ISSN 2278-3091
Volume 10, No.2, March - April 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse441022021.pdf
https://doi.org/10.30534/ijatcse/2021/441022021

Pattern Image based Dynamic Framework for Security in Web Application

Khaled M. Alalayah
Department of Computer Science, Faculty of Science and Arts, Sharurah,
Najran University, Saudi Arabia
Department of Computer Science, Faculty of Science.
IBB University, Yemen
E-mail: kh101ed2005@yahoo.com

ABSTRACT

Web Application Security is one of the most important nonfunctional requirements whenever we build a web application. A chain is just as though it were effective as its weakest relation, as is the case with defence. We might be having very strong applications and might be having very strong infrastructure because one small flaw can break the security. In terms of image-based security the image processing techniques are used. The image can be used for authentication of the system like in the present work the image is being used as "pattern image based authentication system". In majority of the web systems the security can be comprised by visualizing the passwords on runtime or accessing the system by unwanted calls like phishing attacks. In the current work the image based pattern is being used which actually is dynamic in nature like changes after every attempt made by the user either successful or unsuccessful, the patterns are drawn on the basis of the passcode generated. The system works for the manual hacking of the user id and password and also hacking the data by visualization. The developed system is prone to many type of attacks like phishing attack where the attackers shares information over mail, message and other medium to get the secure information from the user. This technique is used to develop an effective web application, and the results show that our method provides a quick and secure image transmission process.

Keywords: *Web Application; Pattern Image; Dynamic; Security; Integrity; Availability; Image.*

1. INTRODUCTION

WWW is the basic part to start saying about the web, things started from a simple static we documents and have changed to the rope of the multiple seed level web document presentation and is most considered technology now a days in almost every of the field. The services using the web technologies are like remote access, compatibility using the cross platforms, fast and efficient development, etc, in the current world the usage of the web application is increasing which actually is due to the several factors. The major factor which are responsible for the popularity of the web based tools are the effectiveness, instructiveness and responsiveness, and also the technologies like AJAX which actually works for improving the user working experience over web applications.

The web application as are used at larger extent are on the target of the attackers; this is due to the web applications as they Increasingly used to provide essential security services. With back-end database systems, several web applications interact, and it may lay up responsive information (like, banking, medical), in the case when the data available over web is compromised which may result in the high risk data loss and also can create legal consequences. The Figure 1. below shows the amount of data available and also the amount of data compromised by the attackers, the data presented is as per the report submitted by Verizon [1].

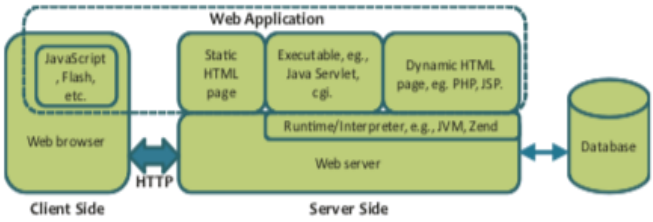

Figure 1: Overview of Web Application.

Web system actually is a highly complex structure which includes technologies like HTTP protocols, web servers, Java script, Flash, etc, and many other connected segments and components. The different components available in the complex web system is having several challenges and related inconsistencies while working with the components of the web system. As the web application framework is highly used in majority of available business frameworks, beyond which the same is having very little to be said about the security of the same. Because of which several security techniques are implanted with the development of the web application to make the process error prone, the techniques provided are to secure the data available and also to provide sufficient alerts to the market and user related to the same. Just because of high usage and availability plenty of applications are available over the internet are not prone to security threats and are not able to counter vulnerabilities related to safety. On the basis of the review study considered by Web Application Security

Consortium [2], for checking the security threats over the available web applications it is been noticed that about 49% of the applications are having high risk and also about 13% of the web data can automatically compromised. On the basis of the recent report [3] it was like 80% of the web content is having serious risk of attacks and can be compromised easily.

Just because of the data presented in the reports about the chances of threats on the web applications, several studies are conducted and also several techniques are presented just to make the web application a secure platform to work with. In the majority of the developed technique a specific type of threat or vulnerability is being considered and rest are considered as an assumption to carry out the further study or process. On the basis of the specific cause the technique is counted that is it working well or not and also the developed technique only provide security to the limited part of the platform. The developed technique which actually works for the specific part or component of the web infrastructure can be better used by integrating them, as they cannot directly applied as addition, a proper integration algorithm is required to make them work together. A clear framework is required to determine the primary cause of the vulnerabilities, challenges, security threats etc. This form of framework will allow both new and seasoned researchers to better understand web application security issues and analyse existing defences, and motivate them with fresh ideas and trends.

2. UNDERSTAND WEB APPLICATION SECURITY PROPERTIES, VULNERABILITIES AND ATTACK VECTORS

A secure web application has to comply with the desired security properties under the given threat model. Generally, in the area of web application defence, the following threat model is considered:

1) *The web application itself is benign and hosted on a hardened and trustworthy infrastructure i.e., the trust computing base, including OS, web server, interpreter, etc.) (i.e., not hosted or owned for malicious purposes);*

2) *The intruder is capable of modifying either the content or the Web request series sent to the web application, however the infrastructure or the application code may not be explicitly breached. In web application implementations, the vulnerabilities can breach the expected safety features and permit acceptable successful exploits.*

Specifically, a secure web application should retain the following stack of security properties, as shown in Figure 2. Data validity means that user feedback should be reviewed before it can be used by the web application; state integrity means that the application's status should be kept untampered; logic correctness means that the application's logic should be properly implemented as expected by the developers. The three properties of protection referred to above are interconnected in such a way that failure to maintain a lower-level security property will have an impact on the higher level of security assurance of the security property. For instance, if the web application fails to maintain the input validity property, a cross-site scripting attack can be carried out by the attacker to steal the victim's session cookie. Then the intruder will hijack and tamper with the victim's web session, resulting in the infringement of state integrity properties. In the following sections, we describe the three security properties and explain how the unique characteristics of web application development complicate the security architecture for web applications.

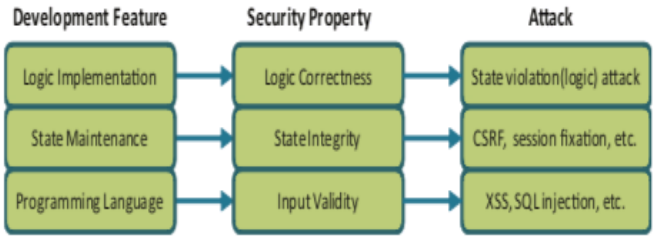

Figure 2: Web Application Security Properties.

A. Input Validity

User input data, given the threat model, cannot be trusted. They must, however, be validated first for untrusted user data (e.g. writing a SQL query or web response) to be included in the application. Thus, we refer to this protection property as the input validity property:

To make sure that it is used by the web application in the intended way, all user feedback should be correctly reviewed.

User input validation is also carried out via sanitization routines that by filtering suspicious characters or constructs within user input, transform untrusted user input into trustworthy data. While simple in principle, it is non-trivial to achieve the completeness and correctness of user input sanitization, particularly when scripting languages are used to programme a web application. First since user input information is propagated across the software, all sanitization points must be monitored to identify it all the way. However the dynamic features of scripting languages have to be carefully handled in order to ensure accurate monitoring of user input data. Second, proper sanitization must take into account the context that determines how the user input is used by the application and is later interpreted by either the web browser or the SQL interpreter. Therefore, different contexts include distinct sanitization functions. However the weak typing feature of the programming language makes context-sensitive sanitization difficult and error-prone.

