https://doi.org/10.48047/AFJBS.6.si2.2024.5955-5967

Probability of Incidents in Determining Forgery Vulnerabilities in Server-Side Request Attack Exposure with Predictive ML Analysis

Dr.K.Aanandha Saravanan[1]

Associate Professor

Department of VeltechRangarajanDr.Sagunthala R&D Institute of Science and Technology. Avadi, Chennai-62

anand23sarvan@gmail.com

Dr Suresh Kumar K[2]

Associate Professor, Department of Information Technology, Saveetha Engineering College, sureshkumar@saveetha.ac.in

Dr. P. Ranjith Kumar[3]

Associate Professor

Department of ECE

KallamHaranadhareddy Institute of Technology (Autonomous), Chowdavaram, Guntur- 522019

ranjithkumar.painam@gmail.com

J.Hymavathi[4]

Assistant Professor,CSE department,K L university,Vijayawada.mailid: janaswamy.hymavathi@gmail.com

Abhijeet Das[5]

Research Scholar Department of Civil Engineering C.V. Raman Global University (CGU), Bhubaneswar, Odisha das.abhijeetlaltu1999@gmail.com

G.Janani alias Pandeeswari[6]

Assistant professor

Department of Artificial intelligence and data science,PSNA college of engineering and technology ,Dindigul

er.gjanani@gmail.com

Volume 6 issue si2 2024
Received:15May2024
Accepted:10June2024
doi:10.48047/AFJBS.6. si2. 2024. 5955-5967

Abstract

The Server-Side Request Forgery (SSRF) exploit enables an attacker to send responses from a server that is weak to other internal and external systems. This is a potent target for attackers, which can be employed to obtain private information, run malicious scripts, or even initiate additional assaults. SSRF operates by using a server's vulnerability to make requests on the attacker's account. The most effective strategy to guard against server vulnerabilities is to routinely check for unusual behaviour and maintain the system updated with the most recent security updates. In this approach, server-side request forgery (SSRF) vulnerabilities are determined by determining the probability of incidents associated with the attack exposure using machine learning. Suspicious activity within the server logs is detected and monitored to attain server-side forgery attacks. Unusual requests, failed login multiple attempts, and also other suspicious activity indicate the attack is in progress. In this research, the dataset is taken from the CISA standard of known exploited vulnerabilities with different attributes. Ensure that the server-side code is properly secured and that server logs are monitored and responded to appropriately for any incidents based on the likelihood of an attack succeeding, which can be significantly reduced.

Keywords: Server-Side Request Forgery, Machine Learning, Malicious Scripts, Server Logs, Server Vulnerabilities

1. Introduction

An adversary can use the queries to obtain sensitive information or perform additional attacks by sending them to either internal or third-party services. While the requests originate from the susceptible server rather than the attacker's host computer, SSRF is also accessible to the perpetrator to get around identification and authorization systems. While the queries are transmitted from the susceptible server rather than the attacker's own workstation, it may also be exploited to get around identity and authorization systems. Moreover, SSRF can be used to identify programs, search for open ports, and gain connection to other internal networks that are normally unreachable. Through providing appropriate input validation, making use of a whitelist of permitted subdomains and interfaces, and restricting the use of internal services, SSRF can be avoided. Furthermore, it's crucial to make sure that a single origin is used for all requests and that identification and permission procedures are in place. Since it can be deployed to obtain confidential information, run malicious scripts, and conduct various operations, SSRF can have significant consequences. While the queries are transmitted from the susceptible server rather than the attacker's personal device, they may also be exploited in order to bypass identification and authorization systems. The purpose of the approach is to obtain access to computer systems or the infrastructure layer, which are normally inaccessible from a network. Ensure the server is correctly setup and each response is verified and sanitised prior to execution in order to avoid SSRF vulnerabilities. The guiding principles for SSRF involve

validating and processing all responses, making sure that only authorised users are permitted to have access to private resources, and establishing access controls that avoid unauthorised access. It is indeed critical to check that the server is appropriately setup, that almost all connections are documented, and that suspicious behaviour is always being observed. Non-secure setup, a lack of request validation and sanitization, and a shortage of access control methods are prevalent SSRF vulnerabilities. A type of cyberthreat characterised as forging web vulnerabilities involves bad actors trying to obtain confidential data or resources while pretending to be an authorised user.

