I. J. Computer Network and Information Security, 2019, 11, 46-59
Published Online November 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2019.11.06

Delivering a Secured Cloud Computing Architecture and Traditional IT Outsourcing Environment via Penetration Tools in Ghana

Umar Sayibu
Department of Mathematics/ICT, Bagabaga College of Education, Tamale, +233, Ghana
E-mail: sumar.sayibu2@gmail.com

Frimpong Twum
Department of Computer Science, KNUST, Kumasi, +233, Ghana
E-mail: {second.author} twumf@yahoo.co.uk

Issah Baako
Department of Mathematics/ICT, Bagabaga College of Education, Tamale +233 Ghana
E-mail: {third.author} issahbaako@gmail.com

Received: 30 August 2019; Accepted: 23 October 2019; Published: 08 November 2019

Abstract—The decision to use either Cloud Computing (CC) applications or Traditional Information Technology Outsourcing (Traditional ITO) environments is a function of the security evaluations of these two options. Hackers are constantly nosing around websites and other computer networks for compromised computers that have some vulnerabilities to exploit them. Vulnerabilities in cloud computing and Traditional ITO environments are leading causes of recent data breaches. These breaches provide opportunities to hackers to attack and gain access to customer information such as credit cards and contact information, passwords, sending of malicious codes to website users or making users computer potential candidates of botnets and to hijack the sessions of authentic users to make unapproved purchases on their behalf. In this paper, security penetration tools have been employed to evaluate the security vulnerabilities of cloud-based solutions and Traditional ITO to discover possible vulnerabilities, their causes and mitigation strategies to securing web applications from the discovered vulnerabilities. Some web applications and a Traditional ITO network were ethically hacked to discover vulnerabilities in them. Analyses of the results obtained through the ZAP scan flagged Remote File Inclusion (RFI) alert were high priority alert. In all, RFI constitutes the most serious potential threat and it needs the fullest attention of CC service providers. Nmap disclosed opened ports in Traditional ITO Virtual Private Network which can make the server of the provider accessible to hackers leading to a considerable disclosure of information to unauthorized users.

Index Terms—Vulnerabilities, Web-based Applications, Traditional ITO, Cloud Computing, Virtual Private Network.

I. Introduction

Over the years, some business organizations in Ghana have relied heavily on Traditional ITO or CC (two outsourcing options) to meet their individual IT needs. While Traditional ITO refers to contracting outside professional services to meet specific local business IT needs, CC allows users to store and access data and programs via the Internet instead of a hard drive which is managed by third parties. However, in terms of choice and implementation, organizations are skeptical of the security evaluation of the two outsourcing options for they fear adversaries will attack their websites and other computer networks for compromised computers that have some vulnerabilities to exploit them.

Vulnerabilities in web applications and network servers are leading causes of recent data breaches. These breaches provide opportunities to hackers to attack and gain access to customer information such as credit cards and contact information, passwords, sending of malicious codes to website users or making users computers potential candidates of botnets and to hijack the sessions of authentic users to make unapproved purchases on their behalf.

[1] Recently, web applications have become very widespread due to the ubiquity of Internet. They are online applications with enormous paybacks coupled with risk to the end-users, brand and data. Intruders continuously launch attacks to take advantage of the vulnerabilities in these applications to gain unauthorized access to mission-critical information.

Besides, changes in the web has changed the threat landscape. The web browser has become a powerful ecosystem on the client-side which attackers abuse to

penetrate the privacy of victims and hijack the sessions of authentic users to make unapproved purchases on their behalf. The resourcefulness of the web has made the server-side a shared podium to continuously replace legacy applications which have become very expensive for intruders to exploit. This has made old-fashion security measures such as firewalls and intrusion detection systems ineffective to detect or prevent web application vulnerabilities.

Web applications are designed using a variety of technologies including JavaScript among others which are used by many people across the globe for their services. The design flaws or vulnerabilities at the design stage in the technologies cause security breaches resulting in the stealing of user's sensitive information [2].

Most of the research conducted in the area of CC focused only on the technology [3] and other studies have recently discussed CC's business implication concerning how an organization can take advantage of it and the associated risks [4,5]. Similarly, a good number of researches have been conducted on CC but the studies do not indicate clearly how CC differs from the Traditional ITO [6,]. At best, the studies on CC only examined the concept of CC separately with little or no regard to the huge literature in Traditional ITO. Besides, [7] came out with determinant factors to consider in vendor selection of CC and ITO but not in the aspect of their security evaluation.

[8] confirmed that the Traditional ITO theories were valid for the CC sourcing decisions too contrary to the security evaluation of the two concepts. An insignificant number of studies only differentiate CC from ITO as outsourcing concepts and not in terms of their security challenge evaluation. This is the knowledge gap that this study identified as a research opportunity and sought to address. Specifically, the following question was addressed: *How can the implementation of a penetration testing tool be used to deliver a secured Cloud Computing Architecture and also a secure Traditional IT Outsourcing environment?*

In this paper ZAP and Nmap which are penetrating testing tools were used to discover vulnerabilities in Cloud-based applications and the Traditional IT Outsourcing environment to address the research question. This paper emphasized the need to deliver secured CC architecture and Traditional ITO via penetration testing tools to discover vulnerabilities, their causes and suggest remedies since discovering alone is not enough.

Other parts of this study are organized as follows: Related work, Materials and Methods, Results and Discussion, Conclusion, Acknowledgement and References.

II. Related Work

CC as a form of outsourcing option allows resources to be delivered as a service to its clients as a utility service through the internet where customers just pay for only what they have used. This method of service delivery renders CC susceptible to several security attacks such as the Login page and cloud APIs.

Web application security concerns the security loopholes found in web applications during security breach analysis or after implementing security applications. It is, therefore, very useful to conduct a thorough security evaluation throughout the application development Life Cycle especially at the initial stages to avoid financial loss to any organization intended to use the application by using penetration testing tools. A detail description of Penetration testing is provided in [9].

Analysis of Cloud Computing Attacks and Countermeasures was conducted in [13] where a framework of penetration testing was carried out to discover possible vulnerabilities in a CC infrastructure and simulated attacks for the exploitation of identified vulnerabilities which included Denial of Service and Man-in-the- Cloud attacks. Their research concluded data breach, fraud identification, compromising data integrity and confidentiality were possible.

A feature of CC that allows data and storage to be outsourced to a third party makes it vulnerable to most security threats as well as vulnerabilities. According to [10] Vulnerabilities refer to the weaknesses of an Information System which could be a web application, a device or component used in a network perimeter. These vulnerabilities engender security threats the consequences of which are unintended outcomes and unknown and more potentially damaging effects. In their report, they concluded web application trajectory is significant and a major threat to the security of any organization whether they are security conscious or not

Even though some of the security challenges found were not new to the computing concept, they were of serious concern to CC environment according to [11] due to features such as virtualization, multi-tenancy, data and sharing of resources which are unique to CC.