Sanitization routines are usually manually placed by developers in an ad-hoc fashion in current web development methods, which can be either incomplete or erroneous, and therefore introduce vulnerabilities into the web application. Missing sanitization allows malicious user input to flow into trustworthy web content without validation; faulty sanitization allows malicious user input to circumvent the validation process. A web application with the above vulnerabilities does not carry out the input validity property, It is also vulnerable to a class of attacks known as data-flow attacks, script injections or attacks on input validation. This method of attack embeds malicious content into web requests that the web application uses and later executes. Examples of input validation attacks include cross-site scripting (XSS), SQL injection, directory traversal, filename inclusion, response splitting and so on.

They are distinguished from each other by the locations where malicious content is executed. Below are the two most common input validation attacks outlined.

1) SQL Injection: A SQL injection attack is successfully launched when malicious content within user input flows into SQL queries without proper validation. The database trusts the web application and executes all the queries provided by the application. Using this attack, the attacker will insert SQL keywords or operators into user input In order to manipulate the structure of the SQL query and proceed to Unintended implementation. SQL injection effects include by-pass authentication, information leakage and even the loss of the entire database. Interested readers can refer to [4] for more information on SQL injection.

2) Cross-Site Scripting: A cross-site scripting (XSS) attack is successfully initiated when malicious content within user input flows into web responses without proper validation. The web explorer interprets all web responses that the trusted web application returns (according to the same-origin policy). Using this attack, the attacker is able to inject malicious scripts into web responses that are executed within the victim's web browser. The most prevalent outcome of XSS is the leakage of sensitive information, such as session cookie theft. Usually, the first stage of more sophisticated attacks (e.g. the notorious MySpace Samy worm [5][14]) acts as XSS. There are several types of XSS, including content-sniffing XSS, stored/persistent XSS (malicious scripts are inserted into persistent storage) DOM-based XSS,, reflected XSS, [6],[15], etc, depending on how the malicious scripts are injected.

B. State Integrity

State management, which requires a secure web application to preserve the dignity of application states, is the basis for creating state-of-the-art web applications. However the untrusted group (clientinvolvement)'s in the management of the state of the application makes it a difficult problem for web applications to maintain state integrity.

A number of attack vectors target the vulnerabilities in session management and state maintenance mechanisms of web applications, session fixation (When the identifier for the session is predictable,), session hijacking (When the identifier for a session is stolen), etc., including cookie poisoning (The tampering of cookie data), Cross-site request forgery (i.e., session riding) is a common attack that falls into this category. In this attack the attacker tricks the victim into sending designed web requests with the victim's correct session identifier, but on behalf of the attacker. This may lead to the victim's session being tampered with, sensitive data being leaked (e.g.[7]), financial losses (e.g., an attacker may file a web request instructing a vulnerable banking website to pass the victim's money to his account), etc.

To protect state credibility, a number of effective strategies have been proposed [8]. Via MAC (Message Authentication Code), integrity verification can protect client-side state information. It is important to build (to defend against session fixation) session identifiers with high randomness and to transmit them (against session hijacking) via secure SSL protocol. They can be web requests reviewed in order to mitigate CSRF attacks by checking headers [9] or related special secret tokens [10], [11], [12]. Due to the relative sophistication of the methods of maintaining the integrity of the state, they fall outside the scope of this study.

C. Logic Correctness

Maintenance of logic consistency is the secret to the functioning of web applications. Because the logic of the application is specific to each web application, a single description does not cover all aspects. Instead, a general concept that covers the most common features of the application. The following is given, to which is referred as a logic correctness property[13]:

Only approved information and activities can be accessed by users and they are required to follow the intended workflow provided by the web application.

It can be difficult to implement and execute application logic correctly because of its state management mechanism and the "decentralised" nature of web applications. Second, the interface's hiding technique, which follows the principle of 'security by obscurity', is obviously flawed in nature, allowing the attacker to discover hidden connections and directly access unauthorised information or operations or breach the intended workflow. Second, developers manually and in an ad-hoc way conduct explicit testing of the application state. Thus, it is quite likely that such state checks on unintended control flow paths are missing because of the various entry points of the web application. Moreover, writing accurate state checks can be error-prone, since It is important to consider not only static safety steps, dynamic knowledge on the state, but also. In both incomplete and faulty state checks, logic bugs are introduced into web applications.

A web application with a logic flaw is vulnerable to a class of attacks, commonly referred to as logic attacks or state infringement attacks. Logic attacks, since the application logic is special to each web application, are also unique to their specific targets. Several attack vectors falling within this range (or partially) include authentication bypass, forceful browsing, requirements tampering etc. Application-specific vectors of logic attack also exist. A weak e-commerce website, for example, can allow multiple applications of the same coupon that can be used by the attacker to decrease his payment amount.

3. CATEGORIZE EXISTING COUNTERMEASURES

To ensure security of the web application and also protect the same from wide type of attacks many techniques are available or are developed. The developed technique works for a specific threat or components or even can incorporate more than that, the developed can be better used at the various stages of web application creation development and execution phase too. Along with two dimensions, we organize existing countermeasures. The security property is the first dimension that these techniques address. Following three classes are regarded as the second dimension is their design principles are as follows [13-15]:

1. **Security by construction:** This class of techniques aims to develop stable web applications to ensure that there are no possible vulnerabilities within the applications. The resulting exploits would then fail and the desired security property would be retained. And they typically use modern frameworks and web programming languages that are developed with the aid of security mechanisms that repeatedly enforce the security features necessary. They are also more robust as these approaches help to address safety issues from the foundation. In addition, they are deemed most appropriate for new web application creation. Finally, it may be impossible to rewrite a significant number of legacy applications.

2. **Security by verification:** Its main objective is to confirm whether a web application maintains the necessary security properties and to identify possible inside the device vulnerabilities. This approach is often known to be vulnerability analysis. The efforts to harden insecure web applications are then expended by swindling the bugs and retrofitting the framework either manually or robotically. This class of techniques can be applied to both modern and legacy web applications.

   In this type of works the previous program analysis works well for the testing and development. For effective and efficient counter measure various technical challenges are needed to be considered. Majorly static and dynamic analysis is being used where the code reviews and analysis of the behaviour of the program works for the detection or even for the counter measure of the several types of the vulnerabilities.

3. **Security by protection:** In this type of counter measure process a specific platform can be developed to ensure the secure execution of a mobile server application. In this type of process, the web application is kept separately from rest of the available components of the web ecosystem. Scale well then, as if these approaches programming languages or platforms may be independent of this. Moreover, it inevitably introduced the runtime performance overhead.

4. PROBLEM WITH WEB SECURITY

It is considered a significant security research issue to allow users who run a client on an untrusted network to interact securely with the web application. As there are some people who want to do business with timid site. There are some companies who don't like to share their information about their own security holes. Nowadays it is too hard to get the consistent information about the state of web security. As there was no need, there are many sites that did not substantiate users. Each user is treated in a same manner & present same information. In web server software, any security threats occurring by hosting a website were largely connected to vulnerabilities. Attacker would not gain access to any sensitive information as if he comprises a web server, this is due to the information alleged on the server was previously open to public view. To deface the website's contents, an attacker typically modify the files on the server. As if the user believes her information will be disclosed with unauthorized parties than in this condition no user wants to use the web application.

a. Web Applications Attacks

Unique vulnerabilities of a Web application should decide the technologies you use to protect it. Figure 3 demonstrates several points that could need security within a system. Often, first of all it is better to use common countermeasure principles Instead of one that claims to tackle the latest hacking process to help ensure that you choose the most effective technology for needs.

