ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 9, Issue 9, September 2022

CVEdB – Common Vulnerabilities and Exploits Database

Kaustubh Rai[1], Fatema Vajhee[2], Divya Shinday[3]

U.G. Student[1,2,3]

Department of Computer Engineering, St. John College of Engineering and Management, Palghar, India

raikaustubh12@gmail.com[1] ,fatmav@sjcem.edu.in[2] , diyash@sjcem.edu.in[3]

Vidya Kawtikwar[4]

Assistant Professor[4]

Department of Computer Engineering, St. John College of Engineering and Management, Palghar, India

vidykw@sjcem.edu.in[4]

Abstract: Digital weaknesses have represented a risk to all nations, and their security has been absolute stress for the whole world for a long time. Despite advancements in IT security, the data security field is at this point in its adolescence. Open-source network security data sources like the National Vulnerability Database (NVD), CERT, and seller sites provide a quantitative exploration approach to give focusing on bits of knowledge that can help the futuristic investigation. These public vaults consider the normalization and exchange of exceptional weak data, with the goal of increasing online security awareness. The dataset was standardized to eliminate excess information and accurately orchestrate the information properties in order to differentiate weakness measurements. In this paper, we led a similar report on the current weakness data sets administrations presented for abuse and weaknesses text.

Keywords: Cybersecurity, Exploits, Vulnerability, CVE

I. INTRODUCTION

The CVEdB - Common Vulnerabilities and Exploits Database – is an application that provides information on weaknesses, exploits them, and provides additional dangerous insight data. This application recognizes flaws, exploits and provides additional risk insight data such as current risk levels, exploits costs, and provides an answer for how to avoid making the plan flaw while fostering a product and how to moderate the CVE. The phrase "plan blemishes" implies that it splits or breaks an application or gadget that was not planned safely and can thus be exploited by digital assailants for malicious purposes. This application is intended for security professionals, security enthusiasts, and individuals interested in hacking and penetration testing. The application contains no malicious code that can be used to hack or harm any device or application. The application has no passwords, keygens, or infections saved. This is not a fun-loving application like an arcade game, but it is presented as instructional material to make people aware of the dangers of shaky programming plans and how to avoid them by adhering to secure programming advancement practises.

The application provides countermeasures for how to avoid making the plan compromised while developing a product, as well as how to resolve the CVE. The flaws in any product are not planned safely all along, which is true for most configuration flaws. In this application, you will learn about untrustworthy programming configuration flaws and solutions for making your product vastly secure by adhering to secure programming improvement practices. CVE was founded in 1999, at a time when many computer security programs had their own vulnerability databases and labels. Because the available services varied so widely, it was difficult to tell whether multiple databases were addressing the same issue. This resulted in security vulnerabilities, making it difficult to develop a solid system for interoperability across multiple databases and technologies.

To tackle these issues, the CVE was created to enable universal identification. As a result, it addresses the root causes of the problems and enabled IT experts to exchange information regarding vulnerabilities, allowing them to

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 9, Issue 9, September 2022

collaborate to recognize and fix issues. As a result, it has become the mainstream technology for exploiting vulnerabilities and exposures, and it is supported by the CVE Numbering Authority, the CVE Board, and a wide range of industry-leading products and services. The MITRE Corporation, a non-profit corporation that oversees federally sponsored research and development institutes that serve U.S. government organizations, maintains the CVE. MITRE is in charge of maintaining the CVE vocabulary and the open site. The Department of Homeland Security's Cybersecurity and Infrastructure Agency is funding this initiative.

II. Related Work

Cybersecurity plays an important role in the survival of any organization. In 2021 Yuning Jiang, Manfred A. Jeusfeld, Jianguo Ding [1] conducted a case study on vulnerability analysis of the IT and OT systems of a company in Sweden. Briefly introduced the systems under investigation, followed by empirical analysis of the main causes and impact of data inconsistencies in the vulnerability analysis result of their investigated systems. Focused on cross-record data inconsistency, especially between NVD and vendor advisories. This paper evaluates the data inconsistencies of open-source vulnerability repositories in vulnerability assessment, particularly from the perspective of cybersecurity awareness enhancement.

