JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 4502-4504
https://publishoa.com
ISSN: 1309-3452

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

Terupally Chandrika

Rapole swetha

Pokala Jyothi

Under the guidance of

Mr. P.Prashanth Kumar

JNTUH

Received 2022 March 15; Revised 2022 April 20; Accepted 2022 May 10.

Abstract

In this project, we propose a methodology to leverage Machine Learning (ML) for the detection of web application vulnerabilities. Web applications are particularly challenging to analyses, due to their diversity and the widespread adoption of custom programming practices. ML is thus very helpful for web application security: it can take advantage of manually labeled data to bring the human understanding of the web application semantics into automated analysis tools. We use our methodology in the design of Mitch, the first ML solution for the black-box detection of Cross-Site Request Forgery (CSRF) vulnerabilities. Mitch allowed us to identify 35 new CSRFs on 20 major websites and 3 new CSRFs on production software.

Keywords: Machine Learning, Web vulnerability, Mitch tool, Cross site request forgery, Black box detection methods**.**

1. Introduction

Web applications became common part of every ones life. They are used in various fields like medical, business, banking etc. Web applications are of different types like static web applications, mobile web applications, e-commerce web applications etc. Some of the applications ask the user to create account where the user enter personal information like email address, password, bank account details etc. This means that they can access to our personal details. Though, Web applications provide security to our personal data there occurs some vulnerabilities which steal personal data. Providing security to personal data is the main challenging task.

Web applications are build using different custom programming practices like Bootstrap, Javascript, ASP.net,PHP, Servlets, CSS etc. Though these practices provide security there occurs some vulnerabilities like CSRF, SQL Injections, Cross site scripting, DOS attack, etc. Vulnerabilities lead to the theft of personal data i.e; sensitive information like credit card information, passwords etc by attackers. Attackers mainly concentrate on shopping web applications, forms, login pages etc. So, Securing web application is the most challenging task. Black Box detection methods which operate at HTTP traffic are most popular detection methods. This approach abstracts complexity and provides uniform interface. And the main challenging to expose automated tools i.e; understanding of web application semantics.

For example: CSRF , a vulnerability that forces person to submit unwanted request towards a web vulnerable application. The person submit request from authenticated website. Hence, attacker routes to web application through the person browser , they cannot be identified by person.

2. Literature Survey

We identify multiple attacks during web session and we classify them based on the type of attack mode i.e; type of vulnerability and the security properties they break. This classification allows us to understand how the attack occur and what security policies it break. And also identify existing security mechanisms and solutions that prevent the attacks or vulnerabilities. When any security policy is accepted under some assumptions explicit. We evaluate compatibility and usability of each security solution. This evaluation helps to understand upto what extent they may be amenable for large scale adoption on the web. We have different proposals to provide security against these attacks. We discuss which attacks it prevents with type of attack.

Cross site request forgery is one of the web vulnerability. In this project , we consider CSRF vulnerability on website authentication and identity management. We started study by segregating many authenticated CSRF attacks and analysed strategies which helps tester for testing process. To check the effectiveness of our security testing strategies, we experimented 300 websites which belong to 3 different ran ranges of the Alexa global top 1500. The results of the are: out of 300 websites we have taken, 133 gave positive results and 90 of them suffered vulnerabilities. So, we further generalized and improved them with our experience during experiment and implemented it as an extension to the open-source penetration testing tool OWASP ZAP. With this CSRF checker we experimented 132 websites, identified 92 vulnerable ones. Our findings include serious attacks among Microsoft, eBay etc websites. We disclosed this to affected vendors.

CSRF is oldest and simplest vulnerability, it is still effective on many websites and leads to serious consequences like economic losses, account takeovers. Though, we have tools to identify CSRF attacks they need manual review by human experts or availability of source code of the web application. In this paper, we design a Mitch tool, the first machine learning solution for black-box detection of CSRF vulnerabilities.

In mitch, we have automated detector of sensitive HTTP requests. We trained mitch detector using supervised machine learning techniques with a dataset of 6000 HTTP requests from popular website. Our solution performs existing detection discovery, allowed us to identify 35 new CSRFs on 20 major websites and 3 new CSRFs on production software.

3. Methods

A.USER MODULE

This is user module, where user first needs to get register to create account. In this module user can perform mitch process to get detected sites of CSRF attacks, can get websites CSRF detected algorithms, accuracy and precision of machine learning classifier and the existing systems accuracy for comparison. User account need to be activated by admin otherwise user cannot login.