Further survey on a well-known framework for cloud security- ENISA, CSA AND NIST to obtain a list of risks, best practices and vulnerabilities provided detailed insight into decurrent privacy, security and trust issues concerning the CC or Web applications current security status. Several privacy and security issues linked to CC were identified including vulnerabilities such as Legal and Compliance issues, Virtualization and Hypervisor Security, Network Security, Data and Storage Security and Identity and Access Management. A research conducted to evaluate OWASP top 10 security threats using W3af, Skipfish, and OWASP ZAP which are vulnerability scanners on a vulnerable application-DVWA, it concluded testing and analysis of these scanners using different parameters, OWASP ZAP showed better results [12].

Some studies show that recent web applications have become very popular and are utilized in very critical security environments like the medical field, military installations, and financial systems. Their applications have heightened sophisticated attacks on them. A counter-approach of attacks to these web applications attacks is the detection and blocking of these attacks by applying application-level firewalls and vulnerability

analysis techniques to identify potential security issues before usage [13]. In this paper, vulnerabilities in cloud-based solutions that pose potential threats to clients of CC and Traditional ITO have been discovered using penetration testing tools with detailed discussions on the causes, prevention and mitigation of these vulnerabilities.

III. Materials and Methods

Methodology is the set of rules, procedures and practices that are followed in the implementation of any information security evaluation environment. Hence, a methodology used in a penetration testing is the roadmap for practical strategy and time-tested practices followed to evaluate the actual security status of an application, a network, a system or a combination of these [14]. Penetration testing involved is the process of evaluating the security issues within a target's perimeter using simulation technique by attackers to discover vulnerabilities that could be exploited by a hacker. Penetration testing involves a detailed analysis of the system designs, technical flaws, configuration and weaknesses. Largely this is done to check the extent of information disclosure and other system flaws that attackers can take advantage of and further make the system vulnerable to external attacks [15].

There are three different penetration testing methodologies which include: White-box penetration testing where the tester possesses an in-depth knowledge and access to the information system to be tested; a Block-box penetration testing which is an approach in which only special information about the system is given to the tester and Gray-box penetration testing approach which provides the tester with limited information about the system [14]. This study utilized Gray-box penetration to answer the research question to avoid biases. A meticulous framework is provided in Fig.1. for conducting comprehensive and genuine penetration testing.

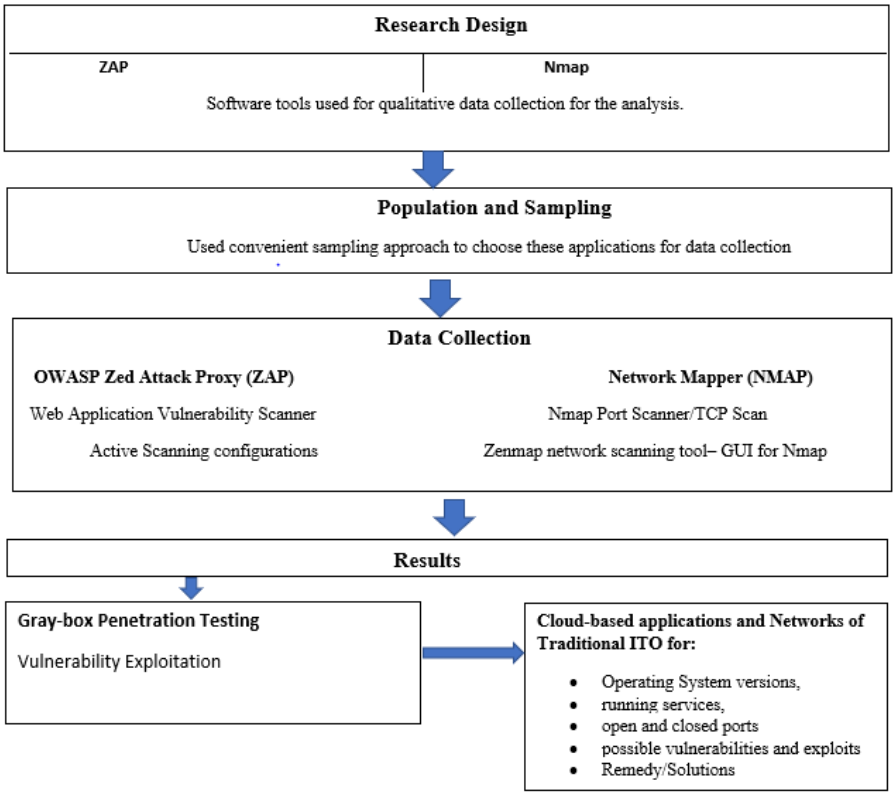

Fig.1. Framework of Methodology

A. Research Design

In terms of primary data, qualitative data will be acquired via ZAP and Nmap software tools for the collection of data required for the analysis. Nmap was a tool used to discover vulnerabilities associated with Network Perimeter such as detecting hosts that are alive and other details namely: the Operating System, open ports and the services being ran [14]. The study discovered vulnerabilities within web applications. Both applications are considered to be an easy to use integrated web application and networking security penetration testing tools that effectively find vulnerabilities in web application [17. 10] and networks. Interestingly, they are easy to use open-source software.

B. Population and Sampling

There are about 30 IT outsourcing companies in Ghana and 7 of them are cloud computing companies including mobile networking companies This number was arrived

at via a search on the internet. Again, 2 well-known cloud computing service applications which are used by some Ghanaians and many people around the world. Besides, these applications are easily accessible and can easily be scanned to get real web-based application effects. This accessibility explains why the researcher used convenient sampling to choose these applications for data collection. Again, the URL of a Ghanaian Traditional ITO provider was scanned in addition to the two applications. This small number of companies or application to be examined does not only make it more manageable but also few companies use cloud computing in Ghana.

According to [18], a large sample size guarantees accurate reflection in the target population. However, care should be taken not to allow the large sample size to negatively affect the quality of the findings. As the exact number of companies in Ghana that utilize IT outsourcing is not known scientifically, the researcher used the following criteria to select credible companies for the study:

- Already utilizing Traditional ITO or making attempts to switch to CC services
- Using CC solutions
- Provide services for outsourcing.

These criteria for selection provided valid and reliable findings that were generalized to all organizations engaged in either providing Traditional ITO or CC outsourcing services.

C. Data Collection

The collection of information from the entire list of relevant sources to find answers to a research question, testing of a hypothesis as well as the evaluation of the results. The sources here are primary and secondary resources of information.