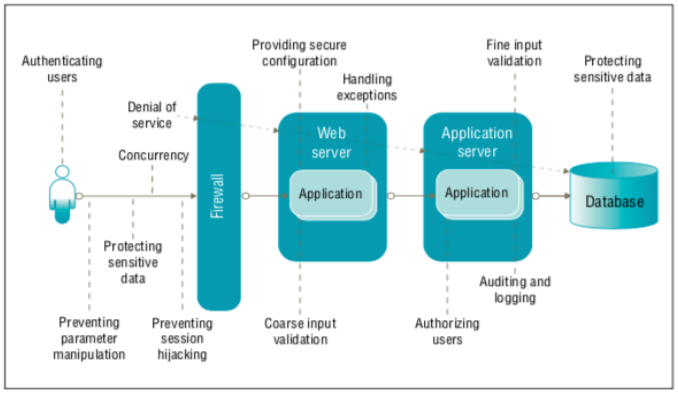

Figure 3: Web application security concerns

Popular threats and preventive steps are shown in Table 1. pecific risks to your application can however be different.

Table 1: Common types of Web application attacks

Description	Common Causes	Preventive Measures
Impersonation		
Typing the credentials of another user or modifying A cookie or an impersonation parameter for a consumer or to pretend that the cookie comes from another server	Using authentication based on communications to allow access to the data of any user Use passwords that can be collected and reused without encryption In cookies or criteria, shop credentials Using unproven methods of authentication or authentication from the wrong realm of trust Not allowing client applications to authenticate the host	Using strict authentication and credential details security using: Operating system (OS)-Frameworks supplied Encrypted tokens, including cookies for sessions Digital signatures
Tampering		
Changing or removing an unauthorised resource (e.g. altering transit data, defacing a Web site)	Trusting sources of data without validation Sanitizing input to stop unnecessary code execution Driving with progressive rights	To lock down files, folders and other tools, use OS protection to Validate your details Hash and sign data in transit (for example, by using SSL or IPsec)

	Leaving unencrypted confidential data	
Repudiation		
Trying to erase, cover or modify proof that an event has taken place (e.g. impersonating a user to request changes, removing logs)	Using a slow or incomplete method for authentication and authorization Improperly Logging Allowing confidential details about unsecured channels of communication	Using strict authentication, digital signatures, and transaction logs Audit
Information Disclosure		
Personally identifiable information (PII) disclosure, such as passwords and credit card information, plus information about the source of the application and/or its host machines	Allowing an authenticated user access to other users' data Allowing sensitive information on unsecured communication channels Selecting poor encryption algorithms and keys	Store PII on a session (transitory) rather than permanent basis Where necessary, use hashing and encryption for confidential data. Match user data to user authentication
Denial of Service (DoS)		
Flooding-sending multiple messages or concurrent requests to overload a server Lockout: submitting an increase in requests to force a slow response from the server by consuming resources or restarting the application	Placing too many apps on a single server or putting competing apps on the same server Neglecting to do rigorous testing of units	Filter packets using a firewall To monitor the number of requests from a single source, use a load balancer to To deal with processing-intensive requests and error recovery, use asynchronous protocols
Elevation of privilege		
Gaining administrative rights or access to sensitive data exceeds standard access privileges	Running processes on Web servers as "root" or "administrator" Using errors in coding to make buffer overflows and lift the application to a debug state	Use fewest privileges context whenever possible To avoid or monitor buffer overflows, use type safe languages and compiler options

b. Basic Guidelines For Web Security

Software development teams can defend against violations of confidentiality, such as those seen in table 2. [70], by using security-specific processes to construct applications. In particular, it is important to apply any specific guidelines to existing applications and new ones. In your method, or re-engineered applications to help achieve higher protection and lower costs of remediation, such as:

- **Discover and create baselines*:*** Run a full inventory of software and services, including technical details (e.g. Domain Name System [DNS], plus business data (e.g. Who approved the deployment? Who should be informed if the application fails?) using the Internet Protocol [IP] OS). Next for common vulnerabilities and exploits, search the Web infrastructure. For any known assaults on your OS, Web server and other third-party products, search list services and bug tracking pages. Ensure that you have scanned, hardened and patched the server before loading the application onto a server. Then, search your application for known attack vulnerabilities, look at HTTP requests, and other data manipulation opportunities. Finally, verify the authentication of the system and monitor the user rights function, and terminate unknown services.
- **Assess and assign risks:** Score applications and frameworks for risk, focusing on data stores, access control, user provisioning and privilege management. Prioritize system flaws identified during tests. Review the implementation of federal, corporate and organisational policies. And recognise acceptable as well as objectionable behaviours.
- **Shield your application and control damage*:*** Keep on top of known security threats and apply available updates to your apps and/or infrastructure. If you are unable to fix a security issue, use an application firewall, restrict access, disable the application or move it to minimise exposure.
- **Continuously monitor and review*:*** Plan reviews as part of the reported change management process. Cause a new discovery process immediately when you shut one out.

5. RESEARCH OBJECTIVES

- To study and analyze various techniques for image-based security in web application.
- To study the various challenges and issues related to image-based security in web application.
- To present a new approach for image based integrated security in web application.
- To carry out the proposed strategy for validating the outcomes and comparative analysis between the existing approach.
- The proposed approach for web application security considering the current and future investigations in security.

6. LITERATURE REVIEW

Dhamija and Perrig [16] presented an authentication technique using the Hash Visualization technique [17]. In the presented study the user of the application needed to choose the images from the generated set of images before granting

the access to the web. The use is then authenticated by the means of comparison of the selection of images with respect to the query made and images generated by the program. The major limitation of the technique was that it is necessary for the server to the seed source of every user image in the form of the plain text.

Akula and Devisetty's algorithm [18] is very much similar to the process provided by [16]. The major difference in the present work is about the use of the SHA-1 which actually generates 20 byte output and makes the process of more safe authentication and also uses less memory as compared to the previous technique. IN the future prediction or further development of the work author stated that the work can be uploaded over the internet, mobiles and over PDAs.

Weinshall and Kirkpatrick [19], in the study have considered various techniques as recognition of pictures, recognition of pseudo words and object recognition and on the basis of which author conducted several researches. In the picture recognition technique the user is trained for the selection large number of image from the dataset available the image to be selected can be like 100-200 and dataset can be of 20000 images. On the basis of the study presented by the author it is quite clear that picture based recognition technique works well over all other presented. The pseudo code recognition can also be used as replacement of the picture based recognition but a serious training and setting is being required.

Jansen et al. [20-22] have represented a graphic password method for the cell phone users. In the methodology at the time of the registration the user is asked to consider some of the thumbnail image and make the sequence of images as password. At the time of the authentication process or for sign in the user is asked to enter the set of images in the same sequence as was entered at the time of registration. The major limitation of the technique is about the limited space and also limitation for the selection of the image as sequence. In the generated sequence the image is considered as numerical value and the set of image will make a numerical password. In the results shown the author in the study it was quite clear that the images in the sequence are less than that of the textual password generated at the time of the registration.

