ISSN 2047-3338

Towards a Perspective of Web Application Vulnerabilities and Security Threats

Nana Yaw Asabere[1] and Wisdom Kwawu Torgby[2]

[1, 2] Computer Science Department, School of Applied Sciences and Arts, Accra Polytechnic, Accra, Ghana

[1]yawasabere2005@yahoo.com, [2]torgby@gmail.com

***Abstract*—The rapid and tremendous growth of Information and Communication Technology (ICT) has increased access to web applications. This increased access has paved the way for disadvantageous security and vulnerable threats in the form of attacks in web applications. Various detection and prevention techniques have been proposed by researchers in the field of web applications and technologies development. Through relevant literature and existing research, this paper presents a viewpoint of different web application vulnerabilities and security threats and also outlines some open research issues in accordance to the state-of-the-art.**

***Index Terms*- Attacks, Security, Threats, Vulnerabilities and Web Applications**

I. INTRODUCTION

THE sudden development and expansion of web technology, has resulted in an upsurge increase of web application vulnerabilities and attackers intending to cause harm are all the time making every effort necessary to obtain unfair advantage of these applications. During this period where web applications are readily available, attackers now find it easy to violate such systems to cause problems and to steal [1].

The rapid growth of the internet World Wide Web (WWW) has led to a situation where more and more people are demonstrating increased interest using web applications. This as a result, has led to direct increase in the frequency of security violations for web applications. At present web application security is getting more prominent attention than before; this is due to the fact that criminal attackers are continuously striving to expose the holes in these applications and try to take advantage of them to their benefit [2].

Regrettably, many web applications have certain characteristic vulnerabilities and security design flaws, at the moment the protection of data and web applications from well thought-out and calculated criminal attacks is getting much more attention, there is no denying the fact that web applications have become the criminal attackers prime object of attack where efforts are directed towards infiltrating and taking advantage of the vulnerabilities present. The real problem is that, more often than not developers of these applications centre all their attention on getting a working application program developed under time constraint and as a result do not put into operation the best security practices. Developing a web application is a very intricate and complicated process; hence it is easy for developmental mistakes to become loopholes where attackers can easily take advantage of to violate the security of web applications technology [3].

In this paper, we present, through literature and existing research the current state-of-the-art of different web application vulnerabilities and security threats.

The paper is presented in five sections. Section II presents a Literature Review of the subject matter, Section III presents Related Work and State-of-the-Art, Section IV presents Discussion, Research Challenges and outlines some Open Issues. The paper is finally concluded with a recommendation in Section V.

II. LITERATURE REVIEW

A. What is Web Application?

At the outset of web development, web pages presented only information that appears in a form which does not change, in a sense that the information remained the same and that no changes could be made. During those early periods of web development it was perhaps possible that hyperlinks were integrated into websites, for direction to other web pages making it easy for end-users to access information from different websites. At those early times, websites did not provide any means of making it possible for end-users to interact with it and thus users have no need to enter data, because the sites did not provide any mechanism for doing so. The improvement of web technology has led to a situation in which a variety of different ways of implementing websites, and new programming languages to the web applications have been developed thus making it possible for web pages that were newly created to have features such as [1]:

- Command buttons
- Text boxes,
- Radio buttons,
- Check boxes and
- Drop-down lists

As a result of such innovations, it is now possible for the end user to enter data, information, and issue commands to a website, this is what is referred to as a dynamic website. A dynamic website describes a situation where the end user and the system interact. Due to the concept of creating dynamic websites it is now feasible to transform, append, make improvements, or make corrections to a site. The basic operation of a website is such that once the user inputs information to the web page, the browser which is the software then loads the page, this is as a result of the deeds and activities of the user visiting the web site. A dynamic website can also be defined as software which characteristically comprises of scripts that changes on constant basis, typically it stays on a web application server and interacts with databases and other sources of dynamic contents and makes it possible for needed changes to be made as and when required. Generally speaking, some of the types of web applications used are webmail, shopping carts and portals. The web applications make it possible for lots of people to have the right to use the internet regardless of where the user is situated in the world [1].

B. Fundamental Architecture

The fundamental architecture of a web application comprises of two elements; that is the organizational structure and the programming language used in designing the applications. The organization of web applications usually involves many levels and the applications can be written in a number of different web programming languages [1].

