



Sri Lanka Institute of Information Technology

IE3042 – Secure Software Systems

Detection of Fake News using Machine Learning

Audit Report

Registration Number	Name
IT20623036	Jayashan H.S.P.C

Introduction

The paper that follows presents the conclusions of a detailed vulnerability study done on the Fake News Detection Web Application. The purpose of this audit report is to identify any security issues and vulnerabilities inside the program, with a focus on ensuring the integrity and trustworthiness of the information supplied.

As part of the continuous fight against the dissemination of false news, the False News Detection Web Application strives to provide customers with access to reliable news sources. To maintain public confidence and faith in the efficacy of this application, its security must be assured, since the prevalence of false news continues to pose a severe danger to the reliability of information on the internet.

To examine the security posture of the Fake News Detection Web Application, we utilized Netsparker, a well-known vulnerability assessment tool known for its extensive scanning capabilities and ability to find potential gaps that may be exploited by hostile actors. Our purpose is to identify and raise attention to any threats, security flaws, or vulnerabilities that may endanger the confidentiality, integrity, or availability of the application.

This report details the vulnerabilities discovered during the assessment, categorizes them according to their severity, and includes detailed explanations of each issue, as well as their potential implications on the application's security. Because it also gives guidance and best practices to address the vulnerabilities discovered, the development team is able to take the required measures to fix and reinforce the application's security posture.

It is critical to remember that this vulnerability assessment does not guarantee the application's security indefinitely, as new weaknesses may emerge over time. Nonetheless, the evaluation provides useful information about the application's current security and serves as a foundation for implementing security measures to combat potential assaults.

By addressing the vulnerabilities revealed in this study, the security of the Fake News Detection Web Application may be reinforced, increasing public faith in its reliability and ensuring the integrity of the information it supplies.

What is owasp

OWASP stands for Open Web Application Security Project. It is a non-profit organization dedicated to improving the security of web apps and software. OWASP provides information, tools, and documentation to help people and organizations understand, detect, and mitigate common security risks and vulnerabilities in web applications.

OWASP's major mission is to make software security clear and accessible to everyone. They advocate for the implementation of industry best practices, safe coding approaches, and understanding of potential security risks and vulnerabilities that may harm online applications. OWASP is a global network of security professionals, programmers, researchers, and organizations committed to online application security. The organization collaborates on a variety of activities and projects aimed at teaching, raising awareness about, and spreading knowledge regarding web application security.

One of OWASP's most important projects is the Top 10 Project, which identifies and stresses the 10 most critical web application security flaws. The OWASP Top Ten can help developers and security professionals identify and prioritize common vulnerabilities.

Owasp top 10

1. Injection:

When untrusted data is regarded as a command or query, injection vulnerabilities take place. This may result in the execution of malicious code, data loss, or unauthorized access to data.

2. Broken Authentication and Session Management:

This category includes issues such as weak authentication mechanisms, improper session handling, and insecure password management, which can result in unauthorized access to user accounts.

3. Cross-Site Scripting (XSS):

XSS flaws let attackers insert harmful scripts onto web pages that other users are seeing. These scripts have the potential to deface websites, hijack user sessions, and steal confidential data.

4. A broken access control:

Allows unauthorized users to access sensitive information or privileged functionality because access constraints are not properly enforced.

5. Security Misconfiguration:

When systems or applications are improperly configured and open to exploitation, security misconfigurations take place. Examples include open directories, unneeded services, and default or weak setups.

6. Cross-Site Request Forgery (CSRF):

These attacks persuade authenticated users to utilize a web application in an unauthorized manner without their knowledge or agreement. This may result in illegal transactions or account breach.

7. Using Third-Party Components with Known Vulnerabilities:

This risk is associated with the usage of out-of-date or susceptible third-party components, such as plugins, frameworks, libraries, or, more specifically, code.

8. Insecure Deserialization:

When untrusted data is deserialized without enough validation, it is insecure deserialization, which can result in remote code execution, denial of service, or other attacks.

9. XML External Entities (XXE):

XXE flaws let attackers to take advantage of XML parsers that handle references to external entities, which can result in the release of private data, a denial of service, or server-side request spoofing.