Recognition based techniques are also termed as the cognometric systems. In this technique the user needed to generate the portfolio of images at the time of the password generation and at the time of sign in the user is supposed to recognise the generated images. Just because of the study which shows that the humans are having high capability of memorising the images, because of which the image based recognition techniques became popular. In many of the studies several recognition based techniques are proposed considering various types of images like facial images, iconic images, objects of every day use, randomly generated arts, and many others [23].

The recall based graphical password technique is also termed as the drawmetric technique where the user is asked to draw an outline over the grid generated at the time of the registration phase. In this technique the user is actually asked to draw the pattern over the grid or even over the blank canvas. The major limitation of the proposed technique is about the memorializing the patterns drawn because there is no clues or suggestions for the same [21][24].

Cued-recall techniques are also termed as the locimetric techniques where the specific location is to be identified. In this type of techniques the user is asked to click over the location of the image. In the proposed system it is quite easy to remember the location selected in the image which is then to be used as the authentication step at the time of sign in. In the discussed methodology the user is needed to select the specific point over the provided image and the same is to be clicked at the time of the sign in process and the being used as the password for the web application. In the authentication process the user is supposed to right click on the point previously selected [20] [25].

In the hybrid schemes the two or more graphical password techniques are used together to generate a single password or authentication system. In this techniques the limitations of the single type of authentication techniques are removed like hotspot problem, shoulder surfing, spyware, etc. In the hybrid type of techniques the several types of recognition techniques like recall based, recognition based are grouped together [26].

Xiong [27] in his work have conducted a review study and presented proposal using the Central Authentication Services. The technique proposed by the author is quite more secure than that of the Xiong's, where the author uses the SPNEGO for the process of authentication. In the case when the CAS is being used for the authentication process the password and id is shared over the network every time the login request is made. While in the case of the SP-NEGO the password and id sharing is not required rather a simple token is being shared.

Kabay [28] in his have conducted a study related to the identification, authentication and authorization over the internet and also the author have not included any specific case study in the work. The author also have reviewed many of the techniques which are currently being used in the market.

In many of the companies dealing in network authentication uses Kerberos. Tagg [29] have presented a study describing about the Kerberos and also have explained how the technique is being used for SSO over the business networks. Samar [30] in his study have described about the cookie based authentication system which actually is a alternate technique for Kerberos [22].

Other than the above presented literature many of the other authors were referred to generated the summarized details about the attacks and techniques for the resolution of the same which is below depicted in tabular form.

Most of the content of the table is a quote from reference [70].

Table 2. Summary of conclusions from the study.

Type of Attacks	Work Summary	Focus areas
XPath injection [31]	An architecture that uses a monitoring framework for run-time to detect malicious queries to avoid XPath Injection.	Attack prevention
SQL injection [32]	Online services detect SQL vulnerabilities based on mutation operator-related automated testing approach.	Vulnerability detection
DOS attack [33]	An adaptable algorithm to test web services by parsing incoming DOS attack XML messages	Attack detection
SQL, XPath injection [34]	To find security vulnerabilities, a comparison of current vulnerability scanners against 300 public web services.	Vulnerability detection
SQL, XPath injection [35]	An automated approach to the discovery of XPath/SQL injection vulnerabilities in web services.	Vulnerability detection
SQL injection [36]	A systematic approach to finding SQL injection vulnerabilities using a penetration testing tool for web services.	Vulnerability detection
DOS attack [37]	Model a web services architecture and build a philtre protection framework to defend against XML-based DOS.	Attack detection/Attack prevention
XML injection attack [38]	A proposal is suggested in order to detect and resolve XML injection attacks with pluggable APIs and middleware protection services..	Attack detection
Spoofing. DOS[39]	An automated pluggable API model was proposed for detecting threats at the network stage.	Attack detection
Dos attack[40]	A proposed classification real-time-based framework agents to detect and prevent DOS attacks on the web services.	Attack detection/ Attack Prevention
SQL,XML, XPath injection[41]	A new proposal to complement the sound and detailed slices of current vulnerability detection farming, thus distinguishing true positives and false.	Other
Spoofing[42]	Suggested a pattern of misuse to avoid the same attack on web services called spoofing web services.	Attack detection
XML injection[43]	A hybrid learning universal approximator model is proposed for detecting XML SOAP-based attacks on Web servers.	Attack detection/ Attack Prevention
SQL, XML, injection Dos attack[44]	There is a proposed intrusion detection approach focused on fuzzy rules to prevent injection and denial of service attacks.	Attack preventions
Dos attack[45]	It requires a process of content introspection to prevent XML-based attacks against Denial of Service on web servers.	Attack preventions
Dos attack[46]	The effectiveness of an access grant cryptographic authentication technique to avoid DOS assault on web services is reviewed.	Other
Dos attack[47]	Deploying identified vulnerabilities of DOS on web services to recognise the effect on resources and efficiency.	Other
Dos attack[48]	An adaptive approach that incorporates Case Based Reasoning(CBR) capabilities and multi-agent frameworks to defend web services from malicious soap messages.	Attack services

XML injection [49]	A web services based WS-inject used to detect XML injection vulnerabilities.	Vulnerability detection
Dos attack[50]	An informative, real-time architecture for using logic in a time-bound way to distinguish DOS attacks against web services.	Attack detection
Dos attack[51]	An automatic plug-in is suggested that can analyse web services based DOS attacks on black box testing.	Other
Dos attack[52]	Propose an approach to classify DOS attacks on intrusion-tolerant web services.	Attack detection
Dos attack[53]	A gateway scheme focused on schema hardening is suggested to combat XML signature attacks.	Attack detection
SQL, XML injection [54]	To prevent attacks on web services such as coercive parsin, injection etc., an architecture and filtering method is suggested and checked.	Attack detection
Dos attack [55]	To achieve better true detection speeds, a vector Quantization-based intrusion detection method for DOS attacks on web services.	Attack detection
Dos attack [56]	With adaptive rule updates and minimising DOS attacks, a method is suggested.	Attack detection
XML signature injection [57]	For the prevention of XML signature attacks, a Schema Hardening approach is addressed.	Attack detection
SQL injection, XPath injection [58]	An approach based on the notion of anomaly detection to stop injection attacks and SQL/XPath.	Attack detection
SQL injection, XPath injection [59]	A learning-oriented technique based on the principle of anomaly detection to detect XPath & SQL Injection attacks.	Attack detection
Dos attack [60]	A suggested approach to evaluating the security of the DOS Attacks service platform by multi-phase testing.	Other
XML injection [61]	A hybrid approach that applies ontology to the information database to detect web services based XML Injection attacks on the basis of knowledge.	Attack detection
XML injection ,Dos attack [62]	Identify the numerous web service attacks, recommend methods to counter the attacks and build a self-adaptive hardening validation mechanism.	Other
Dos attack, XML injection [63]	A series of tests to classify potential attacks on web services was performed on SOAP requests	Attack detection
XML injection [64]	A tool for detecting XML injection attacks based on SOAP message tree verification based on XML.	Attack detection
Dos attack [65]	A scheme is specified for defending DOS as well as XML-based DOS attacks on web services.	Attack detection/ Attack Prevention
XML injection [66]	A framework based on the interception, identification of node counting and logging module is designed to detect XML attacks.	Attack detection

7. RESEARCH METHODOLOGY

a. System Overview

The technique of access control is being used for preventing the unauthorized access to the system. The companies providing different account based services to the user mustMake sure that unauthorised access is not permitted and that unauthorised access is not required. no unauthorized user can make any short of the changes in the data available for specific user. The access control can be ensured in many of the ways like identifying the badges and passwords for the purpose of authentication and ensuring the security of the web. In general form there exist two types of passwords:

- **Static/Dynamic Password**

Static type of the password are the tradition system of password where the password is changed when it is required

otherwise the password remains unchanged, the password can be changed in the cases like password reset where the conditions are like password forget or the expiration of the password. The static type of password are having high risk of cracking because the static password once entered is saved in the cache memory of the system.

