# Security of Web Applications: Threats, Vulnerabilities, and Protection Methods

**Asma Mohammed[1] , Jamilah Alkhathami[2] ,Hatim Alsuwat[3], Emad Alsuwat[4]**
Lele-so@hotmail.com j.ojo2010@hotmail.com Hssuwat@uqu.edu.sa Alsuwat@tu.edu.sa
Department of Computer Science, College of Computers and Information Technology, Taif University[1]
Department of Computer Science, College of Computers and Information Technology, Taif University[2]
Department of Computer Science, College of Computer and Information Systems, Umm Al Qura University[3]
Department of Computer Science, College Of Computers and Information Technology, Taif University[4]

**Summary**
This is the world of computer science and innovations. In this modern era, every day new apps, webs and software are being introduced. As well as new apps and software are being introduced, similarly threats and vulnerable security matters are also increasing. Web apps are software that can be used by customers for numerous useful tasks, and because of the developer experience of good programming standards, web applications that can be used by an attacker also have multiple sides. Web applications Security is expected to protect the content of critical web and to ensure secure data transmission. Application safety must therefore be enforced across all infrastructure, including the web application itself, that supports the web applications. Many *organizations* currently have a type of web application protection scheme or attempt to build/develop, but the bulk of these schemes are incapable of generating value consistently and effectively, and therefore do not improve developers' attitude in building/designing stable Web applications. This article aims to analyze the attacks on the website and address security scanners of web applications to help us resolve web *application* security challenges.
***Keywords:*** *web applications, threats, attacks, security*

## 1. Introduction

Every year, more and more Web-based apps, while figures do not exist on the number of web applications existing worldwide, by the first quarter of 2020, there were approximately 367 million domain names; Each of these areas may be considered as a static or dynamic web application [1].
Such *applications* may exchange and process confidential data of the web users, Therefore, these web applications attract malicious attackers. Web applications are becoming an essential part of our lives, economically and privately. Unfortunately, over 90% of these systems are insecure and there are a total of 13 bugs per program. For web apps, security, therefore, plays an important role [2].

Various current web apps are totally useful platform structures that send clients and businesses and different administrations to different clients. For instance, enterprises used web-based software to strengthen and increase their operations to manufacture, save capital, instruction, and government. The Web also calls for the enhancement of corporate website intranet
applications under each association's boundaries. The excellent deployment of Web apps across correspondence and enterprise regions make it one of the most relevant and vital fields of the productive sector [3].

In the last ten or so months, millions of businesses have adopted the Internet as a cheap medium for contact and knowledge sharing with prospective clients and consumer purchases. In particular, the web gives advertisers a way to get to know and connect with those who visit their websites. One way to do this is to ask online users to subscribe to emails, to send a request form for product information, or to give their browsing experience personalized when they next visit a particular website [4].

Online technologies in modern times play an important role in automating conventional everyday tasks by updating of existing solutions. The Internet and several web *service* providers are used by more than 3.88 billion users worldwide because they can be freely used and readily accessible everywhere [5].

IT security infringements *have caused* major problems for customers, states, businesses, and companies over recent years. Recent daily loss of information and the theft of millions of dollars by multiple cyber threats are typical views. Although sufficient research on cyber and web vulnerability has been conducted. However, we must now

Manuscript received August 5, 2021
Manuscript revised August 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.8.22

consider new ways of reducing risks, malware, and cybercriminals, and so on harm [6].

In the second section of this paper, we present the background information that shows the importance of web applications and their spread around us, which led to many security threats and attacks. We also present the principles of the CIA that must be introduced to guarantee the privacy and security of web applications. We discuss security threats on web applications in Section III. In section IV, we show the security recommendations to mitigate such threats. We then present our discussion I Section V. In section, we conclude and state the contribution of our paper.

## 2. Background

### 2.1 Web Applications Security

Security tests for web applications are a method for validating safe application execution, security testing aims to identify faults, keep them away from the final product and ensure that *application* security is appropriate [7].
The highly dynamic web framework extends its attack surface to target a wide variety of faults. Web apps are not *anomalous*, provided that current HTML5 specifications and the growing functionality of JavaScript are used to create rich web applications, also subsuming the need for conventional desktop applications. A potential way of addressing this increased difficulty is to debug devices**.** While developing on operating systems, libraries and compiled programs has been successfully implemented, its applicability in web applications has still not been investigated [8].

The CIA Principals must be applied as standards that every individual and organization must follow and adopt to ensure the security in web applications. Figure 1 illustrates these Principals.