10. Inadequate recording and Monitoring:

When recording and monitoring are inadequate, it is difficult to identify and address security events. For detecting assaults, looking into accidents, and reducing risks, proper recording and monitoring are crucial.

It's crucial to keep in mind that the OWASP Top Ten list is frequently updated to reflect new threats. For the most latest information and recommendations on web application security, it is advised to consult the most recent edition, which is accessible on the official OWASP website.

Scope

A typical scope of a fake news detection audit report would be to analyze the efficacy and precision of the system or process that a firm has put in place to detect false news. Here are some components that might be addressed by such an audit report.

The technique or algorithm used by the fake news detection system to determine whether news reports are authentic or not would be scrutinized as part of the audit. This would include assessing the methodologies and standards used in the examination of the content, sources, and other relevant factors. The audit would evaluate the informational sources used by the fake news detection system, such as news articles, social media postings, or other textual sources. Examining the dataset's diversity and representativeness, as well as checking the authenticity and legitimacy of the data sources, would be required.

The audit would verify the accuracy of the fake news detection system by analyzing a representative sample of news items or data points. The categorization findings would be compared to a ground truth or independent verification to determine the system's true positive rate, false positive rate, false negative rate, and overall accuracy. The audit will check for any biases or limitations in the fake news detection system. To do so, any biases in the training data, algorithmic biases, or inherent constraints in the system's ability to detect certain types of misleading information would need to be evaluated.

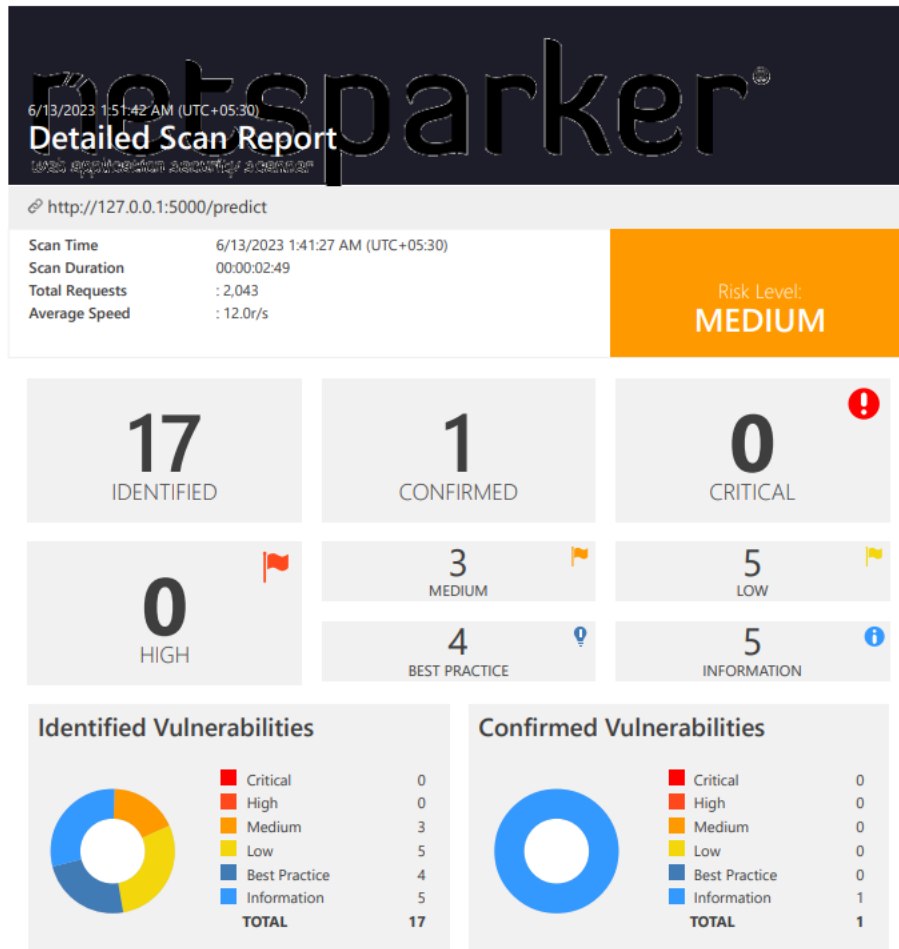
The scope of the audit may include an evaluation of the organization's practices for improving the system for spotting fake news. This would require analyzing the procedures for feedback, monitoring, and improving the system in order to boost the system's efficacy and enable it to address new difficulties.

