Framework for Brute-Force Attack Detection Using Federated Learning

J. Chethana Datta(✉), S. Ananya, Mukund Deepak, Nishanth Mungara, and V. Sarasvathi

PES University, Bengaluru 560100, India
chethandatta2@gmail.com, sarsvathiv@pes.edu

Abstract. Intrusion Detection and Prevention Systems (IDPS) play a pivotal role in safeguarding computer networks by identifying and responding to potential threats. This paper focuses on the implementation of a Federated Learning-based Intrusion Detection and Prevention System which mainly focuses on detecting brute-force attacks. The IDPS captures network packets, predicts anomalies using a Decision Tree model and logs malicious flows for further analysis. The Federated Server holds a pre-trained machine learning model, it also communicates with the IDPS to send and receive model updates facilitating collaborative learning. Additionally, the malicious traffic is redirected to the honeypot service employed in the system. The paper aims to enhance real-time brute-force detection for specific services, such as SSH and FTP, through the federated learning paradigm. By harnessing the collaborative power of multiple nodes in a network, our system showcases improved detection capabilities with minimized communication overhead. Detailed design and experimentation reveals that the IDPS is capable of predicting the nature of interaction while ensuring that data privacy is preserved. The success of this experiment is evident with it's remarkable 99.997% accuracy rate. The system's capacity to provide smooth communication between the various intrusion detection components highlights how effective it is at defending computer networks against a variety of dynamic cyber threats.

Keywords: Federated Learning · IDPS · Decision Tree · SSH · FTP

1 Introduction

The implementation of Intrusion Detection and Prevention Systems (IDPS) remains a key defence mechanism against the rapidly expanding types of sophisticated cyber threats, in this exponentially growing area of cybersecurity. Traditional IDPS systems perform the necessary tasks of monitoring network traffic with the use of predefined attack signatures and heuristics. They also make use of anomaly detection methods to identify potential security breaches in the system. IDPSs thoroughly monitor the communication between the devices, mainly

X. Cheng (Ed): BROADNETS 2024, LNICST 601, pp. 64–73, 2025.
https://doi.org/10.1007/978-3-031-81168-5_7

searching for known patterns of hostile behaviour in the interactions and also looking for any indicators of unusual activity. Therefore, IDPSs serve as a crucial line of defence in digital environments by actively identifying and mitigating security threats in this ever-evolving domain of cyber threats.

The Verizon DBIR 2023 [11] report emphasises that the Small and Medium Businesses (SMBs) are the most vulnerable to brute-force attacks. These attacks can have a devastating impact on their day-to-day operations and overall finances. These attacks can result in the theft of sensitive data like customer records and financial records as well. This can lead to loss of trust among the customers which will eventually lead to loss of business. These businesses may not have the necessary resources to detect and respond to such attacks resulting in prolonged downtime and loss in productivity. The report also states that the average loss occurred during a data breach for SMBs is $149,000, which can be a significant financial burden for the small and mid-size organizations making it crucial for them to employ effective security measures.

However, there is an immediate need for innovative security systems which can dynamically adapt to the emerging diversity and complexity of cyber threats. Machine Learning (ML) has proved to be a revolutionary contributor in the enhancement of the capabilities of a traditional IDPS system. An ML based IDPS systems can categorise and identify patterns in network traffic by using various ML algorithms which significantly improves their efficiency and accuracy. These systems unlike the rule-based systems, enable the detection of new unseen cyber threats and progressively develop the system's defence against them. Recently, Federated Learning has emerged as a ground breaking technique within the ML domain. It proposes a decentralised approach of collective learning which keeps in mind principles of privacy preservation as well. Federated Learning uses power of collaboration among the edge devices to train machine learning models without the need to centralise sensitive data. This concept is highly relevant in the context of an IDPS as well, where maintaining the privacy of network traffic data is of utmost importance.