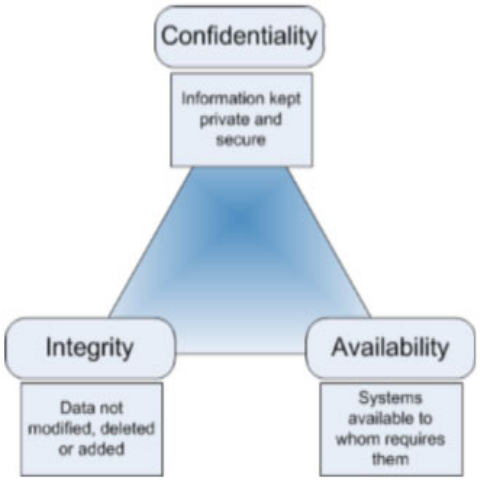


**Fig 1**. CIA Principals [10]

A. Confidentiality

ensures that only individuals who have the authority to view or read information on a website. To foresee the simple hands of foreigners, designers will forecast this by means of designing
positions for all apps, for starters, rather than the administrator, the website, theme or the plugin cannot be altered; thus, a separate user than the administrator cannot adjust the theme. By exchanging roles such as these data or user list records, the manager only would be visible.

Developers can predict this by designing positions from any consumer they have to anticipate ignorant hands of outsiders, The website theme and/or plugin, for example, cannot be modified
other than the administrator. Therefore, a user other than the administrator cannot modify the subject. The only way to see this
admin is to exchange position such as these details or details such as user lists

B. Integrity

recognizes that only those who have permission to do so can modify or erase the data inside a server or a website, this means that the method of moving the file from a server to a computer, or vice versa (can be uploaded or downloaded), is not sure of website use. Similarly, whether a file will change a virus attack, if the name or contents are changed. Often a user with a position that is less than the manager can, even if he does not have access to the changes or destruction of records, do so (in certain cases, even by accident). Perhaps this action is an action that the user does not like, but is nevertheless committed because of a web application bug. This must, of course, be avoided to make a website safer. One approach is to use the procedure, namely the testing system, which needs to be present in a

software engineering process. There are two aspects of the testing procedure: - Inspection of the Black Box - Testing the white box Simply put, the review of the black box is a trial of the software for people who immediately use the website (act as a user). White box testing is the testing process that aims to test functions written in some programming languages such as PHP. Such white box testing process are split into three tests depending on input, function of the program, or input values.

C. Availability

ensures that if the user wishes to use the site, it must be available. If you can visit a website without mistakes, this means that the website satisfies this concept of compatibility. If possible, a website should be open 24 hours 7 weeks (24/7). This ensures, that it needs to be open. If secrecy ensures that such data stored on a computer or database can be accessed only by registered users, functionality implies that the web application is accessible whenever the user wants to access it. It can sound contradictory and not unlike the first theory, but these two rules are somewhat distinct since two views are presented. The usability of a website is emphasized.

Concept for real-time feedback A real-time method based logically on the facts and the timeliness of the effects the reality and timeliness of the performance results. Using such a device, devices such as engines, lines, telescopes, or all other instruments may be monitored and controlled. Real-time power, typically often needed for telecommunications equipment and computer systems, in developing a Web Security Analyzer framework, the idea of real-time reviews would be extended to search the site and deliver outcomes (securitization hole) with suggestions that are able to address the weaknesses.

D. *Algorithm*

This paper provides a method that is able to conduct an computerized scanning procedure, in which data from a number of resource shares are retrieved from the system to match the security whole process on a site by using the model matching principle used in the Micro service Architecture. The following formulation is a website safety deficiency scan scheme

F. Moodle, an open-source framework that is publicly available, can be used and updated by anybody who has a license for the GNU (General Public License) program. At the address: http://www.moodle.org, users can download Moodle. Moodle encourages student-centered learning that encourages the eLearning process, a.k.a. distance learning. This model helps students to not only retrieve the content but also participate in the learning process. Furthermore, instructors can share the teaching material at any time with students, without limiting themselves by the distance and space factors. The instructor may upload information from other websites in the form of sentences, presentations, audio, videos, and links [10].

## 3. Security Threats on Web Applications

Web applications are exposed to many malicious attacks that negatively affect the work of these applications. In this section of our research, we review the most dangerous attacks and vulnerabilities that web applications can be exposed to:

**Table 1:** Attacks and Vulnerabilities in web applications.

| *Ref* | *Attack* | *year* | *Approaches* |
|---|---|---|---|
| [11] | **Cross Site Scripting (XSS)**. | 2021 | The **web-based** attack is often known as the Cross Site Scripting (XSS). This takes place when malicious web code is submitted or run from the computer browser of the victim using their Web apps, usually in script form. This execution may filter personal data or stolen cookies from the user to retrieve the identity in a fraudulent session. It also helps attackers to steal or even take possession of other devices with sensitive info. |
| [12] | **SQL Injection.** | 2018 | Is a security flaw, which leads to a SQL query being passed to a back-end database by an attacker. The syntax and capability of SQL itself and the power and flexibility the data database and features provided can be exploited by an attacker to manipulate what is transmitted to the database. |
| [13] | **Remote File Inclusion (RFI)** | 2020 | Is a web server script attack That enables an attacker to add a file remotely. This attack can allow data theft or malicious code execution, such as JavaScript, to lead to other attacks on the client- side. Due to the user provided input without sufficient confirmation, this limitation exists. |
| [14] | **Denial of Service (DoS).** | 2020 | A *cyber-attack* where the attacker disrupting the services of a host that is connected to the Internet briefly or permanently renders a computer or network infrastructure not accessible to intended users. To overwhelm networks to avoid certain or all valid demands from being met, a service is usually refused by overwhelming the target |

