Machine Learning for Web Vulnerability Detection: The Case of Cross-Site Request Forgery

[1]Prof.Kanchan Umavane,[2]Mr. Kunal Jain , [3]Prathamesh Vishwakarma, [4]Manoj Verma

[1]Asst.Professor,[2,3,4]UG Student,[1,2,3,4]Computer Engg. Dept. Shivajirao S. Jondhle College of Engineering & Technology, Asangaon, Maharashtra, India. *[1]kanchanumavane2020@gmail.com, [2]kunaljain2104@gmail.com, [3]pratham46vishwa@gmail.com, [4]vermamanoj000.mv@gmail.com*

Abstract- In this research, author present an approach for detecting web application vulnerabilities using Machine Learning (ML). Due to their diversity and broad use of bespoke programming approaches, web applications are particularly difficult to analyses. As a result, machine learning highly valuable in web security: It might blend human understanding of web app semantics into mechanized analysis techniques using manually explained data. These approaches were used to create Mitch, the first machine learning sol for black-box diagnosis off Cross-Site Request Forgery's (CSRF) vulnerabilities. Mitch assisted us in discovering 35 new cross-site request forgery's (CSRFs) across 20 major domains, as well as three additional CSRFs in production software.

Keywords: Web application security, Machine Learning(Ml), Cross-Site Request Forgery's.

I. INTRODUCTION

In today's world, web apps are the most popular interface to security sensitive data and functionality. To name a few common use cases, they are frequently used to file tax returns, view the results of medical exams, conduct financial transactions, and express thoughts with friend social circles. On the flipside, this makes web applications ideal targets for malevolent users (attackers) who want to cause financial losses, get unauthorized access to personal data, or embarrass their victims. It is commonly known that securing online applications is difficult. There are various reasons for this, ranging from the web platform's diversity and intricacy to the use of unstructured scripting languages with questionable security assurances and that are not accessible to static analysis. Black-box vulnerability detection approaches are very common in this situation. Black-box methods operate at the level of HTTP traffic, i.e. HTTPs requests and responses, as opposed to white - box techniques which need access to the web application source code. Though this narrow perspective may overlook critical insights, it does have the advantage of providing a language-agnostic vulnerability detection approach that abstracts from the complex of scripting languages and provides a unified edge to the widest range of online applications conceivable. Although this seems enticing, past research has revealed that such an analysis is far from simple. One of the key matters is exposing of critical component of efficient vulnerability detection, namely, an understanding of the web application semantics, to automated techniques.

Cross-Site Request Forgery's(CSRF) is also called as a well-known online attack that causes a genuine user to submit undesired, attacker-controlled HTTP requests to an exposed web application. The essential notion behind CSRF is that fraudulent requirements are routed through the user's browser to the web app, making them indistinguishable from legitimate caring requests permitted by the user.

A typical CSRF attack works as follows:

1) Alice accesses to trustworthy vulnerable web application, such as her favorite social networking site. A session cookie is used to implement session authentication, which is instantly assigned by the browser to any future request to the web application.

2) Alice opens a new tab in browser and go to a vulnerable website, such as a Facebook site, which returns a website page with a malicious ad.

3) The malicious ads send a cross-site request to the social media platform using HTML or JavaScript, for example, asking to "like" a given political party.

It's worth observing that CSRF doesn't need the attacker to intercepting or altering the victim's requests and answers; the victim has to do is visit the attacker's website, from which the assault is launched. As a result, any rogue website on the Internet can take full advantage of CSRF vulnerabilities.