On the other hand dynamic type of passwords are the one which changes whenever the user login to the account. For eg OTP where the OTP is the characters set which can be used for authentication for only once and once the password is used it cannot be used further for any type of authentication process. In the case when the attackers cracks the dynamically generated password then in the most of the cases the saved password is used already and cannot be used further and useless for all. Using the dynamic type of password the system is secure from the hacks like network sniffing, password retrieval, and protects the authenticity of the user [62, 63].

In the proposed system a dynamic password generator technique is being used for authenticating the user, considering the type of attack as phishing where the attacker try to present themselves as legitimate and tries to get some secure information from user's as password, and other related secured information. In the defined technique we are introducing the pattern drawing or "Pattern image based dynamic authentication system" with which the access to the web application which is having the sensitive information like banking can be further secured. In the proposed methodology two layered security mechanism is being used like first system will authenticate via password provided and then will consider the pattern drawn using the passcode generated at the time of signup. The arrangement of the passcode will be dynamic in nature like every time the position of the number in the pattern image will be updated automatically. Authentication based on the image is included in order to provide additional protection integrated withpattern matching where the patterns changes on every successful or unsuccessful attempt.

Points showing the difference and requirement of the current work on the basis of the previous studies conducted,

- In the case when the single user is having multiple accounts then it is not easy for individual to remember all of the passwords.
- Individual may also can forget the password in the case when the account is not frequently in use.
- Generating easy to remember password is the one solution to the above problems but they easily guessable, hence there should be a specific process for defining the security of the system. There are several ways to create passwords but there are many challenges related to security and many of the challenges can be better resolved using the dynamic pattern based image password.[64][65]
- In majority of the password based techniques the password length is of eight characters which is quite hard to remember [66].

The graphical representation shown in Figure 4,5,6 depicts the way how the image patterns are used for the authentication of the web application and also includes the dynamic nature of the same.

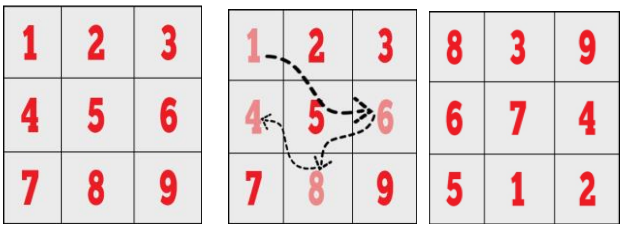

Figure 4: Image pattern-based passcode authentication, (a) shows the initial status of the pattern image, (b) pattern image after drawing the passcode, (c) pattern image after shuffling the numbering of the blocks.

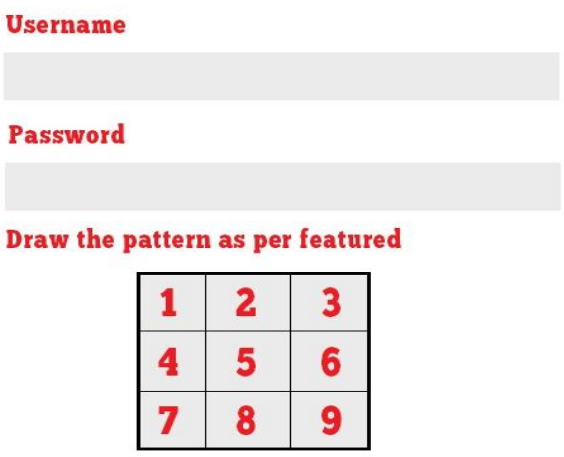

Figure 5: Representation of how the GUI of the web application will look like while requesting for access.

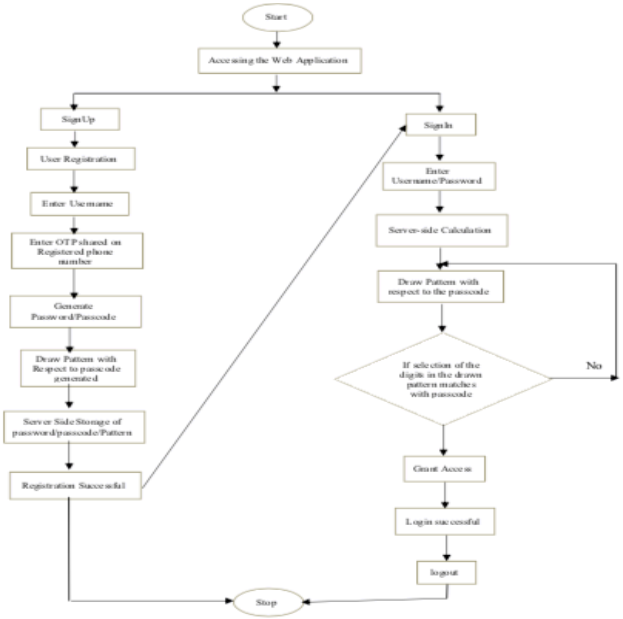

Figure 6: System Architecture of the proposed methodology.

b. Step by Step Execution of the proposed methodology:

The complete proposed system is categorizedinto two phases as signup phase and sign in phase as the sign up is mandatory for new user and already registered can directly go for sign in.

➢ **Phase 1: Sign Up**

Step 1: Open any of the web browser to be used for holding the web application,

Step 2: Drop a specific URL related to the web application,

Step 3: Select the sign up option available,

Step 4: Start registration process by filling the form for options like user name, and after which system will fetch the data related to the username entered and will share an otp to the registered phone number,

Step 5: Enter otp and if otp matches then generate a password and passcode, and if otp mismatches then simply a message will flashed that otp mismatch,

- Password is string with the combination of the numbers, characters, and special characters,
- Passcode is a four digit number generated by the user is available in the server space and no where else, and passcode can only be changed after completing the otp stage on registered number.

Step 6: An pattern image is presented to draw the pattern with respect to the passcode generated and if matches then proceed further,

Step 7: All of the data generated by the user for the entered user id will be saved at server location with proper encryption process and flash a message that the registration process is complete go with sign in to the web application.

➢ **Phase 2: Sign In**

Step 1: Fill in the form available with entries like username, and after which the system will fetch all of the data available related to the entered user id,

Step 2: Enter password and if password mismatches then simply deny the access and if matches then ask to draw the pattern with respect to the passcode generated and after every attempt the pattern image sequence will be dynamically updated using random shuffling technique, so that no one is able to predict the pattern drawn,

Step 3: If the pattern drawn matches with the passcode then grant the access to the user else deny the access to the user, else go back again to step to redraw the pattern for which new shuffled image is available to draw the pattern with respect to the passcode,

Step 4: Login successful message displayed and web application can used for further usage as per user requirements,

Step 5: logout and stop using the web application.

