Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 5-B, pp. 1907–1917
DOI : 10.47974/JIOS-2283

CyberShield : AI powered broken access control detection and mitigation

Aashmit Mckenzie *
Malay Doshi †
Gauri Deoghare §
Ranjeet Bidwe ‡
Sonali Kothari @
Pooja Bagane #
Ankur Goyal $
Department of Computer Science and Engineering
Symbiosis Institute of Technology (SIT)
Symbiosis International (Deemed University)
Lavale
Pune 412115
Maharashtra
India

Abstract

Broken Access Control (BAC) is identified as the most dangerous and common vulnerability in the modern web systems. BAC can cause severe problems, such as data leaks, privilege escalations and unauthorized system manipulations. To address this issue, we propose to manually construct an extensive and well-featured dataset that can represent realistic traffic-based access control behaviors and behaviors of BAC attacks. This dataset serves as a preliminary resource for developing and evaluating machine learning algorithms capable of real-time BAC attack detection. The study has employed this dataset to develop various machine learning techniques such as Logistic Regression, Random Forest, Gradient Boosting, and XGBoost. Double is the contribution of the proposed work, namely the new BAC dataset that serves as a fundamental requirement for research communities, and a machine learning-based detection framework that is dedicated for real-time attacks detection.

The Random Forest classifier achieved the highest overall accuracy (90.5%) and F1-score (0.8627), while XGBoost had the greatest AUC value of 0.9556. The system can also

* *E-mail:* **aashmit.mckenzie.btech2022@sitpune.edu.in (Corresponding Author)**
† *E-mail:* **malay.doshi.btech2022@sitpune.edu.in**
§ *E-mail:* **gauri.deoghare.btech2022@sitpune.edu.in**
‡ *E-mail:* **ranjeet.bidwe@sitpune.edu.in**
@ *E-mail:* **sonali.kothari@sitpune.edu.in**
# *E-mail:* **pooja.bagane@sitpune.edu.in**
$ *E-mail:* **Ankur_gg5781@yahoo.co.in**

evolve and extended in future using the adaptive learning mechanisms, by adding more variants of attack to Produce a comprehensive set of attacks for a large dataset creation, implementing the detection framework to live environment for proactive prevention and automatic response in real time incidents.

Subject Classification: *Primary 94A60, Secondary 68T05.*

Keywords: *Broken access control (BAC), Access control vulnerabilities, Machine learning, Data-driven security, Python-based security tools, Dataset creation for cybersecurity.*

1. Introduction

One of the most wide-ranging and high-impact vulnerabilities in the changing world of web application security has been Broken Access Control (BAC). Ranked as one of OWASP's Top 10 security risks consistently at the first spot, BAC permits unauthorized users to circumvent access constraints, resulting in unauthorized exposure, manipulation, or deletion of sensitive information. These attacks weaken systems' confidentiality and integrity while impacting millions of users and eroding trust in organizations. BAC vulnerabilities, despite their gravity, are currently addressed through manual review of codes, rule-based strategies for access control, and reactive security patches—mechanisms that tend to fail in dynamic and large-scale environments [1].

Academic studies and industry white papers have thoroughly documented BAC's theoretical foundations and exploitation patterns [2], [3]. Among the key limitations that hold back machine learning (ML)–based BAC detection is a shortage of publicly available, annotated datasets that reflect actual access control breaches. Absent these datasets, researchers cannot effectively train, benchmark, and test smart models capable of actively detecting unauthorized access in real-time [4], [5].

Broken Access Control (BAC) permits attackers to evade authorizations and gain unauthorized access to sensitive resources. Despite its prominence in security research, there has been very little or no practical effort aimed at developing ML-based detection systems suited for BAC. This poses an urgent gap in the world of cybersecurity—requiring an innovative, data-driven solution to effectively detect and defend against BAC attacks [6].

The proposed work aims to create and build the first complete, annotated dataset specifically for Broken Access Control (BAC) vulnerabilities, with both request-based features and behavioral features.

2. Literature Review

Over the past few years, the integration of Machine Learning (ML) models into access control systems has gained interest in several studies which demonstrate the potential of ML to address limitations of traditional access control models. BAC target detection is a manual task in traditional access control systems (ACS), and manual methods are unable to identify trend vulnerabilities effectively. Hence, researchers have strived to apply ML techniques to automate and enhance the BAC detection with the tighter and more reliable security mechanisms which are strongly robust, dynamic and secure.

A seminal work in this area is the one by Nobi et al. [4], where they deliver their taxonomy of machine learning applications in access control. In this paper they detail several ML techniques that can be applied to access control, e.g., supervised learning, unsupervised learning and deep leaning. However, they also point out the challenges, including the lack of labeled real-world datasets and the difficulty of training robust and accurate models.