To gain access, including information, an intruder can try to take advantage of holes in a system's authorization process. In order to deceive users into surrendering their information, organisations could also employ phishing methods. Secure authentication procedures, including multi-factor verification and two-factor identity, can help organisations minimise fraud and web vulnerabilities. Furthermore, companies ought to be certain their websites employ secure methods of communication, including HTTPS, and are thus up-to-date on the most recent security fixes. Organizations ought to implement precautions to prevent the attack after discovering forged web vulnerabilities, such as blocking access to personal information and deactivating compromised accounts. Furthermore, organisations ought to investigate the incident to identify the underlying causes and take precautions against future assaults of the same kind. A type of cyberattack known as forgery web vulnerability involves bad actors trying to obtain sensitive information or assets by pretending to be an authorised user. Businesses may take precautions to guard against and identify fraudulent web vulnerabilities, as well as react rapidly and effectively in the event of a breach. The possibility that a system is susceptible to malicious assaults and manipulation is known as server vulnerability. Understanding the different risks and how to defend against them is crucial. Server risks could be brought on by a range of factors, such as out-of-date computer software and hardware, improper system setup, and inadequate security mechanisms. There are two types of server vulnerabilities: known and unknown. In contrast to an undiscovered vulnerability, known vulnerabilities are all those that have yet to be detected and published. SQL injection, cross-site scripting, and cross-site scripting injection are common server vulnerabilities. Attacks on vulnerabilities are a significant threat to the integrity of computer systems. To secure the security of data and devices, it is crucial to comprehend the likelihood of a vulnerability exploit.

The possibility of an attacker attacking a technology is gauged by the probability of a vulnerability being attacked. It depends on the scope of the target machine, the kind of

approach, and the likelihood that it will succeed. The scale of the target device, the sort of attack, the attacker's competence, and the protection precautions taken by the system are some of the variables that have an impact on the likelihood of a vulnerability attack. The overall dimension of the target structure is essential since it affects how many possible targets an attacker can hit based on the target's size. A critical stage in estimating the likelihood of a vulnerability assault is risk assessment. This entails assessing a system's possible dangers and figuring out how likely an attack is to happen. A risk evaluation ought to consider the scale of the victim machine, the attack's design, the attacker's expertise, and the security controls put in place by the framework. An evaluation of the possible harm that a strike might do ought to also be included. When the risk analysis is finished, actions can be taken to lessen the likelihood of a vulnerability attack. To reduce the possibility of an attack, risk evaluation and mitigating measures should be implemented. The likelihood of a vulnerability attack may be considerably decreased by establishing security protocols, routinely upgrading software and system components, and offering training to individuals.

2. Literature Review

(Dhivya 2022) indicates that vulnerabilities on the web-based application platforms within the online applications operate inefficiently with high expenditure. The web application security has to be evaluated with logical query processing to detect the possibilities of attack with injection in the online application portal. Some of the algorithms in the vulnerability injector tool combine with other algorithms to determine the possible attacks using IDS. IDS evaluates the false-positive attack, and even the web application scanners are assessed. The server-side vulnerabilities are implemented, which evaluates the security mechanism. (Bekri 2022) indicated that the IoT creates a larger spectrum in heterogeneous devices where each device consists of a considerable range of data. The data is said to be sensitive and critical to handle. One of the major factors of the IoT is ensuring that data privacy and security threats are investigated. A wide range of security threats and traditional IoT attacks are assessed, and countermeasures are proposed.