The audit would determine if the company's procedures for identifying false news adhere to any applicable standards, directives, or best practices in the area, such as those provided by academic studies or trade associations.

In-Scope Domains

Network URL: http://127.0.0.1:5000





























Overall and summarize of the report



Vulnerability summery

There were many vulnerabilities in my product so among the others there was not any critical vulnerability but there were medium and low vulnerabilities.

Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	 Out-of-date Version (Bootstrap)	GET	http://127.0.0.1:5000/static/js/bootstrap.min.js	
	 Out-of-date Version (jQuery)	GET	http://127.0.0.1:5000/predict	
	 SSL/TLS Not Implemented	GET	https://127.0.0.1/predict	
	 [Possible] Cross-site Request Forgery	GET	http://127.0.0.1:5000/predict	
	 [Possible] Phishing by Navigating Browser Tabs	GET	http://127.0.0.1:5000/predict	
	 Missing X-Frame-Options Header	GET	http://127.0.0.1:5000/predict	
	 Version Disclosure (Python)	GET	http://127.0.0.1:5000/predict	
	 Version Disclosure (Werkzeug Python WSGI Library)	GET	http://127.0.0.1:5000/predict	
	 Content Security Policy (CSP) Not Implemented	GET	http://127.0.0.1:5000/predict	
	 Missing X-XSS-Protection Header	GET	http://127.0.0.1:5000/predict	
	 Referrer-Policy Not Implemented	GET	http://127.0.0.1:5000/predict	
	 Subresource Integrity (SRI) Not Implemented	GET	http://127.0.0.1:5000/predict	
	 Expect-CT Security Header Errors and Warnings	GET	http://127.0.0.1:5000/predict	
	 Out-of-date Version (Modernizr)	GET	http://127.0.0.1:5000/predict	
	 Python Identified	GET	http://127.0.0.1:5000/predict	
	 Werkzeug Python WSGI Library Identified	GET	http://127.0.0.1:5000/predict	

2/62

These are the vulnerabilities in medium level in my product.

I found medium vulnerability as Out-of-date version (Bootstrap).

1. Out-of-date Version (Bootstrap)

MEDIUM ⓘ 1

Netsparker identified that the target web site is using Bootstrap and detected that it is out of date.

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap before 4.1.2, XSS is possible in the data-container property of tooltip.

Affected Versions

1.0.0 to 3.3.7

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

External References

- [CVE-2018-14042](#)

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap 3.x before 3.4.0 and 4.x-beta before 4.0.0-beta.2, XSS is possible in the data-target attribute, a different vulnerability than CVE-2018-14041.

Affected Versions

3.0.0 to 3.3.7

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

External References

- [CVE-2016-10735](#)

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap before 3.4.0, XSS is possible in the tooltip data-viewport attribute.

Affected Versions

1.0.0 to 3.3.7

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

External References

- [CVE-2018-20676](#)

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap before 3.4.0, XSS is possible in the affix configuration target property.

Another out-of-date version vulnerability in (jQuery)

2. Out-of-date Version (jQuery)

MEDIUM  1

Netsparker identified the target web site is using jQuery and detected that it is out of date.

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

jQuery Improper Neutralization of Input During Web Page Generation ("Cross-site Scripting") Vulnerability

jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.

Affected Versions

1.8.0 to 2.2.4

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/E:L/A:N

External References

- [CVE-2015-9251](#)

jQuery Improper Neutralization of Input During Web Page Generation ("Cross-site Scripting") Vulnerability

In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

Affected Versions

1.9.0 to 3.4.1

CVSS

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/E:L/A:N

External References

- [CVE-2020-11023](#)

jQuery Improper Neutralization of Input During Web Page Generation ("Cross-site Scripting") Vulnerability

In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

Affected Versions

1.9.0 to 3.4.1

CVSS

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/E:L/A:N

External References

- [CVE-2020-11022](#)

JQuery Prototype Pollution Vulnerability

SSL/TLS Not Implemented

3. SSL/TLS Not Implemented

MEDIUM  1

Netsparker detected that SSL/TLS is not implemented.