Fig .Cross site request forgery(example)

 DOI : 10.35291/2454-9150.2022.0162

Preventing CSRF

To avoid CSRF, web developers have to implement explicit protection mechanisms. Force re-authentication or use one-time credentials / CAPTCHAs to prevent cross-site requests from going undetected if add additional user engagement does not negatively impact usability. Automated protection is desirable in many cases: the newly available Same Site cookie feature may be utilised to prevent cookie attachment on cross-site requests, which removes the core source of CSRF and is highly recommend for new website applications. Unfortunately, this Défense is not yet widespread and existing web applications typically filter out cross-site request by using any of the following techniques:

1) examining the value of common HTTPs request headers like Referrer and Origin, which indicate the page from which the request originated;

2) checking for custom HTTP request headers, such as X-Requested-With, that can't be set from a cross-site location;

3) Anti-CSRF tokens set by the server into sensitive forms are being checked for the existence of unanticipated Anti-CSRF tokens.The advantages and disadvantages of these various strategies are discussed in a recent work . All three solutions, however, have the same drawback: they all necessitate the precise and fine-grained placement of security checks.

II. AIMS AND OBJECTIVE

a) Aim:

The purpose of this project is to create machine language that can detect and stop Cross-Site Request Forgery's attacks. By utilizing Machine Language, it is possible to identify and mitigate Cross-Site Request Forgery's assaults, hence preventing CSRF attacks and keeping websites safe from them.

b) Objective:

Using machine learning classifiers, design a system that can successfully prevent malicious Link assaults (Cross-Site Request Forgery's), as well as uncover the discriminative properties that define the attack and therefore lower the false positive value. The project's primary goal is to **DETECT WEB VULNERBILITY IN WEBSITE.**

III. LITERATURE SURVEY

1)Surviving the Internet: An Exploration into Web Session Security

The Web is the most prevalent method to access online data and applications. It is incredibly complicated and diverse, since it incorporates a plethora of dynamic material created by many parties in order to provide the best possible user experience. This focus on attacks against web sessions, i.e., attack on authentic web browser users who are attempting to create an authenticated connection with a trustworthy source application. This kind of attack takes use of the Web's inherent complexity by interfering with dynamic content, client-side storage, or cross-domain connections, for example, in order to alter browser activity and/or network traffic. This decision is based on the reality that online session assaults are a significant subgroup of important web security events, and numerous potential countermeasures, working at various levels, have been presented to prevent them.This research examines common attacks against web sessions and organises them based on I their attacker model and (ii) the security characteristics they violate.. This first categorization is important for determining whether of an internet session's intended security features can be broken by an attack and how. When security is guaranteed only under certain assumptions, this make these assumptions explicit. This also consider the influence of each security solution on compatibility and usability, as well as implementation simplicity.

2) Large-Scale Analysis & Detection of Authentication Cross-Site Request Forgeries

CSRF (Cross-Site Request Forgery's) attacks are a serious danger to online applications. In this research, this focus on cross-site request forgery's (CSRF) attacks that target internet sites' authentication & identity management functions. This'll refer to them as Authentication CSRF as a group (Auth-CSRF in short). This began by gathering many Auth-CSRF attacks documented in the literature, analyzing their underlying techniques, and identifying seven security testing strategies that might assist a manual tester in uncovering Auth-CSRF vulnerabilities. This experiment is examining the efficacy of given testing methodology and estimate the prevalence of Auth-CSRF, this conducted an experimental inquiry including 300 web sites from three distinct rank ranges of the Alexa worldwide top 1500 to assess the efficiency of this testing methodologies and estimate the occurrence of Auth-CSRF. The conclusions of this tests are alarming: just 133 of the 301 web sites given evaluated qualified for given tests, and 90 of them had at least one vulnerability that allowed Auth-CSRF to operate (i.e., 68 percent). This summarized the test methods, enhanced them with the wisdom that gained from these experimental tests, and executed them as a CSRF-checker extension towards the open-source penetration vulurablity tool OWASP ZAP. This used CSRFchecker to test 132 extra websites (all from the Alexa world's top 1500) and found 95 that were susceptible (i.e., 72 percent). The given discoveries reveal major flaws in Microsoft, Google, and eBay websites, among others.

3) AUTOMATED BLACK-BOX WEB APP LOOPHOLE TESTING IS THE STATE OF THE FINE ART.