| | | | |
|---|---|---|---|
| | | | equipment with unwanted requests. |
| [15] | **Buffer Overflow**. | 2019 | Buffer Overflow happens when an attacker tries to transfer more data deliberately than can be handled by a program or procedure Such an assault can lead to a crashing system. |
| [16] | **Cross-Site Request Forgery (CSRF)**. | 2021 | is a well-known web-based *attacks* that induces a user to send unsound, attack-controlled HTTP requests to a currently vulnerable web application. CSRF's basic principle is that malicious requests are addressed by the user's browser to the Web Application, and cannot, therefore, be differentiated from the expected user-authorized benign requests. |
| [17] | **Unrestricted File Upload (UFU).** | 2020 | Is a vulnerability That exploited errors on server-side web application content filtering tests. A competitor, who is considered an upload attacker, takes advantage of his restricted right to upload malformed files by taking advantage of the UFU. If a forced file is successfully submitted, a code execution risk may occur. |
| [18] | **SYN flood DDoS attack.** | 2020 | TCP SYN flood (a.k.a. flood) is a type of DDoS attack that exploits part of the traditional 3-way TCP handshake, to absorb resources and make the server unauthorized. The attacker mostly sends TCP link requests faster with SYN flood DDoS as a result of network saturation. |
| [19] | **Broken Authentication and Session Management.** | 2018 | Is a form of web vulnerability due to session management misconfiguration? A session will be set up after an authentication procedure is completed that enables data exchange between the server and a single user. |
| [20] | **Security Misconfiguration.** | 2020 | The security issue of configuration malfunction happens when one or more components of a device are not properly designed, including systems, frameworks, application servers, web server, DB-server, network router, and platform. Specific, execute, and manage safe configurations. |
| | | | The danger is most commonly caused by default settings. This fault may be exploited by the attacker to render multiple assaults. The seriousness of the attack depends on the degree and location of the error. |
| [20] | **Sensitive Data Exposure** | 2020 | Sensitive data includes usernames, credit card details, passwords, etc. Once there is a bug in app security. The attacker can expose this data and there are 3 points of attack. The first is an information leak attack. The Second is transmission attack. The third is the database theft. |

## 4. Security Recommendations

In this section of our research, we review many studies presented by researchers, which dealt with many solutions and methods to detect the security flaws that cause malicious attacks, and we mention the following:

A study group launched a malicious script analysis tool in web apps in 2017, Where to detect XSS-Check. It determines user feedback for a returned web page, *verify* login-functional web pages, and provides information encoded http headers and the DOM parameter. Once specified, both the server and the client are validated across dynamic web pages, Fig.2 shows the flow chart for additional user interface for XSS screening [21]

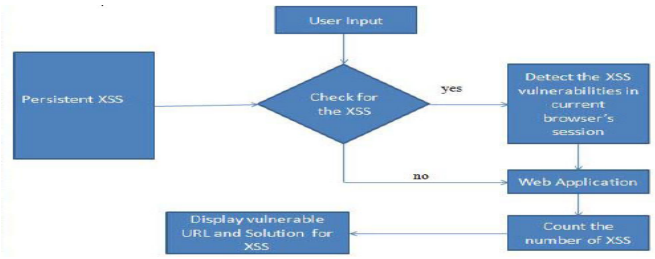


**Fig.2** Flow Diagram of XSS-Check add-on User Interface [21].

In 2019, The researchers introduced a method for finding and subsequently comparing Elastic Pooling CNN-based SQL injection (EPCNN). EP-CNN-based SQL injection recognition can automatically extract occult typical SQL injection traffic characteristics and identify attack traffic

bypassing Fast SQL normal injection. It is routed on a single letter, but credentials can be maintained from all query statements. If the vocabulary is limited, the complexity and costs of teaching can be minimized. The original usual method of identification can be replaced entirely by a good type and modified in real time. Then we will review the assault forms and apply a multi-class model not confined to the SQL attack definition. This approach can generate a static 2D matrix with no data truncation and detect SQL injection efficiently for web applications [22].

In 2020, The researchers listed all the weaknesses that could contribute to web apps, including remote file integration, and researchers used Open-Source software to determine multiple security vulnerabilities and penetration testing (VAPT) in Web applications that could induce security infringement. Security Infringements were identified. VAPT Security Test Tools are very useful for web application defense, so that safety violations do not occur [23].