C. Number of Levels

The constitution of a web application architecture is made up of many levels, what this means is that a number of layers or levels are needed to build a web application and is normally made up of at least three (3) different layers, these are namely [1]:

- The presentation tier or generally what is referred to as the front-end or the graphics user interface,
- The applications tier (business logic) such as the Java Servlets, or Active Server Pages,
- The data tier more often than not referred to as the back end.

1) Presentation Tier

This section of the web application is generally referred to as the Graphical User Interface (GUI). It is this segment of the web application that makes it possible for the end-user to have direct communication with the system and the browser is the software that makes it possible for data and information to be displayed. This is the level that provides a physical medium of connection which makes it possible for the user to interact and communicate with the application server. One of the roles of the presentation tier is to change data in its raw state into human readable form for the end-user.

2) Applications Tier

The applications or business logic tier is that part of the web application architecture where the codes that are used in programming the application is stored, one of its main functions is to operate as if it is the brain of the application where all the commands and instructions are executed, in other words it is that part which handles the logical reasoning. This is the tier that hosts the intelligence of the application. This segment of the web application is where issues such as business rules, data manipulation are defined and other activities take place.

3) Data Tier

The data tier section is very significant because it is that part of the web application architecture that takes care of data storage; this section basically makes it possible for web applications to store data using the Database Management System (DBMS) and uses database creation languages such as:

- MYSQL,
- ORACLE
- MS-ACCESS etc.

The data tier also serves as a medium that makes it possible for the end-user and the data to interact and communicate. Sometimes it is possible to design an application devoid of the data tier; however the advantage and functionality of such an application would experience a drawback in terms of how that application operates. This view has in recent times been extended to include alternate sources of data like XML files and Lightweight Directory Access Protocol (LDAP)[1]. Usually the web browser software at the start sends a request to the application tier; the application tier in turn accesses the database to execute the required task, in this instance it can get back the information from the database, transform it, make needed changes or update the information.

In view of the fact that web applications resides on the web server and they could be updated, improvements can be made or transformed at any point in time without any distribution or setting up on a client's machine, this is the rationale behind the extensive implementation of web applications in many organisations all over the world.

D. The Framework of Web Applications: (Programming Languages)

There are varieties of different programming languages that are used to design web applications; these languages enables web developers to deal with a wide variety of data and make it possible to manage information in a dynamic way. These languages make it possible to write very uncomplicated or complex codes to perform particular functions on the server. When a user usually pays a visit to a website the entire request for a particular activity are processed by the generated codes stored in the web server. The following are brief descriptions of some of the languages used to dynamically design a web site [1]:

1) HTML

HTML stands for Hypertext Mark-up language. It is made

[1] http://www.ietf.org/rfc/rfc2251.txt

up of tags which embeds the text of a document. It is the browser (software) that reads the document and gives interpretation of the mark-up tags to assist in formatting the document for display to the reader. It should be noted that web browsers are available for a wide variety of different computer systems. Consequently the browser displays the document with respect to characteristics that the viewer selects. Features affecting the layout and appearance include:

- the mark-up tags used,
- the physical page width,
- the fonts used to display the text and
- the colour depth of the display.

2) DHTML

Dynamic Hypertext Mark-up Language (DHTML) is an extension of HTML, the idea is to make it a bit more dynamic. The use of DHTML has typically been associated with having the functions of the GUI.

3) XML

XML stands for eXtensible Mark-up language; it is a mark-up language that is very much like HTML. It was designed with the express purpose of carrying data but not to display the data in any form. With XML you must define your own tags; there are no already defined tags to be used. It is designed in such a way that it is self-explanatory and descriptive. It is used in many different areas of web development normally to make the process of storing and sharing data very simple. XML was not designed as a replacement for HTML; they were both developed with different goals in mind. XML on the other hand was developed to transmit and store the data with the attendant result, depending on what the data is. HTML was developed to show the data with the prime focus on how the data looks.

4) XHTML

XHTML, which stands for eXtensible Hypertext Mark-up Language, was developed to solve a situation where the HTML has been put together in a different way as a result of HTML being loose in implementing its inherent structural organisations. One of the prime reasons why XHTML was developed was to bring in imposed structure to the HTML. This led to a situation where XHTML document became stricter and cleaner than HTML. As a result of the development of XHTML, issues with respect to cross-browser concerns have started declining. The coming into being of XHTML has brought in its wake the development of device independence that ensures compliance to the rules set forth by The World Wide Web Consortium (W3C).