Threats are becoming more frequent and varied in the rapidly evolving domain of cybersecurity. Therefore, real-time threat detection and protection of user privacy are undoubtedly required. This paper presents an efficient IDPS framework based on the Federated Learning approach to address these issues. By fusing the advantages of a classic IDPS with the benefits of Federated Learning, this proposed innovative solution presents a paradigm shift in the Intrusion Detection space.

In the proposed framework, the collective effort of the Intrusion Detection and Prevention System, the Federated Server which oversees the global machine learning model and a Honeypot dedicated for further in-depth threat analysis is required to secure users' access to critical resources on the End Server. This federated approach facilitates adaptive learning across distributed components, ensuring the system's resilience in the face of evolving threats. Positioned at the forefront of the evolving cybersecurity landscape, this Federated Learning-based IDPS system aligns with the contemporary requirements for real-time threat detection, adaptability, and privacy preservation in modern network security.

2 Related Works

The field of intrusion detection has witnessed significant advancements, with notable works contributing to the enhancement of security measures. For instance S. Krishnaveni et al.'s research on Classification through Ensemble Method for Network Intrusion Detection on Cloud Computing [6] addresses the challenges of precise intrusion detection in cloud environments. Their ensemble-based approach demonstrates efficiency in recognizing and categorizing network intrusions, offering a promising solution to reduce false positives and false negatives. Nevertheless, the limited evaluation on real-world datasets and the lack of comparison with state-of-the-art methods highlight areas for improvement.

In the survey by Joffrey L et al. [7], the focus is on analyzing intrusion detection research utilizing the CSE-CIC-IDS2018 dataset. The survey identifies gaps in current research and sheds light on performance metrics used in curated works, offering valuable insights for future directions in intrusion detection. However, it lacked sufficient information for a comprehensive evaluation, and the survey does not provide an exhaustive analysis of the performance of different intrusion detection models.

Laurens Hellemons et al.'s flow-based SSH Intrusion Detection System [5] presents SSHCure, a flow-based intrusion detection system designed to efficiently detect brute-force SSH attacks in real-time. The system operates solely on flow data and offers an effective algorithm for real-time detection, contributing to network security. However the limitations include its focus on detecting specific types of attacks and reliance on flow data, which may not be available in all network environments.

Li Yang et al.'s Open Source Code for Intrusion Detection System Development Using Machine Learning [12] introduces IDS-ML, an open-source code facilitating the development of intrusion detection systems using machine learning. This code offers automated procedures for various aspects of IDS development, enhancing network security through the application of traditional and advanced machine learning techniques. However the limitations include need for further evaluation on real-world datasets and the sensitivity of the code to feature selection.

László Göcs and Zsolt Csaba Johanyák [4] propose a comprehensive workflow for feature identification in the development of intrusion detection systems. Their work evaluates different feature selection methods, providing valuable insights into feature relevance. However, the focus on filter-based feature selection methods and lack of a detailed analysis of computational complexity pose considerations for broader applicability.

Yang Qin and Masaaki Kondo introduced Federated Learning-Based Network Intrusion Detection with a Feature Selection Approach [10]. This work employs federated learning and a feature selection approach to enhance network intrusion detection, showcasing a novel application of machine learning in security. However, challenges such as need for robust federated learning models and potential communication overhead pose considerations for practical implementation. Jonathas A. de Oliveira et al. [8] introduce F-NIDS, a network intrusion

detection system leveraging federated learning. The system aims to address privacy concerns while achieving excellent detection performance with a low network communication overhead. Challenges include the detection performance's sensitivity to specific attack types, such as SQL injection attacks, and potential organizational reluctance to share network traffic data due to privacy concerns.

Dasu et al.'s recent contribution on a Risk-Based Authentication [3] explores risk-based authentication as an innovative strategy to fortify security measures against identity threats. Their study complements our emphasis on intrusion detection and brings an important perspective to the current discussion on preventive security measures. Their findings add to the larger field of cybersecurity by highlighting the significance of dynamic risk assessment in verifying identity and adopting risk-based authentication solutions.

These varied contributions offer insight on how intrusion detection is changing and combining cutting-edge methods and tools to tackle new cybersecurity issues. Even if each study offers insightful information, the shortcomings and difficulties that have been noted emphasize the necessity of continued investigation and improvement in the quest for more reliable and efficient intrusion detection systems.