1. OWASP Zed Attack Proxy (ZAP)

This study collected data using ZAP and Nmap for analysis. This tool was chosen because [18] concluded it had better results compared to Skipfish and W3af after testing and analyzing these scanning tools. All available links on the target sites were browsed, filled forms and the values submitted. After browsing all the available links, the spider option was used to crawl through other links automatically by selecting the Spider site option. This was done to ensure that the spider automatically discovers the hidden links.

2. Active Scanning

The Active Scan was used to scan the web application. An Active scan is a tool that attacks the application in all possible ways to discover all available vulnerabilities.

3. Active Scanning

The Active Scan was used to scan the web application. An Active scan is a tool that attacks the application in all possible ways to discover all available vulnerabilities.

4. Network Mapper (Nmap)

Nmap is an open-source network mapping scanning tool used by attackers to explore and discover servers and their associated services. It has a Lua-based script engine feature that is capable of rapidly detecting vulnerabilities over a network thereby positioning you ahead of any security threat [19]

5. Nmap Port Scanner/TCP Scan

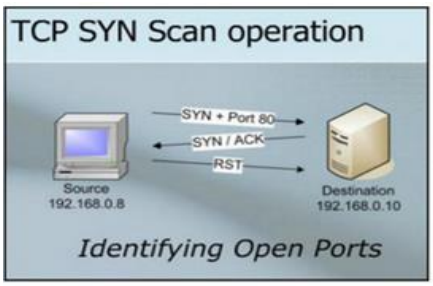

Fig.2. Nmap port scanner/TCP Scan

Nmap port scanner/TCP Scan is activated when an SYN link is established on every port on the target host. Open ports will respond with SYN-ACK. This connection by the initiator when he sends a reset (RST) [14]. If the port is not closed, the host gives feedback with a connection reset [14].

Vulnerabilities within a network perimeter which has the potential of information disclosure were discovered with their causes such as application flaws and misconfigurations, whether the Operating Systems are up-to-date and their versions and suggest preventive measures as well as solutions using the Common Vulnerability Exposure (CVE) data source. CVE offer definitions for cybersecurity, exposures, and vulnerabilities that are publicly disclosed and also suggest remedial actions.

IV. Results and Discussion

A. Overview

The adoption of CC and ITO has been challenged by the security issues in these fields despite the economic benefits that come with them. Organizations fear to lose their data and reputation due to trust issues. Several organizations are concerned with how to make their critical IT needs secure in their quest to evaluate the security of CC and Traditional ITO with the third party in terms access, reliability, availability as well as performance. In this paper empirical study was carried out and discovered vulnerabilities as alerts. Alerts are the vulnerabilities that have the potential of exploits by attackers. They were categorized according to their risk levels. High priority alerts pose serious threats compared to other defined alerts whilst the informational alerts pose little threats.

Fig.3. provided a summary of the scan results. The summary of the alerts is put in a tabular form with the

alert of high priority being the first one. This means is the most serious potential threat which deserves serious attention and the least serious is the informational one which was not found in the scan. The "Alert Detail" section of Fig.1. offered the name and a detail description of the alerts that were discovered during the scan. It also gave suggestions or solutions to the alerts that have been discovered. The alerts are summarized in the following tables and further explained in detail in the sections that follow.

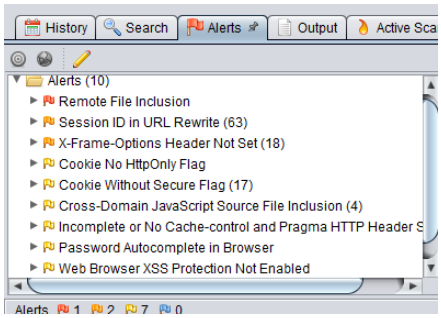

Fig.3. ZAP Scanning Alerts

Table 1. High Priority Alerts

No	*Name of vulnerability*	*Number of vulnerabilities*
1	Remote File Inclusion	1

Table 2. Medium Priority Alerts

1	*Session ID in URL Rewrite*	*107*
2	*X-Frame-Options Head Note Set*	*9*

Table 3. Low Priority Alerts

1	Cookie No HttpOnly Flag	1
2	Cookie Without Secure Flag	19
3	Cross-Domain JavaScript Source File Inclusion	4
4	Incomplete or No Cache-Control and Pragma HTTP Header Set	77
5	Password Autocomplete in Browser	1
6	Web Browser XSS Protection Not Enabled	2
7	X-Content-Type-Options Header Missing	52

B. Evaluating Vulnerabilities of Cloud Application Using Penetration Testing

Two web applications that are used to store documents were tested using ZAP to discover vulnerabilities in them. Fig.3. summarizes the alerts that were discovered. The results in Fig.3. is a grate concern to patrons of such application. These vulnerabilities are discussed in detail with a special attention to Remote File Inclusion (RFI) due to the severity of RFI as shown in the Fig.3. Other alerts would be discussed as well.

1. Remote File Inclusion (RFI)

RFI is a vulnerability used by intruders to attack their targets in a web application reference independent or external script dynamically. The intention of the perpetrator is to take advantage of the referencing function within an application to introduce a malware remotely from a URL in dissimilar domain [20]. When successful, the RFI attack can lead to information theft, taking over the site which can result in content modification. Servers can also be comprised.

Most cloud-based application design support file inclusion. This mechanism is used strictly for integrating similar code into different files that the main application modules later reference. When an include file is referenced by the web application, the code contained in the file could be executed with a call to specific procedures. The web application could be prone to RFI if the module to be used to load depends on elements from the HTTP request.

Attackers use RFI for the following:

* To run harmful code on the server causing the server to run every code within the "include malicious" files. To prevent the code in include executing in the context of one using the server, then the file includes be executed by applying a wrapper. This will avoid an entire system compromise. To try to manipulate the responses intended for the subscribers running malicious code on them. The attacker can achieve this when malicious code is embedded in the response that a client would run. For example, JavaScript could be used to deny the client a session cookie. File include is extensively used in PHP during programming thereby making it susceptible to RFI attacks as well as server configuration (default) the heightened the vulnerability to RFI attacks. Fig.4. below typifies the flow of an RFI attack followed by an algorithm for the attack.