Online application vulnerability scanners, often known as black-box scanners, are automated technologies that analyze web app for security flaws. This used up to eight prominent tools to assess the present state of the sculpture and

 DOI : 10.35291/2454-9150.2022.0162

performed research into: (i) the type of vulnerabilities these scanners evaluate, (ii) their efficiency against target vulnerabilities, and (iii) the relevance of the target vulnerabilities to real-world weaknesses. This employed a bespoke online application vulnerable to known and anticipated flaws, as well as prior versions of widely used web apps with known vulnerabilities, to perform this research. These findings disclose the possible and usefulness of automated technologies as a whole, as well as some drawbacks. Many technologies do not now detect "stored" variants of Cross Site Scripting' (XSS) and SQL Injection (SQLI) vulnerabilities.

They do not offer comparison statistics or give suggestions concerning the purchase of certain products since our purpose is to assess the possibility of future .

IV. EXISTING SYSTEM:

Securing online apps is generally acknowledged to be difficult under the current setup. There are a number of reasons for this, varying from the web platform's diversity and complexity to the employment of unregulated scripting languages with dubious security guarantees that are not available to static analysis. Though this limited outlook might miss important awareness, it has the key advantage of offering a language-agnostic vulnerability detection approach, which abstracts from the complexity of scripting languages and offers a uniform connection to the vast possible range of web applications.

V. COMPARTIVE STUDY

SR NO.	PAPER TITLE	AUTHOR NAME	METHOD	ADVANTAGE	DISADVANTAGE
1.	An Exploration into Web Transaction Security	Stefano Calzavara', Riccardo Focardi', Marco Squarcina', & Mauro Tempesta	Machine Learning	It has clean practical benefits allowing a comprehensive identification of all the attack vectors.	It is complex and variegate being vulnerable to outer attack vectors.
2.	LARGE-SCALE ANALYSIS & DETECTION OF AUTHENTICATION CROSS-SITE REQUEST FORGERIES.	Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessandro Armando, and Umberto Morelli	Machine Learning	Good Approach Explained	Time Consuming
3.	Automated Black Web Applications Vulnerability Test	Jason Baau, Elie Burszteinn, Divij Guptaa, and John C. Miitchell	Machine Learning	Good Approach Explained	Difficult to understand

Fig1.Compartive study table

VI. PROBLEM STATEMENT

Web apps have been a main target for hackers in past year, web apps have been a popular target for hackers, with malware, notably JavaScript, being injected into them to perform malicious behavior such as caricature, years, with malware, particularly JavaScript, has been injected into them for years, causing them to perform dangerous behavior's such as fraud. The number of assaults on users based on browser exploits has escalated at an alarming rate. The existing attack prevention planned have failed dourly in most of the states. Furthermore, people haven't yet taken the time to safeguard their browsers by utilising accessible extensions and snap-ins. Unless the user disables it, the scripts are also performed automatically without the user's consent. By creating a new record system for privileged levels and vulnerabilities levels of the information handed over in the browser, this proposal proposes an enhanced XSS detection approach. Machine learning techniques are used to keep, classify, and evaluate the java scripts that are displayed in browsers. Machine learning may also be used to predict browser quirks and create a pattern of attackers. As a result, machine learning tries to address a user's problem faster than traditional XSS detection systems.

VII. PROPOSED SYSTEM

Cross-Site Request Forgery's (CSRF) is a well-known online attack that causes an authorized user to send undesired, assaulter HTTP requests to a victim's browser. The key concept of CSRF is that the malicious requests are dispel to the web application through the user's browser, hence they might be identical from intended inoffensive requests which were actually authorized by the user**.** The CSRF doesn't really necessitate the attacker intercepting or modifying the victim's queries and answers; all that is required is that the target visit the attacker's webpage, from where the attack is started. Thus, Any rogue website on the Internet is suspicious of CSRF vulnerabilities.

