ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

DEVELOPING BLACK BOX WEB APPLICATION PENETRATION TESTING METHODOLOGY USING COMPARATIVE CRITERIA

By: GEBREKIDAN GEBREMEDHIN MEBRAHTU

Advisor: WORKSHET LAMENEW

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master of Science in Information Science

JUNE 2015

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

DEVELOPING BLACK BOX WEB APPLICATION PENETRATION TESTING METHODOLOGY USING COMPARATIVE CRITERIA

By: GEBREKIDAN GEBREMEDHIN MEBRAHTU

Advisor: WORKSHET LAMENEW

Name and signature of the examining Board

Name	Title	Signature	Date
_____	Advisor	_____	_____
_____	Examiner	_____	_____
_____	Examiner	_____	_____

Contents

List of Main Figures

List of Main Tables

ACKNOWLEDGMETS

My main aim of attending a post-graduate program in Information Science at Addis Ababa University was to learn more about Information Science and conduct a research on ethical hacking. After finalizing this research I believe that I have achieved my goal as conducting this research has enabled me to learn more about ethical hacking, vulnerabilities of different targets, the methodologies to conduct security testing. This research also made me to understand the huge gap in the field of ethical hacking which in turn is motivating me in the future to pursue my PhD study in this field. I hope God will help me my next goal as He had helped me to achieve this one.

This research wouldn't have been successful, had I did not get the necessary resource and knowledge from the office I am currently working as a result I greatly thank the top management of my office.

I also want to acknowledge my advisor Doctor Workshet Lamnew for guiding and setting a direction how the research should proceed. It is through his direction and guidance the research had formulated and guided.

Finally, I want to acknowledge my lovely wife Alemnesh Assefa who not only gave me an ample time to work on my research but also a lovely daughter, Markan Gebrekidan, when I was conducting this research.

ABSTRACT

The impact of exploiting web applications can be from stealing confidential information, loss of confidence to a war between countries and civil unrest. Attackers can deface university websites and post offending messages that target to make fighting between communities and civil war. As a result, developers, website and web application owners should take an appropriate measure from being attacked.

One of the preventive measures to protect from attacks is to identify the possible vulnerabilities that can lead for attackers to exploit them through a black box penetration testing.

To conduct such a test black box assessors need methodologies as without them leads the test to be non-effective and time consuming. That is why different testing methodologies and procedures are being developed.

The security tester that may test for the identification of any vulnerability irrespective of the standard, however, hasn't enough frameworks to compare those testing methodologies. In this research, the set of criteria for selecting and testing black box web application security methodologies was developed and the methodologies was compared based on a set of criteria.

Once the strength and weakness of those standards were known, a complete testing methodology based on the current testing methodologies and additional reference was developed. The testing methodology was then tested on a sample of four Ethiopian universities and two intentionally vulnerable.

From the research findings it can be concluded that NIST, OWASP, ISACA, ISSAF, and Penetration Testing Framework can be used for black box web application testing. However, they incorporate black box, white box, and gray box testing methodologies within one methodology. Hence, a black box security tester can't directly use them. More importantly, a black box tester can't only relay in only one of the testing methodologies as there are areas that each of them don't cover. As a result a new and complete methodology was developed.

Researches such as white box web application security testing, black and white box security testing on other targets besides to web application, risk calculation, knowledge and skill requirement for black and white box security assessors can be further conducted on this area.

CHAPTER ONE

INTRODUCTION

1.1. Background

Websites and web applications have become necessary for organizations to introduce themselves, to contact their customers, and process their customer data. The websites being developed can either be through professional developers who have a deep knowledge on attacks related to a website or through those who have little knowledge on website cyber attacks. Most organizations are concerned on the availability and the graphics of the website but not on the security. With the increase in the skill of attackers and the easy availability of attacking tools, websites have become targets of attackers. It is after being attacked where most organization recognize that they need to have a secured website. [1]

Different companies or associations have developed web application security testing framework or methodology that they claim can be used for any organization's web application security testing. Among them are NIST, OWASP, ISSAF, ISACA, and Penetration Testing Execution Standard. [2, 3, 4, 5, 6]

The methodologies show the techniques, the objectives, examples, countermeasures, tools, and additional reference. Using those methodologies a tester can conduct specific tests on web applications and other targets. The methodologies help a security assessor to identify vulnerabilities (weakness) of the target. Using those methodologies a security assessor can state the relative security level of the target. [7]

The components of information security are confidentiality, integrity and availability as shown figure 1. Confidentiality is providing data only to the privileged user; integrity is making sure that the data in transit or stored not modified; and availability is a way of assurance that the

required data or system is available whenever it is needed by the authorized user.

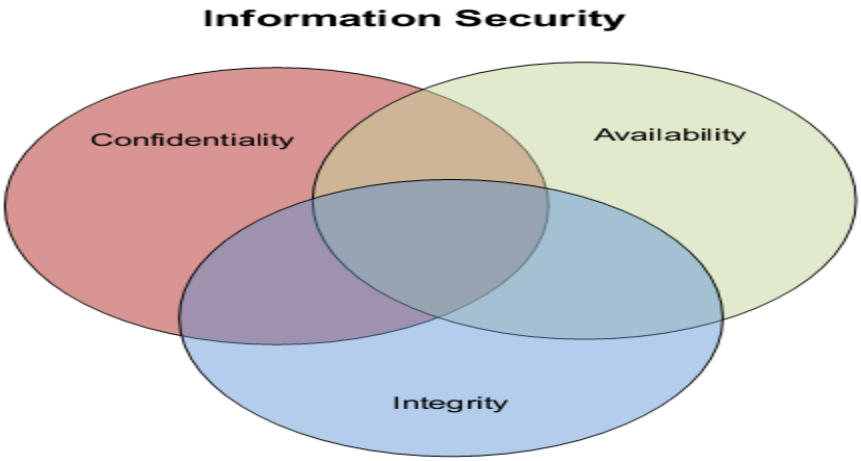


Figure 1: Information Security components

Organizations, security auditors and penetration testers however, face difficulty in selecting the right methodology to test the target. [8, 9, 10, 11]Some testers mix the different testing methodologies and develop their own testing methodology.

To avoid the confusion in selecting the right testing methodology, different researchers have compared the methodologies as can be seen in [8, 9, 10, 11].This research assesses five testing methodologies (NIST OWASP, ISSAF, ISACA, and Penetration testing Framework) that will enable organizations, security testers and internal employees a way to select the testing methodologies, effectively conduct test using a newly complete methodologies.

1.2. Problem Statement

Websites have become one of the channels of communication between organizations and institutes such as universities with public and internal staff. If we take Ethiopian universities as an example, some universities such as Dire Dawa university allow students to access their grade through their website. An attack on those websites can reveal propriety information and may even lead to organizations collapse. As a result, the vulnerabilities should be identified before attackers identify them.

This research was initiated by personal interest to work in the area of Ethical hacking. Then from personal experience in the area one of the challenges that testers face is selecting the right methodology. Taking this challenge, we tried to check if this is global problem.

From the researches reviewed, it can be understood that taking the importance of website security testing different organizations or associations have developed different security testing standards but the testers are getting confused because they do not have a framework to select the existing testing methodologies, some of the testing methodologies focus only on the technical testing only, some of them have limited testing methods. The details of the problem faced are further described in the literature review. It can be concluded that selecting the right methodology and avoiding the limitations of the methodologies is a global difficulty for penetration testers. As a result, this research will focus to solve the challenges faced by penetration testers especially black box web application penetration tester in selecting the right methodology and developing a black box web application penetration testing methodology with by minimizing the limitations of the most widely used security testing methodologies.

This research will answer the following questions

1. Can we have one complete black box web application penetration testing methodologies that can be support the tester in not only providing the technical methods but also in managing the project and delivering, presenting a penetration testing report?
2. Can we have criteria that can be used to compare the existing and future methodologies?
3. What are the targets that should be included in black box web application penetration testing?

As to my current knowledge none of the current researches conducted classified the current widely used testing methodologies into black box and white box testing methodologies. There are security testers that are allowed only to perform a black box security testing with no information or little information about the web application and its defense perimeter. So there must be a methodology to conduct only black box testing so that the tester would not get confused or waste his time in selecting parts of the testing methodologies.

1.3. Objectives

1.3.1. General Objectives

The general objective of this research is to develop a black box web application penetration testing methodology using comparative criteria to enable black web application penetration tester to conduct penetration testing on web application provided with no or little information.

1.3.2. Specific Objective

The specific objectives of this research include

- Identifying web application technologies, and the methodologies used to test them
- Developing a criteria to compare current and future web application testing methodologies
- Synthesizing the current testing methodologies that contain black, gray and white box testing methodologies for testing web applications, infrastructures and other targets to be used for only black box web application penetration testing and the related infrastructure
- Comparing five web application testing frameworks/methodologies and then developing a complete black box web application penetration testing methodology
- Evaluating the methodologies developed on sample Ethiopian universities and intentionally vulnerable web application.

1.4. Research Scope and Limitations

Hackers target technology, people, and process that can enable them to exploit the target for their own motives. The hackers search for easy exploitable and less risky targets. Each target however poses its own obstacles for attackers.

Hackers exploit the general people or employees of an organization through social engineering. However, the following are some of social engineering difficulty for an attackers-

- During social engineering the attacker needs to be smart enough to persuade the target to get the required information.

- There is a possibility of being caught as the tester might need to interact with the target in person
- If the target is aware enough of social engineering, the attacker may not be successful

The process is another target for attackers. Though an organization might have more secured technology and attack technique aware employees, it might be exploited through its weak process. The processes include secured development process, development and implementation of information security, physical security process etc. However, most of the attacks related to process are applicable in the premises of the target. As a result, the attacker can only exploit certain parts of the process remotely.

Technology attack encompasses both people and process that can be inside the organization or remotely. In addition to that, no matter how much secured the technology there is always a possibility of being exposed to an attack. Each day new releases of vulnerabilities (zero day) are discovered and made available to the public. Hence, an attacker might find a way to compromise the target.

Due to the above reasons this research focused on technology security testing, however, the test could be more effective in identifying the possible vulnerabilities had it had incorporated all the targets.

Though vulnerabilities can be identified from interview, observation, source code review and other means, this research focused on black box web application security testing through penetration testing. The research includes the pre-assessment, assessment, and post assessment methodologies necessary to manage, conduct and report black box web application security testing. However, risk calculation which should have been part of the post assessment is not part of this research as by itself can be another research through including it in here would need more time than the allocated time to conduct this research. This research wouldn't also provide organizations a methodology to select black box security assessors but will provide them the test types that should be conducted.

Due to the limitations of the current available tools, it was mandatory to include manual assessment however as tools become more smart the manual test might be replaced by using automated tools.

1.5. Significance of this research

1.5.1. Significance to black box penetration testers

This research is expected to solve the challenges such as which methodology to select when conducting a black box web application security testing. Similar methods are used in different methodologies as a result similar tests might be conducted several times if the testers want to use several methodologies as testers can't only relay on a single methodology. This research would enable them to identify the strength and weakness of each methodology as a result the testers would have the chance to combine the strengths of the methodologies and conduct effective security testing. A new methodology of how to conduct black box web application penetration testing is also developed in this research which will be helpful to not only identify vulnerabilities but also to validate them by identifying false positive and false negative vulnerabilities.

Using the methodology developed in this research the security assessors wouldn't only conduct technical assessment but also manage black box web application penetration testing hence it would enable them to directly conduct their research without wondering here and there.

1.5.2. Significance to Researchers

This research will motivate researchers to conduct further research on the area of black box security assessment methodologies and other methodologies. Researchers can conduct further researches on areas such as black box testing on other targets, white box testing on web applications and others. The researchers can also benefit from this research by extending the criteria set on these methodologies.

1.5.3. Significance to Ethiopian Universities

Critical vulnerabilities were identified on the sample four Ethiopia universities. The universities can take the initiative to fix their vulnerabilities before they are exploited by attackers. Other universities can also get motivated to have their websites tested as their websites might also be vulnerable and could be in the hands of attackers.

CHAPTER TWO

LITERATURE REVIEW

This literature review is organized by first described the concepts related to penetration testing and test types. The related works in comparing and developing penetration testing methodologies are also reviewed. The attacks on different sector web applications and university web applications are reviewed to indicate the type of attacks and their impacts.

2.1. Concepts Review

2.1.1. Penetration testing

Penetration testing also known as ethical hacking is the art of testing a running application remotely to find security vulnerabilities. A penetration tester performs as an attacker but only differs because he is an ethical and hence during a penetration testing the tester works to exploit and find known vulnerabilities. [8]

2.1.2. Test types

Depending on the information provided by the client to the tester, testing can be classified as black box, white box and gray box testing.

In black box testing the client almost doesn't provide the information about the target of test but contrary to this in white box testing the information requested by the tester are provided by the client. Partial information is provided in gray box testing by the client to the tester.

Tests such as design review, source code review, and configuration review can only be conducted in white box testing unless the tester obtains them by herself/himself when conducting black box testing. [2]

2.2. Web attacks

2.3.1. Website attack statistics

In the 2014 assessment performed by whitehat website attacks were ranked as follows. 55% of the website were vulnerable to Information leakage, 53 % to cross-site scripting, 33% to content spoofing, 26% to brute forcing, 26 % to CSRF, 23 % to fingerprinting, 22% to insufficient transport layer protection, 14% to session fixation, 13% to URL Redirector abuse, 11% to insufficient authorization, 11% to directory indexing, 9% to abuse of functionality, 8% to predictable resource location, 7% to SQL injection, and 4% to HTTP response splitting. [12]

According to the study performed by Imperva on 70 websites during the first six months period in 2013, website attacks has doubled as compared to their study in 2012. [13]In their study they have analyzed attack request, attack incident, attack incident magnitude, attack incident duration and battle day. They have identified the median battle days or the number of days encountered is 12 and the maximum is 176. Based on the IP address of the attackers many of the attacks originated from USA and China. As a recommendation, deploying security solutions, learning from peers as attacks in similar business may share similar attacks, detect and block attacks, acquire intelligence on malicious sources and others were proposed.