In 2018, The authors present Rampart, which is a defense against the advanced DOS fatigue DOS attacks by the web applications. By mathematical approaches and program profiling at the functions, Rampar identifies, and prevents advanced CPUs fading DOS attacks. Furthermore, Rampart has been introduced as an enhancement to the PHP Zend engine. It combines and implements the filters to detect and mitigate future adversarial attacks and change to reduce any possible adverse effects on legitimate users. The filter is used to block further threats and
to eliminate potential harm to legal users. The filter is synthesized and used. The overhead effectiveness of Rampart is minimal and can be used without modifying the source code of the program for any PHP application. They show that Rampart offers protection in two of the world's most frequent web applications - Word Press and Drupal - against realworld and synthetic CPU exhaust attacks, and that the Ram component preserves web reliability, meaning that we will achieve not only a low false positive rate but also low false negative rate [24].

Any solutions to avoid buffer overflow attacks are restricted locks, the use of security labraries, and static code review. To restrict the number of characters to be sent as input, developers use JavaScript and HTML. However, you can always change the HTML using a hacker, switching off the Java Script and then send a buffer overflow attack. Client feedback must be scrupulously managed on the server to ensure that the program is secure against this threat. Also, an application may mount a firewall or security gateway to search requests with extraordinary duration [15].

The modern web-based architecture that defends web apps from CSRF attacks was implemented by a team of researchers in 2018. With the historical and *behavioral*-study of consumer queries, the classic WAF methodology was extended. Based on previous activities the user took. The scientists have initiated a classic WAF program fraud attack to defend malware such attacks while downloading them. This approach improves the overall response and feel of the submission. It can be developed more precisely and efficiently in the future [25].

In 2020, researchers developed FUSE, a penetration testing platform to find weaknesses in server-side PHP web applications in the UFU (Unrestricted File Upload). The FUSE purpose is to build download requests; any request becomes a payload exploit that undermines UFU and UEFU. 33 PHP *programming* test this method. 30 UEFU bugs, including 15 CVEs, were observed by FUSE to demonstrate the functional advantages of infecting file uploads with code execution errors [17].

In 2020, researchers suggested a technique for reducing TCP SYN Flood (DoS) attacks by using the Linux operating system CSF and SPI proof of concept (PoC) methods. Three means of conducting the security process: Setting up full IP server link and protecting the incoming SYN packet for each, the number of times a SYN minimum packet basis IP address is being broken by seconds before the firewall blocks it as it was determined that CSF is a smart Linux firewall tool capable of effectively handling the DoS Attack TCP SYN Flood sort, along with SPI methods. The TCP SYN Flood (DoS) attack style can also be mitigated in a quick and easy Furthermore, CSF offers a simpler and quicker way of alleviating a small case attack, the sort TCP SYN Flood (DoS). Other approaches are also available, for example: Advanced Political Firewall (APF), Firewall, the IP Blank Path server system and the Cloudflare service. Yet CSF provides further advantages dependent upon its /etc/csf/csf.conf setup. Attack avoidance, CSF will block dos forms for the professional admin to customize CSF.CONF [26].

In 2020 The researcher has created an algorithm that can safely register or navigate Web applications, the researcher developed an algorithm that could protect registration or access web apps, The hypothesis suggested is founded in zero-based awareness proof, A dynamic random number generation began with the current chaotic 6D-Hyper the bottom line is that all sides have a pin code (web client, user). The registration of these two numbers without a password is completed. The study findings revealed the value of the approach suggested, which securely and effectively manages and distributes the keys. Chaos generated random numbers are very reliable. The method has a very strong random power, since each number is

unpredictable. Via the current authentication scheme, attackers and users cannot obtain member keys since the Zero Knowledge method was paired with the proposed messy mechanism and RSA algorithm. Many of the keys used are dynamic and strong keys in the proposed system. This guarantees that the customer and the web application remain confidential and authenticated [27].

In 2018, investigators reviewed an *operator's* point of view of the first solution to the human implications of malicious protection measures. They noticed that security problems don't always lead to accidents: one-third of respondents indicated that their misunderstandings ended in a safety event and also found that human error was motivated by device operations. By systemic, personal, and interpersonal influences. The researchers proposed urgent action items as required and useful, which are, sadly,

seldom adopted by *organizations*, to minimize the occurrence and effect of security configuration errors. That contain the following.

Documentation, *transparent, transparency*, blurred post-mortems, outsourcing of systems and procedures [28].

In 2017, researchers proposed solutions to exposure to sensitive data attacks which include the protection of sensitive data using encryption techniques, not, if not necessarily, the storage of sensory data, the use of standard algorithms and strong keys, the deactivation of pages containing sensitive data, and the use of sensitive data. Powerful password-saving algorithm. Using SSL as well. [29]

In 2020, a group of researchers proposed a web-based detective model, which is based on in-depth learning since it is qualified to provide the exactness of the receiver launch curve on anomalous and benign web queries. This model uses an automated coding tool so it can learn and measure any word or letter from word sequences. In order to classify anomalous questions according to the type of attack. The classification engine is trained on ECML-KDD results. The findings revealed that this model was able to detect web app attack. The classification engine suggested is less than 100% reliable but in future, a greater data volume could be used to develop the classification engine [30].