This study, among others, sets the context for our work on a Federated Learning-based Intrusion Detection and Prevention System (FL-IDPS). Our focus lies in real-time analysis, federated learning, and an adaptable, scalable architecture to provide an effective solution for network security.

3 Proposed Methodology

In our proposed methodology, we have adopted a systematic approach, encompassing pre-processing, federated learning setup, Intrusion Detection and Prevention System (IDPS) implementation, and virtual machine (VM) configuration for the Federated Learning-based Intrusion Detection and Prevention System. The research began with an in-depth exploration of the CICIDS2018 dataset, laying the foundation for subsequent analyses. Feature extraction was crucial, and after evaluating various techniques, we opted for information gain, ensuring a robust set of features for model training. This stepwise process, from data exploration to feature extraction and model training, facilitated the construction of a resilient predictive model. The establishment of the Federated Learning setup involved configuring the Federated Server with a pretrained model, enabling collaborative machine learning across multiple IDPSs. Socket API communication facilitated the exchange of model parameters between the Federated Server and individual IDPSs, ensuring synchronization and iterative model improvement. The IDPS component was designed to classify network packets as benign or malicious, leveraging the global federated model hosted on the Federated Server. Predicted benign packets were directed to the End Server, while malicious packets underwent further analysis in a Honeypot – a decoy system designed to lure and log information about potential attackers.

3.1 System Design

The architecture illustrated in Fig. 1 shows an advanced system for intrusion detection and prevention which prioritises the availability of a secure and efficient environment for resource access from the End Server. This unified system mainly relies on key components of the system collectively working together to maintain the overall integrity of the system and to ensure that the system can adapt quickly to emerging cyber threats. The heart of the system is the Intrusion Detection and Prevention System (IDPS) which is entrusted with the crucial task of monitoring network traffic packets in real time. This is done using a Machine Learning (ML) model such as Decision Trees to classify the packets as benign or malicious, and if malicious, the type of attack being conducted is also detected. In this framework, we focus on identifying threats like SSH and FTP Brute-force attacks and the model has been trained accordingly. The IDPS smoothly handles the redirecting of packets by directing the legitimate users' benign requests to the End Server for uninterrupted resource access. On the other hand, the attackers, detected due to the malicious packets are routed to a Honeypot for further in depth analysis of the attack patterns which will help in enhancing the overall security posture of the system.

The Federated Server plays a pivotal role in this framework by facilitating the distribution of the global machine learning model and by managing the updates to the model based on communication with various IDPS' present in the network. This further ensures an adaptive resilience to evolving threat landscapes. This architectural approach highlights the system's effectiveness in the face of a dynamic threat scenario by considering the periodic retraining.