5) PHP

Hypertext Pre-processor (PHP)[2], is a general purpose web coding language that in particular is appropriate for the development of the web, this language can be fixed into HTML. It was in the beginning designed to produce dynamic web pages and websites. Before you can use PHP, its interpreter must be available on the web server in order for its pages to serve out. It actually supports many database creation languages such as MYSQL, Informix and Oracle etc. It can be sourced openly and is freely obtainable, it's mode of operation is such that usually the page is sent to the browser, the server then calls PHP to interpret and carry out the operation that is called for in the PHP code.

6) Java Server Pages (JSP) [3]

This is a web technology developed by Sun Microsystems, it is Java based and was designed to implement applications dynamically. It has a rich array of functions, tags and tag support. The development of technology has led to a very simple and speedy way of creating websites dynamically. Comparing Java Server Pages to other languages like ASP and PHP, JSP involves a lot more in terms of processing.

7) ColdFusion MX

This programming language enables designers to create very powerful web applications. The language requires less and smaller amount of training and coding skills unlike other languages such as JSP, ASP and PHP. A ColdFusion application is very simply a collection of pages, similar to a static web site. Cold Fusion is renowned for quick development of web applications[4].

E. Web Applications Architecture

The diagram below (Fig. 1) shows the flow of information within a web application, the user normally would the load browser into the memory of the system, the system then makes a call for the web page from the application server, as a result the application server subsequently will call for information from the database, and the data is then sent back through the application and causes it to be in the browser by the HTML code.

F. Web Applications Security

Security from the angle of web applications is incorporated to ensure that data and resources that dwell on the web server are protected from unauthorized users The resources being referred to in this instance can be items physical in character, such as a web page, customer database, or could be stuff less physical in character, for example status and name of an organizations [4].

Web application security from another standpoint is all about trust, i.e. users and organisations would have the faith and confidence that the necessary data and resources committed to a web server will be secure and safe devoid of those not authorized to get access to such vital resources.

In view of the fact that today there are criminal attackers with intent to cause harm to these applications hovering around websites and making every effort to steal very important information, regulations and standard policies have been designed to help protect and safeguard the confidentiality, integrity and availability of data that is accessible through the World Wide Web.

Web application security is all about ensuring that the associated risks are managed in such a way that the impact of violating these applications will be reduced to the barest

[2] http://www.php.net

[3] http://www.java.sun.com/products/jsp

[4] http://www.macromedia.com/software/coldfusion

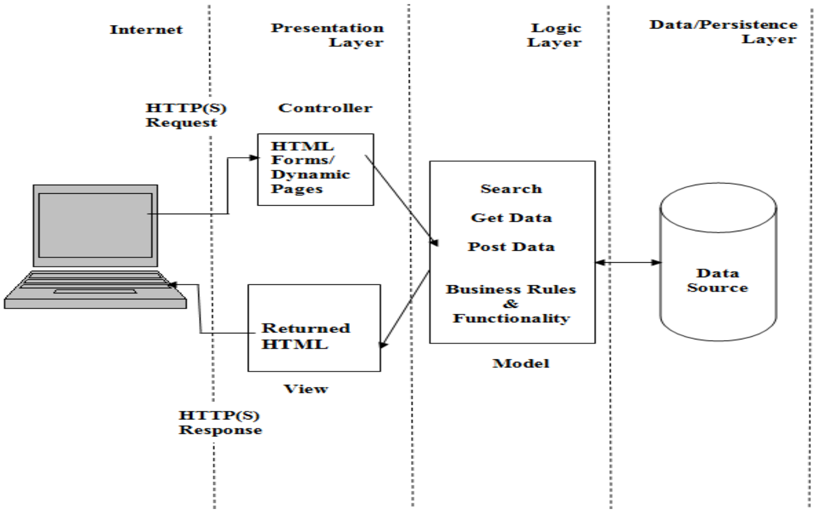

Fig. 1: Pen Testing For Web Applications [1]

minimum, or in other words putting into operations useful and efficient countermeasures to overcome any security breaches that may occur [4].

G. Fundamentals of Web Applications Security

In order to ensure that web applications are secured, there are certain fundamental measures that need to be undertaken to make the security of the applications possible. Enumerated below are some of the procedures and measures that need to be implemented to ensure security and safety of web applications [5]:

1) Authentication

The term authentication means establishing in a particular way whether or not someone or something is in actual fact what it is intended to be in the real sense of the word. With respect to web application, authentication is a mechanism that is used to check whether a user is in actual fact the person he is believed to be, and this is usually done by the use of login password.

