```
root@ilham-virtual-machine:/etc/fail2ban# sudo nano jail.local
root@ilham-virtual-machine:/etc/fail2ban# sudo systemctl restart fail2ban
root@ilham-virtual-machine:/etc/fail2ban# sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
^[[A^[[A^[[Aroot@ilham-virtual-machine:/etc/fail2ban# sudo systemctl enable failsystemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor pres>
     Active: active (running) since Wed 2025-05-07 10:46:50 EDT; 10s ago
       Docs: man:fail2ban(1)
   Main PID: 4920 (fail2ban-server)
      Tasks: 5 (limit: 4551)
     Memory: 13.3M
        CPU: 258ms
     CGroup: /system.slice/fail2ban.service
             └─4920 /usr/bin/python3 /usr/bin/fail2ban-server -xf start
```

Gambar 5.
Cek Status *Fail2ban* Service

Setelah proses instalasi dan konfigurasi selesai, langkah berikutnya adalah melakukan pengecekan untuk memastikan bahwa layanan *Fail2Ban* sudah berjalan dengan semestinya. Tahapan ini penting guna memastikan bahwa sistem proteksi telah aktif dan siap mendeteksi serta merespons aktivitas *login* yang mencurigakan secara otomatis. Hasil pemeriksaan akan menunjukkan apakah layanan berjalan normal, sedang aktif memantau layanan *SSH*, atau justru

mengalami gangguan. Status operasional ini merupakan indikator utama apakah *server* benar-benar telah terlindungi dari upaya serangan *brute force*.(Fachri, 2023)

4. **Pengaturan Jaringan Lokal**

Eksperimen ini dilakukan dalam jaringan lokal menggunakan dua mesin virtual:

1. *Server Ubuntu* (Target): Menyediakan layanan *SSH* dan menjalankan *Fail2Ban*.
2. *Client Ubuntu* (Penyerang): Digunakan untuk menyimulasikan serangan dengan menggunakan alat *brute force*.

Kedua perangkat diatur agar berada dalam satu jaringan lokal menggunakan konfigurasi *bridge adapter*, *host-only* adapter, atau *internal network*, yang memungkinkan keduanya saling berkomunikasi menggunakan alamat *IP* lokal.