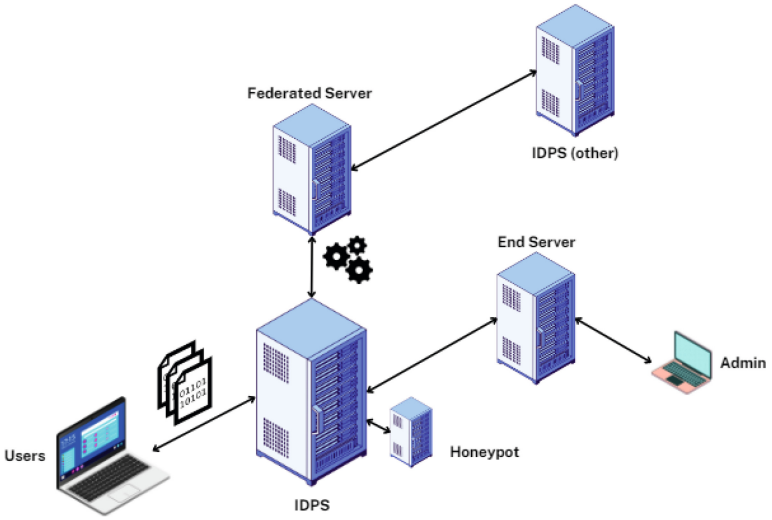

Fig. 1. Architecture Diagram

3.2 Algorithm

The collaborative machine learning process in the intrusion detection system is coordinated by the Federated Server code. The code was implemented using Python with Scapy [1] and Scikit-learn [9] libraries, and it creates a channel of communication with IDPS devices through a Socket API. A global decision tree model, which was pre-trained on the CICIDS 2018 [2] dataset at first, is managed by the server [2]. The decision tree was chosen for its simplicity, speed of training, and ability to handle categorical data well. The model is continuously improved by being updated on a regular basis using real-time data through iterative exchanges with IDPS units. Communication overhead is reduced through a federated approach, exchanging only model parameters via a Socket API between the server and IDPS units. Privacy is preserved by keeping sensitive data on IDPS devices and using a simple yet secure Socket API for communication, despite potential constraints on more complex protocols like gRPC. This federated approach leverages the strengths of collaborative machine learning, providing adaptability to evolving threats while preserving privacy in network security. The model will be refreshed with the latest anomalous flows, enabling the detection of dynamic attacks through the utilization of specific environmental metrics.

Algorithm 1. Federated Learning Algorithm

1: **Initialize:** Federated Server, Local IDPS, Pretrained Global Model
2: **Load:** CICIDS2018 dataset, preprocess, and extract features
3: **for each communication round do**
4: **Send Global Model to IDPS:** Federated Server sends global model to each IDPS
5: **for each IDPS do**
6: **Receive Local Model at Server:** Federated Server receives local model from IDPS
7: **Aggregate Models:** Federated Server aggregates models from all IDPS
8: **end for**
9: **Update Global Model:** Federated Server updates global model based on aggregated models
10: **end for**

The IDPS module is a critical component of the system, responsible for real-time packet analysis and intrusion detection. It was developed using Scapy [1] and Scikit-learn [9] in Python, the IDPS module also incorporates the global machine learning model hosted on the Federated Server. Upon receiving network packets, it leverages the model to classify them as benign or malicious, specifically identifying SSH and FTP brute-force attacks. Benign packets proceed to the End Server for normal resource access, while malicious packets are redirected to a Honeypot for detailed analysis. The implementation includes a routing mechanism within the IDPS, ensuring proper packet redirection based on the model's predictions. This module, operating on a distinct virtual machine, adheres to the principles of isolation and security.

Algorithm 2. Intrusion Detection and Prevention System (IDPS) Algorithm

```
Initialize: IDPS Components, Real-time Packet Capture, Communication Module
while system is active do
    Receive Global Model: IDPS receives global model from Federated Server
    Capture Packet: Capture real-time network packet
    Local Model Prediction: Use local model to classify packet as benign or malicious
    if malicious then
        Send to Honeypot: Route malicious packet to Honeypot for analysis
    else
        Send to End Server: Route benign packet to End Server for normal processing
    end if
    Send Local Model to Server: IDPS sends updated local model to Federated Server
end while
```

The flow-based analysis for SSH and FTP Brute-force prediction involves the IDPS module examining specific features extracted from network packets in real-time. Parameters such as Flow Inter Arrival Time (IAT) Mean, Flow Duration, Destination Port are evaluated to classify sessions as either benign or indicative of a brute-force attack. Benign sessions seamlessly proceed to the End Server for regular resource access, while sessions identified as potentially malicious are rerouted to the Honeypot for detailed analysis. This flow-based approach supported by machines learning models assures the swift and accurate identification of SSH and FTP Brute-force attacks which enhances the overall security of the system.

4 Implementation and Results

The implementation phase consisted of the deployment of separate Virtual Machines (VMs) to emulate various components of the system. This approach was adopted as it can closely simulate real-world scenarios while ensuring a controlled and secure environment.