In a continuation of this work, Nobi et al. [5] put forward Deep Learning-Based access control (DLBAC), a new approach that deploys deep learning models for access control policy enforcement. Experiments on both real and synthetic data demonstrate the suitability of deep learning for the control of dynamic access. The approach is best suitable in contexts where access policies are not static and may dynamically change with the context or user behavior.

More studies by Zhong [7] also give a detailed overview of techniques for detecting and preventing access control vulnerabilities. Zhong highlights the shortcomings of past access control models and the necessity for more intelligent, automatic systems with the ability to recognize and resist access control flaws.

Aside from classical ML models, Haitao et al. [2] also put forward a game-theoretic framework for BAC detection. Its game theory-based modeling to mimic attacks and defenses has more than 90% detection accuracy. ML models excel at pattern recognition, while game-theoretic models help understand the strategic aspect of attacks and defenses, thus offering an added level of security.

Research by [8] expounds on how ML can improve access control in big data environments, with its experiments conducted on the UNSW-NB15 dataset. The research is very applicable to this current study, as it has the same problem of dealing with large datasets as well as is able to emphasize

how ML methods can efficiently handle big data in access control environments [9-10].

3. Methodology

3.1 *Proposed Solution*

The solution proposed involves developing an extensive dataset with a specific aim to improve detection of Broken Access Control vulnerabilities with machine learning. Using a Python script on a platform with a Linux environment, the dataset was created by simulating attacks on three popular vulnerable applications: DVWA, Juice Shop, and WebGoat, all configured in separate Docker containers. The environment enables controlled penetration tests, where user interaction and attacks are recorded in a systematic manner. The created data are organized as CSV files for organized attack logs and JSON files to log detailed user activity [11-12].

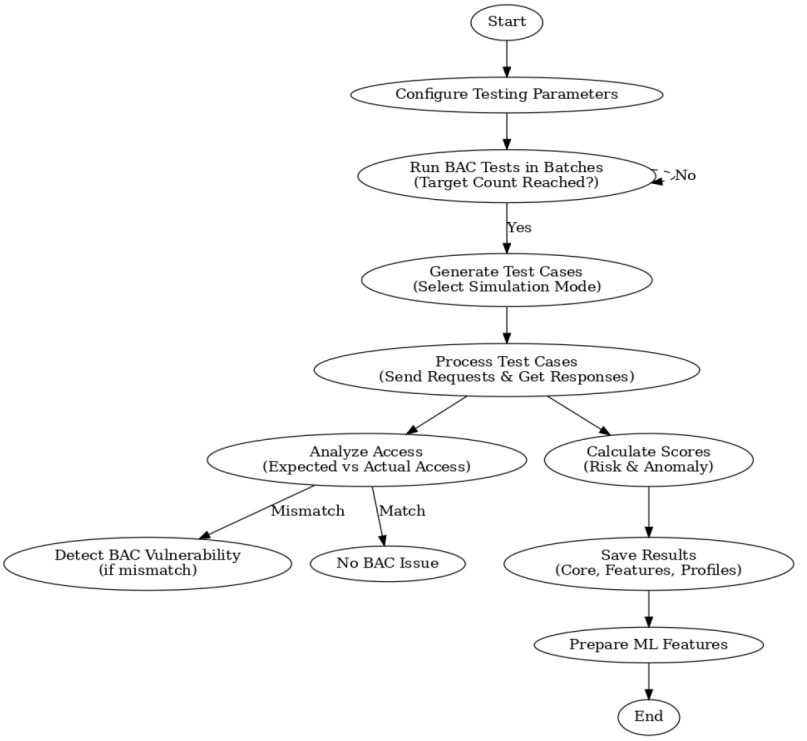

Figure 1

Stepwise details of Dataset Collection

Figure 1 represents the data collections and environment configuration steps for the same.

To further enhance the effectiveness of the solution, several machine learning frameworks were leveraged to create and train models that can identify Broken Access Control vulnerabilities. TensorFlow was used for its native support for deeper learning, and scikit-learn was used to apply more classical machine learning approaches. iLearn enabled model and pipeline management to provide smooth model training and evaluation operations [13].