Jay Jacobs, Sasha Romanosky, Idris Adjerid, and Wade Baker [2] in the 2020 study from Virginia Tech's relies on a corpus of work on the economics of information security. Their finding is significant to a line of research examining the efficacy of vulnerability repair, which is a critical role in how businesses proactively defend against security events. This line of research includes assessments of the correctness of techniques for identifying potential security flaws as well as distributing information to the stakeholders, the design of incentive programs and obligations in relation to known vulnerabilities restoration, and the efficiency of methodologies that firms employ when attempting to prioritize the vulnerabilities they address. The status quo for how businesses manage vulnerability repair is frequently unsatisfactory and has a great opportunity for change, according to results in this stream of study.

Zhang, Su, Doina Caragea, and Xinming Ou in their 2011 paper concluded that it is difficult to create good prediction models. Most Microsoft instances, for example, do not have any version information (especially, Windows instances). Some of the results appear interesting (for example, the Firefox models), but they are not practical. This paper analyzed the information quality of vulnerability databases utilizing state-of-the-art machine learning (ML) and natural language processing (NLP) techniques, searching the free-form description for relevant information missing from structured fields, and updating it accordingly. In 2020, Asra Kalim, C K Jha, Deepak Singh Tomar, and Divya Rishi Sahu [4] suggested a framework for detecting taint attacks. To discover security flaws, it uses a variety of scanning methods, such as taint type, ontology-based, and so on. The record acquired during the data flow analysis step is used to update its instructional database. It can detect online vulnerabilities such as cross-site scripting (XSS) and SQL injection attacks (SQLIAs), as well as frame-jacking and zero-day vulnerabilities. The proposed architecture allows for a more in-depth analysis of the attacker's actions and intentions on the web application. It also makes it simple for security professionals and developers to update the detection database as needed. It creates a thorough report that includes a detailed description of each probable susceptible function that constitutes a web application security risk.

III. Proposed Methodology

3.1 Overview

The Proposed System works as follows; the user goes to the web application. After opening the application, it is introduced with the user-friendly GUI, This GUI consists of all the modules present in the Application. The Modules are CVE-ID, CVSS Score, Exploit link, Exploit Code, Exploit Price Section, Countermeasure, and CAPEC Solutions. The user can access these modules with the help of the interactive GUI.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 9, Issue 9, September 2022

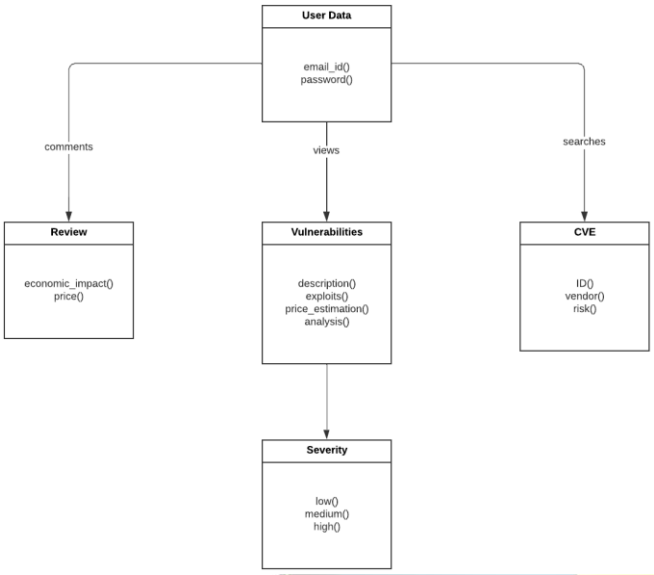

Fig . 1.Class Diagram for CVEdB

The following are the main modules in the GUI:

CVE Title:

The CVE Title is the vulnerability's name, as determined by NVD Data. This title remembers the date for which the information was last produced. The ID component utilizes a CVE-style "timestamp" for the date and time on which the information was last produced, of the structure "YYYYMMDD-hhmmss", with driving zeroes as needs are. This component can be utilized with straightforward string-correlation schedules to naturally decide whether one download document is later than one more one. Since CVRF 1.1 doesn't permit driving zeros in the Version component, the month and day can here and there be a solitary digit.