ADVANTAGES OF PROPOSED SYSTEM:

The value of standard HTTP request headers such as Invoke and Dawn, indicating the page originating the request. Custom HTTP request titles, such as X-Requested-With, that cannot be laid from a cross-site location . The presence of

 DOI : 10.35291/2454-9150.2022.0162

unanticipated anti-CSRF tokens inserted into critical forms by the server.

Algorithm: Logistic Regression, K-Nearest K-neighbours Algorithm (k=3), SVM, Nave-Bayes Algorithm

VIII. ALGORITHM

Step 1 : Start.
Step 2 : Collect the dataset i.e., False Negative, True Positive
Step 3 : Train the Model
Step 4 : Use Machine Language for Prediction
random forest
clf←Model(model←RandomForestClassifie
r(),X←X,y←y) clf.crossValScore(cv←10)
clf.crossValScore(cv←10)
clf.accuracy()
clf.confusionMatrix()
clf.classificationReport()
st_x= StandardScaler()
x_train← st_x.fit_transform(x_train)
x_test ← st_x.transform(x_test)
2 .Naïve bayes
from sklearn.linear_model import load_iris
iris←load_iris()
store the feature matrix (X) and response vector (y)
x←iris.feature
y←iris.target
From_sklearn.model_selection
import train_test_split
x_train, x_test, y_train, y_test ← train_test_split
(feature,target,test_size←0.4,random_state←1)
from sklearn.naive_bayes
import GaussianNB
model = GaussianNB()
model.fit(feature, target)
y_pred ←model.predict(x_test)
from sklearn. metrics import confusion_matrix, classification_report
print("Gaussian NB model accuracy(in %):",
metrics.accuracy_score(y_test, y_pred)*100)
dt←Model(model←DecisionTreeClassifier(),
x←x,y←y)
dt.crossValScore(cv)
dt.accuracy()
dt.confusion_matrix()
dt.classification_report()
Step 5 : Use SVM classify the data & Visualizing result from above algorithm
Step 6 : Test and run the model
Step 7 : End.

IX. MATHEMATICAL MODEL

Four supervised machine learning classifiers were used in this tool for getting the best results:

1. K-Nearest Neighbours Algorithm(k=3)
2. Support Vector Machines
3. Naïve-Bayes Algorithm
4. Logistic Regression

KNN (K- Nearest Neighbor):

It may be used to solve problems involving grouping and regression. However, in the industry, it is more widely employed in categorization difficulties. The K-Nearest Neighbor Technique is a basic ML algorithm that uses a majority vote of its k neighbours to reserve all available situations and codify new ones. As determined by a distance function, the case given to the class is the most common among its K nearest neighbours.This distance problem can be of the Euclidean, Manhattan, Minkowski, or Hamming types. The first 3 functions are used to represent continuous functions, while the fourth (Hamming) is used to represent unambiguous variables. If K = 1, the instance is unambiguously classified as belonging to the class of its closest neighbour. When doing KNN modelling, selecting K might be risky at times.

Euclidean distance function:

$$2\sqrt{\sum k\ x=1\ (xi-yi\ )}\ \mathbf{2}$$

Manhattan distance function:

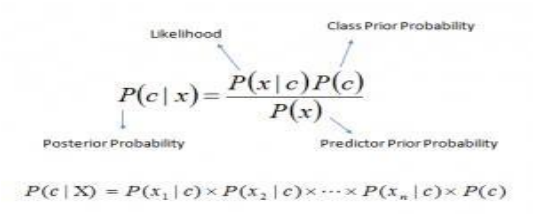
$$\sum \boldsymbol{ki}=\mathbf{1}|\boldsymbol{xi}-\boldsymbol{yi}|$$

Minkowski distance function:

$$(\sum ki=1(xi-yi)q)1/q$$

NAVIE BAYES:

A Naive Bayes Classification algo, in basic words, posits that the appearance of one attribute in a class is independent of the presence of some other feature. The naive Bayesian model is straightforward to construct and is especially effective for huge data sets. Naive Bayes is recognized to exceed even the most advanced categorization systems in terms of clarity. Using P(c), P(x), and P(x|c), the Bayes theorem describes how to compute the posterior distribution P(c|x) from P(c), P(x), and P(x|c).