Under normal circumstances, once the user knows the password, then it is thought that, that user is the genuine or the real person who is assumed to be the owner of the password. Sometimes certain situations can arise where the password can be stolen, revealed inadvertently to someone not authorized or to totally forget it, this has proven to be the ‘achilles’ heel of authentication.

2) Authorization

The term authorization describes a process whereby a user of a system is granted approval or denied the right to use a resource in a network. In a computer system where there are multiple users, the systems administrator usually determines which users have the right to get access to the system; the administrator in this instance determines the privileges of use, for example which files to have access to, how many hours of access is allowed to use the file, the amount of storage space and other rights as and when it becomes necessary.

3) Auditing

Auditing is a checking mechanism put in place to keep a record of transactions that occurs on a daily basis in a computer system, for instance it shows who has accessed the system at a particular point in time, the sort of processes undertaking within a certain period of time. These are normal procedures instituted to keep track of every activity or transaction which goes on in the system. Database Management Systems and Accounting Information Systems incorporate checking audit procedures. Additionally there are auditing software’s that makes it possible for network administrators to keep an eye on the usage of network resources.

4) Confidentiality

The term confidentiality also describes a process which ensures that data, information and resources that stay on a web server are secured and restricted, meaning that only those who have authorized permission are allowed to view and use the available resources, anyone who is not authorized is denied the right to view or use the resource. Putting in such actions and measures ensures that the data, information and the resources stay private and confidential. One other procedure used to put confidentiality in operation is encryption.

Encryption is the process of altering or converting data that is in plain text to what is termed as cypher text or data that is not human readable, by instituting such measures it ensures that the data or information remains private. Using access control lists is also one of the methods of putting into operation effective confidentiality.

The use of access control in this case actually implements a mechanism of defining which user has the right to use an application or resource at a particular point in time; it can be directly built into the application or in other instances provided from outside the application.

5) Integrity

This term integrity (pledge) is used to describe a situation where data and resources are kept safe from harm or injury by malicious cyber attackers who try to either change or modify the resources or data that is on a web server. Integrity in relation to web application also means that the data or information that is in transit gets to the intended recipient in whole without any alteration. Usually when data is in transit, hashing techniques and message authentication codes are used to establish the integrity of the data to prove that the data being transmitted is valid and genuine. Listed below are a few of the threats to data integrity [6]:

- Mistakes caused by users during the process of data input.
- During the course of transmitting data from one system to the other, mistakes that crop up as a result of the transmission.
- A mistake in the software program.
- Failure of the computer equipment to function properly, for instance crashing of hard disk.

The following are some of the means by which dangers to data integrity can be reduced:

- Make duplicate of original data on a regular basis.
- Put in place security measures to determine who has access to data at every point in time.
- Craft user interfaces in such a way that invalid data will not be added to the system.
- During data transmission, softwares that can discover and correct errors can be used to improve data integrity.

6)Availability

Availability from the view point of web application security describes an instance where a system remains accessible to those who are rightfully authorized to use it. Many cyber attackers use denial of service techniques to deny a system of available resources, their end objective is to try and break-down an application or crush it with the intention of overpowering the system so that the user cannot have access to the application. Normally, most of the time, criminal attackers with wicked intent use denial of service attacks which tries to munch through all of a website's available resources of the system like the memory space, storage space, the Central Processing Unit (CPU) and so on and so forth. As soon as one of these crucial resources is made full use of by the denial of service attack, the web site will more often than not be reachable or available to those not authorized to use the system [5].

H. Web Application Vulnerabilities

Vulnerability is an imperfection or deficiency in a web application that exposes the likelihood of the application being attacked by people with malicious intent to cause harm to the system [7]. The vulnerability could happen as a result of:

- poor crafting of the web application,
- mistakes made when configuring the application,
- inappropriate coding,
- and coding methods that are not secure.

During the process when the web application is being crafted, where there is no strong input validation mechanism in place there is every possibility that the web application can be subjected to input attacks by malicious attackers whose main objective is to put the application in harm's way, cause havoc and to steal sensitive company information.

The main challenge facing many business organisations with respect to web application vulnerability for the most part is not the risky security flaws in the web application infrastructure, neither is it the viruses or the worms, it is also not based on the well-known weaknesses in application servers, rather weaknesses in the applications themselves. The weaknesses which are peculiar to each application creates a situation where the web application infrastructure of many business organisations become open to malicious attacks by criminal hackers [5].