CVSS Scores:

CVSS Scoring System, involving the most recent 3.1 variants for ascertaining. We are showing the Base score and an easy-to-use graph for the client to comprehend the CVSS Attack Vector. The base measurement bunch addresses the inherent attributes of a weakness that are steady over the long run and across client conditions. It is made out of two arrangements of measurements: the Exploitability measurements and the Impact measurements. The Exploitability measurements mirror the simplicity by which the weakness can be taken advantage of. That is, they address qualities of what is defenseless, which we allude to officially as the weak part. It comprises Attack Vector, Attack Complexity, Privileges Required, User Interaction, and Scope.

A. Attack Vector

This measurement mirrors the setting by which weakness abuse is conceivable. This measurement of esteem (and subsequently the Base Score) will be bigger the more remote (consistently, and truly) an aggressor can be to take advantage of the weak part.

B. Attack Complexity

The image attributes extracted from the segmentations consist of noise, therefore morphological filtering removes the noise and gives a smooth contour. The output of these filterings is image attributes that are useful for feature extraction and sign recognition. Dilation and Erosion are used for morphological filtering.

C. Privileges Required

This measurement portrays the circumstances unchangeable as far as the aggressor might be concerned that should exist to take advantage of the weakness. It has 2 boundaries, low and high.

D. User Interaction

This measurement catches the prerequisite for a human client, other than the assailant, to partake in the effective split of the difference of the weak part.

E. Scope

The Scope metric catches whether weakness in one weak part impacts assets in parts past its security scope.

The Impact measurements catch the impacts of an effectively taken advantage of weakness on the part that experiences the most terrible result that is most straightforwardly and typically connected with the assault. Examiners ought to compel effects on a sensible, ultimate result that they are certain an aggressor can accomplish. It comprises Confidentiality, Integrity, and Availability.

Exploit code:

An exploit is a piece of code, a lump of information, or a succession of orders that exploits a product weakness or security flaw in an application or a framework to make the unforeseen way of behaving happen. The exploit code for straightforwardness, rather than visiting the Exploit Link and replicating the code from that point, we duplicate the code here.

Exploit Price:

Whenever the CVE was first presented, the cost for taking advantage of this CVE was. A 0-day assault cost and the ongoing cost for taking advantage of this CVE. It permits the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 9, Issue 9, September 2022

expectation of conventional costs by thinking about numerous specialized parts of the impacted weakness. The more specialized subtleties are accessible the higher the exactness of the reproducible estimate. [3] proposed a secure hash message authentication code. A secure hash message authentication code to avoid certificate revocation list checking is proposed for vehicular ad hoc networks (VANETs). The group signature scheme is widely used in VANETs for secure communication, the existing systems based on group signature scheme provides verification delay in certificate revocation list checking. In order to overcome this delay this paper uses a Hash message authentication code (HMAC)

Counter Measure:

A countermeasure to patch the vulnerability affecting the organization.

CAPEC Solution:

The Common Attack Pattern Enumeration and Classification (CAPEC) drive is supported by the Department of Homeland Security. The objective of this work is to create and send to the public an underlying benchmark list of assault designs alongside a thorough blueprint and characterization taxonomy. This inventory will keep on framing the standard system for recognizing, gathering, refining, and sharing assault designs among the product's local area. It provides a solution on how to not make the mistake that leads to this CVE.

Vulnerable Products - Various software products that are affected by the CVE.

Feedback Section – A comment section where the user can discuss anything related to the vulnerability they can use the comment section by simply signing in in order to confirm user identity

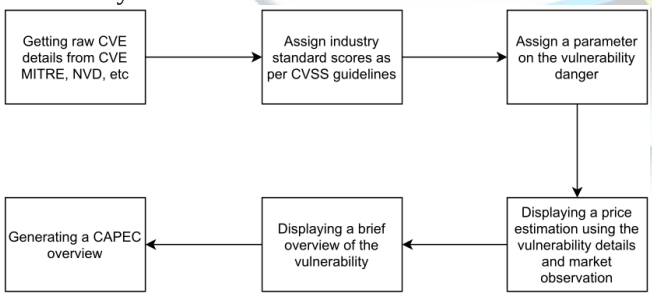

Fig. 2. Block diagram for the proposed system

The framework will get crude CVE subtleties and scratch exploit reports with information cleaning and included designing over the extricated information. Then, at that point, relegate a score and boundary according to industry-standard CVSS rules on the weakness risk. After that carry out a cost assessment of the adventure and a brief outline of the particular weakness to resolve it. Finally, we will show a CAPEC outline of the weakness.

