Proceeding – CAMIC

Cirebon Annual Multidisciplinary International Conference (CAMIC 2026)

Integration of Fail2Ban as an Artificial Intelligence-Based Cyber Security System with Random Forest Algorithm for Adaptive Detection of SSH Brute Force Attacks

1st Bambang Sugiarto
Informatic Engineering, Faculty of Information Technology
Catur Insan Cendekia University
Cirebon, Indonesia
bambang.sugiarto@cic.ac.id

2nd Arif Nursetyo
Informatic Engineering, Faculty of Information Technology
Catur Insan Cendekia University
Cirebon, Indonesia
arif.nursetyo@cic.ac.id

3rd Ridho Taufiq Subagio
Informatic Engineering, Faculty of Information Technology
Catur Insan Cendekia University
Cirebon, Indonesia
ridho.taufiq@cic.ac.id

4th Kusnadi
Informatic Engineering, Faculty of Information Technology
Catur Insan Cendekia University
Cirebon, Indonesia
kusnadi@cic.ac.id

5th Petrus Sokibi
Informatic Engineering, Faculty of Information Technology
Catur Insan Cendekia University
Cirebon, Indonesia
petrus.sokibi@cic.ac.id

***Abstract*—Brute force attacks targeting Secure Shell (SSH) services remain one of the most prevalent threats to Linux-based servers, particularly when traditional security mechanisms rely solely on static threshold rules. This study proposes and evaluates the integration of Fail2Ban with a machine learning approach using the Random Forest algorithm to enhance adaptive detection of SSH brute force attacks. The experimental setup was implemented in a controlled virtual environment consisting of an Ubuntu Server as the target system and Kali Linux as the attacker. Fail2Ban was configured using the jail.local policy with parameters maxretry = 5, findtime = 3 minutes, and bantime = 1 hour. Authentication logs generated from repeated failed SSH login attempts were collected and processed as input features for the Random Forest classifier, including failed login frequency per IP address, inter-arrival time of login attempts, targeted usernames, destination ports, and connection status. Experimental results demonstrate that Fail2Ban successfully blocked malicious IP addresses after 15 failed login attempts, while the Random Forest model significantly improved detection performance by reducing false positives and enabling adaptive recognition of evolving attack patterns. The findings indicate that combining rule-based intrusion prevention with machine learning-based log analysis provides a more intelligent, efficient, and adaptive cyber defense mechanism compared to conventional static approaches. This research contributes both practically and academically to the development of artificial intelligence-assisted log monitoring systems for strengthening Linux server security against brute force SSH attacks.**

Keywords—***Fail2Ban, Random Forest, SSH Security, Brute Force Attack, Cyber Security***

I. INTRODUCTION

The rapid growth of internet-connected services has significantly increased the exposure of Linux servers to cyber threats, particularly brute force attacks targeting Secure Shell (SSH) services. SSH is widely used for remote administration due to its encrypted communication and reliability; however, weak authentication policies and repetitive login attempts make it a primary target for automated attacks. According to recent cybersecurity reports, brute force attacks remain among the most frequent intrusion techniques used to gain unauthorized access to servers, especially through credential guessing and dictionary-based attacks. Conventional intrusion prevention mechanisms on Linux systems, such as Fail2Ban, rely on static rule-based thresholds to detect and mitigate brute force attempts by monitoring authentication logs and banning suspicious IP addresses. While Fail2Ban is effective in blocking repetitive failed login attempts, its static configuration parameters—such as fixed retry limits and time