I. Countermeasures

Countermeasures are protective mechanisms or techniques instituted to discover, prevent, or reject web attacks. The essential countermeasures in a web application should be implemented to make sure that the web application is given the needed protection against ordinary types of web attacks depending on the dangers that faces the application. A mistake or the absence of the required countermeasure will lead to a vulnerability that makes the web application liable to attacks. The countermeasure basically ensures that the dangers associated with attacking the applications are lessened or reduced to the barest minimum [8].

The Open Web Application Security Project (OWASP) has come out with an outstanding catalogue of the web application vulnerabilities, which is called the OWASP Top Ten 2010[5]. Web Application Vulnerabilities. Web attackers frequently take advantage of these application vulnerabilities by digging out data that is sensitive from databases of companies. The OWASP Top Ten 2010 Application Security and Vulnerabilities are as follows:

- A1 - Injection Flaws
- A2 - Cross-Site Scripting (XSS)
- A3 - Broken Authentication and Session Management
- A4 - Insecure Direct Object References
- A5 - Cross Site Request Forgery (CSRF)
- A6 - Security Misconfiguration
- A7 - Insecure Cryptographic Storage
- A8 - Failure to Restrict Uniform Resource Locator (URL) Access
- A9 – Insufficient Transport Layer Protection
- A10 – Invalidated Redirects and Forwards

The OWASP Top Ten 2013[6] Application Security and Vulnerabilities is a released candidate only intended for comments.

III. State-of-The-Art And Related Work

In this section, we present some related work involving the state-of-the-art web application security threats and vulnerabilities. Kong et al. [9] proposed a lightweight static

[5] www.owasp.org/index.php/Top_10_2013-Main

[6] www.owasp.org/index.php/Top_10_2013-T10

analysis approach to detect logic vulnerabilities in Java Web Applications. Their core idea was to discover deviant behaviors among duplication samples. Program slicing technique was leveraged in to extract duplicated invocations targeting similar functionalities. Subsequently, path exploration was conducted to split slices into several path sensitive slices. Then a comparison was made between any two similar slices on their path condition. The slices with abnormal path condition were reported as logic vulnerabilities.

They implemented their approach in a prototype tool named Logic Vulnerability Detector (LVD), and evaluated it on seven real world applications scaled from thousands to million lines of code. The evaluation results in [9] showed that their approach achieved bigger coverage with acceptable cost and better scalability than previous approaches.

Avancini and Ceccato [10] resorted to a search based approach for security testing web applications. They took advantage of static analysis to detect candidate cross-site scripting vulnerabilities. Input values that expose these vulnerabilities were searched by a genetic algorithm and to help the genetic algorithm escape local optima, symbolic constraints were collected at run-time and passed to a solver. They implemented their approach in a prototype and evaluated it on real world PHP code.

Using the web-based tool proposed in [11], the authors were able to evaluate eleven server packages for Apache, PHP and MySQL across three operating system platforms. The evaluation in [11] revealed that the proposed web-based tool was able to audit current security configuration settings and alert users to fix the server environment to achieve the level of safety of security configuration with respect to recommended configurations for real-life web application deployment.

Takamatsu et al. [12] proposed a technique that automatically detects session management vulnerabilities by simulating real attacks. The proposed technique in [12] requires the test operator to only enter a few pieces of basic information about the web application, without requiring a test environment to be setup or obtaining detailed knowledge of the web application. The experiments in [12] demonstrated that their technique could detect five web applications deployed in the real world.

Tian et al. [13] focused on the regression test in web vulnerability detection and presented a strong-association rule based algorithm to make the detection more efficient. In the first step they traversed the whole web site to get the web page collection. Then, in the regression test, they made the association between the pages and expanded the pages to a collection set. Finally, [13] conducted an experiment on their target web site which contains the known vulnerabilities such as XSS and SQL injection, and the result showed that their algorithm can detect almost all the pages that may contains vulnerabilities in their target website.

Holm et al. [14] study the effort that is required by a professional penetration tester to find input validation vulnerability in an enterprise web application that has been developed in the presence or absence of four security measures: (i) developer web application security training, (ii) type-safe API's, (iii) black box testing tools, and (iv) static code analyzers. The judgments of 21 experts were collected and combined using Cooke's Classical Method. The results in [14] showed that 53 hours is enough to find vulnerability with a certainty of 95% even though all measures have been employed during development. If no measure is employed, 7 hours is enough to find a vulnerability with 95% certainty.