In 2019, EPICS researchers suggested a security policy paradigm for dynamic network systems focused on the premise that data are often accessible anywhere but always followed by access policies, processes, access policy and communication. In data transformation, EPICS can be formed. And render it an active agent which protects itself against unwanted detection and handling. In case of threats, engine AB will detect them. In order to avoid identification, EPICS enables data to be lost dynamically. As an example of the suggested system, the scientists used the e-commerce situation. The EPICS structure is a huge advantage. Maintain data protection by maintaining regulated data distribution and minimize transparency. Consumers should exchange data without understanding the direction of data divulgation to determine access management rules, to make sure that the policies are enforced at either end of the relationship. They are TTP-independent and ask the data owner not to post data while the information is exchanged with the primary provider. Furnish the implementation of contextual adaptive data dependent on the application of external facts on the environment (for example confidence values, an emergency or an attack situation) and policy versatility. Reduces customer knowledge management liability for the service by disclosure of only the details *authorized* by the customer's policy to the service. Compatible with current networks, such as RESTful services focused on HTTP. It is policy-free and embraces a wide variety of common authentication and authorization protocols and can be integrated into any data distribution application. Fig. 3 illustrates the work of EPICS.

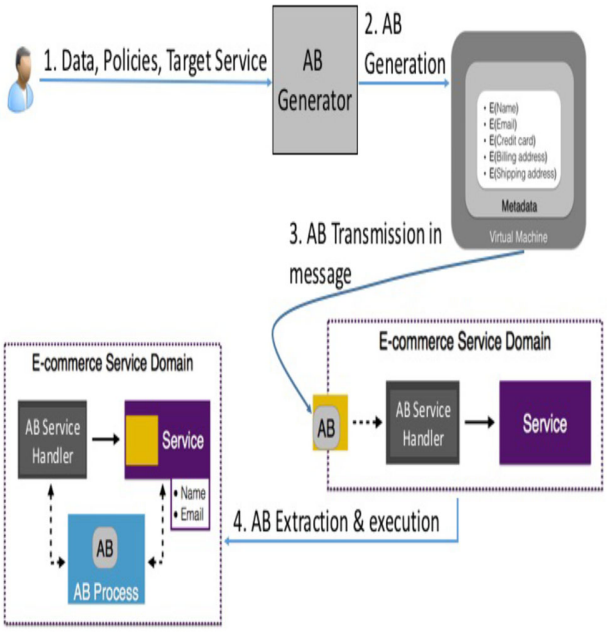


**Fig.3** EPICS operation [31].

In 2020 the developers suggest the SNEAKERS to identify web applications, bugs, which has identified and evaluated bugs in the exploitable network and suggested fixes for the vulnerability, Where ideas are being suggested so that hackers may not have unwanted access to company

information and information, where they search for safety vulnerabilities in web applications SNEAKERS is a testbed application for the Windows and Linux platform, This tool identifies web vulnerability and offers a Web Admin solution SNEAKERS web vulnerability evaluation and scanner [32].

In 2020 the researchers demonstrate how the static, dynamic and collaborative protection methods are designed for technical and algorithm testing ]The goal of this analysis is to explore how efficiency can be improved whilst the number of false positives can be lowered, To research their behavior, two static, two dynamic and two immersive methods of data analysis must be combined to take into account vulnerability vulnerabilities of the OWASP Top Ten and various scripts of separate criticalities in the apps analyzed. This work examines and discusses the principles of the metrics chosen for each n-tool combination that improves web apps safety by the use of an IAST tool combined; Another inference is that the combinations of IAST and DAST tools usually yield good outcomes, whereas the percentage of false- positive is greater than that of IAST and DAsT tools, Combined tools such as Fortify + Arahni+ CCE or Fortify + ZAP+ CCE, integrated with SAST+DAST+IAST, achieve very well in the classifications large, medium and small [33].

In 2018 Vulnerability scanners are used for automated web applications in order to really meet their potential, There is a JARVIS application, called the JARVIS tool, to provide technological alternatives that are feasible to be applied to a variety of vulnerability detectors, so that such restrictions can be resolved, two of the key reasons why they are not available in the system; With JARVIS, it is possible to boost the vulnerability detection efficiency of five scanners which are readily accessible by more than 100% when compared to the basic settings. Given that JARVIS's configuration effort is limited and independent on the scanner type, JARVIS also makes it possible to efficiently practice many scanners at the same time. Therefore, the authors have studied the possibility and drawbacks of parallel the use of multiple scanners, revealing that multiple scanners are beneficial when the number of vulnerabilities found increases, with no significant negative impacts on the study of the false positives. [34].