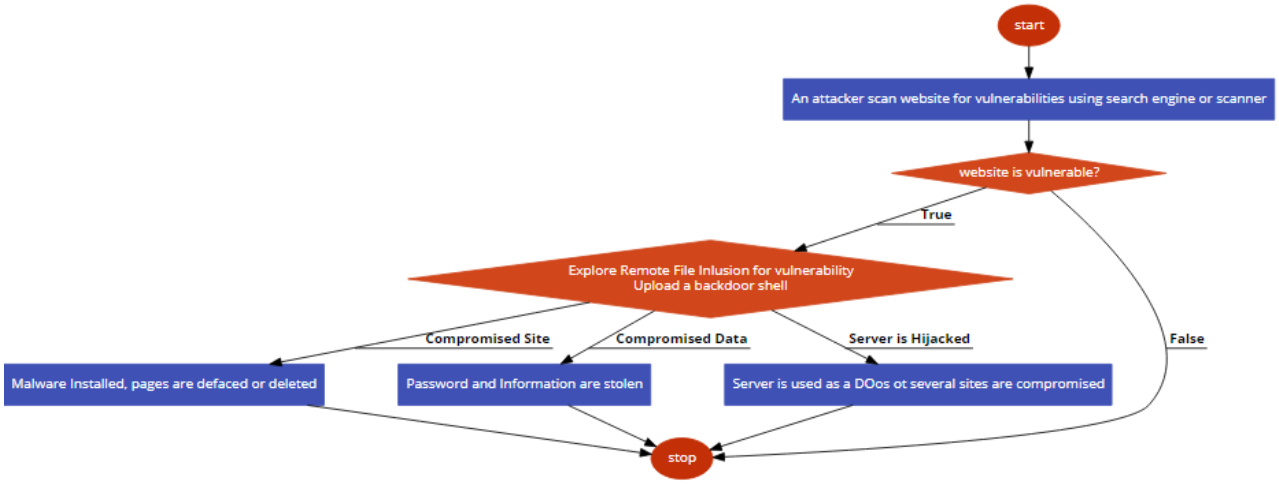

Fig.4. An RFI attack flow

start;
An attacker scan website for vulnerabilities using a search engine or scanner;
if (website is vulnerable?) {
switch (Explore Remote File Inclusion for vulnerability
Upload a backdoor shell)
{
case Compromised Site:
Malware Installed, pages are defaced or deleted;
break;
case Compromised Data:
Password and Information are stolen;
break;
case Server is Hijacked:
Server is used as a DOos at several sites are compromised;
break;
}
}
stop;

RFI attacks are usually carried out when the value of a request parameter is set to a harmful code. For example,

```
$incfile = $_REQUEST["file"];
include($incfile.".php");
```

The value of the file parameter request is extracted from the HTTP request by the code in the first lie. The value that has been extracted is used by the second line to set the file name to be included dynamically. The value of the file parameter should be sanitized appropriately by the web application by checking against a whitelist to avoid the code from being exploited.

Using this URL for example:

http://www.target.com/vuln_page.php?file=http://www.attacker.com/malicous

It implies the file include name will resolve to:http://www.attacker.com/malicous.php

This will include the remote file and the server will run any file within it.

If the *register_globals* variable is set to *On*, it will extract request parameters implicitly. It will then make the code below vulnerable to similar attacks.

```
include($file.".php");
```

Remedy
Architecture and Design

When the group of acceptable objects like filenames or websites (URLs) is known or limited, a map from a collection of fixed input values, for instance, numeric IDs to the real websites or filenames while ignoring all other inputs. For instance, IDa could be mapped to “inbox.txt” while IDb could be mapped to “profile.txt”. ESAPI Access Reference Map is a feature that provides such capability.

A code can be run in a “jail” or sandbox similar to a jail environment that can restrict the boundaries between OS (operating system) and the process. This process effectively puts restrictions on files that can be accessed in a specified directory or commands that can be executed by software.

Examples of the operating systems are AppMrmor, UNIX chrootjail as well as SELinux. Generally, codes that are managed could protect. For example, the Java Security Manager contains java.io.FilePermission enables you to apply specific restrictions on file operations.

This does not necessarily provide a perfect solution but minimizes the impact on the operating system. The application could still be subject to attacks. As much as possible avoid CWE-243 and some other related jails’ weaknesses.

Languages such as PHP has an interpreter that has a restriction like open “basedir” or when in a safe mode can offer better restrictions to avoid hacker attacks of the application. Again, Suhosin is a hardened PHP extension contains several options that may disable harmful features in PHP

Consider every input as a malicious one. In this case, the input validation strategy of “accept known good” is

applied. Create a whitelist of all inputs that are acceptable and meet restriction specifications. Don't accept any input that does not such specifications or can be transformed into anything that does. Don't exert your effort solely in searching for malformed or malicious inputs (that is, depend solely on a blacklist). Blacklist is, however, very useful in detecting possible attacks or identifying inputs that are malformed and should not be accepted.

All possible relevant properties should be considered when carrying out input validation. For example, length, conformity to business rules, a complete range of acceptable values, input types, syntax, consistency over related fields and extra or missing inputs. In terms of validity, "boat" could be syntactically valid as an instance of business rule logic since it contains alphanumeric characters. However, when colors such as "blue" or "red" are expected, it will not be valid.

The Stringent whitelist can be used for file names to limit character collection intended for use. If possible, use only a single "." character within the filename to eliminate weaknesses like CWE-23 and isolate directory separators like "/" to eliminate CWE-36. Again, to avoid CWE-434, apply a whitelist of acceptable file extensions. For secure content, put session ID in a cookie. To be even more secure consider using a combination of cookie and URL rewrite.

2. X-Frame-Options Header Not Set

Table 2. showed two alerts and X-Frame-Option Header Not Set is one of the two medium alerts. The exclusion of X-Frame-Options header will not protect '***ClickJacking'*** attacks in HTTP response. This vulnerability determines the permission a browser has to load in <frame>, <iframe> or <object>. Setting it will block the content of websites from diversion to malicious sites where clickjacking can occur. Attackers use clickjacking to lure unsuspecting visitors to websites they have no intention of visiting where they ask them to click attractive links like 'CLICK ME', and other interesting links. This action deletes all messages with loaded iframe onto email. When successful, the attacker hijacks all clicks that are due for their websites and transfer them to different sites which other programs and domains own. Keystrokes are hijacked by attackers using the same strategies to deceive users to believe that passwords typed into hidden emails are their accounts or bank accounts controlled by these attackers.

Web applications are exploited in this manner. Modern browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use ***SAMEORIGIN***, if you don't to frame your page, then use ***DENY. ALLOW-FROM*** enables websites to frame the web page in supported web browsers). Besides, one can send the appropriate X-Frame-Options HTTP response headers to block browsers to external domains and use defensive codes to ensure that the topmost level window is the frame used.

3. X-Content-Type-Options Header Missing

Table 3. showed 7 low priority alerts including X-Content-Type-Options Header Missing. Browsers handle files differently depending on whether it is a text file or an image file. Microsoft has a built-in feature in Internet Explorer which tries to examine the right content form, whether or not it has been specified by the webserver. MIME Sniffing is the feature. Like the two sides of a coin, this feature enables end-users to browse the web very successfully and also triggers an attack vector.