Fonseca et al. [15] described a set of tools for implementation in their proposed methodology. They allow the automation of the entire process, including gathering results and analysis. They used specified web-based tools to conduct a set of experiments to demonstrate the feasibility and effectiveness of their proposed methodology. The experiments include the evaluation of coverage and false positives of an intrusion detection system for SQL injection and the assessment of the effectiveness of two web application vulnerability scanners. Results in [15] showed that the injection of vulnerabilities and attacks is an effective way to evaluate security mechanisms and tools.

Monga et al. [16] presented a hybrid analysis framework that blends together the strengths of static and dynamic approaches for the detection of vulnerabilities in web applications: a static analysis, performed just once, was used to reduce the run-time overhead of the dynamic monitoring phase. They designed and implemented a tool, called Phan that is able to statically analyze PHP bytecode searching for dangerous code statements; then, only these statements are monitored during the dynamic analysis phase.

As web applications generally adopt input validation and sanitization routines to prevent web security risks, Shar et al. [17] proposed a set of static code attributes that represent the characteristics of these routines for predicting the two most common web application vulnerabilities—SQL injection and cross site scripting. The experiments conducted in [17] showed that vulnerability predictors built from the proposed attributes detected more than 80% of the vulnerabilities in the test subjects at low false alarm rates.

In order to detect known attacks in web applications, some set of attack rules and detections are needed. In [18], a negative security model based on misuse of web applications was used. This negative security model provides a Web Application Firewall (WAF) engine with a rule set, to ensure critical protection across every web architecture. WAFs were deployed to establish an increased external security layer to detect and/or prevent attacks before they occur. The authors in [18] successfully tested almost all the common attacks using their proposed model in an apache web server's log file.

IV. Key Research Challenges, Discussion And Some Open Issues

Through a reflection from Section III and an observation from [2], this section outlines and enumerates some key research challenges, discussion and open issues.

A. What Drives And Pushes Cyber Attackers To Do What They Do?

For every endeavor in life there be must be a compelling desire to undertake a particular course of action. Getting to understand and appreciate what motivates web attackers to do what they do may well lead to knowing about their mindset thereby helping find solutions to stop or minimize such attacks. For purposes and objectives of this paper, enumerated below are some of the reasons that inspire web attackers to get involved hacking web applications and these reasons are key challenges in the field of web applications and vulnerabilities.

1) Ever Present

One of the most important reasons that motivate cyber attackers to do what they do is because web applications are ever present everywhere today; it is easily spread across both private and public networks. Due to the ubiquitous nature of these applications they have become simple and easy targets for cyber attackers.

2) Simple Techniques

As a result of uncomplicated techniques used in attacking web applications it is quite straightforward for persons without the requisite professional qualifications to easily understand the attack techniques since for the most part they are text-based, making it rather inconsequential to control the input of web applications. For instance, when designing a buffer overflow as a case in point, attacking web applications is easy and a child’s play. In contrast to the above, the deep knowledge that is necessary to breach more complex operating systems or applications should be the focus of organisations.

3) Anonymity

It is imperative to appreciate the fact that web application attacks can easily be started without anyone being able to trace where the attack commenced. Usually the attacks are often routed through open Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) proxies. These proxies make it quite difficult for anyone to trace where a request is coming from. This is a major reason for the increase in malicious attacks of web applications, given that this anonymity takes away one of the most important deterrents of being caught and imprisoned, this is what drives attackers to pursue their criminal actions.

4) Evade Firewalls

It is fairly easy for web attackers to very simply bypass firewalls. HTTP and HTTPS that are inbound are normally allowed by most firewall policies. As a matter of fact, this is not a weakness of the firewall, but a policy which is configured by the administrator of a system. What will happen is that as more and more applications move towards HTTP this configuration is most likely to experience an increase in frequency, and this will suit web attackers even more.

5) Custom Code

With the increase usage of web applications for business transactions, a situation has arisen where most developers of these applications have little or no past experience. This is because the technology that drives the web applications are so easy and simple to understand that developers of these applications are not given any proper training in terms of security. The standards of entry for building web applications are quiet low making it easy for all manner of people to start developing these applications without formal training.