## 5.Discussion

Our analysis will examine many ways to safeguard web apps from malware threats, which several researchers have explored and investigated. The OWASP software has a lot of protection capabilities and is one of the most common programs to secure open- source web applications. In some researchers, a method has been developed to measure and evaluate the safety efficiency of SAST instruments, using some various forms of vulnerabilities for test cases in each OWASP group. This allows for the

reproducibility of SAST instruments to be compared and classified, the percentage of bugs found in the instruments analyzed shows a wide margin of change for the tools, Instruments except for spot-bugs with a summon value between 0.34 and 0.57 have bad effects, A skilled individual or team with protection capabilities in the language used in the target code and special security vulnerabilities for any language must constantly thoroughly check the vulnerability outcomes. Generally, the changing design and progression of the technology for online applications many improvements in vulnerability categories over time. Future studies are also essential for the tool to regularly detect the most frequent and relevant categories of vulnerability. Thus, regular modifications should be made to the OWASP Top Ten. The analysis also supports the TP and FP ratios improved with the use of many (SAST) instruments.

Other scientists have suggested an OWASP Stinger-based tool for validating input fields and a collection of regular terms and the sterilizing mechanism, The goal would be to ensure successful security against popular injection attacks for web applications such that fundamental characters (letters, numbers, periods, dashes, marks of doubt, and exclamation) and complex data (JSON and XML les) may be checked for each domain, A protection check for regular injection attacks has been used for the attack tool and the console was built in two stages for routing requests, The first-fiction submission is pointed to late**r**-phi and is forwarded to the web application itself if it is incorrect. The proposal filter was tested in three standardized applications and a genuine, we have reached a secret web app, 98.4 percen**t** accuracy and 50ms total loading time. Thus, the suggested filter is highly stable**,** and more computing power can be concluded. are not required.

Also, we will present features of Twitter spam detection in this section as Twitter is one of the prominent websites with 313 million daily monthly users sending five hundred million tweets every day. Spammers use Twitter to pick up legitime users or to

propagate malicious software and publicity via tweets that are posted in URLs, follow/follow legitimate users aggressively, or pull trendy subjects to draw their interest, propagating pornography, this success draws the attention of spammers. So, it is obligatory to track and filter spammers of legitimate users to provide a spam-free Twitter world. Twitter Spam identification requires different approaches to detect spam than standard email spam methods: Account-driven, tweet-based, graphical and

hybrid spam detections using a mixture of other, and the internet, as spammers prefer not to use the full URL, but use condensed URLs and Twitter builds on an extensive, comprehensive network based on posts, profiles, lists, moments, and links. A rigorous approach is thus essential to detect spam on Twitter, given the variety of legitimate users that, under some conditions, are close to spammers. Even Twitter itself does have false positive detections, as stated, (spammers who are listed as legitimate users).
Account-based methods analyze account with the use of account-related features, some of which may be used by spammers, such as the number of tweets that the account sent, number of lists of accounts created, number of times created by the brand-new account function, and the number of mentions that the account received, number of lists created by the account. Similarly, the following can be exploited by a network of bots: the

number of supporters, the percentage of the number of people following, the number of tweets enjoyed by others, and even a proportion of the number of tweets retweeted. Bots use different instruments to perform automatic functions, including following a person, to submit a tweet. Tweets of an account to detect spam spread by bots, a set of C&C (command and control) accounts, whether their contents are almost similar to the tweets posted recently.
Real-time spam detection, for instant review, is sufficient for the account-based functionality to be used. The number of user lists can be used as a valuable indicator for detecting spammers as this is an evident indication of the effect of the user on others, although it is possible to manipulate it by making false lists and inserting fake C&C accounts in such lists.

Account-based features are light enough to spot real-time spam, which needs immediate review, but spammers can comfortably handle them, the methods of tweet identification use aspects of the tweet including tweets and hashtags, these use the number of recipes sent, the number of recipes received for the tweet, Twitter content, tweet analyzes, tweet URL, the tweet place, the tweet postal date. Because a malicious URL is the most popular way to transmit spam, tweet URLs must be inspected. Consequently, nearly all spam prevention approaches on Twitter inspect tweet URLs. Traditional methods of filtering spam are based on blacklisting, domain, and URL, Spammers are generally unable to filter malicious
URLs on Twitter as they prefer to use shorted URLs, traditional blacklist URL or IP methods. Grier et al. also demonstrate that blacklist-based solutions are too sluggish to secure users, since the malicious URLs appear before the database, Similar to accounts, tweet features are light enough to spot spam, a fact that demands immediate analytical review, in real-time.

Graphic methods of spam identification are used to calculate the connectivity and distance of such accounts in conjunction with the sender, and to measure the connectivity of such accounts so that the probability of spam connectivity may be discovered. Unlike accounts and Tweet-based features, graph-based features are difficult to exploit. However, removing these features requires a detailed study of the large and complicated time and resource-intensive Twitter graph. As a consequence, graphical functionality is not sufficient light to spam in real time. detentions as opposed to account-based and tweet-based features. The graph-dependent approach is often restricted by assuming that tweets from friends are benevolent, irrespective of their content not true when attackers steal accounts from legitimate users for malicious purposes.