(Al-Talak 2021) denotes the SSRF vulnerability that arises from those vulnerabilities. Services are accessed through URLs, which the attacker modifies to access those services. Through machine learning techniques, various SSRF attack detection techniques are used to detect and mitigate those attacks. The main focus of this research is on DL-LSTM networks, which achieve an accuracy rate of about 96%. In 2021, Zakaria uses the CSRF and SSI to typically

indicate the web application vulnerability. Several threats with web applications lack some of the security implications. The primary analysis is to perform the risk assessment along with determining the severity, level of limitations, and penetration testing advantage. Different levels of risk assessment identify the severity level of those attacks with a higher level of critical vulnerability. (Ahmed 2021). Cyberattacks launch a server-side attack that directly listens to the port. Server-side gets compromised and also infringes on the server data. The hybrid approach implements a two-layer security firewall to detect those malicious codes. Using the machine learning approach, accuracy is efficiently analysed to detect SQL detection attacks. (Gupta 2021), SDN applications with virtualization, where the data plane gets separated from the control network. SDN makes it easy to control the network, which is the new threat to security. threats, which include DoS and MiM attacks that affect the server. In this review, it compares the various machine learning techniques to detect those DDoS within SDN.

(Abel 2020) predicts cyber-attacks using ML, which supplicates the adversary nature. Based upon situational awareness, threats are predicted and analysed, including through spam filtering, firewalls, and IDS. Malware detection predicts the malware attacks, which automatically labels the event based on malware penetration such as manipulated nodes. The DT algorithm learns the dataset that identifies the malware threat prediction from the Kaggle dataset. Bahruz 2020 indicates the novel defence approach that protects from SSRF attacks. A prototype extends the functionality of a reverse proxy application, which exposes the vulnerability of those web applications within the prototype. A malicious URL server opens the connection to an internal service where the server is completely vulnerable. (Priya 2020) with high-relentless attacks, which are the crucial distributed DDoS attacks where the methodology gets more advanced. An automated DDoS detector using ML, where the accuracy is about 98%. Classification algorithms normalise the packets that detect DDoS, which requires a large number of features to detect those DDoS. The proposed model detects the DDoS of any type of specific protocol with a smaller set of features.

3. Methodology

Server-Side Request Forgery Attack

Server-side request forgery impacts the target application by handling the remote hosts. Remote hosts are controlled by the target attacker, and the SSRF attacks can target the vulnerable server that uses the target system server by sending intermediate pass requests through external resources. The intruders easily try to hack into the system by bypassing the internal systems

within the internal system.Attackers use SSRF to attack the vulnerable application where the malicious code gets compromised. Attackers exchange the data, malicious commands, and also queries across open ports and network ports within public networks. Invalid requests extract that network information into error messages, where the attackers interpret the overall behaviour of the target server. SSRF is vulnerable within web applications where the services are accessed through a URL and domain.Web applications contain sensitive data that is protected by intrusion, and the SSRF exploits those vulnerabilities within those web applications that enter the server in an illegal manner. Some of the web applications that obtain the username and password for authentication from the client-server side get compromised. The dataset for this analysis is drawn from the CISA standard of publicly disclosed exploited vulnerabilities with different features.