IV. IMPLEMENTATION

CVE datasets from NIST's NVD, CVE MITRE, OpenCV, and VULDB API will assist in obtaining the subtleties of weaknesses from various sources. The brief outline will aid in quickly understanding the flaw. The adventure data for the aforementioned weakness will be scraped from the EXPLOIT-DB and CXSecurity websites. A CVSS (Common Vulnerability ScoringSystem) from various sources and accumulated Meta scores will be assigned, and the boundaries of the weakness will range from low to moderate to high. A cost assessment for the weakness, including the multi-day and ongoing costs. Forecasting conventional costs by taking into account numerous specialized parts of the impacted weakness. The more specialized subtleties available, the greater the precision. The conventionally determined costs are compared to information found on the Internet, including articles, news reports, weakness specialist value records, gatherings, and darknet conversations. The value assessment is primarily a pattern marker that highlights common cost ranges. MITRE dataset for the CAPEC outline. A database of attack designs for analysts, understudies, and engineers to avoid making the same common programming configuration mistake that could prompt aggressors to effectively track down weaknesses in the product. Clients can make a record that makes it conceivable to alter and audit sections for example, whether the value assessment is off-base, or the endeavor for the weakness is wrong, and so forth. Numerous weakness analysts and heads could utilize these elements to submit alters of existing sections or propose new entries to be added to the data set. [5] discussed that the activity related status data will be communicated consistently and shared among drivers through VANETs keeping in mind the end goal to enhance driving security and solace. Along these lines, Vehicular specially appointed systems (VANETs) require safeguarding and secure information correspondences. Without the security and protection ensures, the aggressors could track their intrigued vehicles by gathering and breaking down their movement messages. A mysterious message confirmation is a basic prerequisite of VANETs