The languages used to develop these applications are easy to understand, and the speed with which these applications are developed is also a cause for concern, for instance they just copy and paste codes from websites and books. There is virtually lack of internal coding values that guides developers on how to apply the best security principles. The securities of these applications are not yet well developed and are still in a daily evolution process.

6) Continuous Change

One other problem is that, there are people such as developers, systems administrators, and content managers etc. who are also continuously making changes to these applications. It should be noted that many of these people do not have the requisite security training; nonetheless they have the power and authority to make changes to the applications on a constant basis thereby leading to compromising the security of these applications.

B. Attacking Web Applications: The Mode of Operations

Cyber criminals who attack web applications have numerous different ways and means by which they initiate such attacks. They usually choose the type of attack for the most correct part or most fitting to a particular weakness to the application. The attackers are very methodical in their approach of attacks. They follow an organised plan of activity. They try to gather as much information as possible about their targets. The following are some of the reasons why attackers gather information about their targets.

- As attackers collect information about their targets, it helps them to put together all the appropriate principal facts they need to be able to commence attacks. The more familiar they are with the targeted website or application, the more successful they become in pursuit of attacking their targets.
- For cyber attackers to be successful in achieving their goal, the attacker must get to know, be familiar with and intimate with the exposure points of their targets. They gather all the intelligence necessary thereby getting to know much about the weak exposure points of their targets and then take appropriate steps to take advantage of the vulnerabilities.

The following are some of the steps they embark on in achieving their objectives by way of information gathering.

1) In any normal circumstance the first thing an attacker will do, will be to learn about the infrastructure of the web server and the type of operating system that powers the server. They conduct an in depth study of the properties that are essentially present in both the infrastructure and the operating system. They use scanning tools that enables them to find out all the HTTP and HTTPS ports which are opened, and then pinpoint the particular port that needs striking.

2) An attacker also undertakes a careful test of the website as well as web application with an intended goal of finding out whether there are any weaknesses that can be used as a means of violating the system. Usually they also take time to look at every constituent part available on a web page or website to see if they have any chance of gaining entry to the web server. Cyber attackers also take the trouble to scrutinize the authentication and login pages for an opportunity to get the right to use to the web server. The attacker on the other hand also makes effort to examine the code written for a web application to look for any weaknesses during the process of development and see if it can be ripped.

3) One other thing that attackers would normally also look for is the existence or the availability of input validation. Input validation is a mechanism that is built-in into web application which is used to establish whether a particular data that has been inputted is safe and secure and validated. Usually when the data is entered and is not safe, it is discarded and further processing takes place. In situations where developers of web applications are negligent in incorporating proper and effective security measures, attackers can easily send harmful inputs to the web server to cause problems.

Having gone through the process of doing all the discovery work to gather the required information, from the web server to web application and making known the weak target areas of the application, the attacker will then initiate and launch the attack to overpower the system and begin taking advantage of the system's vulnerabilities.

V. Conclusion

Through relevant literature and related work in the state-of-the-art, this paper presented a viewpoint of web application vulnerabilities and threats and also outlined some key challenges and open research issues in the field of web application vulnerability and security threats. The paper generally discussed details of what a web application is, its architecture, security of web applications, the framework of web applications, what motivates hackers to do what they do and the impacts of web attacks in business organisations. As a future work, this paper recommends that continuous research should be conducted on particular web application vulnerabilities and attacks such as Structured Query Language Injection (SQLI). Correspondingly, the measures that need to be incorporated during the web applications development process should further be examined before implementation.

References

[1] A. Andreu, "Pen Testing for Web Application," *Indianapolis, Wiley Publishing Incorporated*, 2006.

[2] J. Scambray, M. Shema and C. Sima "Hacking Web Applications Exposed," *2nd ed. San Francisco, McGraw-Hill*, 2006.

[3] M. Andrews and J. A. Whitaker, "How To Break Web Software," *1st ed. Addison Wesley*, 2006.

[4] S. Simmons "Hacking Techniques: Web Application Security," *East Carolina University*, 2005, Available [Online] *http://www.infosecwriters.com/text_resources/pdf/HackingTechniques_WebApplicationSecurity.pdf* [Accessed 31/03/2013].