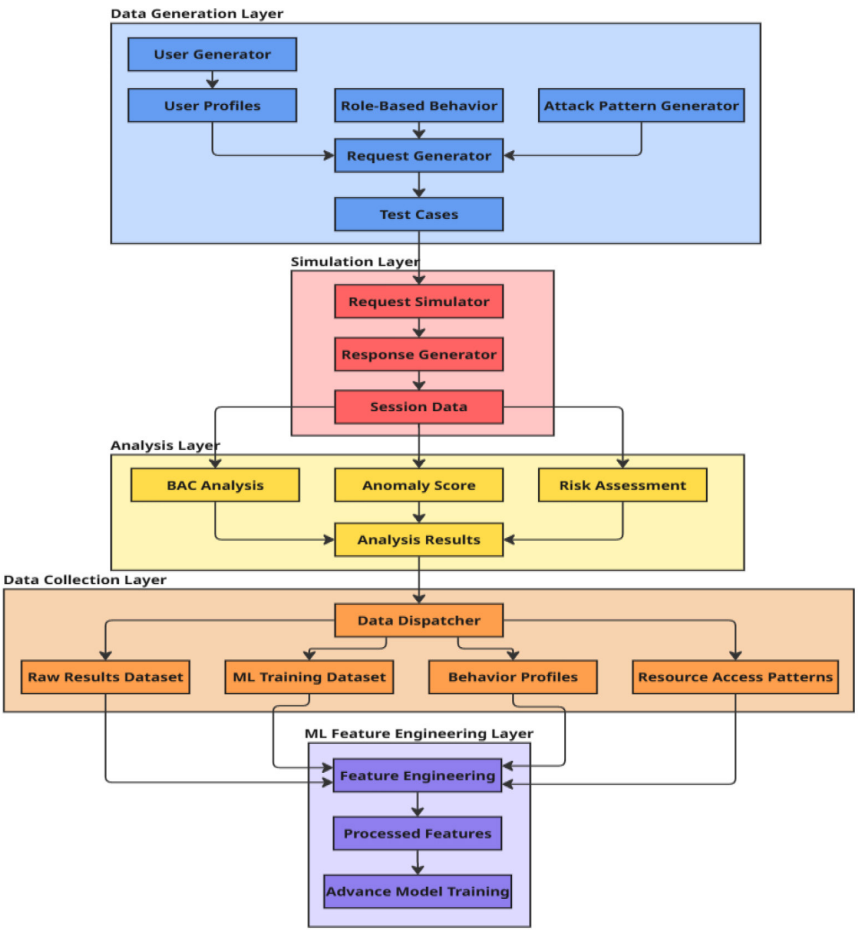

Figure 2

System Architecture

After training, the resultant models are serialized and persisted with Joblib for efficient reuse. The strong methodology permits scalable and automatic detection of access control exploits, offering critical insights into likely security threats. The synergy from Dockerized environments, machine learning models, and rich data logs forms a strong, flexible platform for driving advances in web application security.

3.2 *Environment Setup*

For simulating and identifying Broken Access Control (BAC) vulnerabilities, we implemented and operated a Python-based testing framework on a Kali Linux (Debian-based) setup. The script utilized Python 3.x and was able to run multi-threaded test sets to simulate concurrent users on legitimate and adversarial access use cases. This environment was designed to provide compatibility with popular penetration-testing tools and secure networking configurations.

3.3 *Data Collection Process*

The data was produced with a systematic BAC test run in configurable batches. The test environment involved a simulation engine that produced several types of traffic like valid traffic, attack-oriented traffic, mixed traffic, pattern-based attack traffic. Features considered for dataset creation includes Timestamp, Request_ID, Method, Endpoint, Role, Status_Code, Expected_Access, Actual_Access, BAC_Type, Token_Valid, Geo_Location, Request_Rate, Role_Anomaly, Behavior_Anomaly, Risk_Score, Anomaly_Score.

The proposed methodology follows a layered approach to generate, simulate, analyze, and collect data for security testing and machine learning applications. The system consists of five interconnected layers. Fig. 2 is the system architecture for proposed solution. This figure presents a multi-layered architecture for testing and analyzing Business Access Control (BAC) vulnerabilities using machine learning:

1. Data Generation Layer simulates various user behaviors using profiles, role-based access patterns, and attack scenarios to generate realistic test cases.
2. Simulation Layer processes these test cases by simulating request and response interactions, generating session data for further analysis.

3. Analysis Layer evaluates access behaviors through BAC analysis, anomaly scoring, and risk assessment, producing actionable analysis results.
4. Data Collection Layer organizes outputs into datasets, behavior profiles, and access patterns using a data dispatcher.
5. ML Feature Engineering Layer transforms raw and behavioral data into processed features, supporting advanced model training for anomaly detection.

The entire pipeline enables automated and intelligent analysis of BAC vulnerabilities with ML integration.