The rear probability of category (target) given predictor is : P(C|x) (attribute).

The initial probability of class is P(c).P(x|c) denotes the probability of a predictor in a particular class's.P(x) is the predictor's starting probability.

Logistic Regression is a statistical method for predicting the outcome of Generalized Linear Models are a large class of

 DOI : 10.35291/2454-9150.2022.0162

techniques that includes logistic regression (glm). Nelder and Wedderburn created this model in 1972 as a way of applying linear regression to issues that were not immediately suitable for it. In fact, they provided a set of theories (linear regression, Poisson Regression ANOVA, and etc.) that also included logistic regression as a special instance. The fundamental calculation of generalized linear model is: $g(E(y)) = \alpha + \beta x1 + \gamma x2$ .Here, g () is the link function, E(y) is the presumption of target variable and $\alpha + \beta x1 + \gamma x2$ is the linear predictor (α, β, γ to be predicted). The role of connection function is to 'link' the assumption of y to linear predictor. Based on the results generated, the classifier with the best accuracy score is picked among the four classifiers to generate the result thus accounting to correctness in result prediction.

Support Vector Machines (SVM):

It were created in 1992 by Boser and colleagues. Vapnik and Lerner published the first optimum hyperplane method in 1963. A SVM is a supervised machine learning's technique that may be used to classify high-dimensional data into binary categories. The SVM method works by locating the descriptor with the shortest minimum distance between the training instances. By employing kernel to map the input data into a greater space and separate the data on the mapped dimension, SVM may be extended to data that is not linearly separable.

Performance Evaluation

$$\text{Accuracy} = \frac{No.of classified\ benign\ scripts}{Total\ No.of\ benign\ sample} X100$$

When an attack detection technique wrongly perceives a regular code as harmful code, this is known as a false positive situation. A false negative arises when a malicious code is not recognised despite its unlawful behaviour in a certain view point. Detection rate can be measured by using ambiguity matrix for the assessment of false positives, & false negatives. False positive and False negative detection rate is calculated by where FPR is false positive figure, FNR is false negative figure, FN is false negative, TN is true negative, and TP is true positive. True negative clearly shows a number of correctly identified negative samples, false negative implies a caused by malicious specimen identified as negative, false positive indicates a detection of fake trials acknowledged as negative, and true positive suggests a number of malware samples correctly identified. The pace at which rogue scripts are executed will determine the performance of the suggested detection. The delay time will be measured in the presence and absence of an interceptor while displaying a page, and the system resource consumption will be determined in both circumstances.

Procedure for the experiment: In four phases, the experiment is completed. The first step is to create malicious and benign URL lists. Second, characteristics are separated into training and test groups. Thirdly, the model is generated using the train set. And finally, the generated model is checked using the test information set.

Accuracy: The accuracy is well-defined as the % of successfully identified examples over all examples in the test set. The test set is applied to the four classifiers k-NN (K=3), Naïve Bayes (distribution='Gaussian'), SVM (kernel = 'rbf', random_state = None) and Logistic Regression and the performance is shown.

Features Significance: In this tool, some relevant attack features have been proposed that remain resilient against possible anticipated future attacks. The URL lexical and page content based features rise up the true positive rate where the JavaScript features have a significant effect on the true positive rate. In version 1.0, only the URL lexical and page content features have been used. For real time-data, the tool failed to predict the result correctly due to the presence of lots of false positives leading to lesser accuracy. Thus, in version 3.0, dynamic JavaScript features have been proposed to detect malicious JavaScript pages in real-time. This lead to a very high accuracy of around 98% (Logistic Regression).