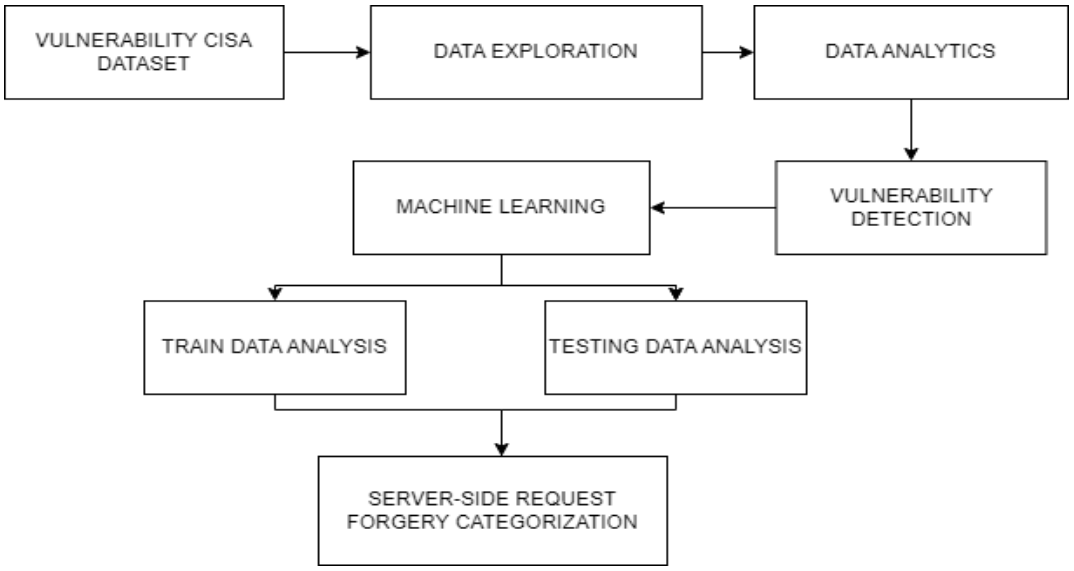

Figure 1: Proposed Architecture Diagram

Using unencrypted HTTP, the information gets compromised when it gets sent. Several security mechanisms and rules are used to protect the web applications from those attacks. SSRF attacks read and control the internal server where server resources are used using the available functions in the web application. A web application tends to be vulnerable where the data is corrupted, lost, or also appropriated. Attackers use internal servers to access the attackers, where the server submits the URL within a web request. Machine learning learns to perform human tasks, which it predicts through a classification and prediction model. It uses a statistical model that analyses data and extracts knowledge. ML produces the training machine models by recognising objects. A SSRF web application vulnerability occurs when the vulnerability exploits the access services. It exploits a software vulnerability that allows an

intruder to trick the server side by modifying the resources accordingly. Attacks succeed if the target application imports the read and write operations over the data without client acknowledgement.