2.3.2. Cyber-attacks on Universities

Many universities are being hacked by malicious users to gain confidential data or disrupt their communication or embarrass them.

Universities are the targets of attacks because they contain not only personal information but also intellectual property. A lot of researches that has been not published might be uploaded in the web servers and attackers might get those confidential data.

In 2002 Yale University website was hacked by Princeton University's admissions staff using the birthdates and social security numbers of the Princeton applicants for the Yale University. The Yale university found out that their website were improperly accessed 18 times.

Mr.LeMenager claimed that he accessed the website because he was curios about its security by stating " IT was really an innocent way for us to check out the security". [1]

A hacking group named Team Ghost Shell claimed that it had hacked servers at 100 major universities around the world. The universities hacked included Harvard, Stanford, Pennsylvania and the university of Michigan. The group posted 120,000 records from the breached servers. Some of the universities including Stanford University confirmed that they were indeed hacked. [14]

2.3. Related works

As described in the problem statement of this research the researchers have recognized the importance of methodologies for a penetration tester, and the challenges posed on selecting the right methodology for penetration testers. The details of the researches reviewed that are similar to this research are as described below.

In [8] it is stated that the security testers have had the difficulty of selecting the right approach to perform security testing as there are no criteria to do so. To tackle this challenge the testers developed a set of criteria for classifying and comparing the security testing approaches. The researchers categorized the testing approaches as static security analysis or white box where the tester has the source code and dynamic security test or the black box testing where the tester has no or little information about the target. To compare those testing approaches the researchers used the covered attacks, test case generation algorithm, white box or black box, test case or test data, source of test cases, tool and automation as the set of criteria. Using those set of criteria the researchers have identified that most of the security testing approaches identify SQL injection attack, Cross site scripting attack and buffer overflow, and most of the testing approaches are white box. However, the researchers should have compared each of the main approaches rather than grouping them as white box and black box testing. Besides to this, if there is an option to test using a white box approach it is known that it is the best approach as compared to black box because in white box the tester has more information on the application and can perform more tests so it is unclear how they help the security tester in solving the challenges faced.

In the paper "About Effective Penetration Testing Methodology" [9], the researcher also agreed that testers have been confused which methodology to choose but he pointed out that it is

a must to have a testing methodology as without it the tester will have an incomplete testing, time consuming test, requiring more effort and the test would in general be ineffective. In this paper the tester claims that the three most popular testing methodologies are OSTMM, NIST, and ISSAF and those testing methodologies are also described. The researcher also developed a methodology to conduct penetration testing by classifying it into three categories. The three categories are Information, Team, and Tools. In the information part the tester should identify whether the test to be conducted is black box or white box. In the team category the testers roles and responsibilities should be known. Finally in the tools part the tester should beforehand choose the tools that will be used during testing. Though the researcher claimed that a new methodology has been developed, this can't be called a methodology because as it does not show how the tests should be conducted like it is stated in the methodologies the researcher reviewed.

The importance and benefits that emerge from methodologies are also described in [10]. In this paper it is stated that if the testers are more experienced they have high probability of selecting the right methodologies otherwise others will get confused. It is also stated that OWASP, NIST SP800-15, OSTMM are the best known testing methodologies. The researchers then tried to highlight the strengths and weakness of the testing methodologies but they failed to answer the question how the tester will be able to solve the challenge of selecting the right methodologies as they have not provided any criteria to compare those methodologies. Rather they jumped into the comparison of testing tools.

Another paper compares the web application testing approaches based on the functional and non-functional requirements as described in [11]. The researchers used Performance, Reliability, Accuracy, testing method, testing criteria and testing targets to compare the approaches used in web application testing such as performance testing, load testing, stress testing, compatibility testing, security testing and others. The security testing approach was given little focus in this paper. In addition to this, the comparison is on the approaches but not on the methodologies of web application security testing.

In [15] it is stated that web application security testing is becoming complicated and there is a search for security testing methodology. The researcher compared agile security testing, OWASP, and Penetration testing web application security testing methodologies. The comparison criteria used are as explained below

- Reducing complexity- if the security testing methodologies reduce the complexity of the testing process.
- Increasing efficiency- if the methodology increase efficiency by decreasing the time spent to conduct the test
- Mitigating false-positives: how does the methodology mitigate false positives
- Increasing knowledge: does the methodology increase the knowledge of the security tester
- Postmortem evaluations: if the testing methodology has a phase to initiate the security tester to reflect over the vulnerabilities, the development process and the security testing process?
- Repository of knowledge- a phase for the security tester to document the security decisions

Using those criteria the researcher concluded as shown in the following table

Criteria / Security testing methodology	Reducing complexity	Increasing efficiency	Mitigating false-positives	Increasing knowledge	Postmortem evaluations	Repository of knowledge
Agile Security Testing	✓	✓	✗	✓	✗	✗
A Penetration Testing Approach	✗	✗	✗	✓	✓	✓
OWASP Testing Framework	✗	✗	✗	✓	✓	✗

Some of the results of this paper are wrong as agile security testing is more related with the developer and the developer is expected to know the code and the design behind the web application and hence the rate of mitigating false positive should be higher compared to the other testing methodologies. Besides to this, the research might enable tester to select one methodology from the other methodology based on the criteria set out but which testing methodology should the tester select to test a specific vulnerability such as SQL injection will not be answered by this comparison.

From the researchers reviewed above the following can be concluded

1. Security testers are getting confused which methodology to select from the most well-known and being used by many web application security testers (OWASP, ISSAF, NIST, and OSTMM). As a result, testers should have the right benchmarks where they can use the specific testing methodologies whenever they want to test a specific target
2. The existing methodologies can be used for both black box and white box testing.
3. Organizations that need to have their website audited also get confused when they try to compare the proposal of one tester from the other because each tester claim they use either their own testing methodology or any of the five testing methodologies. So based on what criteria should organizations select testers is still unanswered question.
4. There are repetitions in some of the testing methodologies and some of them are difficult to understand though they can be represented in an easier way. In addition to this, there can be attack type that may not be covered by some of the methodologies.
5. As has been criticized under each paper reviewed the current researches conducted still do not answer the challenges being posed to the security tester such as which methodology should the tester use to conduct an effective and efficient security testing? And which part of the methodologies can be used for black box and white box testing?
6. The papers have also failed to incorporate ISACA' s security testing methodology, the penetration testing framework that are considered as one of the best known testing methodologies.[5,6]

2.4. Website testing Methodologies

From the researches reviewed in this research the most common methodologies that can be used to test the fourteen targets selected in this research are OWASP, ISSAF, PTES, ISACA IS audit guideline, and NIST SP 800-115. To compare those methodologies, it is necessary to review them hence this section will provide a short summary of the five testing methodologies.

2.4.1. OWASP web application penetration testing[2]

OWASP divides its test methodology into a passive and active testing. In the passive test also known as the information gathering, the tester tries to understand the application's logic and to

identify all the access points of the application. In the active test, the tester tries to exploit the target. OWASP categorized the active test into 11 sub-categories for a total of 91 controls. A short description of the 11 sub-categories is provided below

i. Information gathering

The tests during the information should try to gather as much information as possible about the target and should test spiders, robots and crawlers, search engine discover/reconnaissance, identify application entry points, testing for web application fingerprint, application discovery, and analysis of error codes.

ii. Configuration and deployment management

In this subcategory the tests include SSL/TLS testing, DB Listener testing, infrastructure configuration management testing, application configuration management testing, testing for file extensions handling, old, backup and unreferenced files, infrastructure and application admin interfaces, testing for HTTP Methods and XST. In a similar way, the other subcategory includes other security controls test types.

iii. Identity Management testing- this test includes testing for roles, testing for user registration process, testing account provisioning, testing for account enumeration and guessable user account and test for weak or unenforced username policy.

iv. Authentication testing

The digital identity of the user should be authenticated in a secure way and any attempts to invalidate that should be mitigated.

At this the tests include

a) testing for credentials transport over an encrypted channel- here the tester would investigate if the data credentials are sent over an encrypted channel
b) Testing for user enumeration- here the user interacts with the application to enumerate usernames
c) Testing for guessable user account- the availability of default usernames in the application are tested

d) Other tests include brute forcing testing, testing for bypassing authentication schema, testing for vulnerable remember password and password reset, testing for logout and browser cache and others

v. Authorization testing-

Testing for authorization is testing for access to resources based on the permission. At this stage the tests include testing for path traversal, testing for bypassing authorization schema, and testing for privilege escalation.

vi. Session management testing

The session management of web application enables them to maintain and control user interaction with the site, hence this must be tested thoroughly. The testing include testing for session management schema, testing for cookies attributes, testing for session fixation, testing for exposed session variables.

vii. Input validation testing

Many of the web application vulnerabilities are due to validation problem and hence they should be properly assessed. The tests include cross-site scripting, reflected cross site scripting, stored cross site scripting, DOM based cross site scripting, cross site flashing, SQL injection, oracle testing, MySQL testing, SQL server testing, MS Access testing, testing PostgreSQL, LDAP injection, ORM injection, XML injection, SSL injection,

XPath injection, IMAP/SMTP injection, code injection, OS commanding, buffer overflow.

viii. Error Handling-

The test in this sub-category include testing for error code, and testing for stack traces.

ix. Cryptography

The testing in this sub-category testing for weak SSL/TLS, testing for padding oracle, and testing for sensitive information sent via unencrypted channel.

x. Business logic testing

This is more of a manual testing to identify the business logic flaws implemented in an application. Though this is more of a white box testing OWASP proposes the following a systematical black box testing. This includes understanding the application, creating raw data for designing logical tests, designing the logical tests, standard prerequisites and executing of logical tests. In understanding the application the tester should assess application manuals, requirements documents, functional specifications, use or abuse cases.

xi. Client side testing

The client side testing is done frequently in gray or white box testing.

In general OWASP's testing guide has a summary of the control, objectives of the test, black and white box testing techniques, references, tools, and countermeasures.

2.4.2. ISSAF Assessment [3]

The Information System Security Assessment Framework (ISSAF) is a framework to test different systems. The objective of ISSAF are to act as end-to-end reference document for security assessment, to standardize the information system security assessment process, to set the minimal level of acceptable process, to provide a baseline on which an assessment can be performed, to asses safeguards deployed against unauthorized access etc.

The goals of ISSAF include the following

- Audit organizations and its related infrastructure to assure that they meet industry security requirements
- Use its framework to identify any weakness such as mis-configurations and rectifying them, identifying risks related to technologies, people, business process and addressing them through vulnerability assessment and penetration testing.

Its target of audience include internal and external vulnerability assessors, penetration testers, security auditors and security assessors, security engineers and consultants, web security administrators.

ISSAF has four phases defined as planning, assessment, treatment, and accreditation as shown in figure.

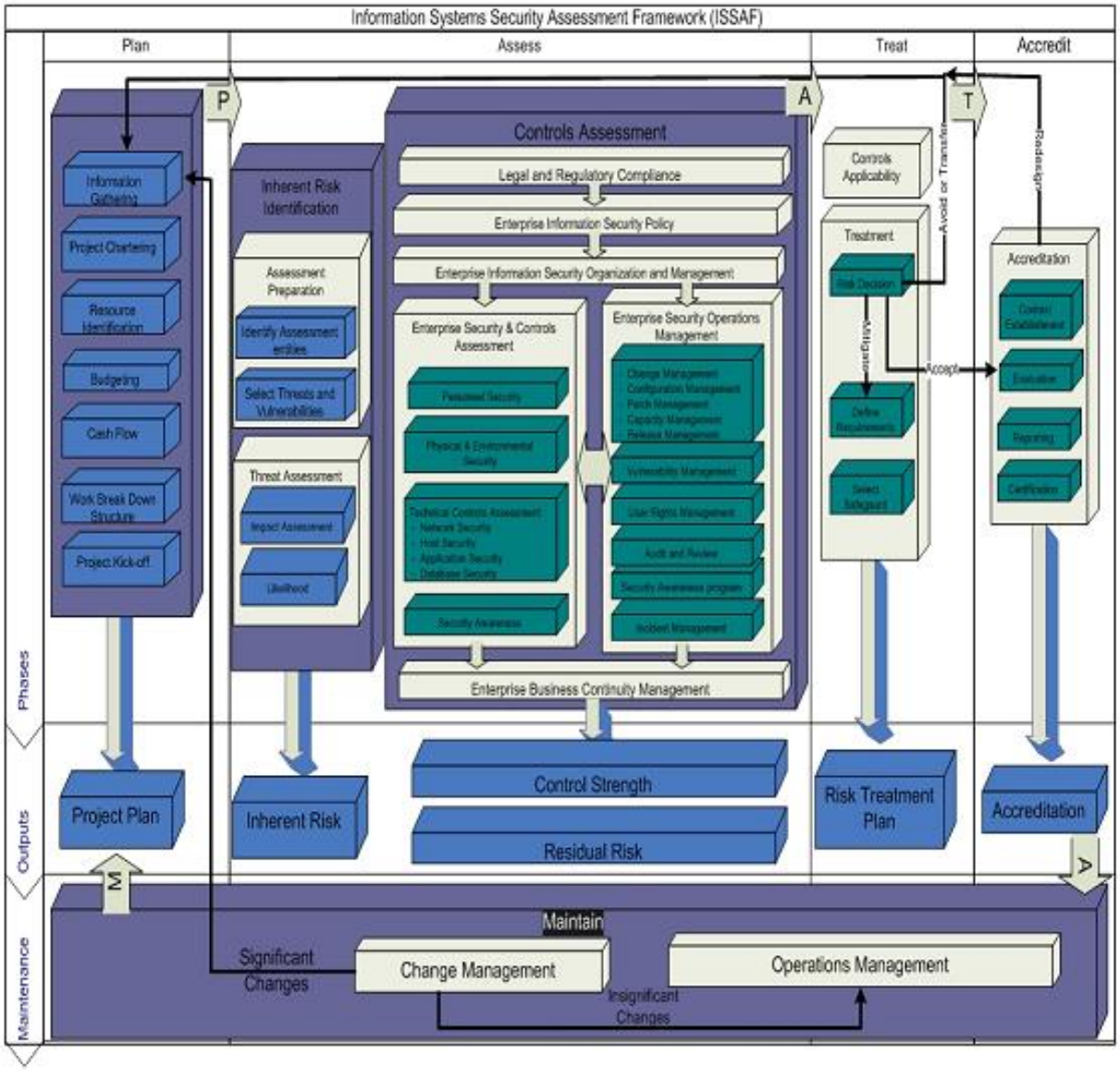


Figure 2: ISSAF Testing Framework

In the planning phase the auditor aim is to get prepared by integrating schedules, and resources which will in turn be used to monitor and control the execution of the project. This phase contains Information gathering, project chartering, resource identification, budgeting.