X. SYSTEM ARCHITECTURE

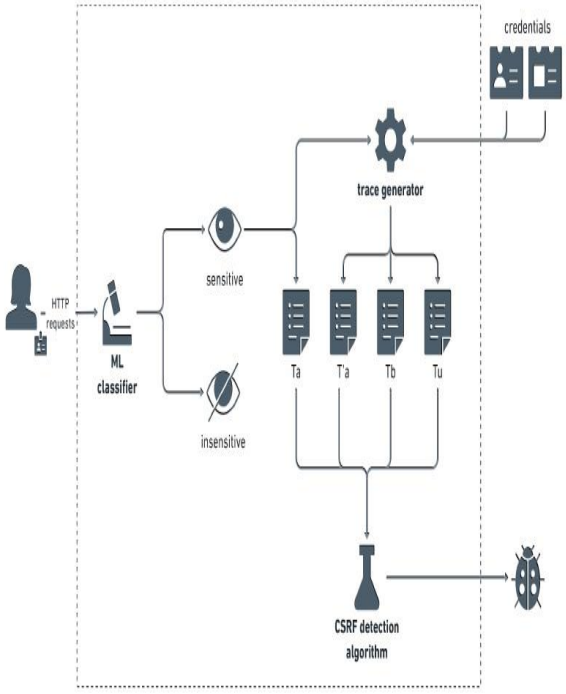


Fig.1: System Architecture

Description: There are Two phases:

1.Training phase 2.Testing phase

1)Training Phase: In which system take data from all Alexa global top 1500 website for conducting an experiment to determine a CSRFs.

 DOI : 10.35291/2454-9150.2022.0162

2)Testing Phase: In testing phase it verify the data of the new website. Then It will cross check with Alexa global top 1500 website and it will show the result whether the website is genuine or forged.

XI. ADVANATGES

1) Helps solve complex real-world problems with several limitation.

2)Tackle problems like having small or almost no labeled data availability**.**

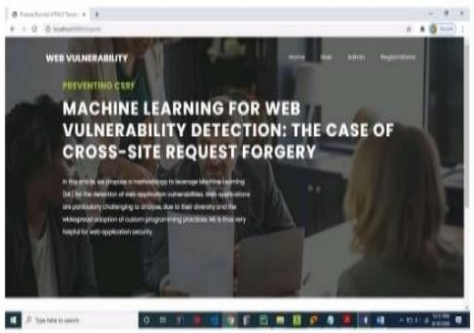
3)The ease with which information may be transferred from one model to another based on provinces and issues.

4) Provides a roadmap for developing Artificial General Intelligence in the future..

XII. DESIGN DETAILS

SCREEN SHOTS

Home page:



User Registration Form

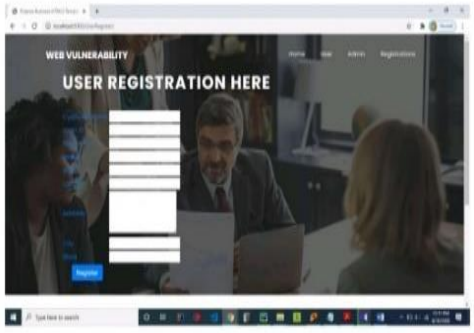


Fig 1: User Registration form

XIII. CONCLUSION

Thus, we have tried to implement the paper of Author “Stefano Calzavara, Mauro Conti, Riccardo Focardi and Alvise Rabitti, Gabriele Tolomei”, “Machine Learning For Web Vulnerability Detection-The Case Of Cross-Site Request Forgery” and IEEE version 2020. This diversity and widespread use of bespoke programming approaches, web - based applications are particularly difficult to analyse. In the online scenario, ML is particularly useful since it can utilise manually labelled data to demonstrate to automated analysis tools the human comprehension interpretation of web applications That proved this assertion by creating Mitch, the first machine learning solution for blackbox detection of CSRFs vulnerabilities, and testing its effectiveness. This expect that other researchers will be able to use our method to discover different types of web application vulnerabilities.