5. **Instalasi Hydra di Ubuntu Client**

```
root@ilham-virtual-machine:/# sudo apt install hydra
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  firebird3.0-common firebird3.0-common-doc libapr1 libaprutil1 libbson-1.0-0
  libfbclient2 libmemcached11 libmongoc-1.0-0 libmongocrypt0 libmysqlclient21
  libpq5 libserf-1-1 libsnappy1v5 libsvn1 libtommath1 libutf8proc2
  mysql-common
Suggested packages:
  hydra-gtk
The following NEW packages will be installed:
  firebird3.0-common firebird3.0-common-doc hydra libapr1 libaprutil1
  libbson-1.0-0 libfbclient2 libmemcached11 libmongoc-1.0-0 libmongocrypt0
  libmysqlclient21 libpq5 libserf-1-1 libsnappy1v5 libsvn1 libtommath1
  libutf8proc2 mysql-common
0 upgraded, 18 newly installed, 0 to remove and 259 not upgraded.
Need to get 4,650 kB of archives.
After this operation, 17.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Gambar 6.
Instalasi *Hydra*

Untuk melakukan simulasi serangan *brute force*, digunakan alat bernama *Hydra* yang diinstal pada mesin *Ubuntu* kedua (*client*). *Hydra* adalah salah satu perangkat yang dikenal luas dalam pengujian penetrasi atau peretasan etis untuk mengevaluasi keamanan sistem, terutama dalam hal pengujian ketahanan kata sandi atau serangan terhadap kata sandi.(Force, 2024). Meskipun biasanya digunakan pada sistem *Kali Linux*, *Hydra* juga dapat dijalankan dengan baik di *Ubuntu*, menunjukkan fleksibilitas alat ini yang dapat digunakan pada berbagai distribusi *Linux*.

6. **Penyusunan Wordlist untuk Simulasi Serangan**

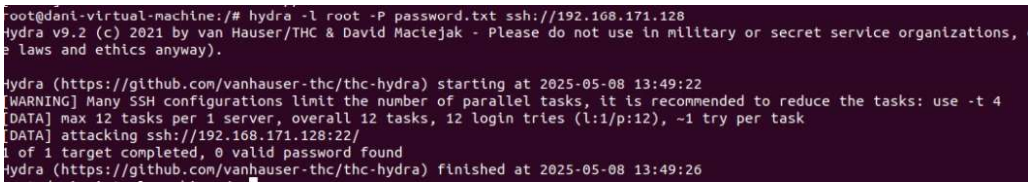
```
 GNU nano 6.2
123093820983
093213890283
asmklasndknaknd
123i01poj1
lekdpokdpoapo
lkpofpojpdj
apoajfpojpodjfsopsjf
odjofjidosjf
[iijj09\\dsfdsf
jidjiofjdsiojf
s;dflsfpojdposfjd
```

Gambar 7.
Menyusun *Wordlist*

Simulasi *brute force* membutuhkan sebuah *wordlist* yang berisi kombinasi *username* dan *password* yang akan diuji ke *server*. *Wordlist* ini dapat disusun secara manual dengan kombinasi umum atau diambil dari sumber terbuka. *Wordlist* tersebut disimpan pada direktori lokal di *client Ubuntu* yang menjalankan *Hydra*, dan akan digunakan untuk percakapan *login* selama proses serangan.(Sampurna, 2022)

7. **Simulasi Serangan Brute Force Menggunakan Hydra**

Setelah semua perangkat dan konfigurasi siap, serangan *brute force* dilakukan menggunakan *Hydra* dari *client Ubuntu* ke *server SSH*. *Hydra* akan membaca *wordlist* dan mencoba melakukan *login* dengan berbagai kombinasi sampai menemukan yang cocok. Pada tahap ini, sebelum *Fail2Ban* diaktifkan, *Hydra* dapat mencoba banyak kombinasi *login* tanpa halangan, yang bisa berpotensi membobol *server* jika ada kecocokan. Namun, setelah *Fail2Ban* diaktifkan, sistem akan memblokir alamat *IP* penyerang setelah mencapai jumlah kesalahan yang ditentukan, misalnya tiga kali, sehingga tidak ada percakapan *login* lebih lanjut yang bisa dilakukan.(Az Zahra et al., 2024)



Gambar 8.
Simulasi Serangan Menggunakan *Hydra*

8. **Analisis Hasil dengan Fail2Ban**

Gambar 9.
Status *IP* yang Diban

Setelah alamat *IP* penyerang diblokir, tidak ada lagi koneksi *SSH* yang dapat dilakukan dari *client* tersebut. Hal ini menunjukkan bahwa *Fail2Ban* efektif dalam menghalangi upaya *brute force* sejak awal. Catatan pemblokiran dapat ditemukan di *log* sistem, dan administrator dapat memverifikasi alamat *IP* yang diblokir melalui log *Fail2Ban* atau status layanan yang tersedia. Hasil tersebut membuktikan bahwa implementasi *Fail2Ban* memberikan perlindungan tambahan yang sangat penting untuk layanan *SSH* yang terbuka di jaringan.(Tambunan & Neyman, 2024)

KESIMPULAN

Kesimpulan dari penelitian Keamanan *SSH Server* Berbasis *Linux* melalui Implementasi *Fail2Ban* dan Uji Serangan *Brute Force* bahwa penerapan *Fail2Ban* pada layanan *SSH server* berbasis *Linux* mampu secara signifikan meningkatkan perlindungan sistem dengan cara mengenali dan memblokir alamat *IP* yang melakukan percobaan *login* berulang dalam waktu singkat. Berdasarkan