8. RESULT AND DISCUSSION

As per the technique discussed in the research methodology the things are implemented using the web development tool (php) for designing the front end visibility, as per the stages defined in the research technique the web application considers the things two different stages as registration and login. In both of the stages the security technique discussed in the paper is implemented and considered to process the authentication process. The work considers the pattern image which changes dynamically on every successful and unsuccessful attempt of login and have to be counted for the passcode and if the pattern considered matches with the passcode generated at the of registration then access is allowed else will be denied. In the complete process two stage security mechanism is considered as using the password and second using the image based dynamic pattern technique to ensure secure login. Below are the GUI images of the implementation:

In the stage 1 the user is asked to fill the registration form with basic information about the user account and user, like name, login id, password, passcode, etc. At the login time, users are asked to register to generate password and passcode differently and once the passcode is generated then it can only be changed on authentication using mail or mobile number, the passcode generated is accessed over the pattern image shown in figure 7 and figure 8.

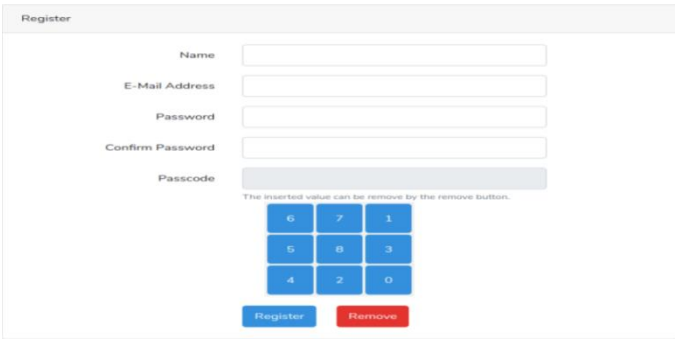

Figure 7: Image 1, Registration form.

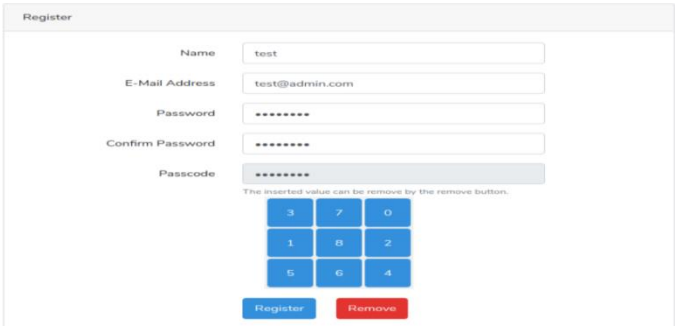

Figure 8: Image 2 showing the initial stage of registration with information fields for user.

Second stage of the technique (shown in figure 9) is login into the user account, as in previous stage all of the information filled is stored in server space allocated to the web application and all short of comparison is processed from the entered data and data available inside the storage.

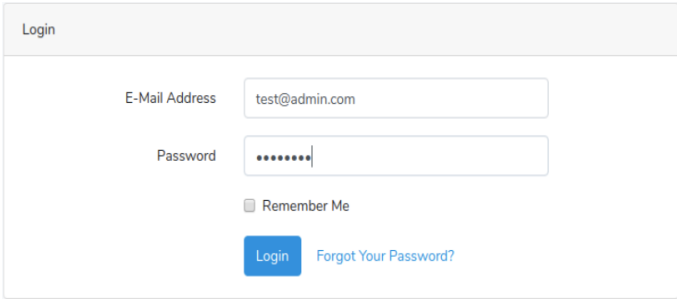

Figure 9: Image 3, representation of the stage 2 processing.

In the login stage the user is asked for first level authentication for which the user id is used authenticate the account for which data is to be fetched and then password is being used authenticate the user that the legitimate user is using the account and in the case when the password entered is not correct then the access is denied and also after unsuccessful 5 attempts the account is freezed from online platform for security reasons.

After successful authentication of the user using the password the user is asked for second level authentication where the user is needed to enter the passcode via pattern image shown in figure 10,11 which will generate a pattern type of image as output and the same will be matched with the data available in the background and if matches then access is granted else the user is asked again to enter the passcode via pattern image for which the pattern block position is randomly altered. The same makes the account prone to visibility type of attacks and also if the pattern is shared to attackers then it is quite hard for then to identify the passcode as for every attempt the pattern changes (as shown in section above).

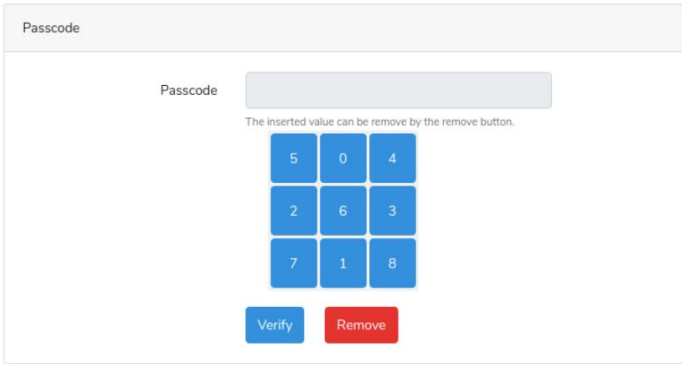

Figure 10: Image 4, enter the passcode via pattern image shown.

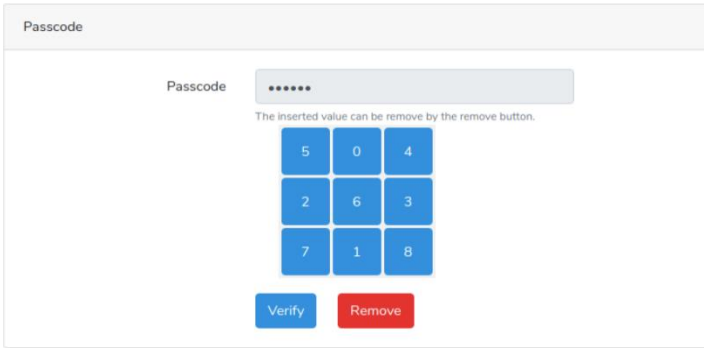

Figure 11: Image 5, showing the entry of six digit passcode.

After successful and unsuccessful attempt of passcode entry the image will look like image shown in the figure 12 given below:

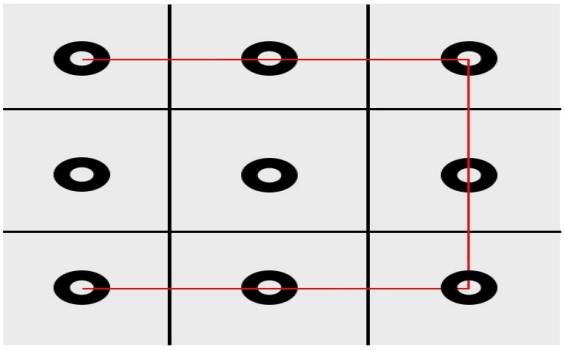

Figure 12: Image 6, representation of the pattern image.

The image show in the figure 12 represents the pattern generated after entering the passcode and the same is being stored in the database for comparison and also at the time of sharing the same over public network. The presented methodology is checked for efficiency and accuracy for many users and is found 100% accurate for most of the cases or even for almost all of the cases. The methodology picked can be further considered for real time representation considering the different factors of attacks and network over which the application is executed.