Impact

An attacker who is able to intercept your - or your users' - network traffic can read and modify any messages that are exchanged with your server.

That means that an attacker can see passwords in clear text, modify the appearance of your website, redirect the user to other web pages or steal session information.

Therefore no message you send to the server remains confidential.

Vulnerabilities

3.1. <https://127.0.0.1/predict>

Certainty

These are the vulnerabilities in low level in my product.

Cross-site Request Forgery

4. [Possible] Cross-site Request Forgery

LOW  1

Netsparker identified a possible Cross-Site Request Forgery.

CSRF is a very common vulnerability. It's an attack which forces a user to execute unwanted actions on a web application in which the user is currently authenticated.

Impact

Depending on the application, an attacker can mount any of the actions that can be done by the user such as adding a user, modifying content, deleting data. All the functionality that's available to the victim can be used by the attacker. Only exception to this rule is a page that requires extra information that only the legitimate user can know (such as user's password).

Vulnerabilities

4.1. <http://127.0.0.1:5000/predict>

Form Action(s)

- [/predict](#)

Certainty

Request

```
GET /predict HTTP/1.1
Host: 127.0.0.1:5000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
```

Phishing by Navigating Browser Tabs

5. [Possible] Phishing by Navigating Browser Tabs

LOW  1

Netsparker identified possible phishing by navigating browser tabs but was unable to confirm the vulnerability.

Open windows with normal hrefs with the tag `target="_blank"` can modify `window.opener.location` and replace the parent webpage with something else, even on a different origin.

Impact

While this vulnerability doesn't allow script execution, it does allow phishing attacks that silently replace the parent tab. If the links lack `rel="noopener noreferrer"` attribute, a third party site can change the URL of the source tab using `window.opener.location.assign` and trick the users into thinking that they're still in a trusted page and lead them to enter their sensitive data on the malicious website.

Vulnerabilities

5.1. <http://127.0.0.1:5000/predict>

External Links

- <https://colorlib.com>
- <http://unsplash.co/>
- <http://pexels.com/>

Certainty

Missing X-Frame-Options Header

6. Missing X-Frame-Options Header

LOW  1

Netsparker detected a missing `X-Frame-Options` header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options` HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a frame or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities

6.1. <http://127.0.0.1:5000/predict>

Certainty

Version Disclosure (Python)

7. Version Disclosure (Python)

LOW  1

Netsparker identified a version disclosure (Python) in the target web server's HTTP response.

This information can help an attacker gain a greater understanding of the systems in use and potentially develop further attacks targeted at the specific version of Python.

Impact

An attacker might use the disclosed information to harvest specific security vulnerabilities for the version identified.

Vulnerabilities

7.1. <http://127.0.0.1:5000/predict>

Extracted Version

- 3.11.3

Certainty



Version Disclosure (Werkzeug Python WSGI Library)

8. Version Disclosure (Werkzeug Python WSGI Library)

LOW  1

Netsparker identified a version disclosure (Werkzeug) in the target web server's HTTP response.

This information can help an attacker gain a greater understanding of the systems in use and potentially develop further attacks targeted at the specific version of Werkzeug.

Impact

An attacker might use the disclosed information to harvest specific security vulnerabilities for the version identified.

Vulnerabilities

8.1. <http://127.0.0.1:5000/predict>

Extracted Version

- 2.3.4

Certainty



Conclusion

This audit study highlights a number of serious flaws in the fake news detection system, highlighting possible dangers and holes in its security framework. The system's integrity and capacity to precisely identify and reduce false news information are seriously threatened by the discovered flaws. The creators and managers of the false news detecting system must immediately remedy these flaws. Failure to do so can leave the system vulnerable to attack by bad actors, resulting in the spread of false information or undermining the system's efficacy. The false news detection system may improve its security posture, guarantee the accuracy of its findings, and make a more dependable and trustworthy contribution to the battle against misinformation by addressing these weaknesses and putting the suggested steps into practice.