In the assessment phase a holistic approach to assessing information security risks to an enterprise is provided. The assessment phase is classified as inherent risk identification and controls assessment. In the inherent risk identification impact assessment and likelihood assessment are performed. In the impact assessment an assessment of the impact of a threat to the business is identified. In the likelihood assessment the probability of occurrence of the threat is estimated.

At the controls assessment stage the compensating controls are audited. The controls to be audited include the evaluation of legal and regulatory compliance, evaluation of enterprise information security policy, audit of enterprise information security organization and management. It is classified as enterprise security and controls assessment and operations management assessment. In the enterprise information systems security and controls assessment physical and environmental controls, technical controls such as network security, host security, application security, database security, and security awareness assessment through interviews, observation, structured walk through, and social engineering.

The evaluation of enterprise security operations management is an assessment of capacity management, vulnerability management, release management, enterprise incident management, user management, and certification and accreditation.

During phase three ISSAF proposes how to treat risks. Risks identified at the second stage might be accepted , rejected or transferred. If the risks are rejected then a control must be selected.

ISSAF also has an accreditation process at its fourth phase. If organizations pass the assessment according to ISSAF framework, the organization might be accredited that it is safe based on the standard.

ISSAF has a penetration testing that is inside the second phase of ISSAF assessment framework with three phases and nine assessment steps. The phases are planning and preparation; Assessment; Reporting, clean-up and destroy artifacts as shown in the following figure

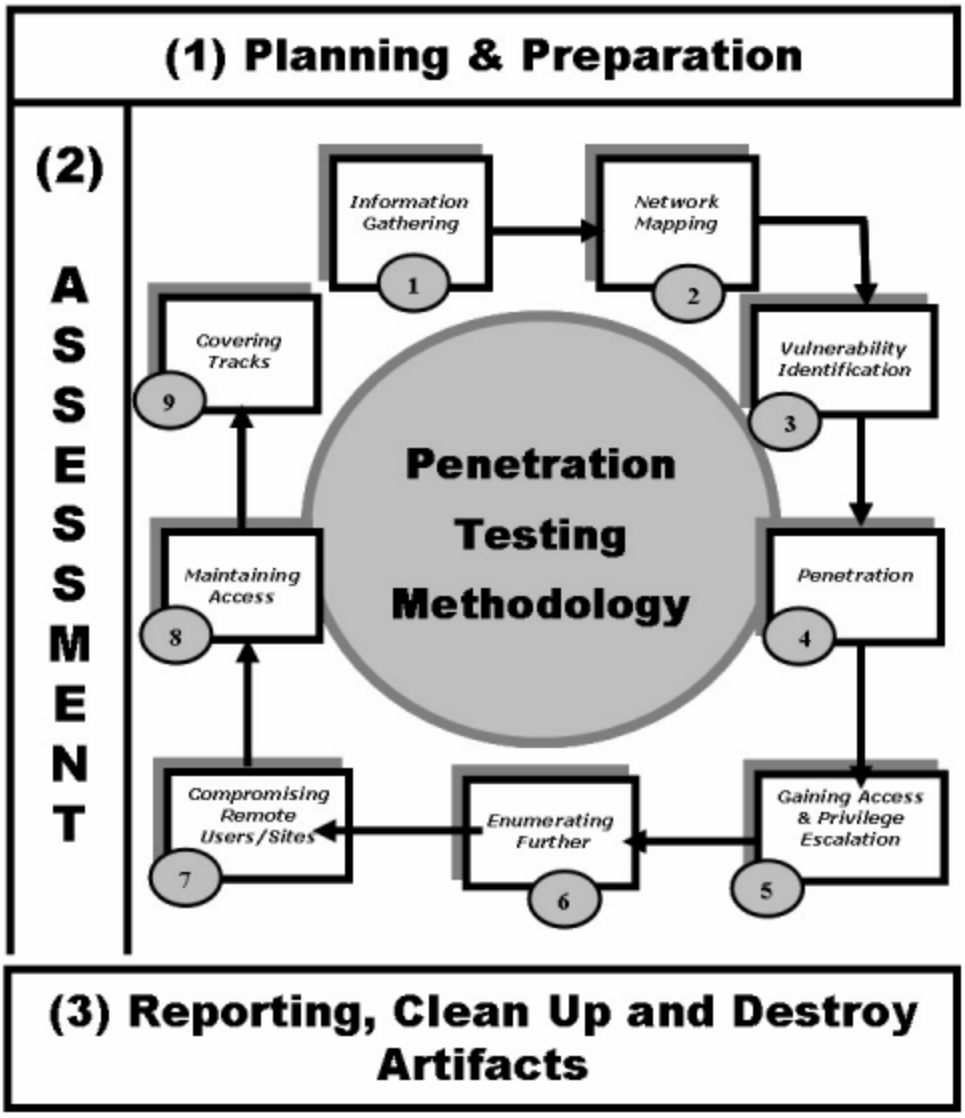


Figure 3: ISSAF Penetration testing phases and steps

ISSAF's penetration testing planning is for the identification of contact individuals from both side, opening meting to confirm the scope, approach and methodology, and agreeing to specific test cases and escalation paths

In the assessment phase an actual penetration testing is conducted and this phase contains information gathering, network mapping, vulnerability identification, penetration, gaining access and privilege escalation, enumerating further, compromise remote users/sites, maintaining access and covering tracks.

Though ISSAF can be used to test different systems, this research's focus is on web application assessment. The purpose of ISSAF's web application assessment is to make the web application and the hosting servers secure by identifying vulnerabilities that might expose them to attack or to provide unnecessary information. It is designed to get access of the remote machine and its related application by even escaping the perimeter network such as a firewall. The pre-requisite to use their methodology is a basic knowledge of HTTP protocol.

The methodologies used starts by identifying the web server vendor and version through banner grabbing, web server detection using automated tools, default file detection, and checking the file extension on the server. To help them assess the methodology has used different tools such as netcat, httprint, HTTRACK, Black widow and other automated and manual testing methodologies. In addition to this, the methodology suggest to copy the website and investigate it to obtain credential and other necessary information such as usernames, default password, E-mail address, Auto redirection information, and external Links.

The next tests conducted during the web application assessment included common gateway interface test, directory traversal test, testing product specific issues, attacking secure HTTP, brute forcing attacks, browsable directories' attack, test invalidated parameters such as cross site scripting, cross-site tracing, URL manipulation such as hidden form fields manipulation, cookie manipulation, vulnerability identification, input validation tests that include validate data tests, and test for buffer overflow, testing SQL injection using SQL injection vulnerability, bypassing user authentication, getting control of database, getting control of hosts etc.

ISSAF has also provided a global countermeasures that need to be implemented in the client side and server side. The client side protection include loging out all sessions when done, not selecting the " Remember Me" option, protecting cookies desktop security, implementing SSL, patching browser etc. The server side countermeasures included patching the web server regularly, making sure that the web server exposes as less details as it can, checking logs, using multiple 404 pages and sending them randomly etc.

In each of the testing methodologies the framework has description of the evaluation criteria, its aims and objectives, pre-requisite for conducting the evaluations, the process for the evaluation, the expected results, recommended countermeasures, and references to external documents.

2.4.3. Information Systems Audit and Control Association (ISACA) IS Auditing Guidelines [5]

ISACA's testing includes both manual and automated testing of the portal site as an external penetration tester with no login information or as an internal penetration tester with login details.

ISACA's testing guideline includes information gathering, vulnerability analysis and exploitation of different web application security controls. The web information gathering starts by first analyzing and crawling through the web pages to gather information of the target. During the information gathering stage testers are expected to gather information on technologies such as Session ID, authentication method employed, certificates used, entry points, hidden contents, robots.txt and others.

ISACA's vulnerability identification includes injection testing, information leakage and error message analysis. ISACA proposed to run commercial and open source tools including Nikto, WebInspect, ScanDo and Appscan to identify vulnerabilities of web application.

The guideline has an exploitation techniques including insertion of code into text field, Directly access a web page that can ordinarily only be reached through authentication by a brute force attack, alternate contents and others.

2.4.4. Penetration Testing Execution Standard (PTES)[6]

PTES has seven phases that starts from the initial communication and reasoning of the penetration testing through intelligence gathering and threat modeling which are helpful to understand the target of test. After understanding the target of test PTES's next phases are vulnerability research, exploitation, post exploitation. In the vulnerability research the main weakness that could lead to attacking the target or exposing it to provide information for the attacker is attained. This is then exploited at the exploitation phase and if the tester wants to continue controlling the target a post exploitation to enhance his privilege and further assessment will be performed at the post exploitation phase. PTES concludes the test by recommending how a tester should write the report of testing.

2.4.4.1. Pre-assessment

The main activities in its pre-assessment phase are scope formulation, understanding the goals of the test (classified as primary, secondary and business analysis), Establishment of lines of communication, agreement on rules of assessment , estimating the duration of the test and Pricing terms.

To formulate the scope of the test the standard uses questionnaires that are grouped as network penetration test, web application penetration test, wireless penetration test, physical penetration test, social engineering, questions for business unit managers, and questions for systems administrators. In addition to that in limiting the scope the IP ranges and domains are determined and validated, how to deal with third parties are also determined.

2.4.4.2. Intelligence gathering

In PTES intelligence gathering is an information gathering against the target. PTES goal of intelligence gathering is to determine various entry points of the target. The information gathering consists of passive reconnaissance, and active foot printing. PTES also recommends and sets a way to identify the network, host, application, storage and user protection mechanisms.

2.4.4.3. Threat Modeling

In this phase the standard recommends the model to be used for threat modeling be consistent in terms of threat representations, their capabilities and their qualification with respect to the target of test.

2.4.4.4. Vulnerability Analysis

It is the process of identifying flaws in the target of test which can enable attacker to use it as hole to access the target. The principles of the standard in vulnerability analysis to first scope the test for applicable depth and breadth to meet the goals and requirements of the desired outcome.

According to the standard the vulnerability analysis can be active, passive, and manual testing. The active test is a direct interaction of the tester with the target of test by using automated tools. The standard categorized the vulnerability scanners as general vulnerability scanners, web application scanners, Specific vulnerability scanners. The general vulnerability scanners are classified as port based scanners which check open and closed ports, service based scanners which identify the services on the open ports, banner grabbers. The web application scanners are general application flaw scanners, directory listing/brute forcing, and web server version/vulnerability identification. The network specific protocols scanners include VPN scanners, voice network scanners.

The passive vulnerability analysis includes metadata analysis that looks at data that describes a file, traffic monitoring that includes sniffing data and making an offline analysis.

After both active and passive analysis the standard proposes to validate the gathered vulnerability. To validate the vulnerability is a correlation between the vulnerabilities identified using different tools. The correlation can be done by vulnerability ID, CVE, OSVDB, vendor indexing numbers, known issue with a software product.

The manual testing includes VPN fingerprinting and authentication, Citrix enumeration, and gathering information from DNS, Web, and Mail servers.

After vulnerabilities are identified through active, passive or manual means an attack avenue should be developed. The attack avenue creation includes creation of attack trees, isolated lab testing and visual confirmation.

Researching for vulnerabilities is the next phase in the vulnerability analysis as recommended by the standard. The researching of the vulnerability can be a research on the identified vulnerabilities from vulnerability databases, vendor advisories, exploit databases and frameworks modules, common/default passwords, Hardening guides/common misconfigurations, and conducting private research by setting up a replica environment, testing configurations, fuzzing, Identifying potential avenues/vectors and disassembly and code analysis.

2.4.4.5. Exploitation

After the vulnerabilities are identified the next phase in PTES is exploiting the target through those vulnerabilities. To avoid attacks the standard proposes to find a way to bypass controls including anti-virus, encoding, packing, encrypting, whitelist bypass, process injection, purely memory resident, purely memory resident, human, data execution prevention, address space layout randomization, and web application firewall.

During the attack the tester is expected to evade IDS/IPS controls by obfuscation or encoding requests/responses to circumvent web application firewalls. The tester is also expected to customize the attacks and exploits according to the target of test. If the existing exploits doesn't work the tester should conduct fuzzing.