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 9, Issue 9, September 2022

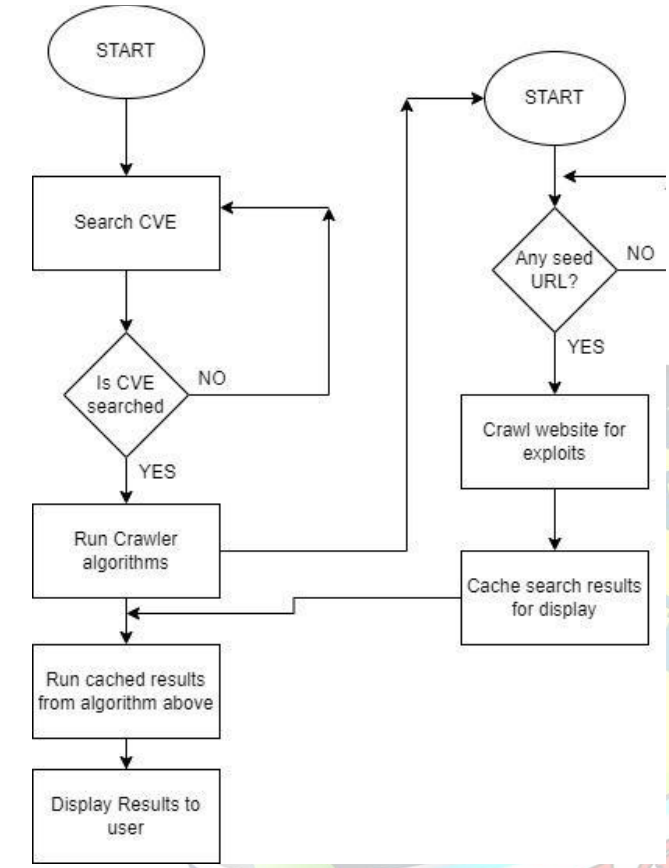

Fig. 3. Flow Chart of the proposed system

4.1 ALGORITHM

A. **Context Processing Algorithm**: When handling a JSON-LD information structure, each handling rule is applied utilizing data given by the dynamic setting. This segment portrays how to create a functioning set. The dynamic setting contains the dynamic term definitions which determine how properties and values must be deciphered as well as the current base IRI, the jargon planning and the default language. Each term definition comprises an IRI planning, a boolean banner converse property, a discretionary kind planning or language mapping, a discretionary setting, discretionary next esteem, a discretionary prefix banner, and a discretionary holder planning.

B. **Expansion Algorithm**: This calculation extends a JSON-LD archive, to such an extent that all setting definitions are eliminated, all terms and minimal IRIs are extended to outright IRIs, clear hub identifiers, or catchphrases, and all JSON-LD values are communicated in clusters in an extended structure.

C. **Compaction Algorithm**: This calculation compacts a JSON-LD report, to such an extent that the given setting is applied. This should bring about shortening any pertinent IRIs to terms or conservative IRIs, any appropriate catchphrases to watchword false names, and any pertinent JSON-LD values communicated in extended structure to basic qualities like strings or numbers.

4.2 OUTPUT

After applying the above-mentioned algorithms, the expected results are obtained. The snapshots attached below give a clear idea of the results

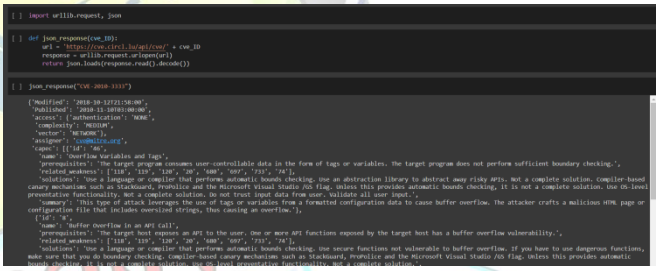

Fig. 4. Importing the dataset

Data is sorted to obtain accurate results

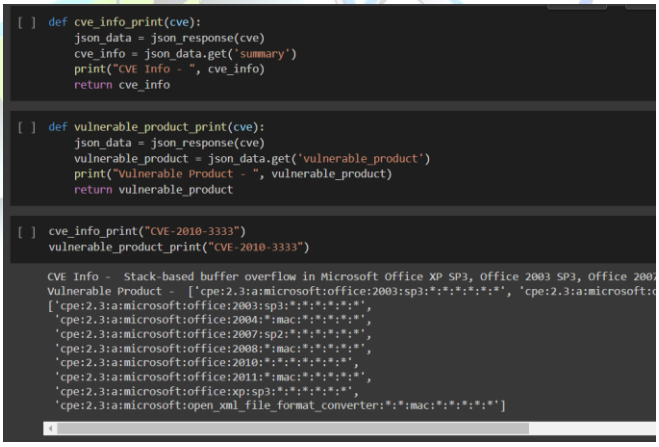

Fig. 5. Sorting the dataset

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 9, Issue 9, September 2022

Getting the exploit prices for the vulnerability

```
from urllib.parse import urlencode
from urllib.request import Request, urlopen
import json

def vulrul(cve_ID):
    cve_id = cve_ID
    url = 'https://vuldb.com/?api'
    post_fields = { 'apikey': 'd31c748851065a1e5168367c029ac527', 'search': cve_id, 'details':'1' }
    request = Request(url, urlencode(post_fields).encode())
    data = json.loads(urlopen(request).read().decode())
    json_result = data['result']
    json_response = data['response']
    prices = json_result[0]['exploit']['price']
    # attack_vector = json_result[0]['vulnerability']['cvss3']['vuldb']['basevector']
    # base_score = json_result[0]['vulnerability']['cvss3']['meta']['basescore']
    # new_attack_vector = attack_vector.replace("\\", "")
    # summary = json_result[0]['entry']['summary']
    # countermeasure = json_result[0]['entry']['details']['countermeasure']
    # remaining = json_response['remaining']
    vul_list = [prices]
    print(vul_list)
    return vul_list

vulrul("CVE-2010-3333")

[{'0day': '$25k-$100k', 'today': '$0-$5k'}]
[{'0day': '$25k-$100k', 'today': '$0-$5k'}]
```