The virtual machines were interconnected through a NAT (Network Address Translation) network to ensure isolation and security of the system. The VMs were allocated 2 GB of RAM and 20 GB of storage each to run the Ubuntu OS. The interconnected VMs formed a unified subnet with IP addresses that ranged from 10.0.3.X band. The above network configuration allowed controlled communication and interaction between the different components of the system.

In this simulated environment, the attacks were carried out using Hydra, which is an open source password brute-forcing tool generally used to emulate malicious activities. Hydra supports various network protocols including SSH and FTP. The attacker module in the system aims to replicate a malicious user attempting to obtain unauthorised access through brute-force attacks.

The attacker module utilised Hydra, specifically targeting SSH and FTP services to systematically and exhaustively try different combinations of usernames and passwords until a valid set of credentials were identified, essentially exploiting we ak or easily guessable login credentials. This type of attack, known as a brute-force attack, is a common method employed by attackers to compromise systems by repeatedly attempting different login credentials until a correct combination is found (Figs. 2 and 3).

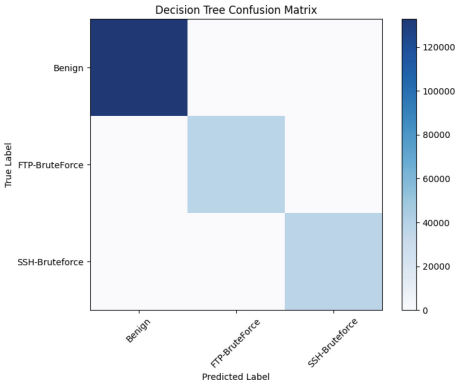

Fig. 2. Confusion Matrix

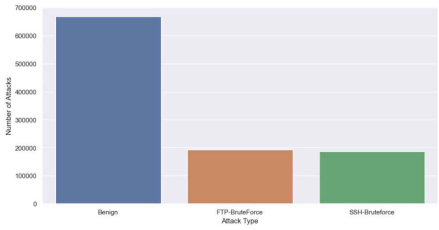

Fig. 3. Attack Type Plot

We used a Decision Tree for model training and Information Gain for feature selection in our study technique. A metric called Information Gain assesses how well a feature classifies data. As indicated in Table 2, we carefully examined the CICIDS2018 dataset, choosing suitable variables and used Information Gain to evaluate their importance. Upon conducting extensive pre-processing and exploratory data analysis, we determined that Information Gain was the most suitable feature extraction method for our analysis.

Subsequently, we used the selected features to train our model, opting for a Decision Tree due to its interpretability and effectiveness with categorical features. The training process involved fitting the model with the training data, evaluating its performance metrics (such as accuracy, precision, recall, and F1 score), and extracting feature importance using Gini indices. This step-by-step approach, from feature extraction to model training, enabled the development of a robust predictive model for intrusion detection. Despite limitations such as resource constraints and dataset usability, we could enhance our approach by considering a packet-based approach over a flow-based one, potentially mitigating these challenges while refining our intrusion detection model.

When tested using performance criteria such as accuracy, precision, recall, and F-score, the Decision Tree model exceeds 99.99% as shown in Table 1, demonstrating its excellent results in accurately categorizing network packets as benign

Table 1. Feature Importance

Feature	Importance
Flow IAT Mean	0.0
Bwd IAT Tot	0.0
Bwd Pkts/s	1.70×10^{-10}
Flow Duration	9.31×10^{-10}
Flow Pkts/s	3.47×10^{-9}
Dst Port	9.03×10^{-6}
Bwd IAT Mean	4.52×10^{-5}
Flow IAT Max	1.03×10^{-3}
Init Fwd Win Byts	0.35
Fwd Seg Size Min	0.65

Table 2. Performance Metrics

Metric	Value
Accuracy	0.9999714776705814
Precision	0.9999714806204552
Recall	0.9999714776705814
F-Score	0.9999714776477965

or malicious. The model's remarkable precision and recall indicate its ability to reliably identify positive occurrences and relevant examples, respectively, while its high accuracy demonstrates its precision in making accurate predictions. Recall highlights the model's capacity to catch all real positive events, whereas precision highlights the accuracy of positive predictions. When combined, precision, recall, and the F-score give a detailed assessment of the Decision Tree model's effectiveness and a thorough grasp of how well it can differentiate between legitimate and malicious network packets.