2.4.4.6. Post Exploitation

If the target of test is exploited, the tester should understand the value of the compromised target. In PTES the next phase after exploitation is the post exploitation. PTES proposed methods to help the tester to identify and document sensitive data, identify configuration settings, communication channels and relationships with other network devices that can enable further access to the network and setup methods for accessing the exploited target

at later stage. The standard proposes to have rules of assessment to protect the client and the tester.

After using different methods at the post exploitation phase the standard proposed a cleanup process. The cleanup process is to delete all the vulnerabilities identified during the assessment once the report is provided to the customer, remove all executable, scripts and other techniques.

2.4.4.7. Reporting

PTES recommends the report to be classified into executive summary and technical report. The executive summary includes the goals of the penetration test, high level findings of the testing exercise.

The technical report includes an introduction part that contains personnel involved in the testing, contact information, assets involved in testing, objectives of test, scope of test, strength of test, approach and threat/grading structure. After the introduction part the standard proposed to incorporate the results of each of the previous phases of the standard. This includes information gathering report, vulnerability report, exploitation report, and post exploitation report. Finally the standard proposes to provide a conclusion of the report.

2.4.5. NIST SP800-115[4]

The National Institute of Standards and Technology (NIST) publishes different standards, guidelines and procedures. One of the technical guides is NIST SP800-115 that provides technique guide to information security testing and assessment. According to the technical guide there are three ways to conduct security assessment as described below.

1. Examination- analyzes the assessment objectives and studies the internal directives of the target organization, analyzes log files of critical infrastructure to find possible security vulnerabilities.
2. Testing- Here the tester acts as malicious hacker to exploit vulnerabilities obtained in the target organization

3. Interviewing- The assessor asks questions with the target organization to identify weakness.

In black box web application testing both the examination and interviewing won't work unless the assessor wants to conduct social engineering. However, social engineering is out of the scope of this research and hence both examination and interviewing won't be used in this research.

NIST's technical guide to information security testing and assessment has six phases. The framework starts from security assessment planning, then proceeds to review techniques or target identification and analysis to target vulnerability validation techniques and the framework finalizes by post-testing techniques. The framework also provides how to execute the test.

Since the review techniques only works for white box testing, only five phases of the framework has been incorporated in this research.

2.4.5.1. Security Assessment Planning

In its security assessment planning phase the framework provides guidance on creating an assessment policy, prioritizing and scheduling assessments, selecting and customizing the appropriate assessment approach, and addressing logistical considerations, addressing legal considerations.

Since the security assessment policy is the policy that should be developed by the target to be audited, and prioritizing and scheduling assessments are also done by the target to be audited, they will not be included as a methodology for black box testing.

2.4.5.1.1. Assessment Plan Development

The assessment plan development uses NIST SP 800-53A as a reference to develop the plan. The suggested steps are

(i) determining the type of security control assessment;

(ii) determining the security controls and control enhancements to be included in the assessment;

(iii) selecting the appropriate assessment procedures to be used during the assessment based on security controls and control enhancements in the system security plan;

(iv) tailoring the selected assessment procedures for the information system impact level and organization's operating environment;

(v) developing additional assessment procedures, if necessary, to address other security controls and control enhancements;

(vi) developing a strategy to apply the extended assessment procedure;

(vii) optimizing assessment procedures to reduce duplication of effort and provide cost-effective assessment solutions; and

(viii) finalizing the assessment plan and obtaining the approvals needed for its execution

Since this research is to develop a methodology which has embedded in it a procedure only steps (iv)-(viii) will be considered when comparing the frameworks/standards for black box web application testing.

2.4.5.1.2. Determining the logistics of the assessment

The framework suggests the needed resources including Laptop, assessment team, where to conduct the test should be selected.

2.4.5.1.3. Addressing Legal Considerations

The framework suggests what the target of audit organization should consider when being audited by an external auditor. Since the auditor might not have any involvement in drafting or signing such an agreement the framework's suggestion won't be used by the auditor. However, the auditor should develop through its legal representatives should develop a legal document.

2.4.5.2. Target Identification and Analysis Techniques

In its second phase the framework suggests to conduct network discovery, network port and service identification, and finally conducting vulnerability scanning.

2.4.5.2.1. Network Discovery

During network discovery the tester should identify active and responding hosts through either active or passive techniques. However the passive technique is conducted from internal network where in black box this may not be the case hence will not be used as a technique.

In the active network discovery the tester should send traffics to the target website or the hosting server. To avoid detection by firewalls and IPS/IDS the framework suggests conducting stealth scanning or other appropriate mechanisms.

The framework suggested Netcat, and Nmap as a testing tools for network discovery.

2.4.5.2.2. Network port and service identification

At this stage the framework suggests to identify open ports, services running on the target of audit using service identification. In addition to this, through OS fingerprinting the tester can identify the running operating system on the target of audit.

The framework suggests to use tools that allow fragmentation, duplication, overlap, out-of-order, and timing techniques so that the test packets won't be detected by an IPS/IDS.

2.4.5.3. Vulnerability Scanning

Though the above stages of the framework can help identify live hosts and host attributes such as operating systems, applications, and open ports, it is through vulnerability scanning that can enable the assessor to identify vulnerabilities of the target. The assessor will be to identify outdate software versions, missing patches, misconfigurations and depending on the scanner the assessor might also validate whether the target complies with a given standard. The vulnerability scanners can also identify hosts and open ports that are done in the previous stages.

The vulnerability scanners can level the identified vulnerabilities as informational, low, medium, high, critical etc depending on the scanner type. The vulnerabilities scanners can also provide a recommendation on the identified vulnerabilities.

2.4.5.4. Target vulnerability validation techniques

At this stage the framework suggests validating the identified vulnerabilities in the previous stages using password cracking, and penetration testing.

2.4.5.4.1. Password cracking

If the tester identify any password hashes during the previous stages, the password can be cracked using a dictionary attack, brute force and rainbow tables. During a dictionary attack the tester should build a dictionary to crack the hashed passwords. Whereas during the bruteforce attack the testing tools generates all possible passwords and compare it with the hashed password and accordingly may crack the obtained password.

2.4.5.4.2. Penetration testing

During this stage the tester mimics real-world attacks to control the target of assessment. The framework has the following four phases that will be used for penetration testing.

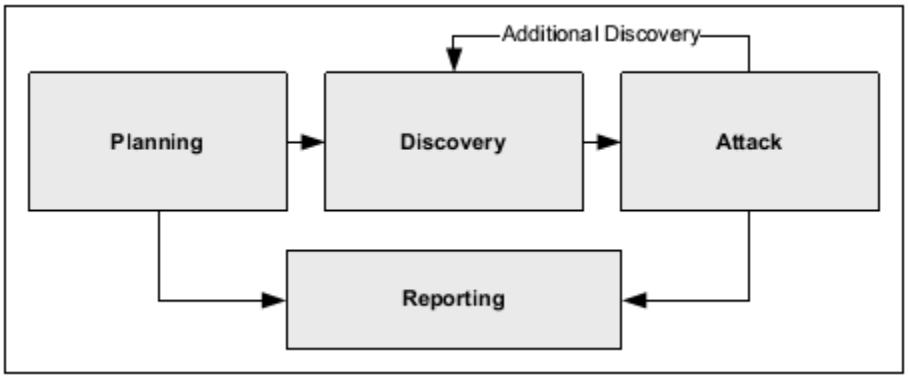


Figure 4: NIST Four Stage Penetration testing Methodology

In the planning phase, rules are identified, management approval is finalized and documented and the goals of the tests are set. According to NIST the planning phase sets groundwork to conduct a successful penetration testing.

NIST's discovery phase has two parts. In the first part information gathering and scanning are recommended to take place. This includes network port and service identification, host name and IP address information gathering, employee name and contact information gathering, and others.

In its second part of the discovery phase vulnerability analysis is performed. This will be comparing the services, applications, and operating systems of scanned hosts against vulnerability database and the testers' own knowledge of vulnerabilities.

Executing an attack is the third phase in NIST 800-115. The following figure shows the steps is expected to be taken at this phase.

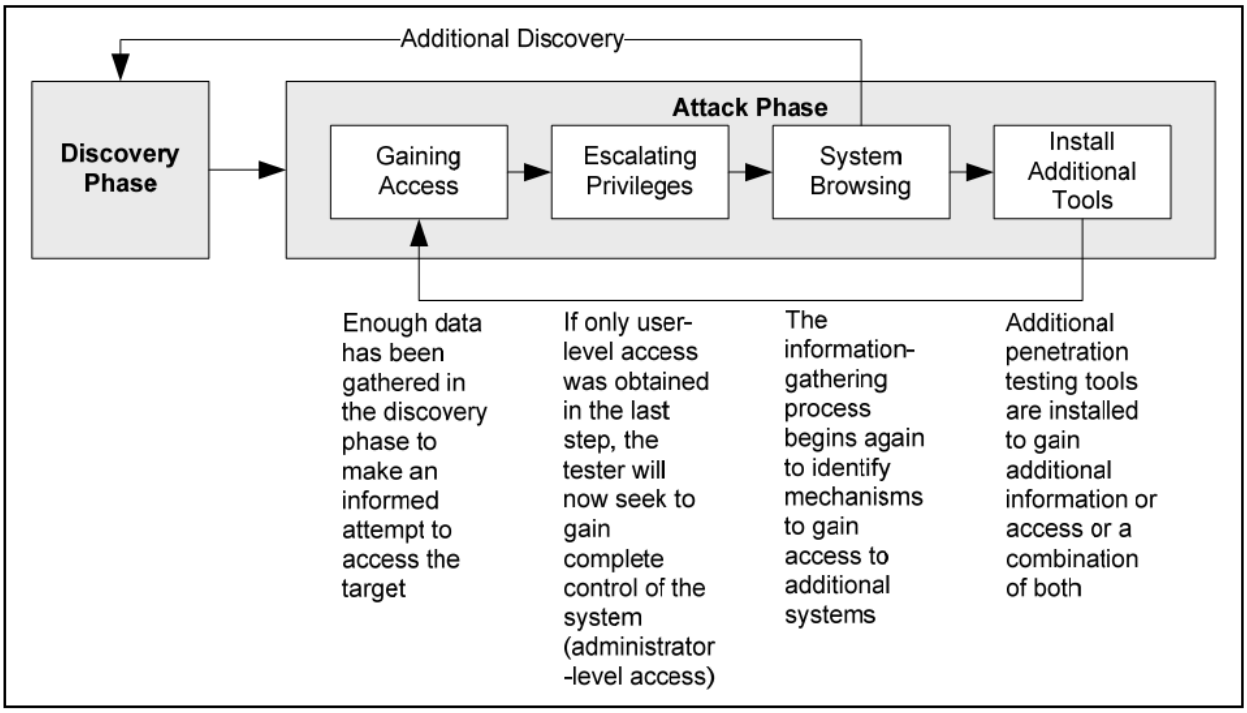


Figure 5: NIST's attack phases with loop back to discovery

2.4.5.5. Reporting

As stated in the framework after the completion of the test a report appropriate according to the standard of compliance of should be documented. The report format can vary based on the standard type.

CHAPTER THREE

RESEARCH METHODOLOGY

As described in the literature review of related work, the researchers developed a set of criteria to compare and develop new penetration testing methodologies. The researchers then evaluated the developed methodologies against those criteria.

By taking the input of previous researchers on related area, this research was conducted by formulating the research problem, performing data collection, selecting testing methodologies, developing comparison criteria, comparing and developing black box web application penetration testing methodology, evaluating the methodologies on six targets as shown in figure 6.