The flaw that was revealed in the attack is the missing "x-content-type-options header". This means that Anti – MIME Sniffing header x-content-type-options failed to set to 'nosniff'. In this case, some old versions of browsers such as Explorer and Chrome are capable of performing MIME-sniffing on the response body which eventually leads to the decoding of the response body and display as a content type rather than the content type that has been declared.

When the web application server sets the "Content-Type header" correctly, it will eventually set the "X-Content-Type-Options header" to 'nosniff' for the entire web pages.

Guarantee the end-user employs the latest browser that complies with standards and will never perform MIME-sniffing or will prevent web servers from performing MIME-sniffing. Error type pages such as 401, 403, 500, etc are still affected by MIME-sniffing for injections issues affect those pages which create a lot of concern for browsers that sniff pages away from their real content type. The following sub-sections explained in detail the rest of the alerts in Table 3.

4. Cookie No HttpOnly Flag

A cookie is a text file that is left by browsers on computers that contain information about a particular website that has been visited by a user. In other words, it is a message that is contained in a browser from a server that is usually sent back to the server anytime the browser needs a page from the server. JavaScript can access a cookie that has not been set to HttpOnly flag.

The HttpOnly Flag is added to a Set-Cookie HTTP responses header in a browser that is supported to generate a cookie reduces the risk of access being gained by the client-side script of a protected cookie. Any browser that can support HttpOnly can detect a cookie that contains the HttpOnly flag and can prevent a client-side script code from trying to read access the cookie and returns give back an empty string as the result thereby attacking to fail by preventing dangerous code from being sent to the criminal's website.

If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible. However, setting the HttpOnly attribute on a cookie implies its value is not readable and cannot be set by a client–side JavaScript and renders it hard for attacks like cross-site scripting which usually capture the cookie's value through an injection of a script.

One does not easily find the reason for which the HttpOnly Flag should not be set on cookies if any. Developers may intentionally allow client-side scripts in their applications to access the cookie's value. However, for best practice, HttpOnly flag attributes should be set in a significant cookie directive. This does not guarantee HttpOnly restriction to provide all the security. Potential circumvention and very serious attacks are possible by client-side script injection in addition to cookie theft. When HttpOnly is set on a cookie, the instruction to the browser is that it is only the server that can read the cookie's value and not the client-side scripts. This is a serious and important security measure for session cookies. Ensure that the HttpOnly flag is set for all cookies

5. Cross-Domain JavaScript Source File Inclusion

In this case, the page includes one or more script files from a third-party domain. In other words, a JavaScript Source file is included to implement attacks in which malicious scripts are injected into websites that are trusted when the attacker decides to use a dangerous code via a web application such as a browser to other websites of end-users. This is made possible where a web application makes use of an input which generates output from a user and does not validate or encode it.

Even though in their quest to provide rich and very current information from sources across the globe by developers, security and web professionals are also skeptical about the potentials of this development in crippling the internet, by making it more prone to exploits and attacks by criminals. It is not the request to third-party domains which is worrying, but an attacker can use the request as coming from a trusted user. The user unsuspectedly executes the script which can access cookies and tokens of sessions or even very sensitive information could be retained by the application or browser which is eventually used.

There are three types of Cross-site Script (XSS). ***These include Stored XSS, Reflected XSS and DOM XSS***. The Stored XSS is also known as Persistent or Type I, is triggered by storing user input on the target server which in this case could be in a database. A victim could retrieve such unsafe data like message fora, visitor's log, and comment fields and load it in a browser. The Reflected XSS also called the Non-Persistent or Type II occurs as a result of an immediate return of input by a web application as error message, search results or responses which incorporates inputs that have been provided by the user as a request which is not safe to be loaded in the browser and does not permanently store the user's data. The DOM Based XSS or Type-0 is where the source and destination of the malicious code or data are in the same domain. The source could be the page's URL or HTML element and the destination could be the method that triggers the execution of such malicious code.

To avoid this problem, ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end-users of the application. Again, domain requests could be prevented by browsers imposing restrictions on cross-domain source file inclusion. A policy that prevents communication of dissimilar domains known as "Same Origin Policy is being implemented by many browsers. However, the details and the exact manner in which the same origin is being implemented is beyond the scope of this research.

6. Password Autocomplete in Browser

Authentication is a very significant aspect in terms of the security of an application since it is the key that opens up conversations. Once a user is authenticated it starts the security of the transaction and goes further to confirm that the user is who he/she claims to be.

Most developers include features to ease the effort exerted by users to complete web forms, ensuring secure connectivity, management of web pages and fixing of bugs in web applications. All these are to help the user to remember passwords with ease, URLs, completion of web forms that contain personal data. This capability also generates security and vulnerability issues. Besides, the data browsers save and manage are very sensitive as they contain some of form of data that have been autocompleted in the form of phone numbers, emails, addresses, user names, browsing history of previous visits that have been collected with URLs describing the personal experiences in relation to user information requirements including Autocomplete passwords which provide access to web resources that have been restricted as revealed in the dropbox attack. If the AUTOCOMPLETE attribute is not disabled on an HTML FORM/INPUT element containing password type input. Passwords may be stored in browsers and retrieved. What is disheartening is that passwords are not given strong protection in several instances.

To avoid this security challenge, turn off the AUTOCOMPLETE attribute in forms or individual input elements containing password inputs by using AUTOCOMPLETE='OFF'.

7. Web Browser XSS Protection Not Enabled

Web Browser XSS Protection is not enabled or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web-server. Cross-Site Scripting (XSS) is the means of injecting code to perform a task in a user's browser for a website. The action performed could be seen or hidden in the background. The attack in the cloud application showed that it is not protected from such malicious code from executing. As mentioned earlier, there are many categories of XSS vulnerabilities but the common ones are ***Reflective XSS and Persistent XSS.*** The former is usually within an HTTP query parameter used to parse and display results of a page for the user by server-side scripts. The latter is actual data of an attacker saved on a server which is displayed to the user as if it were a normal page.

Cross-Site Scripting (XSS) is an issue that results from poor data/code isolation. The main concern here is that a developer might want the user's input interpretation as data. On the contrary, manipulation by an attacker can lead to the browser interpreting the input as commands or

tags. The X-XSS-Protection header is built to enable cross-site scripting filters which are built-in current web browsers to enforce it. The browsers that support this are Safari, Chrome and Explorer version 8 and beyond. It is recommended that the header should be configured to a value that will enable the XSS protection and require of the browser not to allow a response from malicious scripts that have been injected from the user's input and should not attempt to sanitize it. The syntax should be as follows:

x-xss-protection: 1; mode=block

Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.

The X-XSS-Protection HTTP response header allows the web-server to enable or disable the web browser's XSS protection mechanism. The following values would attempt to enable it:

X-XSS-Protection: 1; mode=block

X-XSS-Protection: 1;
report=http://www.ghanaedu.com/xss

The following values would disable it:

X-XSS-Protection: 0

The X-XSS-Protection HTTP response header is currently supported on Internet Explorer, Chrome and Safari.