4. Results and Discussion

Of all the models, MLP scored the best on both accuracy (91.828%) and ROC AUC (0.949573), but took the longest to train (562 seconds). XGBoost was also very accurate (91.409%) and trained very quickly (20

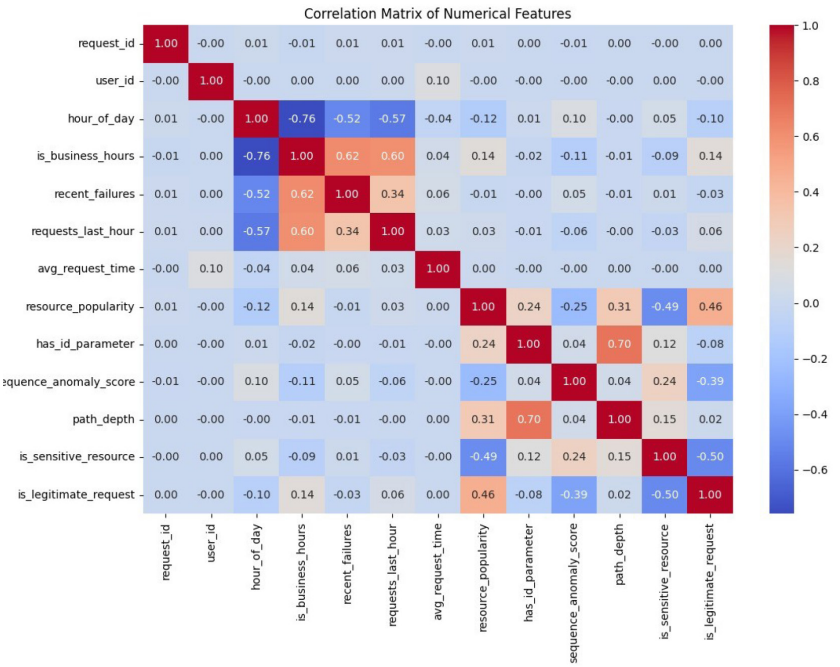

Figure 3

Correlation matrix

seconds), so was a solid and speedy option. LightGBM was not so accurate (91.320%) but trained quickest (4.36 seconds), so was great for big or time-critical jobs. Random Forest possessed high accuracy (91.257%) but slower time to train than the models that used boosting. Logistic Regression also possessed acceptable accuracy (90.196%) but an unexpectedly high time to train (330 seconds). KNN possessed the poorest accuracy (89.9225%) but trained very quickly at only 0.13 seconds, although its predictive ability was weak. In general, MLP performed the best, LightGBM provided the best trade-off between speed and accuracy, and XGBoost performed the best among traditional models. Table 1 shows the comparison of all trained ML models on the curated dataset.

Table 1
Comparison of proposed model on created dataset

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Training Time (secs)
RF	0.91	0.91	0.92	0.91	0.94	66.80
LR	0.90	0.90	0.90	0.90	0.90	330.74
MLP	0.92	0.92	0.92	0.92	0.95	562.36
KNN	0.90	0.90	0.90	0.90	0.90	0.13
XGB	0.92	0.91	0.91	0.91	0.92	19.95
LGBM	0.91	0.91	0.91	0.91	0.95	4.37

Figure 3 is the correlation matrix which yields useful insights regarding the interrelationship between the numerical attributes included within this study. There is a high negative correlation between hour_of_day and is_business_hours (-0.76), which makes sense since business hours are normally constrained within particular time intervals. Likewise, has_id_parameter and path_depth is highly correlated (0.70), reflecting that URLs with high depth are more likely to contain recognizable parameters. As for the target feature is_legitimate_request, it is moderately correlated with resource_popularity (0.46), reflecting that legitimate requests are more often aimed at popular resources. On the other hand, is_sensitive_resource is moderately negatively correlated (-0.50) to is_legitimate_request, reflecting that sensitive resources are more frequently visited illegitimately.

Figure 4 represents the model performance heatmap**.** The heat map showcases the performance measures of six models of classification on

major parameters of evaluation: accuracy, precision, recall, F1 score, ROC AUC, and training time. Accuracy, precision, recall, F1, and ROC AUC scores are all similar to one another across models and are thus making the background predominantly light yellow. The scores are all high (approximately 0.89–0.92), meaning all models are very good at predicting correctly.