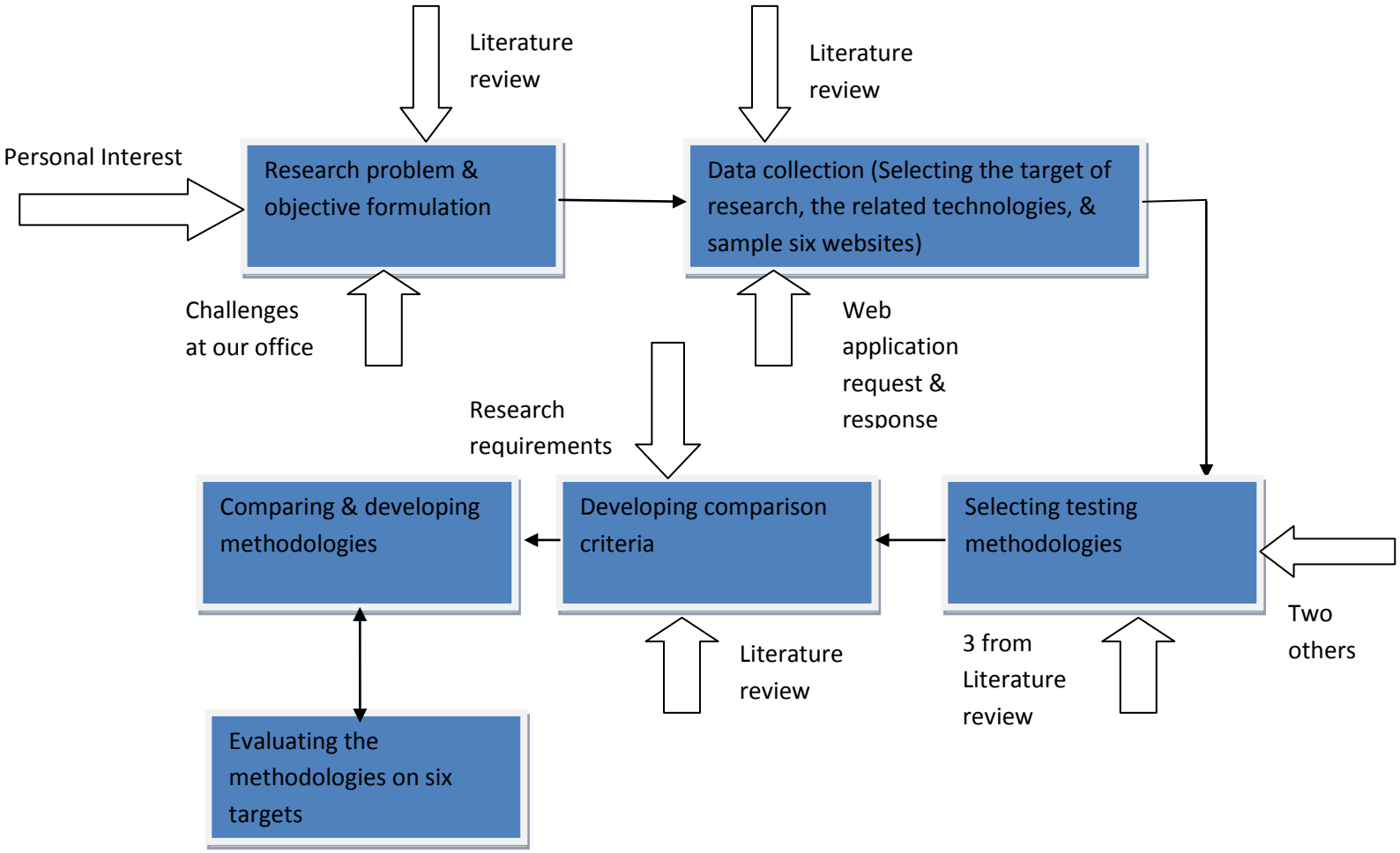


Figure 6 Research Process Flow

The first phase of the research was formulating the research problem. The research problem was initiated by personal interest to conduct a research on the area of Ethical hacking. One of the challenges that we face as penetration testers is selecting the right methodology to be used during penetration testing, security auditing or security evaluation. Since Ethiopia has no standard that companies/organizations need to comply selecting the right methodology depends on the penetration tester. However there are large number of methodologies that the penetration testers to select from but there is not set of criteria that enable them to select one of them.

Taking the challenges that we face in our office, related literature review was conducted. From the reviewed researches, it could be concluded that similar challenges are also faced by other international penetration testers. Hence a complete black box penetration testing methodology development was chosen to be conducted in this research.

To develop the penetration testing methodology it was necessary to select the target and the technologies used to design, develop, implement and operate it. After selecting the targets, it was necessary to select the most common methodologies used to test it. Three methodologies was selected from literatures reviewed and additional two methodologies that have detailed testing methods were added.

To compare the five testing methodologies set of criteria was used from researches conducted previously. However it was not possible to use all of the methodologies from previous researches as some of them don't fit with this research. Hence the previous criteria were filtered and additional criteria were added based on the problem that this research was expected to solve.

The methodologies were compared based on the set of criteria and the strengths and limitations of them were identified. The complete testing methodology was then developed by taking the strengths of the existing methodologies and adding additional methodologies on their limitations from other sources.

The methodologies were then evaluated on four Ethiopian universities websites and two additional vulnerable web applications. The test on the six targets aims to provide an example for testers how to conduct tests using the methodologies and to evaluate the effectiveness of the methodologies.

Since the research problem was already described at the problem statement section, the remaining methodologies used in this research will be described below.

3.1. Data collection

This research requires selecting existing methodologies that are necessary for black box web application security testing; sample web applications to check if the testing methodologies work;

3.1.1. General Target Selection

As has been described in the scoping and limitation sub section this research focuses on technologies penetration testing. However, the technology by itself might be categorized into different categories. One of the ways to categorize technology is based on the OSI layer. Accordingly, the targets might be categorized as physical layer, data link layer, network layer, transport layer, session layer, presentation and application layer technologies.

This research however focused on web application for the following reasons as described in [16].

Ubiquity – Web application are accessed and easily found in the world wide web. Hence, attackers have a very high chance of obtaining vulnerable web application with high value

- Simple techniques- web applications are easy to understand and attack than other targets as they take inputs that could be easily understood by users and attacker than other technologies
- Anonymity- web application attacks are difficult to be traced as there are proxies that can enable attackers to hide themselves and there are many unaccountable regions
- Bypasses firewall- Firewalls permit HTTP/S requests and attackers can embed their attacks in the HTTP/S request and hence the protection provided by firewalls can be bypassed
- Custom Code- most web applications are developed by developers who have little knowledge on security and hence there is a high probability of easily exploited by attackers
- Money- Web applications are processing financial transaction of banks. As a result attackers who have the intention of stealing money might target internet and mobile banking.

3.1.2. Web Application Technology Targets

Web applications can be attacked through the technologies they use or through the infrastructure their deployed on. Hence, an attacker can target the infrastructure or the web application and it would be mandatory to test both of them. As a result it should be known how the request from the user browser passes through different infrastructure.

To identify the target of test it would be mandatory to understand the request and response between the client and the servers. As a result the following is a summary of the request and response

1. A web application provides an interface to the user to access the backend database or data hosted statically on a website. In general it provides the presentation layer, perform business logic and an interface to access data from database. The presentation layer can be html pages, Activex, applet, javascript, and other client script. The dynamic content is accessed using common gateway interface(CGI), ASP, PHP, JSP and others.
2. The application might validate the input data using input validation techniques. In addition to this the request and response might be encrypted.
3. Browser extensions can be used to extend browser built-in capabilities in different ways. The most common browser extension technologies are Java applets, ActiveX controls, Flash objects, silverlight objects.
4. Request for a web page is loaded into a web browser. The web browser provides user access to functions through menus, shortcut keys. The technologies used in client side include HTML, Hyperlinks, forms, CSS, Javascript, VBScript, and others
5. The web browser communicates with the network devices. The network devices check if the requested web page is in the network and if the user is allowed to access the network with the provided URL/IP address, protocol, port number etc. The network security protection devices might include firewall, Web Application firewall (WAF), and IPS/IDS.
6. The requests might pass through load balancers to enhance performance of the application.
7. To elude and catch attackers the user requests might pass through honeypots.
8. A proxy server might further process the request and response
9. If authentication and authorization is required a separate authentication server such as an active directory might process the authentication. Some applications may not use separate

authentication server. Users might be assigned credentials and the application might have a way to transmit/register those credentials. Users might also have the chance to recover forgotten credentials.

10. If the user is permitted the http/https request goes into the web server.
11. If the requested web application is a static website, the web server might send an HTML page. However, if the request is for dynamic content then CGI, ASP or other languages will be used to process the request and communicate with the database. Other back-end components such as file systems, SOAP-based web services, and directory services.
12. In the request and response between client and server additional activities including generation of cookies by web server, opening sockets, encoding and decoding, log activities of the user might happen.
13. The infrastructure and the web application might be managed locally or remotely using for example telnet, SSH, FTP and others

From the request and response described above a penetration tester needs to test the following technologies to identify vulnerabilities in web applications hence the testing methodology should have the right methodologies to test those technologies.

No	**Category**	**Technologies**
1.	Web presentation	HTML(hyperlinks, CSS, forms), Activex, applet, javascript, and other client script
2.	Web application Platforms	ASP.NET and Java
3.	Content Management Systems	Joomal, wordpress, koboo etc
4.	Scripting Languages	PHP, VBScript, Perl
5.	HTTP requests and Response	HTTP Methods, Status Codes, Banners, Cookies, Sessions, Encoding, message body etc
6.	Authentication	Active Directory and other technologies
7.	Browser extensions	Java applets, ActiveX controls, Flash objects, silverlight objects
8.	Client Side Controls	Validation (such as javascripts),
9.	VPN	L2F, IPSec, PPTP, L2TP
10.	Network security devices	Honeypots, Proxies, load balancers, firewall, IPS/IDS

11.	Web server	Apache, IIS, or Netscape Enterprise and others
12.	Database	MS-SQL, Oracle, and MySQL
13.	Other backend technologies	File systems, SOAP-based web services, and directory services
14.	Remote Management	telnet, SSH, FTP and others

Table 1 Testing Targets

3.1.3. Sample Web Application Selection

To conduct this research it was necessary first to select specific sector in Ethiopia so that it is possible to compare between each of the targets and make the more secured web application as benchmark to the other applications. In addition to that, the sector should be one of the high assets values of the countries who have websites that can be accessed remotely.

The sectors might be classified as financial sector, educational sector, industry sector (mining, manufacturing, food processing etc), infrastructure sector (telecommunication, Power generation, Road) etc.

To conduct this research it would be good as compared to the other sectors if the sector to be selected has very high number of websites, very high asset value and low impact if the website is damaged during testing. The following is the relative comparison of the websites.

No.	Sector	Website Asset Value	Possible number of websites	Impact of damage during testing
1.	Financial	Very high	A few	Very high
2.	Educational	High	High	High
3.	Industry	Small	Very High	Small
4.	Infrastructure	Small	Small	Small

Table 2. Relative comparison of sector websites

From the above table though financial sector has a high asset value for attackers, in Ethiopia only a few number of banks exist and the impact of damage of the financial sector is very high. So it would not be recommended to use the financial sector for this research. The next candidate would be the education sector. The education sector fits with this research as there are 30 public university who have websites, the asset value if being attacked is high. The only drawback with the Educational sector is the impact of damage. If registrar office are being attacked students grades can be modified, an offending messages can be posted. With its limitations, this research will select targets from the educational sector.

To conduct an assessment on one website it takes more than two weeks by an individual penetration tester. So it is not possible to conduct the security test on more than six websites as additional time is required. As a result, this research would use six sample websites to finalize the research during the research period.

It would also be mandatory to have an intentional vulnerable websites as those websites would help to identify additional vulnerabilities that may not be obtained on the Ethiopian university websites. So it would be necessary to have two vulnerable websites.

The four Ethiopian websites should have relatively high value as compared to the remaining 26 websites. The most valuable assets that can be accessed by websites in universities is students grades; research papers that can be accessed by users who have credential; and finally the website itself.

Among the 30 public universities Hawassa university and Diredawa university permit to view students grades online and hence they were incorporated in the research. Adigrat university, Arbaminich university, Assosa University, Debremarkos university, Mekelle university, and Wollo university are the only universities that have a login page to access confidential data by users. Assosa university was unique as it was the only website that was developed by Content Management System. As a result, it was incorporated in this research. Wollo university was another target that has an admin page that could be easily guessed hence it was incorporated in this research. So the universities websites that were selected in this research

were Assosa university, Diredawa University, Hawassa university register website, and Wollo University.

The next stage was selecting intentionally vulnerable website. Among the large number of vulnerable websites that a tester can practice penetration testing the following are the most common. An exhaustive list can be found in http://www.amanhardikar.com/mindmaps/Practice.html

Since only two vulnerable websites were needed Cenzic and acublog were selected for this research. In conclusion the websites used in this research are

No.	Name	URL
1.	Assosa University	http://www.asu.edu.et
2.	Diredawa university	http://www.ddu.edu.et
3.	Hawassa university registrar office	http://213.55.76.172/
4.	Wollo University	http://www.wu.edu.et
5.	Acublog	http://testaspnet.vulnweb.com
6.	Cenzic	http://crackme.cenzic.com

Table 2: Research Target web application

3.1.4. Test type selection

The web application test types can be black box, gray box or white box. In a black box testing the assessor is provided with little or no information about the target. Contrary to this in gray box and white box testing, the assessor has some or full information about the target.

Black box security testing is more challenging than gray box and white box testing. Besides to this, most testing methodologies currently developed are white box. As a result black box security testers have more challenges than white box security tester and this research took the initiative to minimize the challenges of those who are more challenged. That is why black box security testing was selected in this research.

3.1.5. Selecting testing Methodology for black box web application penetration testing

As stated in the problem statement the researchers have identified that the best known testing methodologies are NIST 800-115, OWASP Web application testing guide, OSTMM, ISSAF. However, the researchers have not included other well known testing methodologies that are widely used by penetration testers. Those are ISACA's IS Audit Procedure, and Penetration Testing Execution Standard (PTES).

The OSTMM doesn't have a methodology to test web application hence it is excluded in this research. ISSAF has additional testing methodologies besides to web application and the related infrastructure and hence this research will use part of the framework in comparing the black box testing methodologies. Similarly the PTES and ISACA also cover additional methodologies for different targets and hence a similar approach as ISSAF will be used in this research. Though NIST 800-115 has no specific testing methodologies for web application, it would be used in this research as the methodology is expected to work for every target of a penetration testing.

As a result NIST 800-115, OWASP Web Application testing guide, ISSAF 0.2.1, ISACA IS Audit Procedure and Penetration Testing Execution Standard (PTES) were compared in this research.

3.2. Comparison Criteria development

This research was conducted by developing comparison criteria as described below. To develop the comparison criteria previous researches was used as a benchmark.

To compare and select a black box web application penetration testing methodology, set of criteria must be developed. The set of criteria should enable for the penetration tester to identify possible vulnerabilities and do not miss them.

In the literatures reviewed the researchers compared different testing methodologies by using the criteria covered attacks, test case generation algorithm, white box or black box, test case or test data, source of test case, tools and automation, reducing complexity, increasing efficiency, mitigating false positives, increasing knowledge of the tester, repository of knowledge, and postmortem evaluations.

From the above criteria some of them have been used in this research and others were not used as explained below.

Criteria that were not used from other researchers

- As this is black box security testing the criteria white or black box wouldn't be applicable in this research.
- The methodologies compared in this research are not for generating test cases so test case generation algorithm, test case or test data, source of test case would not also applicable in this research.
- Efficiency can't be used in this research as the test methodologies compared in this research doesn't cover equal amount of attacks hence some of them might take more time than others but taking longer time might enable them to uncover more attacks.