8. Inadequate or No Cache-control and Pragma HTTP Header Set

A browser and a proxy can cache content if the cache-control and pragma HTTP header are not set appropriately or missing. The cache-control and pragma HTTP header have not been set properly or are missing allowing the browser and proxies to cache content. If possible, the following parameters namely: "no-cache", "must-revalidate", "no-store", and "private" are set to the cache-control HTTP header as well as the pragma HTTP header set with no-cache.

9. Cookie Without Secure Flag

A secure flag set on a cookie ensures that browsers do not submit a cookie when requested without encrypted HTTP connectivity that remediates against cookie interception by criminals surveying network traffic. Failure to set a secure flag on a cookie, the cookie will be moved in a clear-text upon every visit to and to any HTTP address in a cookie's scope. Hackers may take advantage of this to lure users to appropriate links either directly or indirectly. A domain may issue a cookie but does not have a content to be accessed over HTTP. Hackers can still manipulate links like http://ghanaedu.com:443 to carry out similar attacks.

This vulnerability is well exploited by an attacker positioning him/herself to eavesdrop a victim's network traffic. Communication via a server over insecure connections typifies this scenario. Switched networks are not adequate to fight against this. Attackers within an ISP of the client or an application that host infrastructure can also carry out such an attack. A test run on *google drive* using ZAP revealed that a cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.

Remediation steps such as setting a secure flag on every cookie used to transmit very sensitive data when you need to access content over HTTPS. If there is a clear intention to use cookies to transmit session token, the application area for accessing over HTTPS must apply their way of handling the session and its token must not under any circumstance be communicated without encryptions. Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.

V. Vulnerabilities Associated with Traditional ITO Environment

During information gathering or reconnaissance of a target network, a hacker tries as much as possible to gather the information that will help to make an attack. To block an attacker's malicious attempt, IT professionals look for such vulnerable information to patch the vulnerabilities discovered.

A. Port Scanning

The study discovered open ports, protocols, firewalls, Operating Systems details and many more using the Nmap program. The 'target' was entered in the target field. This was the URL of the network, for example, www.xxxxxxxxx.com. There are many scan profiles but this study used "nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53 --script "default or (discovery and safe)" <target>". When a scan was run on the URL of a Traditional ITO provider, the vulnerabilities in Figure 4. were discovered.

Port	Protocol	State	Service	Version
21	tcp	open	ftp	Pure-FTPd
22	tcp	open	ssh	OpenSSH 5.3 (protocol 2.0)
25	tcp	open	smtp	
53	tcp	open	domain	ISC BIND 9.8.2rc1
80	tcp	open	http	Apache httpd 2.4.33 ((cPanel) OpenSSL/1.0.2o mod_bwlimited/1.4)
110	tcp	open	pop3	Dovecot pop3d
143	tcp	open	imap	Dovecot imapd
443	tcp	open	http	Apache httpd 2.4.33 ((cPanel) OpenSSL/1.0.2o mod_bwlimited/1.4)
587	tcp	open	smtp	Exim smtpd 4.91
993	tcp	open	imap	Dovecot imapd
995	tcp	open	pop3	Dovecot pop3d
3389	tcp	closed	ms-wbt-server	
20	udp	closed	ftp-data	
21	udp	closed	ftp	
53	udp	open	domain	ISC BIND 9.8.2rc1 (RedHat Enterprise Linux 6)

Fig.5. NMAP Scan Report

From Fig.5, ports 389, 20 and 21 discovered were

closed during the scan. Even though the ports are closed, but they are still accessible but there is no application listening on them. It meant that the hosts were up on IP addresses and as part of OS detection. Because the closed ports are reachable, they could be scanned later to see if they were open up. Administrators could consider blocking such ports with a firewall so that they could appear in filtered. The open ports were discussed in detail in the following sections.

B. Associated Vulnerability with tcp/ ftp 21 Pure-FTPd

CVE-2017-12170

Downstream version 1.0.46-1 of pure-ftpd as shipped in Fedora was vulnerable to packaging error due to which the original configuration was ignored after update and service started running with the default configuration. This has security implications because of overriding security-related configuration. This issue doesn't affect the upstream version of pure-ftpd.

CVSS Score	**7.5**
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	

Fig.6. TCP/ FTP 21 Pure-FTPd Vulnerability

C. Associated Vulnerability with tcp/ ssh 22 OpenSSH 5.3 (protocol 2.0)

CVE-2014-1692

The hash_buffer function in schnorr.c in OpenSSH through 6.4, when Makefile.inc is modified to enable the J-PAKE protocol, does not initialize certain data structures, which might allow remote attackers to cause a denial of service (memory corruption) or have unspecified another impact via vectors that trigger an error condition.

CVSS Scores & Vulnerability Types

CVSS Score	7.5
Confidentiality Impact	Partial (There is considerable informational lisclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service Overflow Memory corruption

Fig.7. TCP/ SSH 22 OpenSSH 5.3 (protocol 2.0)

D. Associated Vulnerability with tcp/smtp 25

CVE-2010-0025

The SMTP component in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP2, and Server 2008 Gold, SP2, and R2, and Exchange Server 2000 SP3, does not properly allocate memory for SMTP command replies, which allows remote attackers to read fragments of e-mail messages by sending a series of invalid commands and then sending a STARTTLS command, aka "SMTP Memory Allocation Vulnerability."

CVSS Scores & Vulnerability Types

CVSS Score	**5.0**
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Obtain Information

Fig.8. tcp/smtp 25

E. Associated Vulnerability with tcp/ domain 53 ISC BIND 9.8.2rc1

CVE-2012-5689

ISC BIND 9.8.x through 9.8.4-P1 and 9.9.x through 9.9.2-P1, in certain configurations involving DNS64 with a Response Policy Zone that lacks an AAAA rewrite rule, allows remote attackers to cause a denial of service (assertion failure and named daemon exit) via a query for an AAAA record

CVSS Scores & Vulnerability Types

CVSS Score	7.1
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial of Service

Fig.9. TCP/ DOMAIN 53 ISC BIND 9.8.2rc1 Vulnerability

F. Associated Vulnerability with tcp/ pop3 110 Dovecot pop3d

CVE-2010-3706

plugins/acl/acl-backend-vfile.c in Dovecot 1.2.x before 1.2.15 and 2.0.x before 2.0.5 interprets an ACL entry as a directive to add to the permissions granted by another ACL entry, instead of a directive to replace the permissions granted by another ACL entry, in certain circumstances involving the private namespace of a user, which allows remote authenticated users to bypass intended access restrictions via a request to read or modify a mailbox

CVSS Score	5.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Single system (The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).)
Gained Access	None
Vulnerability Type(s)	Bypass a restriction or similar

Fig.10. TCP/ POP3 110 Dovecot pop3d Vulnerabilty

G. Associated Vulnerability with tcp/http 80 Apache httpd 2.4.33

CVE-2017-15715

In Apache httpd 2.4.0 to 2.4.29, the expression specified in <FilesMatch> could match '$' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where uploads of some files are externally blocked, but only by matching the trailing portion of the filename.