B. ADMIN MODULE

In admin module, admin gets logged in using username and password. Admin can get CSRFs of a websites (dataset which is uploaded) and can display post and get view of data. Admin need to activate account of the registered user so that user can access application otherwise it shows invalid username or password.

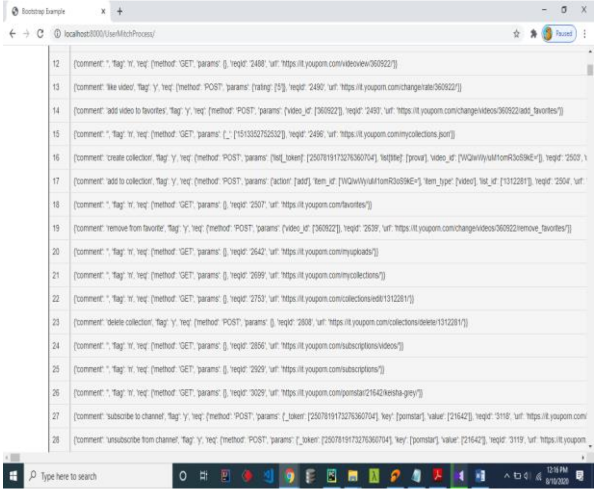
C.FALSE POSITIVES AND FALSE NEGATIVES

In false positive and false negative module, mitch results a false positive when it cannot identify candidate CSRF. This identification can be done by manual testing but it is tedious and time consuming. In common, to find out false is impossible, because we need to know all CSRFs on tested websites. To calculate this, we track all sensitive HTTPs requests returned by classifier and focus on manual testing. This makes the analysis traceable as the classifier uses standard measures.

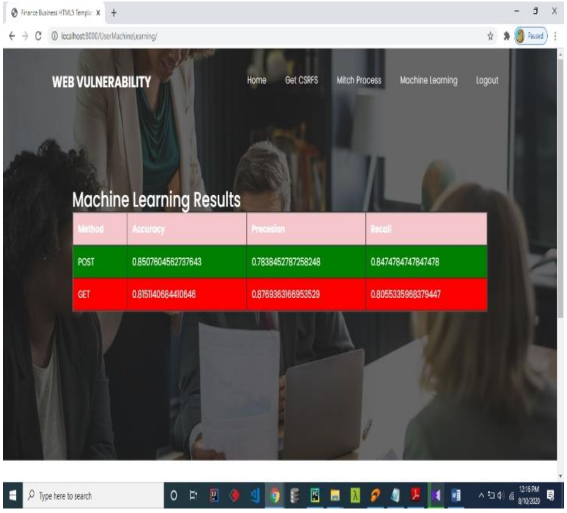
D. MACHINE LEARNING CLASSIFIER

In machine learning classifier, machine learning is trained with 6000 https requests by two human experts. The feature space has 49 dimensions and each one has specific HTTPs request property, the numerical features are classified into following like numofparams, numofbools, numofIds, etc. Machine learning classifier also gives accuracy, precision and recall of tool.