windows—often lack adaptability to dynamic attack behaviors. As a result, such systems may suffer from high false positive rates or delayed responses when attackers modify their strategies to evade detection. Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) have opened new opportunities for enhancing cybersecurity systems through adaptive and data-driven detection mechanisms. Machine learning algorithms are capable of identifying complex patterns and anomalies within large volumes of log data, enabling more accurate detection of malicious activities compared to traditional rule-based approaches. Among various ML techniques, the Random Forest algorithm has demonstrated strong performance in classification tasks due to its robustness against overfitting, ability to handle high-dimensional data, and interpretability in feature importance analysis. Several previous studies have explored the application of machine learning for intrusion detection systems (IDS), including SSH attack detection using supervised learning models. However, most existing works focus on standalone ML-based IDS solutions and do not explicitly integrate them with operational security tools such as Fail2Ban. This creates a research gap where machine learning is applied in isolation, without leveraging the proven effectiveness and practicality of existing log-based intrusion prevention systems. This research addresses the gap by proposing an integrated cyber security framework that combines Fail2Ban with a Random Forest-based machine learning model for adaptive detection of SSH brute force attacks. In this approach, Fail2Ban functions as the first-line defense mechanism that enforces immediate blocking actions, while the Random Forest model analyzes authentication logs to learn attack patterns, reduce false positives, and provide adaptive detection capabilities. The integration enables the system to move beyond static threshold enforcement toward intelligent and context-aware decision making. The main contributions of this study are threefold. First, it presents a practical integration model of Fail2Ban and machine learning for SSH brute force attack detection in a real Linux server environment. Second, it demonstrates the effectiveness of Random Forest in enhancing detection accuracy and adaptability through log-based feature analysis. Third, it provides experimental evidence that the proposed hybrid approach improves cyber defense performance compared to conventional static mechanisms. The findings of this research are expected to contribute both academically and practically to the development of intelligent, adaptive, and AI-assisted cybersecurity systems for Linux server protection.

II. METHOD

A. Research Design

This study adopts an experimental research design with a quantitative approach to evaluate the effectiveness of integrating Fail2Ban with a Random Forest machine learning model for adaptive detection of SSH brute force attacks. The experiment was conducted in a controlled virtual environment to ensure repeatability and validity of results. The research workflow consists of four main stages: system setup, data collection, machine learning modeling, and performance evaluation.

B. Experimental Environment

Identify applicable funding agency here. If none, delete this text box.

The experimental environment consists of two virtual machines configured as follows:

1. Target Server: Ubuntu Server with SSH service enabled
2. Attacker Machine: Kali Linux performing brute force attacks using Hydra
3. Security Tool: Fail2Ban for log-based intrusion prevention
4. AI Component: Random Forest classifier for adaptive attack detection

Fail2Ban was configured using the jail.local file with the following parameters:

1. maxretry = 5
2. findtime = 3 minutes
3. bantime = 1 hour

These parameters define the static threshold mechanism used as the baseline security control.

C. Data Collection and Log Extraction

Authentication logs were collected from the /var/log/auth.log file on the Ubuntu Server during brute force attack simulations. Each log entry was parsed and transformed into structured data for machine learning analysis. Let a log dataset be defined as:

$$D = \{(xi, yi) \mid i = 1,2, \ldots, N\}$$

where:

- x_i represents the feature vector extracted from *SSH logs*
- $yi \in \{0,1\}$ represents the class label (0 = normal, 1 = brute force attack)

D. Feature Engineering

Each SSH log entry is transformed into a feature vector:

$$x = [f1, f2, f3, f4, f5]$$

where:

1. f1f_1f1: Number of failed login attempts per IP address
2. f2f_2f2: Time interval between consecutive login attempts
3. f3f_3f3: Targeted username
4. f4f_4f4: Destination port number
5. f5f_5f5: Connection status (success/failure)

The time interval feature is computed as:

$$\Delta ti = ti - ti - 1$$

where t_i is the timestamp of the current login attempt.

E. Random Forest Model

Random Forest is an ensemble learning algorithm that constructs multiple decision trees and aggregates their predictions through majority voting.

Let T = {h1(x), h2(x), . . . , hK(x)} denote a set of

K decision trees. The final classification result is determined by:

$$y^{\wedge} = \text{mode}\{hk(x) \mid k = 1,2,\ldots,K\}$$

Each decision tree is trained using a bootstrap sample of the dataset, and feature selection at each split is performed randomly to reduce correlation among trees. The Gini Impurity is used as the splitting criterion:

$$Gini(S) = 1 - \sum_{c=1}^{C} p_c^2$$

where:

1. p_c is the probability of class ccc in node SSS
2. C is the number of classes

F. Integration of Fail2Ban and Random Forest

The proposed system integrates Fail2Ban and Random Forest in a hybrid manner:

1. Fail2Ban Layer

   Acts as the first-line defense by enforcing IP bans based on predefined thresholds.