9. CONCLUSION

As can be seen that the major working related to finance, entertainment, social networking, and many more are available over the internet, which makes the things more prone to the attacks as the account over any of the digital platform is the identity of the individual which can be compromised in several ways. Hence the major part into this is to secure the access of the accounts related to the digital platforms like of banks and others. In the proposed methodology two level access authentication is being considered and phishing and visible type of attacks are considered. The first level security is achieved using the authentication password and in the second phase which is major in the technique discussed, considers the dynamic pattern image which works on the passcodes. The passcode is static in nature but the pattern image changes on every successful and unsuccessful attempts and clear consideration of the methodology is shown in the sections described above. The methodology makes the system more secure and authentication process is prone to the attacks which compromise the identity of the user. In the research methodology the only factor that remains un-touched is about consideration of the network issues and related aspects which can compromise the security of the web application for the future aspect the methodology can be further evaluated considering the various attacks and network related aspects.

REFERENCES

1. Verizonhttp://www.verizonbusiness.com/resources/reports/rp 2010-**data- breach-report** en xg.pdf.
2. K.Yanai, M. Shindo, and K. Noshita, "**A fast image gathering system from the World-Wide Web using a PC cluster," Image and Vision Computing**, Vol. 22, Issue 1, pp. 59-71, January 2004.
3. **Web Application Security Statistics**" http://projects.webappsec.org/w/page/13246989/ Web Application SecurityStatistics." ,2008,book.

4. W. G. Halfond, J. Viegas, and A. Orso, "**A Classification of SQL- Injection Attacks and Countermeasures**," in *Proc. of the International Symposium on Secure Software Engineering*, March 2006.
5. MySpace Samy Worm, "http://namb.la/popular/tech.html," 2005. A. Barth, J. Caballero, and D. Song, "**Secure content sniffing for web browsers, or how to stop papers from reviewing themselves**," in *Oakland'09: Proceedings of the 30th IEEE Symposium on Security and*
6. *Privacy*, 2009, pp. 360–371."**Gmail CSRF Security Flaw**, "http://ajaxian.com/archives/gmail-csrf-
7. security-flaw," 2007. M. Johns, "**Sessionsafe: Implementing xss immune session handling,"** in *ESORICS'06: Proceedings of the 11th European Symposium On*
8. A. Barth, C. Jackson, and J. C. Mitchell, "**Robust defenses for cross-site"***Research In Computer Security*, 2006.
9. N. Jovanovic, E. Kirda, and C. Kruegel, "**Preventing cross site request forgery attacks,"** in *SecureComm'06: 2nd International Conference on*
10. *Security and Privacy in Communication Networks*, 2006, pp. 1 –10. M. Johons and J. Winter, "**Requestrodeo: Client-side protection against session riding**," in *OWASP AppSec Europe*, 2006.
11. Z. Mao, N. Li, and I. Molloy, "**Defeating cross-site request forgeryattacks with browser-enforced authenticity protection,"** in *FC'09: 13 th International Conference on Financial Cryptography and Data Security*, 2009, pp. 238–255.
12. J. Bau and J. C. Mitchell, "**Security modeling and analysis**," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 18–25, 2011.
13. H. J. Wang, C. Grier, A. Moshchuk, S. T. King, P. Choudhury, and H. Venter, "**The multi-principal os construction of the gazelle web browser**," in *USENIX'09: Proceedings of the 18th conference on USENIX security symposium*, 2009, pp. 417–432.
14. R. Dhamija and A. Perrig, "Deja Vu: **A User Study Using Images for Authentication,**" in *Proceedings of 9th USENIX Security Symposium*, 2000.
15. A. Perrig and D. Song, "**Hash Visualization: A New Technique to Improve Real-World Security**," in *Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce*, 1999.
16. S. Akula and V. Devisetty, "**Image Based Registration and Authentication System,"** in *Proceedings of Midwest Instruction and Computing Symposium*, 2004.
17. D. Weinshall and S. Kirkpatrick, "**Passwords You'll Never Forget, but Can't Recall,**" in *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. Vienna, Austria: ACM, 2004, pp. 1399-1402.
18. W.Jansen,"**AuthenticatingMobileDeviceUsersThroughImageSelection,"**in*DataSecurity*,2004.
19. W. Jansen, S. Gavrila, V. Korolev, R. Ayers, and R. Swanstrom, "**Picture Password: A Visual Login Technique for Mobile Devices**," National Institute of Standards and Technology Interagency Report NISTIR 7030, 2003.
20. W. A. Jansen, "**Authenticating Users on Handheld Devices,**" in *Proceedings of Canadian Information Technology Security Symposium*, 2003.
21. H. Gao, X. Liu, S. Wang, and R. Dai**, "A new graphical password scheme against spyware by using captcha**.", in Soups, 2009.
22. I. Jermyn, A. J. Mayer, F. Monrose, M. K. Reiter, A. D. Rubin, et al., "**The design and analysis of graphical passwords."**, in Usenix security, 1999, pp. 1–14.
23. S. Ramanan and J. Bindhu, "**A survey on different graphical password authentica- tion techniques**", Volume-2, page, no. 7, 2014.
24. K. Renaud and E. Smith, "**Jiminy: Helping users to remember their passwords"**, in Annual conference of the south african institute of computer scientists and infor- mation technologists. saicsit, 2001, pp. 73–80.
25. Si Xiong. **"Web single sign-on system for wrl company".** Master's thesis, KTH Royal institude of Technology, June 2005.
26. M. E. Kabay. **"Identification, authentication and authorization on the world wide web".** http://www.mekabay.com/infosecmgmt/iaawww.pdf, 1997.
27. Asmawi, Aziah, et al. "**Model-based system architecture for preventing XPath injection in database-centric web services environment.**" *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*. IEEE, 2012.
28. Appelt, Dennis, et al. "**Automated testing for SQL injection vulnerabilities: an input mutation approach."** *Proceedings of the 2014 International Symposium on Software Testing and Analysis*. 2014.
29. Altmeier, Christian, et al. **"AdIDoS–Adaptive and Intelligent Fully-Automatic Detection of Denial-of-Service Weaknesses in Web Services."** *Data Privacy Management, and Security Assurance*. Springer, Cham, 2015. 65-80.
30. Vieira, Marco, NunoAntunes, and Henrique Madeira. "**Using web security scanners to detect vulnerabilities in web services."** *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. IEEE, 2009.
31. Antunes, Nuno, et al. "**Effective detection of SQL/XPath injection vulnerabilities in web services**." *2009 IEEE International Conference on Services Computing*. IEEE, 2009.
32. Antunes, Nuno, and Marco Vieira. **"Detecting SQL injection vulnerabilities in web services."** *2009 Fourth Latin-American Symposium on Dependable Computing*. IEEE, 2009.
33. Chonka, Ashley, Wanlei Zhou, and Yang Xiang. "**Defending grid web services from xdos attacks by sota."** *2009 IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2009.
34. Rajaram, A. Kanchana, and B. ChitraBabu. "**API based security solutions for communication among web services."** *2013 Fifth International Conference on Advanced Computing (ICoAC)*. IEEE, 2013.