Criteria that were used directly or indirectly from other researchers

- Covered attack- is decomposed as target coverage and technical controls assessment comparison
- Reducing Complexity- is replaced with explainable
- Mitigating false positives- is replaced with repeatable as they mean the same
- Increasing knowledge of the tester- this is also used as it is in this research
- Repository of knowledge and Postmortem evaluation is part of the holistic process

In addition to the criteria used by other researchers the criteria manual and automated test, checklist, and futuristic were used in this research. In general the criteria used in this research are as explained below.

1. Holistic Process- Should cover all the processes in the black box testing process
2. Target coverage- should have the methodology to test all the targets technology that protect the web application
3. Repeatable- should avoid false positive results
4. Explainable- the methodology should be easy to understand and explain
5. Technical controls assessment coverage-should cover as much as possible the known attacks

6. Manual and Automated testing- should provide a way to conduct both manual and automated tests
7. Increasing knowledge of the tester- if the knowledge of the tester increases by using the test methodologies
8. Checklist-should provide a checklist of the testing methodology
9. Futuristic- should have a direction to test for new attack types

The criteria can be grouped as non technical and technical criteria. The technical criteria are those criteria that enable the tester to find vulnerabilities using the methods described in it. Except the technical controls assessment coverage the others has been categorized in the non technical criteria.

3.3. Methodology Development

After knowing the strengths and shortcomings of those methodologies an completed testing methodology was developed based on them, from other resources and personal experience in the area.

The test methodologies was compared using the criteria set in section 3.2 .To compare the non technical controls or technical criteria controls, first each methodology is summarized in table. The summary describes the objectives of the testing methodologies, the method name, how to conduct and in the technical controls methods comparison countermeasures are also incorporated. Then each methodology is compared in another table which contains the objectives, the method name, countermeasure, tools to conduct, references and the test methodology. Under each test methodology a Y symbol indicates that the testing methodology contains the stated method and an N symbol indicates the methodology doesn't contain the stated method. Then the methods are summarized under each controls or stages.

3.4. Evaluation

The technical controls were assessed on the four Ethiopian universities websites and two intentionally vulnerable websites. First each of the methods was analyzed against the targets. Sample of the vulnerabilities were also reported. Using the vulnerabilities identified an exploit

was attempted on the targets. The vulnerability assessment and exploit indicate the methods can work any target of assessments.

CHAPTER FOUR

NON TECHNICAL CONTROLS ASSESSMENT COMPARISON & DEVELOPMENT

Out of the nine criteria set in section 3.2, eight of them can be considered as non technical criteria but technical controls is different from the rest because it will assess each of the technologies that are expected the web application to use. Hence the non technical controls criteria will be used to compare the five testing methodologies and develop new non technical methodologies in this chapter and the technical controls will be used in the next chapter.

4.1. Holistic Process

Prince2 is one of the best known project management methods in the world. Prince2 can help to manage any project and hence using it to manage black box web application security assessment will be possible.

According to Prince2 project management method should manage cost, timescales, quality, scope, risk and benefits. The four main phases of Prince2 are pre-project stage, initiation stage, sub sequent delivery stage and final delivery stage.

As prince2 permits tailoring, the project management will be tailored to black box penetration testing. Since most of the testing methodologies combine both the pre-project stage and initiation stage, they are also considered as one in this research and hence this stage is named as Pre-assessment stage. The Sub sequent delivery stage of prince2 is named the assessment stage. The final delivery stage in black box penetration testing is reporting and presentation. As result, this research tailors prince2 into three phases as Pre-assessment, Assessment, and Post-Assessment.

4.1.1. Pre-assessment

Though most of penetration testing methodologies encompass most of the above phases, they have different components at each phase. As it can be seen in the methodologies literature review, each of the five methodologies has different methods at the pre-assessment phase.

Note: To make a comparison between the different methodologies some of the phases of the different testing methodologies has been re-organized. For example- Data handling in NIST is at the execution stage, however, it has been included here at the Pre-assessment stage.

4.1.1.1. Pre-assessment Comparison

I) Scope formulation

After the project initiation stage Prince2 recommends to formulate the scope. Similarly most of the testing methodologies also proposed formulation of the scope of test. This will help the penetration tester to avoid scope creep. However, the details of the scope varies from one methodology to another. The scope to be formulated should contain the parameters described in the table 3.

No.	Expected output	How to conduct	Standards that have the procedure					Remarks
			OWASP	NIST	PTES	ISSAF	ISACA	
1.1.	➢ Primary and secondary objectives of the test	➢ Written objectives from client ➢ Understanding business requirements if it can be provided ➢ The objectives should be Specific, Measurable, achievable, Realistic, and Time based (SMART)	**Y**	**Y**	**Y**	**Y**	**N**	In black box web application testing, the business requirements may not be provided
	➢ Techniques to be applied	➢ Discuss with the client if the technique to be applied is vulnerability identification, exploitation, post exploitation, and compliance	**N**	**Y**	**Y**	**Y**	**Y**	Though PTES and ISACA have mentioned selection of techniques to be applied, it is NIST that has the detail information. So the tester should refer NIST more on this point.

1.2.	➢ Where the test to be conducted	➢ Discuss with the client if the target is in production or test environment ➢ Discuss the possibility of deploying the target in test environment	**N**	**Y**	**N**	**N**	**N**	The test might be conducted on test environment or production. It is better if the customer prepares a similar environment as the production so that risks are minimized.
1.3.	➢ Number and type of targets	➢ Is the test going to incorporate the infrastructure where the application is hosted? ➢ Asking the following questions for web applications i. How many web applications are being assessed ii. How many login systems are being assessed iii. How many static pages are being assessed iv. How many dynamic pages are being assessed	**N**	**N**	**Y**	**Y**	**N**	During black box testing, the client may only provide the URL and hence the assessor may not obtain an answer to such questions. However, the assessor may determine this information during information gathering stage. In addition to that, though the assessor focus is on web application, it is mandatory to incorporate the infrastructure that the web application traverses.
1.4.	➢ Determine the type of data, and deliverable	➢ Identify the type of data that the customers wants to incorporate ➢ Agree on the type of deliverables	**N**	**N**	**N**	**Y**	**N**	Type of data and deliverables

Table 3: Scoping Comparison & Development

II) Project Management

After formulating the scope of the test the client and the tester should agree on how to manage the test to be conducted. The main concerns for a penetration tester and the client are how to handle the evidence; the start and end date since this will have an impact on the project budget; the assumptions and risks; and the project leaders and members should be identified.

No.	Expected output	How to conduct	Standards that have the procedure					Remarks
			OWASP	NIST	PTES	ISSAF	ISACA	
1.5.	➢ When the test should be conducted ➢ The start and end dates of the test	➢ Discuss with the client if the test can be conducted during business hours? ➢ Discuss and arrange the expected start and end dates	**N**	**N**	**Y**	**Y**	**N**	Black box security assessment of web application during working hours might damage the web application and hinder providing service. As a result, it is recommended to conduct during non-business hours or else to conduct it in a test environment.
1.6.	➢ Identify the assumptions and risks	➢ State the main assumptions and the associated risks ➢ Provide mitigation for the identified risks	**N**	**N**	**N**	**Y**	**N**	The assessor can develop the risks from the risks that are imposed by the assessor lack of knowledge, the risks that are posed while exploitation etc

1.7.	➢ Agree on lines of communication and meeting schedule	➢ Assigning emergency contact person and sharing his contact information ➢ Listing the participant testers ➢ Agreeing on incident reporting process ➢ Agreeing on status report frequency ➢ Agreeing on the type of encryption to be used for communication during testing ➢ Conduct Assessment meeting at mid-planning and end-of-planning, kick-off, progress meetings, emergency, assessment end meeting	**N**	**N**	**Y**	**Y**	**N**	Both PTES and NIST provide detailed information on this point so the assessor is recommended to review them when designing the lines of communications and meeting schedules.

1.8.	➢ Evidence Handling	➢ The assessor should have a log during the assessment ➢ The assessor should handle evidences obtained during assessment in a secured way and should protect it from being accessed by other people ➢ The assessor should transmit data if possible in person or otherwise through strong encryption mechanisms	**N**	**Y**	**Y**	**N**	**N**	Attackers can target the black box assessor than the target web application hence the evidences obtained by the assessor should be kept in a secure way. Though NIST's data handling on section 7.4 is on the test execution, it has been included at this stage as the tester should be aware of them before the test is conducted.
1.9.	➢ Develop responsibility matrix	➢ Identify who approves the deliverables, responsible for creating deliverables, notified when deliverable is complete, manages the deliverable, facilities timely resource allocation, responsible for acceptance and signoff, participate in archiving the deliverable	**N**	**N**	**N**	**Y**	**N**	The assessor can check the responsibility matrix through the assessment period. If new developments are made, the responsibility matrix should be modified accordingly
1.10.	➢ Work breakdown	➢ Divide the work into small chunks so that the responsible parties can take specific jobs	**N**	**N**	**N**	**Y**	**N**	Depending on the human resource the assessor project leader has the jobs should be divided into smaller chunks.

Table 4: Project Management

III) Tester Preparation

After agreement on project management and the scope formulation the penetration tester should prepare based on the requirements and the type of the target to be tested. This preparation should include acquiring the necessary knowledge and assisting tools. The tools to be used should be an updated version as new vulnerabilities would not be incorporated otherwise.

No.	Expected output	How to conduct	Standards that have the procedure					Remarks
			OWASP	NIST	PTES	ISSAF	ISACA	
1.11.	➢ Resource Identification and preparation	➢ Identify and prepare the necessary resources ➢ Update the necessary tools ➢ Assess knowledge gap of the assessors and fill the gap ➢ Select testing methodologies according to the objective and other requirements ➢ Customize tools, methodologies as per the target and the objective ➢ Develop new methodologies if the existing methodologies don't align with the requirement	**N**	**Y**	**Y**	**Y**	**Y**	The existing methodologies may not align with the requirement of the targets. Besides the assessor should not test all of the testing methodologies provided in this research or other methodologies as it should be customized as per the client requirement

1.12.	➢ Checking for legal violation	➢ The assessor should check if testing the target as per the objectives set doesn't violate any local or international laws.	**N**	**Y**	**Y**	**Y**	**Y**	As web application might be hosted in third part countries the assessor should not only check local laws but also international laws. If the target is hosted in another country, the client must bring permission to conduct the test.

IV) Project Kick-off

The assessor and the client should sign after finalizing assessors internal preparation as described in the table below

No.	**Expected output**	**How to conduct**	**Standards that have the procedure**					**Remarks**
			OWASP	**NIST**	**PTES**	**ISSAF**	**ISACA**	
1.13.	➢ Launching project	➢ The client and the assessor should sign an agreement. The agreement should include rule of engagement ➢ Kick-off meeting should be hold ➢ The assessor start as per the start date	**N**	**N**	**N**	**Y**	**N**	

Table 5 : Project Kick-Off

4.1.1.2. Summary of pre-assessment

Customer and black box web application assessor can start the assessment if both parties agree first on the objectives of the testing, the techniques to be applied, where the test to be conducted, how many targets are going to be assessed, when the test to be conducted, type of data and deliverables. Those factors will be used to formulate the scope of the test.

After determining the scope of the test, the assessor and the client should agree on project management. The project management should included agreement on evidence handling, expected start and end dates, lines of communication, assumptions and risks of the project, responsibility matrix, and detail activities work break down.

The third stage in the pre-assessment phase is an internal preparation stage for the assessor. Before signing an agreement with the client, the tester should first check legal issues and prepares the necessary resource. The tester needs to check if it is legal to conduct such a test and if the client has a legal ground to have the test conducted for it. There may be targets that the client requests to be tested but not owned by itself, hence the client should provide a legal foundation for the devices that doesn't own but wants to be tested. The black box tester should also prepare according to the client requirement. As a result the tester should check if the test can be conducted with the existing knowledge or needs further reading and training, the testing methodology should also be tailored as per the objectives of the clients, and the required resources should be identified and updated.

After the project scope determination and project management agreement, the black box web application assessor and the client should sign an agreement to kick off the project. The assessor should be ethical to fulfill the duties as per the agreement.

From the 13 methods at the pre-assessment stage, OWASP only provides a method for the objective development. ISACA has 3 NIST has 5 , PTES has 9 methods, and ISSAF has 11 methods. It can be seen from the above that one methodology is not enough as its misses critical methods.

4.1.2. Assessment

The comparison of security testing methodologies assessment phase can be divided into principles comparison and stage comparison.

4.1.2.1. Assessment stage comparison and development

The assessment stage comparison is organized by first determining the similarities between the methodologies, then the difference and additional methods that should be incorporated will follow. Similar to the other methods development, the assessment stage comparison was developed by first summarizing the assessment stage of each methodologies then the comparison and development followed it.

4.1.2.2. Similarities between the methodologies and its analysis

- **Information gathering** - All the testing methodologies have an information gathering though PTES named it as intelligence gathering and ISSAF incorporated it in the penetration testing. The methodologies included the information gathering because without gathering an information the tester would waste his/her time on tests that are not related to the target. Hence, this should be included in black box web application security assessment. The information gathering can be conducted in active/passive. The comparison details of information gathering will be discussed in the next stage of this research.
- **Vulnerability identification and analysis**- OWASP's vulnerability identification and analysis is in the controls assessment. NIST has named it as vulnerability scanning. PTES has a vulnerability analysis and named it as passive and active scanning. ISSAF has included vulnerability scanning in penetration testing. ISSAF's vulnerability scanning has six steps. The five steps are vulnerability identification through the gathered information, vulnerability scanning through automated tools, identify undisclosed vulnerabilities, make a list of vulnerabilities found, perform false positive and false negative verification, make a final list of vulnerabilities and recommend countermeasure. From the above discussion it can be concluded that the five methodologies have a vulnerability scanning and analysis. As the main goal of black box web application security assessor is to identify vulnerabilities it has been included as one phase in this

research. The vulnerability identification should start from the vulnerabilities obtained during information gathering. This should be followed by manual and automated vulnerability identification and scanning for known vulnerabilities. Since there may be new vulnerabilities that are not identified during the previous steps it may be necessary to identify new vulnerabilities. However, the vulnerability validation that has been suggested by ISSAF and PTES will be discussed in the differences between the standards.