## 6.Conclusion

Web apps have now become an important and everyday part of our lives, as we communicate with multiple Internet-connected Web applications every day because these applications have been conceived according to various requirements. We examined the protection of web applications in our analysis and examined the most harmful threats and security flaws that will damage web applications and their consumers in the workflow and safety of these. We have analyzed numerous studies that include

alternatives to the security of web apps and the prospects of researchers. This method is positive, and they will still be improved in the future, like efficient solutions, as we discussed many tools and scanners that help in detecting security vulnerabilities that cause attacks on web applications, we finally came up with some recommendations for developers and users.

**Our contributions** are from the developers' side, in our research, we recommend conducting future studies through which detection and protection tools and scanners are developed, many of which, we have reviewed in our research, due to the frequent development of web applications. So that these studies enlighten the development of standards for detecting the security vulnerabilities that cause the occurrence of malicious attacks. On the side of users, we always recommend using strong passwords and changing these words from time to time. We also recommend not trusting links, notices or advertisements, or downloading any cookies from any sites unless you are sure of the reliability of this site. We also recommend using recognized and safe browsers. The user should also update their web browsers to the latest version.

# REFRENCES


[1] Sönmez, F. Ö., & Kiliç, B. G. (2021). Holistic Web Application Security Visualization for Multi-Project and Multi-Phase Dynamic Application Security Test Results. *IEEE Access*, *9*, 25858-25884.

[2] Zech, P., Felderer, M., & Breu, R. (2019). Knowledge-based security testing of web applications by logic programming. *International Journal on Software Tools for Technology Transfer*, *21*(2), 221-246.

[3] Raveena, K., Elavarasi, K., & Kaaviyapriya, M. (2018). Survey-web application development.

[4] Dhivya, K., Kumar, P. P., Saravanan, D., & Pajany, M. (2018). Evaluation of Web Security Mechanisms Using Vulnerability & Sql Attack Injection. *International Journal of Pure and Applied Mathematics*, *119*(14), 989-996.

[5] Shahzad, F. (2017). Modern and responsive mobile-enabled web applications. Procedia Computer Science, 110, 410-415.

[6] Biswas, S., Sajal, M. M. H. K., Afrin, T., Bhuiyan, T., & Hassan, M. M. (2018). A study on remote code execution vulnerability in web applications. In International Conference on Cyber Security and Computer Science (ICONCS 2018).

[7] Mohanty, S., Acharya, A. A., Mishra, D. B., & Panda, N. (2019). Security Testing of Web Applications UsingThreat Modeling: A Systematic Review. IJCSMC International Journal of Computer Science and Mobile Computing, 8(1), 50-57.

[8] Azad, B. A., Laperdrix, P., & Nikiforakis, N. (2019). Less is more: Quantifying the security benefits of debloating web applications. In 28th {USENIX} Security Symposium ({USENIX} Security 19) (pp. 1697-1714).

[9] Ali, A. N. M. B. M., & Elshoush, H. T. Secure Web Application Service Detecting–XSS Attacks.

[10] Andrian, R., & Fauzi, A. (2020). Security scanner for web applications case study: Learning management system. Jurnal Online Informatika, 4(2), 63-68.

[11] Wibowo, R. M., & Sulaksono, A. (2021). Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd. Indonesian Journal of Information Systems, 3(2), 149-159.

[12] Akbar, M., & Ridha, M. A. F. (2018). SQL Injection and Cross Site Scripting Prevention using OWASP ModSecurity Web Application Firewall. JOIV: International Journal on Informatics Visualization, 2(4), 286-292.

[13] Rahman, M. A., Amjad, M., Ahmed, B., & Siddik, M. S. (2020, January). Analyzing web application vulnerabilities: an empirical study on e-commerce sector in Bangladesh. In Proceedings of the international conference on computing advancements (pp. 1-6).

[14] Rajakumaran, G., Venkataraman, N., & Mukkamala, R. R. (2020). Denial of Service Attack Prediction Using Gradient Descent Algorithm. SN Computer Science, 1(1), 1-8.

[15] Awad, M., Ali, M., Takruri, M., & Ismail, S. (2019). Security vulnerabilities related to web-based data. Telkomnika, 17(2), 852-856.

[16] Khodayari, S., & Pellegrino, G. (2021). JAW: Studying Client-side CSRF with Hybrid Property Graphs and Declarative Traversals. In USENIX Security Symposium.

[17] Lee, T., Wi, S., Lee, S., & Son, S. (2020, February). FUSE: Finding File Upload Bugs via Penetration Testing. In 2020 Network and Distributed System Security Symposium. Network & Distributed System Security Symposium.