5 Conclusion

In this paper, we have proposed a highly efficient brute-force detection setup which takes into account the need for a better Federated Network framework with minimal overhead and of a real time threat detection system which we have achieved using Hydra to simulate real world attacks and breach scenarios. This system was designed keeping in mind the salient features of using the collective power of multiple IDPS' for more accurate classification and results of ever-evolving brute-force attacks. The Decision Tree model was trained using the CICIDS 2018 dataset with a test accuracy of 99.997%. We were able to establish communication between the IDPS modules by enabling the exchange of the updated parameters of the locally retrained machine learning model in an IDPS with the Federated Server to further update the global ML model, which aligns with the principles of a traditional Federated Learning setup.

The aspects we would like to work on in the future include identifying and mitigating diverse attacks. By comprehensively understanding and addressing a broader spectrum of threats, our goal is to enhance the currently existing intrusion detection and prevention mechanisms. Incorporating real-world datasets can enhance the model's robustness and applicability to diverse scenarios. Additionally, focusing on a lower level approach will significantly enhance the operational

efficiency in terms of speed. Enhancing defenses at the foundational level help us fine-tune and streamline our security mechanisms. Considering scalability in terms of handling larger datasets or increasing system demands will contribute to the effectiveness of the system in practical environments.

References

1. Biondi, P.: Scapy. The Scapy Development Community (2022). https://scapy.net/
2. Canadian Institute for Cybersecurity, N.: CIC-IDS 2018 dataset (2018). https://www.unb.ca/cic/datasets/ids-2018.html
3. Dasu, L.S., Dhamija, M., Dishitha, G., Vivekanandan, A., Sarasvathi, V.: Defending against identity threats using risk-based authentication. Commun. Appl. Ind. Math. **23**(2), 105–123 (2023). https://doi.org/10.2478/cait-2023-0016. Received 08 Dec 2022. Accepted 12 May 2023
4. Göcs, L., Johanyák, Z.C.: Identifying relevant features of CSE-CIC-IDS2018 dataset for the development of an intrusion detection system. J. Big Data **7** (2023). https://doi.org/10.1186/s40537-023-00523-0
5. Hellemons, L., Hendriks, L., Hofstede, R., Sperotto, A., Sadre, R., Pras, A.: SSHCure: a flow-based SSH intrusion detection system. In: Proceedings of the 6th IFIP WG 6.6 International Autonomous Infrastructure, Management, and Security Conference on Dependable Networks and Services (2012). https://doi.org/10.1109/DSN.2012.6263955
6. Krishnaveni, S., Sivamohan, S., Sridhar, S.S., Prabakaran, S.: Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. Clust. Comput. **24**, 1–13 (2021). https://doi.org/10.1007/s10586-021-03450-8
7. Leevy, J.L., Khoshgoftaar, T.M.: A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data. J. Big Data **7**, 104 (2020). https://doi.org/10.1186/s40537-020-00379-5
8. de Oliveira, J.A., et al.: F-NIDS - a network intrusion detection system based on federated learning. Comput. Netw. **236**, 110010 (2023). https://doi.org/10.1016/j.comnet.2023.110010
9. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
10. Qin, Y., Kondo, M.: Federated learning-based network intrusion detection with a feature selection approach. In: Proceedings of the 3rd International Conference on Electrical, Communication and Computer Engineering (ICECCE), Kuala Lumpur, Malaysia, pp. 12–13 (2021). https://doi.org/10.1109/ICECCE51249.2021.9483204
11. Verizon, C.: DBIR report 2023 - Master's guide (2023). https://www.verizon.com/business/resources/reports/dbir/2023/master-guide/. Accessed 30 Dec 2023
12. Yang, L., Shami, A.: IDS-ML: an open source code for intrusion detection system development using machine learning. Softw. Impacts **14**, 100446 (2022). https://doi.org/10.1016/j.simpa.2021.100446