- **Exploitation**- similar to vulnerability analysis and identification both OWASP and ISACA has included this stage in the controls assessment. NIST has named this as vulnerability validation (password cracking and penetration testing). PTES has both vulnerability validation and exploitation. However, PTES's vulnerability validation is different from NIST's vulnerability validation as NIST's vulnerability validation is similar to PTES's exploitation. PTES's vulnerability validation will be discussed in the difference between the methodologies. ISSAF exploitation is in the technical penetration testing and it is named as penetration. According to ISSAF to conduct penetration the tester needs to find proof of concept code/tool, test proof of concept code/tool, write your own proof of concept code/tool, use proof of concept code/tool. It is known that black box assessors need to have a proof that the vulnerability identified can be exploited. Hence it would be mandatory to include the exploitation phase in black box web application assessment. Though ISSAF has four steps in conducting exploitation, it had not incorporated manual exploitation. For example, guessing user name and password can be done manually or through automated tools. So depending on the target and type of exploit manual testing should be incorporated. Hence, the exploitation should include the exploitation phase with five steps.
- Information gathering, vulnerability identification and analysis, and exploitation are the three phases that are common in the five testing methodologies. However, there are additional phases included in some of the testing methodologies and hence are described below.

4.1.2.3. Differences between the methodologies and its analysis

- **Target Identification and analysis, network port and service identification** have been included before vulnerability scanning in NIST. Similar to NIST, ISSAF has network mapping. However, the other methodologies included this in information gathering and hence in this research those points have been included in information gathering.
- **Threat modeling**- PTES included threat modeling as one of the phases during testing. However, in black box web application assessment it is difficult to include threat modeling as the black box assessor has no full knowledge of the target. It is through the assessment period the assessor understand the target in depth. However, for each target that are identified, it is possible to build a threat modeling. This would be less efficient and hence has not been included in this research as one of the phases of black box web application security assessment.
- **Vulnerability Validation**- vulnerability validation can be done before exploitation and during exploitation. Vulnerability validation in NIST is exploitation and password cracking. ISSAF has included false positive and false negative analysis as one step during vulnerability analysis. However, except for the detail understanding of the target, the details described in the appendix of the framework are specific and some of them require additional details to be used by the assessor and hence only the idea of validation through understanding of the target is taken from the framework in this research. PTES has detailed steps necessary for vulnerability validation before exploitation. The vulnerability validation suggested by PTES are correlation between tools, manual/protocol analysis, public research (validation of vulnerabilities in public vulnerability databases, and vendor advisory), exploit database and frameworks, default/common password , system hardening guides and private research through fuzzing. Searching for exploit database and framework suggested by PTES can be used for verification during exploitation and hence has been excluded in this phase. Default/common password and system hardening guides should be used to identify vulnerabilities but not to verify them and hence has been excluded in this phase. As described in ISSAF one of the ways for verifying the vulnerability identified is to understand the target in depth. If the assessor understand the target in depth, it is possible to omit false positive vulnerabilities as those vulnerabilities may not be related to the target. The assessor can also validate vulnerabilities from

vendor advisory websites or other resources. Different tools might provide different vulnerabilities for the same target so correlating vulnerabilities and comparing them using atleast three tools would be a recommended step to validate vulnerabilities from tools. After filtering some of the vulnerabilities using the techniques described above, there are may be still some false positive vulnerabilities. To avoid more false positive vulnerabilities the assessor should conduct manual assessment. All the above processes are useful for avoiding false positive vulnerabilities but there may also be vulnerabilities that are not identified by manual or automated tools. Such vulnerabilities that are not identified by manual or automated tools are called false negative vulnerabilities. False negative vulnerabilities can be identified by fuzzing or reverse engineering of the target and making analysis of the target. In conclusion vulnerability validation should be conducted before and after exploitation. Before exploitation the vulnerability validation should take in depth understanding of the target and avoiding the vulnerabilities identified from the assessor understanding, checking in vendor advisory, correlation between tools, manual assessment, fuzzing and reverse engineering to identify zero day vulnerabilities as necessary steps. It should be noted that manual assessment and vendor advisor can also be used to identify vulnerabilities. In addition to this, zero day vulnerabilities analysis should only be used for critical targets as it requires large resource, time and expert assessor.

- **Privilege escalation and maintaining access** - the access obtained during the exploitation phase may not be a root access and as a result the assessor's exploitation might be limited, hence it would be mandatory to escalate privilege. Similarly, the client might want a live demo of the compromised target, it may be necessary to maintain access to the compromised target until the test is finalized. It would also be necessary to pivot from one compromised target to another target. Only NIST, PTES and ISSAF has included privilege escalation and maintaining access as one of their phases in security assessment.
- **Compromising remote users**- ISSAF has included compromising remote users as one phase of the testing procedure. However, in black box security assessment it is highly probable the test is to be conducted from remote location hence this is part of the exploitation phase described in the similarities between standards.

- **Controls assessment**- OWASP, ISSAF and ISACA have controls assessment in their testing methodologies. As described in the above comparison between the methodologies both OWASP and ISACA have included vulnerability analysis and exploitation as parts of the controls assessments. ISSAF however states that each controls should be assessed by gathering information, and other phases described in the above similarities and difference between the methodologies. The process suggested by ISSAF would enable to identify vulnerabilities of each controls and hence it has been selected as a methodology in this research.

4.1.2.4. Limitations of the five methodologies

Information gathering- the methodologies suggest the output of the information gathering and mapping stage to be the input of vulnerability identification analysis stage however vulnerabilities might be identified at the information gathering stage, targets might be exploited from the information provided at the information stage, privilege might be escalated if for example hardcoded accounts are obtained at the information gathering stage. Hence the outputs of information gathering stage should not only the inputs of vulnerability identification but also the inputs of vulnerability validation and privilege escalation

Enumerating further-The understanding of the target that is obtained during the information gathering phase might be enhanced during vulnerability analysis, exploitation, privilege escalation and other phases. The assessor as a result should update the target during each phases of the assessment. This will help the assessor to conduct additional tests that was not planned before. Both ISSAF and NIST have included enumerating further after gaining access and privilege escalation. However, the methodologies assume that this should be done after exploitation.

Quality Assurance- the methodologies should have included quality assurance as one component of the test as beginner testers might not have the detailed knowledge on testing and their assessment might contain both false positive and negative results.

4.1.2.5. Summary of Assessment Comparison

Based on the above similarities and difference between the methodologies the assessment phase will have an

Information gathering stage- all the necessary information of the target will be identified and mapped. This includes the information about the target web application, the network security controls. After the information is gathered about the target the target application and its infrastructure should be mapped. The information gathered should be quality assured by the lead assessor or another expert in the area.

Vulnerability Identification- the Quality assured gathered information will be used an input to vulnerability identification. The assessor will test the technologies identified during the information gathering. However, additional targets might be identified during this stage hence it should be feed to the information gathering stage and further information should be gathered upon it. The vulnerability identification can be done manually, using automated tools and a research for new vulnerabilities. The research for new vulnerabilities can be zero day vulnerability identification through fuzzing, reverse engineering or other techniques. However, researching for new vulnerabilities can time and resource as a result the assessor and the client should first agree the necessities of conducting the research. Similar to information gathering the outputs of vulnerability identification should also go to quality assurance. As the next stage of black box web application is vulnerability validation, the quality assurance at this stage is only checking if the tester has followed the necessary methods to identify vulnerabilities

Vulnerability Validation- includes identifying false positive and false negative vulnerabilities. The false positive vulnerabilities validation should be done by verifying vulnerabilities from information gathering (understanding of the target), comparing the vulnerabilities outputs of multiple tools for example based on CVE, and exploitation. However since only the tester should only conduct sample of exploitations, validation through understanding of target and comparing the outputs of the tools should come first before the exploitation. As this research proposes manual testing and researching as additional ways of vulnerability identification then false negative vulnerabilities are expected to be discovered using those means. Besides, false negative vulnerabilities will also be discovered during the quality assurance phase. During the

exploitation phase new targets might be discovered, this should pass to the information gathering and further vulnerability identification should be conducted based on the value of the target.

Privilege Escalation- After certain exploits are done the privilege obtained may not enable the assessor to control the target up to the level necessary to tune the target in the interest of the assessor. Hence, the assessor should escalate the privilege obtained by analyzing more about the target or installing additional tools.

Quality assurance- the assessment results should be approved by another tester at each phases of the assessment.

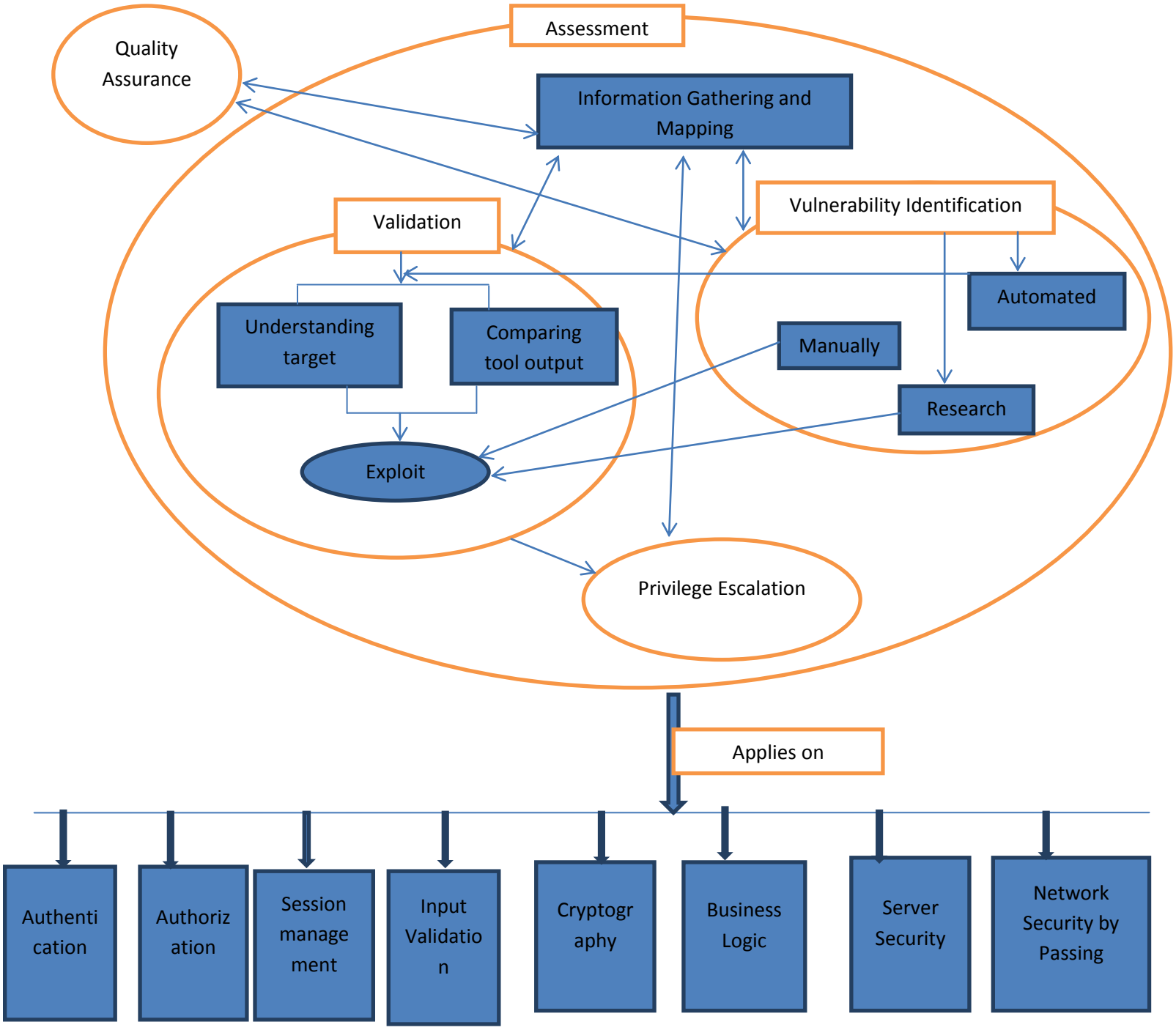


Figure 7. Assessment Stage Methodology

4.1.3. Post-Assessment

The post assessment stage should enable the tester to analyze the outputs of the assessment and accordingly provide countermeasures for the vulnerabilities identified. After the vulnerabilities are identified the tester should be able to calculate the risks. Finally the tester should be able to write a report, presents the results and follow up the mitigation of vulnerabilities. Hence the post assessment should have the necessary methods for countermeasure selection, risk calculation, reporting, presentation and follow up.