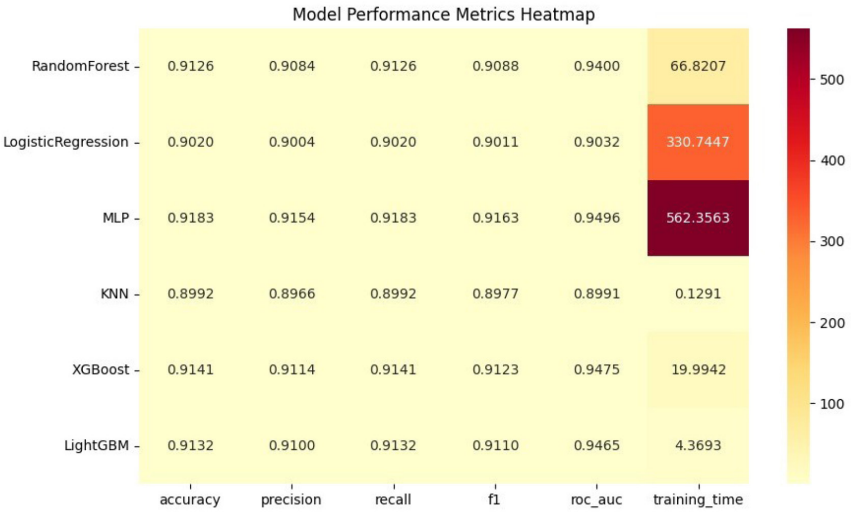

Figure 4

Heatmap of model's performances

5. Conclusion and Future work

XGBoost performed best overall with the best AUC value (0.9556 while RandomForest produced best accuracy (0.9050), and GradientBoost performed best on recall (0.8824). This integrated BAC vulnerability detection method incorporates data generation by simulation, simulation-based testing, and machine learning-based analysis for an automatic framework to identify potential vulnerabilities. The modularity and flexibility are ensured by a layered system design, and different simulation modes enable precise targeting of particular types of vulnerabilities. The strong performance rates for all models demonstrate that machine learning techniques can successfully identify BAC vulnerabilities with high precision and recall, and are an asset for security professionals to use as

tools to spot potential weaknesses in access control before they may be exploited.

Future versions of this system may include more advanced techniques for threat modeling to create even more realistic threat patterns. Using MITRE ATT&CK framework or other such knowledge bases may provide feed for generating advanced persistent threat (APT) scenarios that are representative of sophisticated attackers. Using genetic algorithms or reinforcement learning for evolutionary generation of attacks may also provide the means for uncovering novel vectors of vulnerabilities that have not been reported in security research

References

[1] OWASP Foundation, "OWASP Top 10: Broken Access Control," (2021).

[2] H. Haitao, X. Ke, Y. Shuailin, Z. Bing, Z. Yuxuan, and L. Jiazheng, "Game-based detection method of broken access control vulnerabilities in web applications," *Journal of Communications*, vol. 45, no. 6 (2024).

[3] A. Anas, S. Elgamal, and B. Youssef, "Survey on detecting and preventing web application broken access control attacks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 772–781 (2024).

[4] M. N. Nobi, M. Gupta, L. Praharaj, M. Abdelsalam, R. Krishnan, and R. Sandhu, "Machine learning in access control: A taxonomy and survey," arXiv preprint, arXiv:2207.01739 (2022).

[5] M. N. Nobi, R. Krishnan, Y. Huang, M. Shakarami, and R. Sandhu, "Toward deep learning based access control," in *Proc. 12th ACM Conf. Data and Application Security and Privacy*, pp. 143–154 (Apr. 2022).

[6] N. Shanthi and J. Shreyas, "Research on dataset creation methods for security learning," *SN Computer Science* (2022).

[7] L. Zhong, "A survey of prevent and detect access control vulnerabilities," arXiv preprint, arXiv:2103.04553 (2021).

[8] H. Es-Samaali, A. Abou El Kalam, A. Outchakoucht, and S. Benhadou, "Machine learning enhanced access control for big data," arXiv preprint, arXiv:2109.03200 (2021).

[9] A. K. Sharma and R. Singh, "Secure flow authentication and CNN-based intrusion detection system," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 2, pp. 2628–2645 (2024).

[10] S. Verma, P. Gupta, and M. K. Sharma, "Machine learning-based intrusion detection framework for network security," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26, no. 5, pp. 1987–2003 (2023).

[11] R. K. Jain and S. Bansal, "An efficient cryptographic approach for secure data transmission in web applications," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 4, pp. 1453–1468 (2022).

[12] N. Agarwal and V. Saxena, "Analysis of access control mechanisms in distributed systems," *Journal of Interdisciplinary Mathematics*, vol. 24, no. 6, pp. 1567–1582 (2021).

[13] P. K. Mishra and A. Tiwari, "A hybrid machine learning approach for anomaly detection in cybersecurity," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26, no. 3, pp. 1121–1138 (2023).

Received April, 2025