hasil uji coba serangan *brute force* menggunakan aplikasi *Hydra*, terbukti bahwa *Fail2Ban* secara efektif menurunkan jumlah *login* yang berhasil oleh pihak tidak sah dan mempercepat sistem dalam memberikan respons terhadap ancaman tersebut. *Fail2Ban* terbukti menjadi opsi mitigasi yang efisien dan layak digunakan untuk mencegah serangan *brute force* pada *server*, dan disarankan untuk diintegrasikan sebagai bagian dari strategi keamanan sistem; riset lanjutan dapat mengeksplorasi perbandingan efektivitasnya dengan solusi keamanan lainnya.

DAFTAR PUSTAKA

Al Amien, J. (2020). Implementasi Keamanan Jaringan Dengan Iptables Sebagai Firewall Menggunakan Metode Port Knocking. *Jurnal Fasilkom*, *10*(2), 159–165.

Ariyadi, T., Rizky, M., Hadi, M. K., & Widodo, A. A. (2023). Implementasi Firewall Pada Protokol SSH Linux Ubuntu Menggunakan Iptables. *Seminar Riset Mahasiswa-Computer & Electrical (SERIMA-CE*, *1*(1), 170–175.

Az Zahra, D. R., Ilham, F. P., Ramdhani, H. N., & Setiawan, A. (2024). Penerapan dan Pengujian Keamanan SSH Pada Server Linux menggunakan Hydra. *Journal of Internet and Software Engineering*, *1*(3), 10. https://doi.org/10.47134/pjise.v1i3.2627

Batistuta, A. D., Hendrawan, A. H., & Ritzkal. (2024). Analisis Keamanan Jaringan Server Terhadap Serangan Dictionary Menggunakan Tools Fail2Ban Dengan Notifikasi Telegram. *INFOTECH Journal*, *10*(1), 64–73. https://doi.org/10.31949/infotech.v10i1.8730

Fachri, F. (2023). Optimasi Keamanan Web Server Terhadap Serangan Brute-Force Menggunakan Penetration Testing. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *10*(1), 51–58. https://doi.org/10.25126/jtiik.20231015872

Farhannullah, & Hardjianto, M. (2022). Sistem Monitoring Serangan SSH dengan Metode Intrusion Prevention System (IPS) Fail2ban Menggunakan Python Pada Sistem Operasi Linux. *Jurnal Ticom: Technology of Information and Communication*, *11*(1), 33–38. https://doi.org/10.70309/ticom.v11i1.68

Force, B. (2024). *Arus Jurnal Sains dan Teknologi (AJST) Optimalisasi Sistem Keamanan SSH dari Serangan Brute Force*.

Ikhsan, M. A., & Handaga, I. B. (2023). *Penerapan Keamanan Server Menggunakan Security Information Event And Management Pada Sistem Operasi Ubuntu Server (Doctoral dissertation, Universitas Muhammadiyah Surakarta).*

Likmi, S. (2020). *Penerapan Keamanan Remote Server Melalui Ssh*. *4*(1), 133–138.

Mulyanto, Y., Herfandi, H., & Candra Kirana, R. (2022). Analisis Keamanan Wireless Local Area Network (Wlan) Terhadap Serangan Brute Force Dengan Metode Penetration Testing (Studi kasus:RS H.Lmanambai Abdulkadir). *Jurnal Informatika Teknologi Dan Sains*, *4*(1), 26–35. https://doi.org/10.51401/jinteks.v4i1.1528

Pratama, R., Orisa, M., & Ariwibisono, F. (2020). Aplikasi Monitoring Dan Controlling Server Menggunakan Protocol Icmp (Internet Control Message Protocol) Dan Ssh (Secure Shell) Berbasis Website. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *4*(1), 397–403. https://doi.org/10.36040/jati.v4i1.2310

Sampurna, M. R. (2022). NetPLG Journal of Network and Computer Applications Implementasi Hydra, FFUF, dan WFUZZ dalam Brute Force DVWA. *Journal of Network and Computer*, *1*(2), 102–112. https://jurnal.netplg.com/

Tambunan, M. R. H., & Neyman, S. N. (2024). Implementasi Firewall pada Linux untuk Pencegahan Dari Serangan DoS. *Journal of Technology and System Information*, *1*(4), 10. https://doi.org/10.47134/jtsi.v1i4.2648

Utomo, B. R., Jati, N. H., Jati, A. K., Saputro, I. A., & Purwidiantoro, M. H. (2024, December). Analisis Implementasi Keamanan Jaringan dengan Fail2ban Terhadap serangan Bruteforce. *In Prosiding*

Seminar Nasional Amikom Surakarta (Vol. 2, pp. 1211-1223).

Desmira, D., & Wiryadinata, R. (2022). Rancang Bangun Keamanan Port Secure Shell (SSH) Menggunakan Metode Port knocking. *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI),* 5(1), 28-33.