35. Kumar, R. Kishore, R. Kanchana, and ChitraBabu. "**Security for SOAP based Communication among Web Services.**" *International Journal of Computer Applications* 975 (2013): 8887.
36. Pinzón, Cristian, et al. "**A Security Proposal Based on a Real Time Agent to Protect Web Services against DoS Attack.**" *Soft Computing Models in Industrial and Environmental Applications, 5th International Workshop (SOCO 2010)*. Springer, Berlin, Heidelberg, 2010.
37. Thomé, Julian, LwinKhinShar, and Lionel Briand. "**Security slicing for auditing XML, XPath, and SQL injection vulnerabilities.**" *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2015.
38. Muñoz-Arteaga, Jaime, Eduardo B. Fernandez, and HéctorCaudel-García**. "Misuse pattern: spoofing web services."** *Proceedings of the 2nd Asian Conference on Pattern Languages of Programs*. 2011.
39. Chan, Gaik-Yee, Chien-Sing Lee, and Swee-HuayHeng. "**Policy-enhanced ANFIS model to counter SOAP-related attacks.**" *Knowledge-Based Systems* 35 (2012): 64-76.
40. Chana, Gaik-Yee, Fang-Fang Chuaa, and Chien-Sing Leeb. "**Fuzzy association rules vs fuzzy associative patterns in defending against web service attacks.**" *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2015.
41. Padmanabhuni, Srinivas, et al. "**Preventing service oriented denial of service (presodos): A proposed approach.**" *2006 IEEE International Conference on Web Services (ICWS'06)*. IEEE, 2006.
42. Suriadi, Suriadi, et al. **"Defending web services against denial of service attacks using client puzzles."** *2011 IEEE International Conference on Web Services*. IEEE, 2011.
43. Suriadi, Suriadi, Andrew Clark, and Desmond Schmidt. **"Validating denial of service vulnerabilities in web services.**" *2010 Fourth International Conference on Network and System Security*. IEEE, 2010.
44. Pinzón, Cristian I., et al. **"S-MAS: An adaptive hierarchical distributed multi-agent architecture for blocking malicious SOAP messages within Web Services environments."** *Expert Systems with Applications* 38.5 (2011): 5486-5499.
45. Salas, Marcelo Invert Palma, Paulo Lício De Geus, and Eliane Martins. "**Security Testing Methodology for Evaluation of Web Services Robustness-Case: XML Injection.**" *2015 IEEE World Congress on Services*. IEEE, 2015.
46. Pinzón, Cristian, et al. "**Protecting web services against dos attacks: A case-based reasoning approach.**" *International Conference on Hybrid Artificial Intelligence Systems*. Springer, Berlin, Heidelberg, 2010.
47. Falkenberg, Andreas, et al. **"A new approach towards DoS penetration testing on web services."** *2013 IEEE 20th International Conference on Web Services*. IEEE, 2013.
48. Ficco, Massimo, and MassimilianoRak. **"Intrusion tolerant approach for denial of service attacks to web services."** *2011 First International Conference on Data Compression, Communications and Processing*. IEEE, 2011.
49. Gruschka, Nils, and Norbert Luttenberger. **"Protecting web services from dos attacks by soap message validation."** *IFIP International Information Security Conference*. Springer, Boston, MA, 2006.
50. Loh, Yin-Soon, et al. **"Design and Implementation of an XML Firewall."** *2006 International Conference on Computational Intelligence and Security*. Vol. 2. IEEE, 2006.
51. Zheng, Jun, and Ming-zeng Hu. **"Intrusion detection of DoS/DDoS and probing attacks for web services."** *International Conference on Web-Age Information Management*. Springer, Berlin, Heidelberg, 2005.
52. Mainka, Christian, et al. **"XSpRES-Robust and Effective XML Signatures for Web Services."** *CLOSER*. 2012.
53. Im, EulGyu, and Yong Ho Song. **"An adaptive approach to handle DoS attack for web services."** *International Conference on Intelligence and Security Informatics*. Springer, Berlin, Heidelberg, 2005.
54. Laranjeiro, Nuno, Marco Vieira, and Henrique Madeira. **"A learning-based approach to secure web services from SQL/XPath Injection attacks."** *2010 IEEE 16th Pacific Rim International Symposium on Dependable Computing*. IEEE, 2010.
55. Laranjeiro, Nuno, Marco Vieira, and Henrique Madeira. **"Protecting database centric web services against SQL/XPath injection attacks."** *International Conference on Database and Expert Systems Applications*. Springer, Berlin, Heidelberg, 2009.
56. Oliveira, Rui André, NunoLaranjeiro, and Marco Vieira. "**Assessing the security of web service frameworks against Denial of Service attacks.**" *Journal of Systems and Software*109 (2015): 18-31.
57. Rosa, ThiagoMattos, Altair OlivoSantin, and AndreiaMalucelli. **"Mitigating XML injection 0-day attacks through strategy-based detection systems."** *IEEE security & privacy*11.4 (2012): 46-53.
58. Patel, Vipul, Radhesh Mohandas, and Alwyn R. Pais. "**Attacks on web services and mitigation schemes.**" *2010 International Conference on Security and Cryptography (SECRYPT)*. IEEE, 2010.
59. Siddavatam, Irfan, and JayantGadge. **"Comprehensive test mechanism to detect attack on Web Services."** *2008 16th IEEE International Conference on Networks*. 2008.
60. Tao, Zhao. **"Detection and service security mechanism of xml injection attacks."** *International Conference on Information Computing and Applications*. Springer, Berlin, Heidelberg, 2013.
61. Ye, Xinfeng. **"Countering DDoS and XDoS attacks against web services."** *2008 IEEE/IFIP International*

Conference on Embedded and Ubiquitous Computing. Vol. 1. IEEE, 2008.

62. Gupta, AbhinavNath, and P. SanthiThilagam. **"Detection of XML signature wrapping attack using node counting."** *Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16')*. Springer, Cham, 2016.
63. Pauli Kaila. **from end-to-end to trust-to-trust. in proceedings of the seminar on net- work security,** pages 18–22, 2008.
64. Vipin Samar. **Single sign-on using cookies for web ap- plications.** In WETICE '99: Proceedings of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises, pages 158–163, Washington, DC, USA, 1999. IEEE Computer So- ciety.
65. A. Basso, S. Sicco, **"Preventing massive automated access to web resources,"** computers & security 2 8 (2009) 17 4 – 188. Elsevier; 2008.
66. Basso, M. Miraglia**," Avoiding massive automated voting in internet polls,"** STM2007. Electron. Notes Theor. Comput. Sci.2008; vol. 197(2). Elsevier
67. A Survey on Recognition-Based Graphical User **Authentication Algorithms FarnazTowhidi Centre for Advanced Software Engineering,** University Technology Malaysia Kuala Lumpur, Malaysia
68. Susan Wiedenbeck Jim Waters,**",Authentication Using Graphical Passwords: Basic Results** ,College of IST Drexel University Philadelphia, PA, 19104 USA.
69. G. Agarwal, S. Singh and R.S. Shukla,**"Security Analysis of Graphical Passwords over the Alphanumeric Passwords by G. Agarwal, 1Deptt.of Computer Science"**, IIET, Bareilly, India 2,3 Deptt. of Information Technology, IIET, Bareilly, India 27-11-2010.
70. Varsha, Ravichandra Mouli & Kp, Jevitha. (2016), "**Web Services Attacks and Security- A Systematic Literature Review. Procedia Computer Science**," 93. 870-877. 10.1016/j.procs.2016.07.265.