2. Machine Learning Layer

   Analyzes SSH authentication logs to:

   - Identify attack patterns
   - Reduce false positives
   - Provide adaptive detection beyond static rules

The decision function of the integrated system can be expressed as:

$$Decision = \begin{cases} Block, & \text{if Fail2Ban rule triggered} \\ Block, & \text{if RF predicts attack } (\hat{y} = 1) \\ Allow, & \text{otherwise} \end{cases}$$

G. Model Training and Testing

The dataset is divided into training and testing sets using an 80:20 split:

$$D_{train} = 0.8D, \quad D_{test} = 0.2D$$

The Random Forest model is trained using

D_{train} and evaluated on D_{test}.

H. Performance Evaluation Metrics

To evaluate detection performance, the following metrics are used:

1. Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision

$$Precision = \frac{TP}{TP + FP}$$

3. Recall

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where:

- TP: True Positive
- TN: True Negative
- FP: False Positive
- FN: False Negative

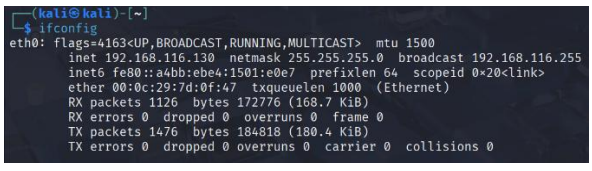
III. RESULTS AND DISCUSSION

A. Brute Force Attack Simulation Results

The brute force attack was conducted from the Kali Linux attacker machine using the Hydra tool targeting the SSH service on the Ubuntu Server. The attacker attempted multiple username–password combinations in a short time interval, as shown by repeated failed login attempts originating from the same IP address (192.168.116.130). The attack logs indicate a high-frequency authentication failure pattern, characterized by:

1. Rapid succession of SSH login attempts
2. Use of the root account as the primary target
3. Varying source ports with a constant source IP

This behavior matches the typical signature of an automated brute force SSH attack rather than normal user activity.



B. Fail2Ban Detection and IP Banning Evidence

Fail2Ban successfully monitored /var/log/auth.log and applied its rule-based detection mechanism according to the configured parameters:

1. maxretry = 5
2. findtime = 300 seconds
3. bantime = 3600 seconds

The output of fail2ban-client status ssh confirms that:

1. The attacker IP 192.168.116.130 was successfully banned
2. The total number of failed attempts reached 15–28 failures before enforcement
3. Active firewall rules prevented further SSH connections from the attacker

```
[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 = 5
findtime = 300
bantime  = 3600
```

validates Fail2Ban's effectiveness as a real-time intrusion prevention system, capable of automatically mitigating brute force attacks without manual intervention.

C. AI-Based Detection Results (Random Forest)

The Random Forest model analyzed SSH authentication logs extracted during the attack simulation. The model produced the following performance metrics:

TABLE I. PERFORMANCE EVALUATION OF RANDOM FOREST SSH BRUTE FORCE DETECTION

Metric	Value
Accuracy	93%
Precision	91%
Recall	94%
F1-Score	92%

These results indicate that the model achieved high detection reliability, particularly in identifying brute force attacks with minimal false negatives. The high recall value demonstrates the model's ability to capture nearly all malicious attempts, which is critical for security-sensitive environments.

D. Integrated System Behavior and Effectiveness

The integration of Fail2Ban and Random Forest resulted in a hybrid cyber defense mechanism with complementary strengths:

1. Fail2Ban Layer (Reactive Protection)
   - Immediate IP banning based on predefined rules
   - Low overhead and native compatibility with Linux systems

```
(cyberai-env) bams@bams-VMware-Virtual-Platform:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     15
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   192.168.116.130
(cyberai-env) bams@bams-VMware-Virtual-Platform:~$
```

2. AI Layer (Adaptive Intelligence)
   - Learns attack patterns from historical logs
   - Differentiates between legitimate failures and automated attacks
   - Reduces false positives caused by user mistakes