[5] J.D. Meier A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla and A. Murukan, "Improving Web Application Security: Threats and Countermeasures," 2003, Available [Online] From *http://msdn.microsoft.com/en-us/library/ff649874.aspx* [Accessed 03/04/2013]

[6] Data Integrity, Available [Online] From: *http://www.webopedia.com/TERM/D/data_integrity.html* [Accessed 05/04/2013]

[7] Web Application Vulnerabilities, Available [Online] From: *http://www.webopedia.com/TERM/vulnerability.htm* [Accessed 06/04/2013]

[8] Countermeasure, Available [Online] From: *http://www.webopedia.com/TERM/c/countermeasure.htm* [Accessed 06/04/2013]

[9] Y. Kong, Y. Zhang, Z. Fang and Q. Liu, "Static Detection of Logic Vulnerabilities in Java Web Applications," *in Proceedings of 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1083-1088, Nat. Comput. Network Intrusion Protection Center, GUCAS, Beijing, China, 2012.

[10] A. Avancini and M. Ceccato, "Security Testing of Web Applications: A Search-Based Approach for Cross-Site Scripting Vulnerabilities," *in Proceedings of the 11th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pp. 85-94, 25-25 Sept. 2011.

[11] B. Eshete, A. Villafiorita and K. Weldmariam, "Early Detection of Security Misconfiguration Vulnerabilities in Web Applications," *in Proceedings of the IEEE 6th International Conference on Availability, Reliability and Security (ARES)*, pp. 169-174, 22-26 Aug. 2011.

[12] Y. Takamatsu, Y. Kosuga and K. Kono, "Automated Detection of Session Management Vulnerabilities in Web Applications," *in Proceedings of the 10th Annual International Conference on Privacy, Security and Trust (PST)*, pp. 112-119, July, 2012.

[13] H. Tian, J. Xu, K. Lian and Y. Ying, "Research on Strong-Association Rule Based Web Application Vulnerabilty Detection," *in Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology, (ICCSIT)*, pp. 237-241, 2009.

[14] H. Holm, M. Ekstedt and T. Sommestad, "Effort Estimates of Web Application Vulnerability Discovery," *in Proceedings of the 46th IEEE Hawaii International Conference on System Sciences (HICSS)*, pp. 5029-5038, 7-10 Jan., 2013.

[15] J. Fonseca, M. Vieira and H.S. Madeira, "Vulnerability Attack and Injection for Web Applications," *in Proceedings of the IEEE/IFP International Conference on Dependable Systems and Networks*, pp. 93-102, June 29 -July 2, 2009.

[16] M. Monga, R. Paleari and E. Passerini, "A Hybrid Analysis Framework for Detecting Web Application Vulnerabilities," *IWSESS '09 Proceedings of the ICSE Workshop on Software Engineering for Secure Systems, Pages 25-32, IEEE Computer Society*, Washington, DC, USA, 2009.

[17] L.K. Shar and H.B.K. Tan, "Predicting Common Web Application Vulnerabilities from Input Validation and Sanitization Code Patterns," *in Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE)*", pp. 310-313, 2012.

[18] M. Auxilia and D. Tamilselvan, "Anomaly Detection Using Negative Security Model in Web Application," *in Proceedings of the IEEE International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, pp. 481-486, 8-10 Oct. 2010.

Nana Yaw Asabere received his BSc in Computer Science from Kwame Nkrumah University of Science and Technology (KNUST), Kumasi, Ghana in 2004 and MSc in ICT from Aalborg University, Denmark in 2010. He has nine (9) years of teaching/lecturing experience at the tertiary level of education in Ghana and is currently on Lectureship Study Leave granted by Accra Polytechnic, Ghana pursuing his PhD in Computer Science at School of Software, Dalian University of Technology, Dalian, P.R. China. Nana Yaw has a number of publications to his credits in International Journals and his research interests include: Artificial Intelligence (AI), Software Engineering, Expert Systems, Mobile Learning, E-learning, ICT in Education, ICT for Development, Information Systems, Multimedia, Recommender Systems, Social Computing, Wireless/Data/Mobile Communication and Computing Technologies.

Wisdom Kwawu Torgby received his MSc in IT from De Montfort University (DMU), Leicester, UK in 2009. He has fourteen (14) years of teaching/lecturing experience at the tertiary level of education in Ghana. Wisdom’s research interests include: Computer/Information/Network Security, Web Applications Development and Security, Network Software, E-Government, E-Learning and Software Development Using Web Programming Languages. He is currently a Lecturer, Academic and School Board Representative as well as an Examinations Officer of the Computer Science Department of Accra Polytechnic, Ghana.