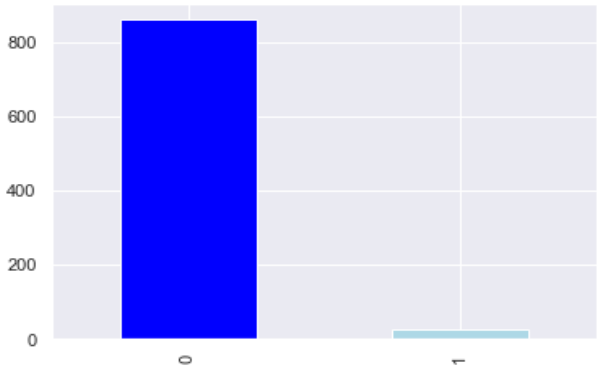

Figure 2: SSRF Attack Vulnerability State

4. Construction

Vulnerability Analysis

Due to its capacity to swiftly analyse massive volumes of data, spot patterns, and predict outcomes, it has grown increasingly useful in server-side vulnerability analysis. Machine learning may aid in the rapid identification of possible system vulnerabilities, including weak credentials, out-of-date software, and unpatched platforms. ML focuses on the development of access data, where it is used to learn each piece of data within a system. Moreover, it may be used to identify harmful behaviour, including malware and brute-force assaults, and notify administrators to take appropriate action. There are several benefits to using machine learning to examine server-side vulnerabilities. Administrators could respond to possible risks right away because of its swift identification capabilities.Evaluating server-side vulnerabilities can be a significant task for machine learning. Automated threat identification, automated vulnerability analysis, and sustaining the constantly evolving characteristics of risks may all be facilitated by it. Because malicious behaviour can be cloaked in seemingly innocent data, it might be challenging to precisely identify possible risks.Vulnerability analysis can be employed to find flaws in corporate procedures in addition to computer software and hardware platforms. It may also be used to identify potential consequences that an adversary might try to employ against the victim. Businesses can more effectively defend themselves against unauthorised assaults by comprehending the potential hazards and weaknesses related to a platform. The technique of identifying, calculating, and controlling hazards that could arise from diverse threats is characterised as vulnerability assessment.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 887 entries, 0 to 886
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   cveID              887 non-null    object
 1   vendorProject      887 non-null    object
 2   product            886 non-null    object
 3   SSRF               887 non-null    int64
 4   vulnerabilityName  887 non-null    object
 5   dateAdded          887 non-null    object
 6   shortDescription   881 non-null    object
 7   requiredAction     887 non-null    object
 8   dueDate            887 non-null    object
dtypes: int64(1), object(8)
memory usage: 62.5+ KB
```

Figure 3: Vulnerabilities Attributes Parameters

	cveID	vendorProject	product	SSRF	vulnerabilityName	dateAdded	shortDescription
3	CVE-2021-27103	Accellion	FTA	1	Accellion FTA SSRF Vulnerability	03-11-2021	Accellion FTA 9_12_411 and earlier is affected...
116	CVE-2016-3718	ImageMagick	ImageMagick	1	ImageMagick SSRF Vulnerability	03-11-2021	The (1) HTTP and (2) FTP coders in ImageMagick...
211	CVE-2015-4852	Oracle	Oracle WebLogic Server	1	Oracle WebLogic Server Remote Code Execution V...	03-11-2021	Allows remote attackers to execute arbitrary c...
212	CVE-2020-14750	Oracle	Oracle WebLogic Server	1	Oracle WebLogic Server Remote Code Execution V...	03-11-2021	Easily exploitable vulnerability allows unauth...
213	CVE-2020-14882	Oracle	Oracle WebLogic Server	1	Oracle WebLogic Server Remote Code Execution V...	03-11-2021	Easily exploitable vulnerability allows unauth...