Fig. 6. Getting the exploit prices for any CVE

After getting the CVSS scores, the attack vector parameters are implemented into a user-friendly chart

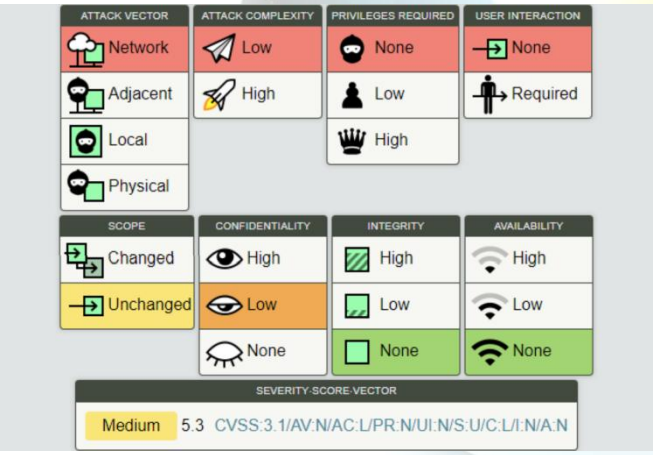

Fig. 7. Parameters for the vulnerability CVE-2020-3452 converted into a chart

V. RESULTS

Fig. 8. Homepage of the proposed system

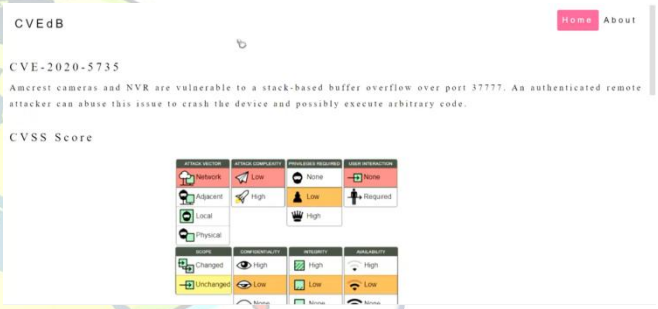

Fig. 9. Details of the CVE—The details of that vulnerability are displayed like a detailed description of the vulnerability

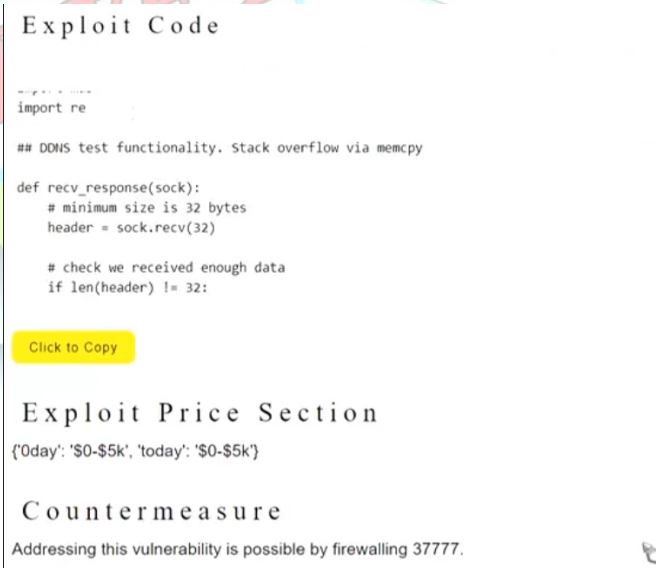

Fig. 10. Exploit Price Section and Countermeasure – It shows the 0day price and today's price of searched

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 9, Issue 9, September 2022

vulnerability. The countermeasures are also provided in order to avoid future causes.