XIV. REFERENCE

[1] Stefano Calzavara, Mauro Conti, Riccardo Focardi and Alvise Rabitti, Gabriele Tolomei (2020), Machine Learning For Web Vulnerability Detection-The Case Of Cross-Site Request Forgery . Published by the IEEE Security & Privacy,vol.18, doi:10.1109/MSEC.2019.2961649 .

[2] Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessandro Armando, and Umberto Morelli. Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessandro Armando, and Umberto Morelli.Analysis and detection of authenticated cross-site request forgeries on a large scale. EuroS&P 2017, Paris, France, April 26-28, 2017, pages 350–365, in 2017 IEEE European Symposium on Security and Privacy.

[3] Alvise Rabitti, Alessio Ragazzo, and Michele Bugliesi are Stefano Calzavara, Alvise Rabitti, Alessio Ragazzo, and Michele Bugliesi. Web sessions are being tested for integrity problems. ESORICS 2019, Luxembourg, Luxembourg, September 23–27, 2019, pages 606–624, in Computer Security - 24th European Symposium on Research in Computer Security

[4] OWASP. OWASP Testing Guide. https://www.owasp.org/index.php/ OWASP Testing Guide v4 Table of Contents, 2016.

[5] Jason Bau, Elie Bursztein, Divij Gupta, and John C. Mitchell are among the participants. Automated black-box web application vulnerability testing is the state of the art. In 31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berkeley/Oakland, California, USA, pages 332–345, 2010.

[6] Giovanni Vigna, Adam Doup'e, and Marco Cova. An examination of black-box online vulnerability scanners and why Johnny can't pentest. 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA 2010, Bonn, Germany, July 8-9, 2010. Proceedings, pp 111–131, 2010.

[7]Adam Barth, Collin Jackson, and John C. Mitchell are the authors of this book. Cross-site request forgery protections that are strong. CCS 2008, Alexandria, Virginia, USA, October 27–31, 2008, pages 75–88, in Proceedings of the 2008 ACM Conference on Computer and Communications Security.

[8] Ameet Talwalkar, Mehryar Mohri, and Afshin Rostamizadeh Machine Learning Foundations is a course

 DOI : 10.35291/2454-9150.2022.0162

that teaches you the fundamentals of machine learning. The MIT Press published this book in 2012.

[9] Michael W. Kattan, Dennis A. Adams, and Michael S. Parks are the authors of this book. Machine learning and human judgement are compared. March 1993, Journal of Management Information Systems, 9(4):37–57.

[10] Ferrucci, D. A. "This is Watson" begins with an introduction. May 2012, IBM Journal of Research and Development, 56(3):235–249.

[11] Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Deep neural networks and tree search are used to master the game of Go. Jan 2016, Nature 529(7587):484–489.

[12] Wilayat Khan, Michele Bugliesi, Stefano Calzavara, Riccardo Focardi Cookiext is a browser patch that protects against session hijacking attacks. Journal of Computer Security, Vol. 23, No. 4, 2015, pp. 509–537.

[13] Salvatore Orlando, Stefano Calzavara, Gabriele Tolomei, Andrea Casini, Michele Bugliesi, and Gabriele Tolomei On the web, a supervised learning strategy to safeguard client authentication. TWEB, 9(3), 2015, 15:1–15:30.

[14] Gabriele Tolomei, Stefano Calzavara, Mauro Conti, Riccardo Focardi, Alvise Rabitti Mitch: A machine learning technique to detecting CSRF vulnerabilities in the blackbox. EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019, pages 528–543, in IEEE European Symposium on Security and Privacy,

[15] Martin Johns, Simon Koch, Michael Backes, and Christian Rossow. Giancarlo Pellegrino, Martin Johns, Simon Koch, Michael Backes, and Christian Rossow. Deemon: Using dynamic analysis and property graphs to detect CSRF. CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pages 1757–1771, in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.

 DOI : 10.35291/2454-9150.2022.0162