214	CVE-2020-14883	Oracle	Oracle WebLogic Server	1	Oracle WebLogic Server Remote Code Execution V...	03-11-2021	Easily exploitable vulnerability allows high p...
274	CVE-2021-22005	VMware	vCenter Server	1	VMware vCenter Server File Upload	03-11-2021	VMware vCenter Server file upload vulnerabilit...
275	CVE-2020-3952	VMware	vCenter Server	1	VMware vCenter Server Info Disclosure Vulnerab...	03-11-2021	Under certain conditions, vmdir that ships wit...
276	CVE-2021-21972	VMware	vCenter Server	1	VMware vCenter Server Remote Code Execution Vu...	03-11-2021	The vSphere Client (HTML5) contains a remote c...
277	CVE-2021-21985	VMware	vCenter Server	1	VMware vCenter Server Remote Code Execution Vu...	03-11-2021	The vSphere Client (HTML5) contains a remote c...

(1)

	cveID	vendorProject	product	SSRF	vulnerabilityName	dateAdded	shortDescription	requiredAction	dueDate	
0	CVE-2021-27104	Accellion	FTA	0	Accellion FTA OS Command Injection Vulnerability	03-11-2021	Accellion FTA 9_12_370 and earlier is affected...	Apply updates per vendor instructions.	17-11-2021	
1	CVE-2021-27102	Accellion	FTA	0	Accellion FTA OS Command Injection Vulnerability	03-11-2021	Accellion FTA 9_12_411 and earlier is affected...	Apply updates per vendor instructions.	17-11-2021	
2	CVE-2021-27101	Accellion	FTA	0	Accellion FTA SQL Injection Vulnerability	03-11-2021	Accellion FTA 9_12_370 and earlier is affected...	Apply updates per vendor instructions.	17-11-2021	
4	CVE-2021-21017	Adobe	Acrobat and Reader	0	Adobe Acrobat and Reader Heap-based Buffer Ove...	03-11-2021	Acrobat Reader DC versions versions 2020.013.2...	Apply updates per vendor instructions.	17-11-2021	
5	CVE-2021-28550	Adobe	Acrobat and Reader	0	Adobe Acrobat and Reader Use-After-Free Vulner...	03-11-2021	Acrobat Reader DC versions versions 2021.001.2...	Apply updates per vendor instructions.	17-11-2021	
6	CVE-2018-4939	Adobe	ColdFusion	0	Adobe ColdFusion Deserialization of Untrusted ...	03-11-2021	Adobe ColdFusion Update 5 and earlier versions...	Apply updates per vendor instructions.	03-05-2022	
7	CVE-2018-15961	Adobe	ColdFusion	0	Adobe ColdFusion Remote Code Execution	03-11-2021	Adobe ColdFusion versions July 12 release (201...	Apply updates per vendor instructions.	03-05-2022	
8	CVE-2018-4878	Adobe	Flash Player	0	Adobe Flash Player Use-After-Free Vulnerability	03-11-2021	A use-after-free vulnerability was discovered ...	The impacted product is end-of-life and should...	03-05-2022	
9	CVE-2020-5735	Amcrest	Cameras and Network Video Recorder (NVR)	0	Amcrest Camera and NVR Buffer Overflow Vulnera...	03-11-2021	Amcrest cameras and NVR are vulnerable to a st...	Apply updates per vendor instructions.	03-05-2022	
10	CVE-2019-2215	Android	Android OS	0	Android "AbstractEmu" Root Access Vulnerabilities	03-11-2021	NaN	Apply updates per vendor instructions.	03-05-2022	

(2)

Figure 4: (1) CVE – Existence of SSRF Vulnerability (2) CVE – Non-existence of SSRF Vulnerability

5. Experimental Analysis