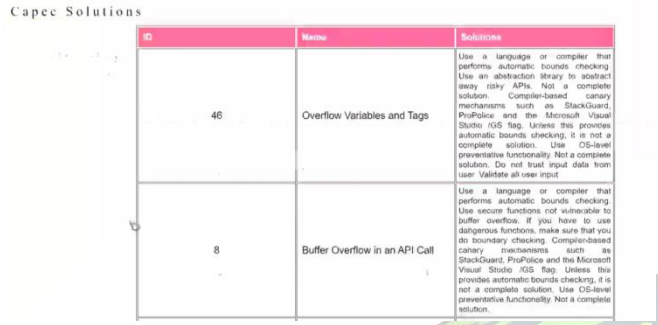

Fig. 11. CAPEC Solution—The System also provides numerous CAPEC solutions for a searched vulnerability

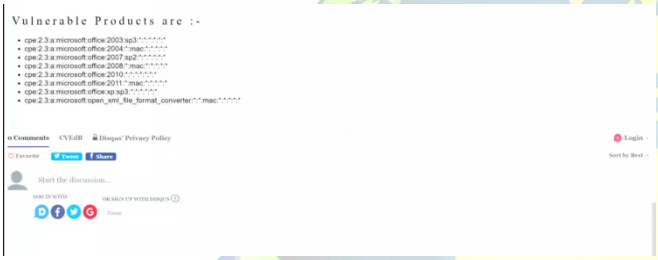

Fig. 12. Discussion Section – A simple comment section for user feedback

VI. Conclusion

We investigated a couple of papers and arrived at a resolution that there is no vulnerability database that is perceived by all ages and makes sense of everything in a nitty-gritty and basic way. We then, at that point, deliberately gather the dataset from different sources and compose a rationale to separate information for the endeavors and weaknesses that the client needs. To additional upgrade the public network safety information sources, we intend to investigate unwavering quality approval computerization. Weakness archives of believed merchants could be fused into a cross-connected neighborhood weakness information base to naturally identify irregularities.

VII. Future Work

We intend to convey the Cyber Intimidation Intelligence Information more exact and point by point. A Cyber Threat Intelligence to do drill-down of IP addresses as IOC (Indicators of Compromise). A custom investigation outline made into a playbook for organizations where we give Tactics, Techniques, Procedures, and Indicators of Attacks too, and finally, an AI bot to examine insights regarding weaknesses, exploits, and danger knowledge.

References

1. Yuning Jiang, Manfred A. Jeusfeld, Jianguo Ding, "Evaluating the Data Inconsistency of Open-Source Vulnerability Repositories," 16th International Conference on Availability, Reliability and Security, Association for Computing Machinery, 2021, pp. 1-10, DOI: 10.1145/3465481.3470093.
2. Jay Jacobs, Sasha Romanosky, Idris Adjerid and Wade Baker, "Improving vulnerability remediation through better exploit prediction", Journal of Cybersecurity, Volume 6, Issue 1, (2020), DOI: 10.1093/cybsec/tyaa015
3. Christo Ananth, M.Danya Priyadharshini, "A Secure Hash Message Authentication Code to avoid Certificate Revocation list Checking in Vehicular Adhoc networks", International Journal of Applied Engineering Research (IJAER), Volume 10, Special Issue 2, 2015,(1250-1254).
4. Kalim, Asra, C. K. Jha, Deepak Singh Tomar, and Divya Rishi Sahu, "Novel Detection Technique For Framejacking Vulnerabilities In Web Applications.," 2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM), pp. 265-270. DOI: 10.35940/ijeat.C4778.029320.
5. Christo Ananth, Dr.S. Selvakani, K. Vasumathi, "An Efficient Privacy Preservation in Vehicular Communications Using EC-Based Chameleon Hashing", Journal of Advanced Research in Dynamical and Control Systems, 15-Special Issue, December 2017,pp: 787-792.
6. Williams, Mark A., Sumi Dey, Roberto Camacho Barranco, Sheikh Motahar Naim, M. Shahriar Hossain, and Monika Akbar. "Analyzing evolving trends of vulnerabilities in national vulnerability database.", IEEE International Conference on Big Data (Big Data), pp. 3011-3020. IEEE, 2018, DOI: 10.1109/BigData.2018.8622299