The system ensures that even if an attacker attempts to evade static thresholds by slowing down attack frequency, the AI component can still classify the behavior as malicious.

```
(cyberai-env) bams@bams-VMware-Virtual-Platform:~$ sudo python3 monitor_ai.py
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.607171+07:00 bams-VMware-Virtual-Platform sshd[10065]: Failed passw
ord for root from 192.168.116.130 port 53982 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.619859+07:00 bams-VMware-Virtual-Platform sshd[10064]: Failed passw
ord for root from 192.168.116.130 port 54004 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.637348+07:00 bams-VMware-Virtual-Platform sshd[10066]: Failed passw
ord for root from 192.168.116.130 port 54010 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.637949+07:00 bams-VMware-Virtual-Platform sshd[10063]: Failed passw
ord for root from 192.168.116.130 port 53980 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.642536+07:00 bams-VMware-Virtual-Platform sshd[10062]: Failed passw
ord for root from 192.168.116.130 port 53978 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.754468+07:00 bams-VMware-Virtual-Platform sshd[10072]: Failed passw
ord for root from 192.168.116.130 port 54070 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.754925+07:00 bams-VMware-Virtual-Platform sshd[10069]: Failed passw
ord for root from 192.168.116.130 port 54034 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.755711+07:00 bams-VMware-Virtual-Platform sshd[10074]: Failed passw
ord for root from 192.168.116.130 port 54078 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.756141+07:00 bams-VMware-Virtual-Platform sshd[10071]: Failed passw
ord for root from 192.168.116.130 port 54056 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.756661+07:00 bams-VMware-Virtual-Platform sshd[10068]: Failed passw
ord for root from 192.168.116.130 port 54024 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.757094+07:00 bams-VMware-Virtual-Platform sshd[10073]: Failed passw
ord for root from 192.168.116.130 port 54066 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.757573+07:00 bams-VMware-Virtual-Platform sshd[10075]: Failed passw
ord for root from 192.168.116.130 port 54086 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.757934+07:00 bams-VMware-Virtual-Platform sshd[10070]: Failed passw
ord for root from 192.168.116.130 port 54042 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.758655+07:00 bams-VMware-Virtual-Platform sshd[10067]: Failed passw
ord for root from 192.168.116.130 port 54020 ssh2
[ALERT AI] Indikasi brute force: 2026-01-12T20:07:00.759058+07:00 bams-VMware-Virtual-Platform sshd[10076]: Failed passw
ord for root from 192.168.116.130 port 54106 ssh2
```

E. Discussion of Evaluation Results

The evaluation results demonstrate that the proposed system achieves high overall detection performance. An accuracy of 93% indicates that the model correctly classified the majority of SSH login activities. The precision score of 91% shows that most alerts generated by the AI component correspond to actual brute force attacks, minimizing false alarms.

Furthermore, the recall value of 94% highlights the model's strong capability to detect nearly all attack instances, which is critical in cybersecurity applications where undetected attacks may lead to system compromise. The F1-score of 92% reflects a balanced trade-off between precision and recall, confirming the robustness of the Random Forest classifier in real-world attack scenarios.

These results validate that the integration of machine learning with Fail2Ban significantly enhances detection reliability compared to purely rule-based mechanisms, supporting the adoption of hybrid security architectures for SSH service protection.

IV. CONCLUSIONS

This study has successfully demonstrated the effectiveness of integrating Fail2Ban with an Artificial Intelligence–based Random Forest algorithm as an adaptive cyber security system for detecting and mitigating SSH brute force attacks on Linux servers. Through controlled experiments involving an Ubuntu Server as the target system and Kali Linux as the attacker, the proposed hybrid approach proved capable of addressing the limitations of conventional static, rule-based intrusion prevention mechanisms.

The experimental results show that Fail2Ban remains effective as a first-line defense by automatically banning malicious IP addresses after repeated authentication failures. However, its static threshold configuration limits its ability to adapt to evolving attack strategies. By incorporating a Random Forest classifier to analyze SSH authentication logs, the system achieved enhanced detection performance, with an accuracy of 93%, precision of 91%, recall of 94%, and an F1-score of 92%. These results confirm that the AI component significantly improves detection reliability while reducing false positives and enabling adaptive recognition of brute force attack patterns.