Malicious vulnerability evaluation is employed to identify vulnerabilities that attackers might exploit, whereas non-malicious risk assessment is intended to find flaws that authorised users may exploit. Moreover, vulnerability evaluation may aid businesses in creating stronger security protocols and effectively defending them against illegitimate attempts. In advance of being used against them, companies may uncover possible vulnerabilities in systems with the use of vulnerability assessment. In this manner, businesses can lower their danger of information theft, financial compromise, and other malicious attack-related consequences.Security should include vulnerability assessments, as they could assist organisations in identifying and addressing possible threats prior to them becoming an issue. Organizations can defend themselves against malicious assaults by comprehending the potential hazards and weaknesses related to a platform. Focusing on the possibly susceptible service in addition to the potential SSRF attacker targets is important for SSRF mitigation. To find weaknesses in organisational procedures and software platforms, it is often done using automated technologies.

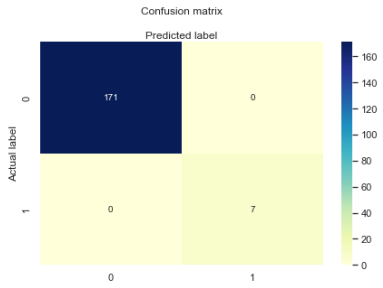

Figure 5: Confusion Matrix of ML algorithm

Random Forest	precision	recall	f1-score	support
0	1.00	1.00	1.00	171
1	1.00	1.00	1.00	7
accuracy			1.00	178
		KNN		
0	0.96	1.00	0.98	171
1	0.00	0.00	0.00	7
accuracy			0.96	178
		SVM		
0	1.00	1.00	1.00	171
1	1.00	1.00	1.00	7
accuracy			1.00	178
		Naïve Bayes Classifier		
0	1.00	1.00	1.00	171
1	1.00	1.00	1.00	7
accuracy			1.00	178

Logistic Regression				
0	1.00	1.00	1.00	171
1	1.00	1.00	1.00	7
accuracy			1.00	178
Decision Tree				
0	1.00	1.00	1.00	171
1	1.00	1.00	1.00	7
accuracy			1.00	178

Table 1: Comparative Analysis

Machine learning interprets data to make predictions and decisions. Metrics used to assess the effectiveness of a machine learning method include precision, recall, the F1 measure, and accuracy. Several metrics assess a model's capacity for accurate categorization and forecasting. Algorithms are employed to evaluate many versions and determine appropriate versions. The reliability of a model's prediction is assessed using the metrics of recall and precision. Recall is the percentage of accurate true positives, whereas precision estimates the percentage of right true positives.90% of the true positives provided by a model with a precision of 0.9 and a recall of 0.8 were accurate, and 80% of the positive cases mostly in the data were properly recognised.

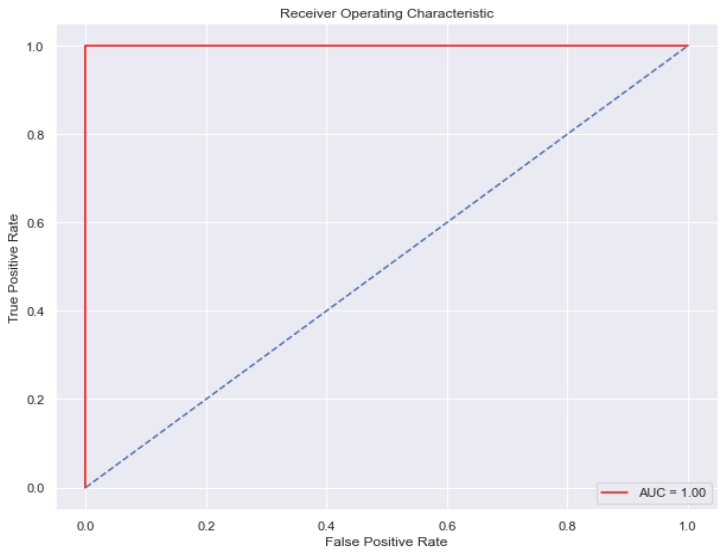

Figure 6: ROC Curve Characteristics

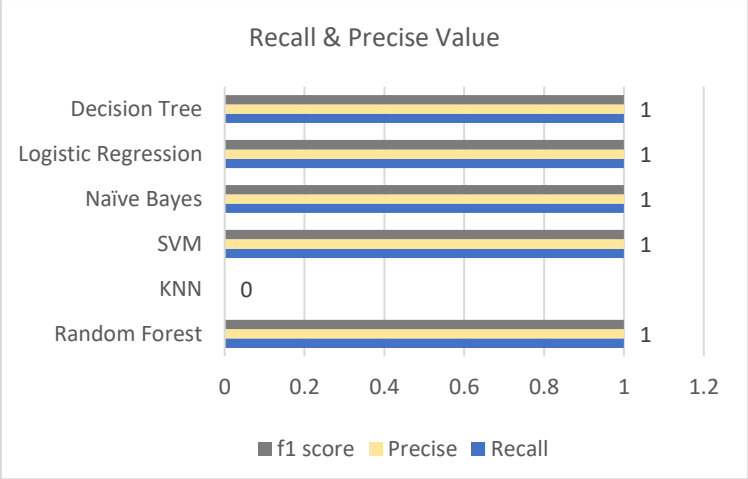

Figure 7: Recall & Precise Value Analysis

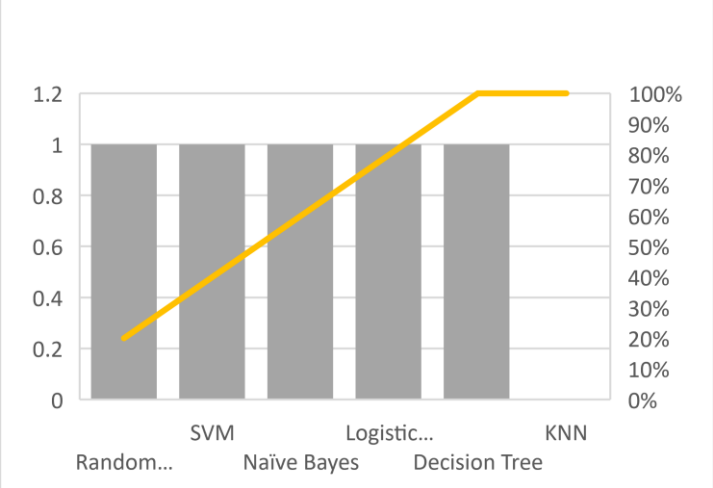

Figure 8: Accuracy Curve Features

6. Conclusion

Vulnerabilities for Server-Side Request Forgery (SSRF) are identified by utilising ML algorithms to calculate the likelihood of incidences involving the attack vulnerability. To prevent server-side forgery attacks, abnormal behaviour in the server logs is identified and tracked. An attacker can also use SSRF to avoid identifying and authorising mechanisms. The primary mechanism of the SSRF attack makes use of the compromised backend systems trust entities in web applications. Organizations can better defend themselves against malicious assaults by comprehending the hazards and vulnerabilities related to a system.The attacker exploits this and replaces the original URL with a malicious URL that points out the server file system. If the server accepts those requests, the attacker can easily penetrate the server's file system, which could cause potential loss and damage.The server side receives the request

through a local service, which looks legitimate.SSRF vulnerability uses the user input that creates the request where the parameter value is vulnerable software on the same server.