[18] Zeebaree, S. R., Jacksi, K., & Zebari, R. R. (2020). Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers. Indones. J. Electr. Eng. Comput. Sci, 19(1), 510-517.

[19] Hassan, M. M., Nipa, S. S., Akter, M., Haque, R., Deepa, F. N., Rahman, M., ... & Sharif, M. H. (2018). Broken authentication and session management vulnerability: a case study of web application. International Journal of Simulation Systems, Science & Technology, 19(2), 6-1.

[20] Fredj, O. B., Krichen, M., Hamam, H., & Derhab, A. (2020). An OWASP Top Ten Driven Survey on Web Application Protection Methods.

[21] Jasmine, M. S., Devi, K., & George, G. (2017). Detecting XSS Based Web Application Vulnerabilities. International Journal of Computer Technology & Applications, 8(2), 291-297.

[22] Xie, X., Ren, C., Fu, Y., Xu, J., & Guo, J. (2019). Sql injection detection for web applications based on elastic-pooling cnn. IEEE Access, 7, 151475-151481.

[23] Malekar, V., & Ghode, S. A Review on Vulnerability Assessment and Penetration Testing Open Source Tools for Web Application Security.

[24] Meng, W., Qian, C., Hao, S., Borgolte, K., Vigna, G., Kruegel, C., & Lee, W. (2018). Rampart: Protecting Web applications from CPU-exhaustion denial-of-service attacks. In 27th {USENIX} Security Symposium ({USENIX} Security 18) (pp. 393-410).

[25] Meng, W., Qian, C., Hao, S., Borgolte, K., Vigna, G., Kruegel, C., & Lee, W. (2018). Rampart: Protecting Web applications from CPU-exhaustion denial-of-service attacks. In 27th {USENIX} Security Symposium ({USENIX} Security 18) (pp. 393-410).

[26] Pratama, I. P. A. E. (2020). TCP SYN Flood (DoS) Attack Prevention Using SPI Method on CSF: A PoC. Bulletin of Computer Science and Electrical Engineering, 1(2), 63-72.

[27] Mohammed, S. J., & Mehdi, S. A. (2020). Web application authentication using ZKP and novel 6D chaotic system. Indonesian Journal of Electrical Engineering and Computer Science, 20(3), 1522-1529.

[28] Dietrich, C., Krombholz, K., Borgolte, K., & Fiebig, T. (2018, October). Investigating system operators' perspective on security misconfigurations. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 1272-1289).

[29] Vamsi Mohan, V., & Malik, S. (2017). DEBUNKING OF COMMON.

[30] Alma, T., & Das, M. L. (2020). Web Application Attack Detection using Deep Learning. arXiv preprint arXiv:2011.03181.

[31] Ranchal, R., Bhargava, B., Angin, P., & ben Othmane, L. (2018). Epics: A framework for enforcing security policies in composite web services. IEEE Transactions on Services Computing, 12(3), 415-428.

[32] Darus, M. Y., Omar, M. A., Mohamad, M. F., Seman, Z., & Awang, N. (2020). Web vulnerability assessment tool for content management system. International Journal, 9(1.3).

[33] Mateo Tudela, F., Bermejo Higuera, J. R., Bermejo Higuera, J., Sicilia Montalvo, J. A., & Argyros, M. I. (2020). On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications. Applied Sciences, 10(24), 9119.

[34] Esposito, D., Rennhard, M., Ruf, L., & Wagner, A. (2018). Exploiting the potential of web application vulnerability scanning. In *ICIMP 2018 the Thirteenth International Conference on Internet Monitoring and Protection, Barcelona, Spain, 22-26 July 2018* (pp. 22-29). IARIA.

**Asma Mohammed** received her BA from Al Baha University, Kingdom of Saudi Arabia, majoring in Information Systems in 2013. currently. I joined Al-Taif University to obtain a master's degree in cybersecurity. Asma's interests include internet security, network security, and information security.

**Jamilah alkhathami** She received her Bachelor's degree in Computer Science from King Khalid University in Saudi Arabia in 2014. Jamila is an employee at the Ministry of Interior for Civil Affairs in the Kingdom of Saudi Arabia. She is currently studying at Taif University for a master's degree in cybersecurity. Jamila's research interests include cybersecurity, Internet of things, and cloud computing.

**Hatim Alsuwat** is an assistant professor of Computer Science in the College of Computers and Information Systems at Umm Al-Qura University. He received his Ph.D. from the department of Computer Science and Engineering at the University of South Carolina (USC) in 2019. His research interests include Information Security, Cryptography, Model Drift, and Secure Database Systems.

**Emad Alsuwat** is an assistant professor of computer science in the College of Computers and Information Technology at Taif University. He received his Ph.D. from the department of Computer Science and Engineering at the University of South Carolina (USC) in 2019. His research interests include Probabilistic Graphical Models (esp. Bayesian Networks), Artificial Intelligence, Information Security, and Secure Database Systems.