4. Results



```
12  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '2488', 'url': 'https://it.youporn.com/videoview/360922/'}}
13  {'comment': 'like video', 'flag': 'y', 'req': {'method': 'POST', 'params': {'rating': ['5']}, 'reqid': '2490', 'url': 'https://it.youporn.com/change/rate/360922/'}}
14  {'comment': 'add video to favorites', 'flag': 'y', 'req': {'method': 'POST', 'params': {'video_id': ['360922']}, 'reqid': '2493', 'url': 'https://it.youporn.com/change/videos/360922/add_favorites/'}}
15  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {'_': ['1513352752532']}, 'reqid': '2496', 'url': 'https://it.youporn.com/mycollections.json'}}
16  {'comment': 'create collection', 'flag': 'y', 'req': {'method': 'POST', 'params': {'list[_token]': ['250781917327636070 4'], 'list[title]': ['prova'], 'video_id': ['WQlwWy/uM1omR3oS9kE=']}, 'reqid': '2503', 'u
17  {'comment': 'add to collection', 'flag': 'y', 'req': {'method': 'POST', 'params': {'action': ['add'], 'item_id': ['WQlwWy/uM1omR3oS9kE='], 'item_type': ['video'], 'list_id': ['1312281']}, 'reqid': '2504', 'url': '
18  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '2507', 'url': 'https://it.youporn.com/favorites/'}}
19  {'comment': 'remove from favorite', 'flag': 'y', 'req': {'method': 'POST', 'params': {'video_id': ['360922']}, 'reqid': '2639', 'url': 'https://it.youporn.com/change/videos/360922/remove_favorites/'}}
20  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '2642', 'url': 'https://it.youporn.com/myuploads/'}}
21  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '2699', 'url': 'https://it.youporn.com/mycollections/'}}
22  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '2753', 'url': 'https://it.youporn.com/collections/edit/1312281/'}}
23  {'comment': 'delete collection', 'flag': 'y', 'req': {'method': 'POST', 'params': {}, 'reqid': '2808', 'url': 'https://it.youporn.com/collections/delete/1312281/'}}
24  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '2856', 'url': 'https://it.youporn.com/subscriptions/videos/'}}
25  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '2929', 'url': 'https://it.youporn.com/subscriptions/'}}
26  {'comment': '', 'flag': 'n', 'req': {'method': 'GET', 'params': {}, 'reqid': '3029', 'url': 'https://it.youporn.com/pornstar/21642/keisha-grey/'}}
27  {'comment': 'subscribe to channel', 'flag': 'y', 'req': {'method': 'POST', 'params': {'_token': ['2507819173276360704'], 'key': ['pornstar'], 'value': ['21642']}, 'reqid': '3118', 'url': 'https://it.youporn.com/
28  {'comment': 'unsubscribe from channel', 'flag': 'y', 'req': {'method': 'POST', 'params': {'_token': ['2507819173276360704'], 'key': ['pornstar'], 'value': ['21642']}, 'reqid': '3119', 'url': 'https://it.youporn
```



Machine Learning Results

Method	Accuracy	Precesion	Recall
POST	0.8507604562737643	0.7838452787258248	0.8474784747847478
GET	0.8151140684410646	0.8769363166953529	0.8055335968379447

5. Discussion

Web applications are particularly challenging to analyse, due to their diversity and the widespread adoption of custom programming practices. ML is thus very helpful in the web setting, because it can take advantage of manually labeled data to expose the human understanding of the web application semantics to automated analysis tools. We validated this claim by designing Mitch, the first ML solution for the black box detection of CSRF vulnerabilities, and by experimentally assessing its effectiveness. We hope other researchers might take advantage of our methodology for the detection of other classes of web application vulnerabilities.

6. Refrences

1. Stefano Calzavara, Riccardo Focardi, Marco Squarcina, and Mauro Tempesta. Surviving the web: A journey into web session security. ACM Comput. Surv., 50(1):13:1–13:34, 2017.
2. Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessandro Armando, and Umberto Morelli. Large-scale analysis & detection of authentication cross-site request forgeries. In 2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017, pages 350–365, 2017.
3. Stefano Calzavara, Alvise Rabitti, Alessio Ragazzo, and Michele Bugliesi. Testing for integrity flaws in web sessions. In Computer Security - 24rd European Symposium on Research in Computer Security, ESORICS 2019, Luxembourg, Luxembourg, September 23-27, 2019, pages 606–624, 2019.
4. Jason Bau, Elie Bursztein, Divij Gupta, and John C. Mitchell. State of the art: Automated black-box web application vulnerability testing. In 31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berkeley/Oakland, California, USA, pages 332–345, 2010.
5. Adam Doup´e, Marco Cova, and Giovanni Vigna. Why johnny can't pentest: An analysis of black-box web vulnerability scanners. In Detection of Intrusions and Malware, and Vulnerability Assessment, 7th International Conference, DIMVA 2010, Bonn, Germany, July 8-9, 2010. Proceedings, pages 111–131, 2010.
6. Adam Barth, Collin Jackson, and John C. Mitchell. Robust defenses for cross-site request forgery. In Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008, pages 75–88, 2008.
7. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning. The MIT Press, 2012.
8. Michael W. Kattan, Dennis A. Adams, and Michael S. Parks. A comparison of machine learning with human judgment. Journal of Management Information Systems, 9(4):37–57, March 1993.
9. D. A. Ferrucci. Introduction to "This is Watson". IBM Journal of Research and Development, 56(3):235–249, May 2012.