7. References

1. K. Dhivya, V. Kannagi, M. Rajkumar, I. Chandra and S. J. Ganesh, "An Effective Server-Side Attack Identification and Prevention Scheme using Logical Query Processing Strategy," 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2022, pp. 945-951, doi: 10.1109/ICEARS53579.2022.9752277.
2. W. BEKRI, T. LAYEB, R. JMAL and L. C. FOURATI, "Intelligent IoT Systems: security issues, attacks, and countermeasures," 2022 International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 2022, pp. 231-236, doi: 10.1109/IWCMC55113.2022.9825120.
3. Al-talak, Khadejah; OnytraAbbass. International Journal of Advanced Computer Science and Applications; West Yorkshire Vol. 12, Iss. 12, (2021). DOI:10.14569/IJACSA.2021.0121230
4. M. Z. Zakaria and R. Kadir, "Risk Assessment of Web Application Penetration Testing on Cross-Site Request Forgery (CSRF) Attacks and Server-Side Includes (SSI) Injections," 2021 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2021, pp. 85-90, doi: 10.1109/ICoDSA53588.2021.9617554.
5. A. S. S. Ahmed, M. Shachi, A. A. Brishty, N. Siddiqui and N. Sakib, "A Hybrid Approach to Detect Injection Attacks on Server-Side Applications Using Data Mining Techniques," 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/STI53101.2021.9732599.
6. S. Gupta and D. Grover, "A Comprehensive Review on Detection of DDoS Attacks using ML in SDN Environment," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 1158-1163, doi: 10.1109/ICAIS50930.2021.9395987.
7. Yeboah-Ofori, Abel (2020) Classification of malware attacks using machine learning in decision tree. International Journal of Security, 11 (2). pp. 10-25. ISSN 1985-2320
8. Calzavara, Stefano & Conti, Mauro &Focardi, Riccardo &Rabitti, Alvise&Tolomei, Gabriele. (2020). Machine Learning for Web Vulnerability Detection: The Case of Cross-Site Request Forgery. IEEE Security & Privacy. PP. 10.1109/MSEC.2019.2961649.

9. Jabiyev, Bahruz&Mirzaei, Omid &Kharraz, Amin &Kirda, Engin. (2020). Preventing Server-Side Request Forgery Attacks. 10.1145/3412841.3442036.
10. S. S. Priya, M. Sivaram, D. Yuvaraj and A. Jayanthiladevi, "Machine Learning based DDOS Detection," 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2020, pp. 234-237, doi: 10.1109/ESCI48226.2020.9167642.
11. Chachra, Anjali and Sharma, Deepak, Applications of Machine Learning Algorithms for Countermeasures to Cyber Attacks (April 8, 2019). 2nd International Conference on Advances in Science & Technology (ICAST) 2019.
12. A. Yeboah-Ofori and C. Boachie, "Malware Attack Predictive Analytics in a Cyber Supply Chain Context Using Machine Learning," 2019 International Conference on Cyber Security and Internet of Things (ICSIoT), Accra, Ghana, 2019, pp. 66-73, doi: 10.1109/ICSIoT47925.2019.00019.
13. H. Luo, "SSRF Vulnerability Attack and Prevention Based on PHP," 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 2019, pp. 469-472, doi: 10.1109/CISCE.2019.00109.
14. H. Shahriar and M. Zulkernine, "Client-Side Detection of Cross-Site Request Forgery Attacks," 2010 IEEE 21st International Symposium on Software Reliability Engineering, San Jose, CA, USA, 2010, pp. 358-367, doi: 10.1109/ISSRE.2010.12.
15. N. Jovanovic, E. Kirda and C. Kruegel, "Preventing Cross Site Request Forgery Attacks," 2006 Securecomm and Workshops, Baltimore, MD, USA, 2006, pp. 1-10, doi: 10.1109/SECCOMW.2006.359531.