CVSS Scores & Vulnerability Types

CVSS Score	6.8
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	

Fig.11. TCP/HTTP 80 Apache httpd 2.4.33 Vulnerability

H. Associated Vulnerability with tcp/imapd 143 Dovecot imapd

CVE-2013-2111

The IMAP functionality in Dovecot before 2.2.2 allows remote attackers to cause a denial of service (infinite loop and CPU consumption) via invalid APPEND parameters.

CVSS Scores & Vulnerability Types

CVSS Score	5.0
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial of Service

Fig.12. TCP/IMAPD 143 Dovecot imapd Vulnerability

I. Associated Vulnerability with tcp/http 443 Apache httpd 2.4.33

CVE-1999-0496

A Windows NT 4.0 attacker can acquire administrative rights by forcing NtOpenProcessToken to succeed irrespective of the user's permissions, aka GetAdmin

CVSS Scores & Vulnerability Types

CVSS Score	7.2
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	User
Vulnerability Type(s)	

Fig.13. TCP/HTTP 443 Apache httpd 2.4.33 Vulnerability

J. Associated Vulnerability with tcp/smtp 587 Exim smtpd 4.91

CVE-2017-16943

The receive_msg function in receive.c in the SMTP daemon in Exim 4.88 and 4.89 allows remote attackers to execute arbitrary code or cause a denial of service (use-after-free) via vectors involving BDAT commands.

CVSS Scores & Vulnerability Types

CVSS Score	7.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of ServiceExecute Code

Fig.14. TCP/SMTP 587 Exim smtpd 4.91 Vulnerability

K. Associated Vulnerability with tcp/imapd 993 Dovecot imapd

CVE-2009-3235

Multiple stack-based buffer overflows in the Sieve plugin in Dovecot 1.0 before 1.0.4 and 1.1 before 1.1.7. It was derived from Cyrus libsieve permits context-dependent attackers to cause a denial of service (crash) and likely run arbitrary code using a crafted SIEVE script as demonstrated by sending e-mail to a large number of recipients, a different vulnerability from CVF-2009-632

CVSS Scores & Vulnerability Types

CVSS Score	7.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of ServiceExecute CodeOverflow

Fig.15. TCP/IMAPD 993 Dovecot imapd Vulnerability

L. Associated Vulnerability with tcp/pop3 995 Dovecot pop3d

CVE-2011-2166

script-login in Dovecot 2.0.x before 2.0.13 does not follow the user and group configuration settings, which might allow remote authenticated users to bypass intended access restrictions by leveraging a script.

CVSS Scores & Vulnerability Types

CVSS Score	6.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Single system (The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface)
Gained Access	None
Vulnerability Type(s)	Bypass a restriction or similar

Fig.16. TCP/POP3 995 Dovecot Vulnerability

M. Associated Vulnerability with udp/domain 53 ISC BIND 9.8.2rc1

CVE-2013-4854

The RFC 5011 implementation in rdata.c in ISC BIND 9.7.x and 9.8.x before 9.8.5-P2, 9.8.6b1, 9.9.x before 9.9.3-P2, and 9.9.4b1, and DNSco BIND 9.9.3-S1 before 9.9.3-S1-P1 and 9.9.4-S1b1, allows remote attackers to

cause a denial of service (assertion failure and named daemon exit) via a query with a malformed RDATA section that is not properly handled during construction of a log message, as exploited in the wild in July 2013.

CVSS Scores & Vulnerability Types

CVSS Score	**7.8**
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service

Fig.17. UDP/DOMAIN 53 ISC BIND 9.8.2rc1 Vulnerability

VI. Conclusion

In this paper, some CC applications and network of Traditional ITO in Ghana were evaluated using penetration testing tools to ascertain their levels of vulnerabilities. Analysis of the results obtained found the following vulnerabilities; Remote File Inclusion (RFI), Session ID in URL Rewrite, X-Frame-Options Header Not Set, Cookie No HttpOnly Flag, Cookie without secure Flag, Cross-Domain JavaScript Source File Inclusion, Incomplete or No Cache-control, and Pragma HTTP Header Set, Password Autocomplete in Browser and Web Browser XSS Protection Not enabled in form of alerts in CC. Opened and closed ports as well as the Operating Systems and their services were also discovered. Remote File Inclusion (RFI) alert was discovered as a high priority alert and as such, it is recommended that CC service providers should pay more attention to RFI in deploying their services to the country since a successful RFI attack can lead to information theft and taking over of websites by adversaries. The effective way of eliminating RFI vulnerabilities is to ensure that dynamically including files using user input is avoided or relying on whitelisting of files that are to be utilized. CC service providers need to sanitize their inputs to avoid XSS. For Session ID in URL Rewrite, use filter as a countermeasure and for best practice, validation on the server side in all cases is recommended. Security headers can be used for solving X-Frame-Options Header Not Set problems. HttpOnly flag should be set by including this attribute in a relevant set-cookie directive. Generally, web application vulnerabilities are caused by poor design preferences in the applications' development and implementation stages of the development cycle. Developers should meticulously roll-out their applications devoid of flaws and make them secure.

The analysis also revealed opened ports, operating systems and services ran on webservers of the host machine creating considerable disclosure of information to unauthorized users which can be exploited. This was due to the fact that the webservers used were found to be obsolete and patches were not installed to bring them up-to-date. Again, majority of the webservers were poorly designed with flaws and misconfiguration of applications which made them susceptible to remote attacks leading to denial of service or having other unspecified impact via vectors that trigger an error condition. Consequently, the most recent patched or upgraded should be used to fix such design problems.

This paper emphasized the need to deliver secured CC architecture and Traditional ITO via penetration testing tools to discover vulnerabilities and fix them since discovering alone is not enough.

Among the top ten vulnerabilities of OWASP [18]; only Cross-Site Scripting and Sensitive data exposure were discovered by the study. Our future research will be a vigorous research in the remaining threats or vulnerabilities.

Acknowledgment

My first thanks go to the Almighty Allah for bestowing on me all the favors to have come this far. I was fortunate to have the guidance and honest criticism of Mr. Alhassan Salaamudeen for his immeasurable support. My sincere gratitude also goes out to anyone who has taught me anything right from A-Z, to whom I will always be indebted.