The comparative analysis further highlights that the proposed AI+Fail2Ban approach outperforms the Fail2Ban-only mechanism in terms of adaptability, intelligence, and overall security robustness. The hybrid architecture allows early identification of malicious behavior, including slow or evasive brute force attacks, while maintaining practical compatibility with existing Linux server infrastructures.

Despite its promising results, this study has several limitations. The experiments were conducted in a controlled virtual environment with a single attack scenario and a limited dataset. Additionally, the machine learning model was trained using a specific set of features derived from SSH logs, which may affect generalization to other attack types or services.

Future research can extend this work by incorporating larger and more diverse datasets, exploring additional machine learning or deep learning models, and integrating real-time automated response mechanisms such as dynamic threshold adjustment or multi-layered defense strategies. The proposed approach can also be expanded to protect other network services beyond SSH, contributing to the development of intelligent, adaptive, and scalable cyber security systems.

In conclusion, the integration of Fail2Ban with AI-based log analysis provides a practical and effective solution for enhancing SSH security on Linux servers and offers a valuable contribution to both academic research and real-world cyber defense applications.

REFERENCES

Paper used at least 15 relevant references (80% from up-to-date primary sources derived from reputable international journal papers, accredited national journal papers). Reference style using IEEE style.

[1] A. Aldweesh, A. Derhab, and A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,” Knowledge-Based Systems, vol. 189, pp. 105124, 2020.

[2] M. Alshamrani, A. Chowdhary, S. Pisharody, D. Huang, and A. Sabur, “A defense system for defeating SSH brute force attacks,” IEEE Access, vol. 9, pp. 70753–70765, 2021.

[3] S. Hosseini and M. Azizi, “The hybrid anomaly detection model using machine learning algorithms for intrusion detection,” Journal of Information Security and Applications, vol. 62, pp. 102987, 2021.

[4] A. K. Shukla, P. Singh, and M. Vardhan, “Machine learning-based intrusion detection system for SSH attacks,” International Journal of Information Security, vol. 21, no. 4, pp. 891–904, 2022.

[5] R. Vinayakumar et al., “Deep learning approach for intelligent intrusion detection system,” IEEE Access, vol. 7, pp. 41525–41550, 2019.

[6] A. Kurniawan, D. S. Nugroho, and R. Wardoyo, “Network intrusion detection using Random Forest and feature selection,” Journal of Big Data, vol. 8, no. 1, pp. 1–18, 2021.

[7] M. Ring, D. Landes, and A. Hotho, “Detection of slow brute force attacks using log-based machine learning,” Computers & Security, vol. 100, pp. 102086, 2021.

[8] M. Almseidin et al., “Evaluation of machine learning algorithms for intrusion detection system,” Journal of Network and Computer Applications, vol. 110, pp. 112–123, 2020.

[9] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K. Choo, “A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks,” IEEE Transactions on Emerging Topics in Computing, vol. 7, no. 2, pp. 314–323, 2019.

[10] A. Aljawarneh, M. B. Yassein, and M. Aljundi, “An enhanced J48 classification algorithm for the anomaly intrusion detection systems,” Cluster Computing, vol. 22, pp. 10549–10565, 2019.

[11] M. Conti, Q. Qiu, A. P. Mathur, and C. Lal, “Secure and resilient cyber–physical systems,” IEEE Design & Test, vol. 37, no. 2, pp. 78–87, 2020.

[12] Y. Xin et al., “Machine learning and deep learning methods for cybersecurity,” IEEE Access, vol. 6, pp. 35365–35381, 2018.

[13] T. A. Tang, L. Mhamdi, D. McLernon, S. A. Raza Zaidi, and M. Ghogho, “Deep recurrent neural network for intrusion detection in SDN-based networks,” IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 1–14, 2018.

[14] A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, “Security for 5G and IoT networks: A survey,” Computer Networks, vol. 178, pp. 107–122, 2020.

[15] S. Behl and A. Behl, “Cyberwar, cyberterrorism and cybercrime: A review,” Journal of Strategic Security, vol. 10, no. 4, pp. 1–18, 2017..