References

[1] Kaur, Daljit, and Parminder Kaur. "Empirical analysis of web attacks." *Procedia Computer Science* 78 (2016): 298-306.

[2] Jasmine, M. S., Kirthiga Devi, and Geogen George. "Detecting XSS Based Web Application Vulnerabilities." *International Journal of Computer Technology & Applications* 8, no. 2 (2017): 291-297.

[3] Chang, Victor, Yen-Hung Kuo, and Muthu Ramachandran. "Cloud computing adoption framework: A security framework for business clouds." *Future Generation Computer Systems* 57 (2016): 24-41.

[4] Jones, Steve, Zahir Irani, Uthayasankar Sivarajah, and Peter ED Love. "Risks and rewards of cloud computing in the UK public sector: A reflection on three Organisational case studies." *Information systems frontiers* (2017): 1-24.

[5] Wang, Nianxin, Huigang Liang, Yu Jia, Shilun Ge, Yajiong Xue, and Zhining Wang. "Cloud computing research in the IS discipline: A citation/co-citation analysis." *Decision Support Systems* 86 (2016): 35-47.

[6] Haried, Peter J., and Craig C. Claybaugh. "Evaluating information systems offshore project success: can success and failure coexist?." *Journal of Global Information Technology Management* 20, no. 1 (2017): 8-27.

[7] Schneider, Stephan, and Ali Sunyaev. "Determinant factors of cloud-sourcing decisions: reflecting on the IT outsourcing literature in the era of cloud computing."

Journal of Information Technology 31, no. 1 (2016): 1-31.
[8] Schneider, Stephan, and Ali Sunyaev. "Determinant factors of cloud-sourcing decisions: reflecting on the IT outsourcing literature in the era of cloud computing." *Journal of Information Technology* 31, no. 1 (2016): 1-31
[9] Schneider, Stephan, and Ali Sunyaev. "Determinant factors of cloud-sourcing decisions: reflecting on the IT outsourcing literature in the era of cloud computing." *Journal of Information Technology* 31, no. 1 (2016): 1-31.
[10] Acunetix Web Application Vulnerability Report 2019. https://www.acunetix.com/acunetix-web-application-vulnerability-report/ (Retrieved on 7/10/19)
[11] Hol k, Filip, and Sona Neradova. "Vulnerabilities of modern web applications." In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1256-1261. IEEE, 2017.
[12] Islam, Tariqul, D. Manivannan, and Sherali Zeadally. "A classification and characterization of security threats in cloud computing." *Int. J. Next-Gener. Comput* 7, no. 1 (2016).
[13] Sagar, Deepika, Sahil Kukreja, Jwngfu Brahma, Shobha Tyagi, and Prateek Jain. "Studying open source vulnerability scanners for vulnerabilities in web applications." *IIOAB JOURNAL* 9, no. 2 (2018): 43-49.
[14] Parasram, Shiva VN, Alex Samm, Damian Boodoo, Gerard Johansen, Lee Allen, Tedi Heriyanto, and Shakeel Ali. Kali Linux 2018: Assuring Security by Penetration Testing: Unleash the full potential of Kali Linux 2018, now with updated tools. Packt Publishing Ltd, 2018.
[15] Jabir, Raja Mohamed, Salam Ismail Rasheed Khanji, Liza Abdallah Ahmad, Omar Alfandi, and Huwida Said. "Analysis of cloud computing attacks and countermeasures." In *2016 18th International Conference on Advanced Communication Technology (ICACT)*, pp. 117-123. IEEE, 2016.
[16] Paudel, Samir. "Vulnerable Web Applications and how to Audit Them: Use of OWASP Zed Attack Proxy effectively to find the vulnerabilities of web applications." (2016).
[17] Omeiza, Daniel, and Jemima Owusu-Tweneboah. "Web Security Investigation through Penetration Tests: A Case study of an Educational Institution Portal." *arXiv preprint arXiv:1811.01388* (2018).
[18] Sagar, Deepika, Sahil Kukreja, Jwngfu Brahma, Shobha Tyagi, and Prateek Jain. "Studying open source vulnerability scanners for vulnerabilities in web applications." *IIOAB JOURNAL* 9, no. 2 (2018): 43-49
[19] A Hacker's View of Your Network—Analyzing Your Network with Nmap https://www.usenix.org/conference/lisa18/presentation/schottman (Retrieved on 23/09/19)
[20] Remote file inclusion (RFI) https://www.imperva.com/learn/application-security/rfi-remote-file-inclusion (Retrieved on 19/09/19)

Authors' Profiles

Umar Sayibu, born in 1973 and a Tutor at Bagabaga College of Education Tamale - Ghana. He holds an M.Sc. degree In Information Technology from Kwame Nkrumah University of Science and Technology, Kumasi – Ghana in June 2018. He received his BA degree in Information Studies in 2010 from University of Ghana Accra -Ghana. His research area of interest is Cloud Computing, Information Systems and Computer Networks.

Frimpong Twum received his B.Sc. (Hons) degree in Electrical and Electronic Engineering and MSc. Internet and Multimedia Engineering from London South Bank University in 2004, and 2007 respectively. He also received MSc. Degree in Information System from Roehampton University, London in 2011. He completed his PhD in Computer Science from KNUST, Ghana, in 2017 with specialisation in Computer Security. He is a **SENIOR LECTURER** at the Department of Computer Science, KNUST. Prior to his appointment at KNUST, he worked as a Lecturer and Systems Engineer at Roehampton University in London and also at PC World in UK. He has 25 articles to his credit including: 1. **Twum F**., Hayfron-Acquah J. B, Panford J.K. A Proposed New Framework for Securing Cloud Data on Multiple Infrastructures using Erasure Coding, Dispersal Technique and Encryption, International Journal of Computer Applications, Vol. 181, No. 50, pp. 38-49, April 2019 and 2. **Twum F**., Hayfron-Acquah J. B, Morgan-Darko W., A Proposed Enhanced Transposition Cipher Algorithm Based on Rubik's Cube Transformations, International Journal of Computer Applications, Vol. 182, No. 35, pp 18-26, January 2019. His areas of research interest include: Computer Networks, Computer Security, Cloud Computing, E-Commerce, and Software Engineering.

Issah Baako, born in 1977. and a Tutor at Bagabaga College of Education in Ghana. He completed a B. Ed. Information Technology programme in June, 2011 and M. Sc. IT in June 2018. His main research interests include E-Commerce security and E-payment systems.

How to cite this paper: Umar Sayibu, Frimpong Twum, Issah Baako, "Delivering a Secured Cloud Computing Architecture and Traditional IT Outsourcing Environment via Penetration Tools in Ghana", International Journal of Computer Network and Information Security(IJCNIS), Vol.11, No.11, pp.46-59, 2019. DOI: 10.5815/ijcnis.2019.11.06