4.1.3.1. Analysis of Post Assessment

- **Countermeasures**- If the security assessment is for compliance depending on the standard the target expected to be complied with the standard may or may not provide a countermeasure. However, as security assessment is to identify vulnerabilities so that the client will cover them, it is necessary to provide a countermeasure. The countermeasure provided by the black box web application may be directly implemented or the client might implement other controls. Based on this, ISACA and PTES has not clearly defined the countermeasure/ mitigation recommendation. However, OWASP, NIST and ISSAF has a countermeasure for most of vulnerabilities identified though there are some vulnerabilities that do not have a countermeasure provided by the methodologies.
- **Reporting**-The final output of a security assessor is either a compliance certificate or a report. The compliance certificate can be if the assessor is complying against a specific standard. So it would be mandatory for black box web application assessor to include reporting. The methodologies that are compared in this research have used different formats. Only NIST has not provided the format of a report. It is mandatory to include an executive summary so that the client top management can understand the test results. That is why PTES, OWASP and ISSAF has included an executive summary. Since an executive summary should include a summary of the core report, the executive summary should include short background of the assessment, scope of the assessment, objectives of the assessment, summary of findings with graphical chart, the root causes of the vulnerabilities, overall risk level of the target, and summary of the recommendation. After the executive summary the details of the assessment should be written. PTES suggested to write a report for each phases of its assessment methodologies. However, this is not recommended as the report would be too much and unnecessary information

would be included in the report. Hence, PTES report mechanism is not suitable for clients and for assessors. The report should also contain an introduction part. The introduction part should contain a summary of the targets controls, the strengths of the existing security controls, and summary of methodology used. The third part of the report should contain the findings of the assessment. The findings of the assessment should contain vulnerability name, vulnerability description, risk rating, impact of the vulnerabilities, countermeasures, and sample exploits for each of the vulnerability identified. However, as ISSAF suggested a verbal reporting might be useful in addition to the technical reporting as critical vulnerabilities might be exploited until the final report is prepared. The report produced should be reviewed by a lead security assessor before the final reports is sent to a client.

- **Presentation**- After the report is prepared and provided it might be necessary to provide a presentation for managers and technical team. When presenting the report to managers of client, presenting the executive summary is enough as the executive summary summarizes the core causes of the vulnerabilities, summary of the recommendation. Presenting every findings to the technical team might be cumbersome however, presenting vulnerabilities that are high, very high or critical vulnerabilities is necessary. Since the report will be provided for the assessors before the presentation the assessors will ask if they do not understand the report produced.

- **Follow up**- the assessor should fellow up if the client has covered the vulnerabilities using a reassessment.
- **Cleanup**- the assessor should destroy all the evidences gathered during the assessment phase after the presentation and report delivery. In addition to that the assessor should return all the changes made during the assessment to their previous settings. If changing to the previous settings is not possible, the black box assessor should inform the client the changes made during the assessment phase.

In conclusion the three phases will interact as shown in the following figure below.

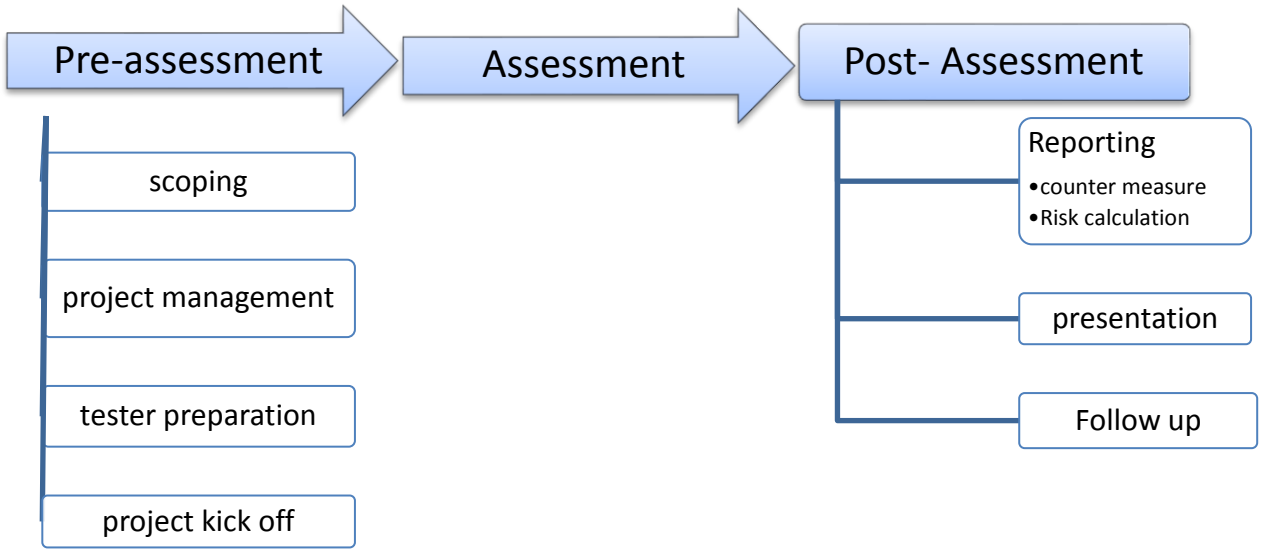


Figure 8. Stages of Testing Methodology

4.2. Other comparison Criteria

4.2.1. Target coverage

As stated in the web application technology target selection the methodologies should have a testing technique for the following targets.

No	Category	Technologies
1.	Web presentation	HTML(hyperlinks, CSS, forms), Activex, applet, javascript, and other client script
2.	Web application Platforms	ASP.NET and Java
3.	Content Management Systems	Joomal, wordpress, koboo etc
4.	Scripting Languages	PHP, VBScript, Perl
5.	HTTP requests and Response	HTTP Methods, Status Codes, Banners, Cookies,

		Sessions, Encoding, message body etc
6.	Authentication	Active Directory and other technologies
7.	Browser extensions	Java applets, ActiveX controls, Flash objects, silverlight objects
8.	Client Side Controls	Validation (such as javascripts),
9.	VPN	L2F, IPSec, PPTP, L2TP
10.	Network security devices	Honeypots, Proxies, load balancers, firewall, IPS/IDS
11.	Web server	Apache, IIS, or Netscape Enterprise and others
12.	Database	MS-SQL, Oracle, and MySQL
13.	Other backend technologies	File systems, SOAP-based web services, and directory services
14.	Remote Management	telnet, SSH, FTP and others

Table 6: Target Coverage

When comparing the technology target coverage, the strength of the testing techniques was not evaluated. This would be evaluated in the subsequent phases of this research. However, the main criteria in here is if the testing methodologies have at least a minimum technique to test the specified technologies.

No	Category	OWASP	NIST	PTES	ISSAF	ISACA
1.	Web presentation	Y	N	N	Y	N
2.	Web application Platforms	Y	N	N	N	N
3.	Content Management Systems	N	N	N	N	N
4.	Scripting Languages	Y	N	Y	Y	N
5.	HTTP requests and Response	Y	N	Y	Y	Y
6.	Authentication	Y	Y	Y	Y	Y
7.	Browser extensions	Y	N	N	N	Y
8.	Client Side Controls	Y	N	Y	Y	Y
9.	VPN	N	N	Y	Y	N
10.	Network security devices	Y	Y	Y	Y	N
11.	Web server	Y	N	Y	Y	Y
12.	Database	Y	N	Y	Y	Y
13.	Other backend technologies	Y	N	Y	Y	N
14.	Remote Management	N	Y	Y	Y	N

Table 7: Target Coverage Comparison table

4.2.1.1. Target Coverage Comparison Analysis

Though OWASP, PTES, and ISSAF have a methodology for testing many of the target technologies as shown in the above table, none of them have placed as a requirement for security tester to include target in addition to the web application. PTES and ISSAF have methodologies to test different category of a technologies. As a result, the black box web application security assessor might only assess the web application part. But vulnerabilities that lead to compromise of the web application through the infrastructure must also be assessed as the controls placed by

web application alone can't protect the web application. Hence, in this research it is proposed to for black box web application assessor to conduct the test for technologies that protect the web application.

Out of the 14 expected technologies to be assessed by black box web application assessor OWASP has 11 methodologies, PTES has 10 methodologies, ISSAF has 9 methodologies, ISACA has 6 methodologies, NIST SP 800-115 has three methodologies. From the results, it can be seen that one methodologies is not enough in conducting black box web application though some of the methodologies especially OWASP cover most of the target technologies.

4.2.2. Repeatable

Black box web application security methodologies should have a way to mitigate false positive results. If the methodology is dependent on tools it is highly likely it won't be repeatable. As described in the holistic comparison of the methodologies, the different methodologies have set different validation mechanisms to mitigate false positive results.

As described in the assessment stage comparison of the holistic comparison, there are two ways of mitigating false positives. The first way of mitigating false positive is before exploitation and the second is during exploitation. Before exploitation only PTES and ISSAF have placed a way to mitigate, however, PTES's way of removing false positive is more efficient.

All the other methodologies have a technique to mitigate false positive during exploitation. The methodologies suggest that false positive vulnerabilities can be identified if that vulnerability can't be exploited. However, there are information vulnerabilities that don't lead to directly attack the target, and checking to exploit for every vulnerabilities found is cumbersome as it requires large resource and time. The main goal of the assessor should also be to identify vulnerabilities but not to exploit each one of them. Hence, the black box web application assessor is advised to conduct false positive vulnerability reduction before exploitation though exploitation can assure the client that indeed the vulnerabilities can lead the target to be exploited.

4.2.3. Explainable

A methodology can be explainable or easy to understand by the assessor if the methodology explains why the technique to conduct a specific test is necessary, the associated vulnerabilities, the details to conduct the assessment, examples of assessment results, the way to analyze the results, the necessary tools that can assist the test, and additional references.

Based on those criteria the methodologies comparison is as shown below

No	The methodology has	OWASP	NIST	PTES	ISSAF	ISACA
1.	Objectives of the test	Y	Y	Y	Y	N
2.	Vulnerabilities	Y	Y	Y	Y	N
3.	How to conduct the test	Y	N	Y	Y	Y
4.	Examples	Y	N	Y	Y	N
5.	How to analyze results	Y	N	Y	Y	N
6.	Assisting tools	Y	Y	Y	Y	Y
7.	Additional references	Y	N	N	Y	N

Table 8: Explainable comparison results

From the above table it can be concluded that both OWASP and ISSAF are explainable according to the comparison points. However, this doesn't mean that those methodologies have explained each of the points at each testing technique. ISSAF have many testing techniques that don't have some of the points mentioned in the above table. Contrary to OWASP and ISSAF, ISACA only sets what to conduct and how and provides some assisting tools. PTES is similar to both OWASP and ISSAF as it meets six of the seven comparison points. Since NIST SP 800-115 is generic, it is not expected to be generic and directly be used.

4.2.4. Futuristic

Every day new technologies are developing and vulnerabilities are released for those new technologies. In addition to this, the existing technologies might be vulnerable to new vulnerability types. As a result the methodologies should have the future of looking for future

technologies and new vulnerabilities for the existing technologies used by web applications otherwise they need to update themselves frequently.

One of the ways to look at future technologies in black box web application security testing is to categorize technologies and provide guidance on future technologies security testing that will be in that category. For example, Content Management System have some common future so the black web application tester should develop the testing methodologies should have a common methodologies for content management system. Of curse, there are some vulnerabilities specific to a certain content management system. Though OWASP's categorization is not complete, the testing methodologies have been categorized into different domains. For example- the identify management testing might work for different technologies. However, none of the rest testing methodologies have categorized the technologies of web application hence they may not be used directly for future technologies.

The testing solutions to be used should also be updated regularly. So in the preparation stage the testing methodologies should put updating testing solutions as a requirement before the test is conducted. However, none of the testing methodologies have a requirement in the planning phase that requires the testing to be updated before testing proceeds.

The testing methodologies should also put for references that focus on security vulnerabilities for new technologies. The reference can be vendor advisories, Common Vulnerability exposure (CVE) and other references. It is only PTES that has such a recommendation.

Finally the testing methodologies should suggest for the black box web application security assessor to conduct research in the areas of new technologies and new vulnerabilities. Again in here only PTES placed a requirement to conduct research during assessment.

4.2.5. Manual and automated

Conducting a black box web application security assessment using automated tools can reduce the time and the resource required to assess the target. However, as testing solutions don't understand the business logic of the target to be tested, it is highly probable that for the existence of false positive and false negative vulnerabilities. So the methodologies should also provide a

way to analyze the automated testing tools results and a way to discover vulnerabilities manually. So there must be a balance between manual and automated testing methodologies.

Among the methodologies compared in this research, OWASP's testing guide provides both manual and automated tests techniques. Contrary to OWASP, NIST depends on automated tools. Though PTES provides for some of technique to test manually, it depends more on automated tools. Similar to OWASP, ISSAF and ISACA has also manual and automated testing technique.

In conclusion the testing methodologies can be ranked interms of manual and automated criteria from high to low as OWASP, ISSAF, ISACA, PTES and NIST.

4.2.6. Checklist

Checklists can assist the assessor not to forgot the testing methodologies and to proceed sequentially whenever it is required for sequential testing. Among the testing methodologies, OWASP, NIST and ISSAF have a checklist that enables the assessor to easily follow them. In addition to the checklists that are necessary during the assessment phase, ISSAF has also examples of documents that are necessary during the pre-assessment and post-assessment phases.

4.2.7. Increasing Knowledge of the tester

As explained previously PTES, and NIST are more dependent on tools and hence the tester will be more dependent on the tools without the indepth knowledge of what is going on. ISACA doesn't provide detailed methods on how to conduct black box penetration testing hence it would not enable the tester to know more about black box web application. Though ISSAF is also dependent on tools, it tries to explain more the outputs of the test by providing examples on analysis of the output. OWASP provides more in depth knowledge of how to conduct black box web application however it only focuses on web application. In conclusion, OWASP and ISSAF might increase of the tester but they have their own limitation. PTES will give the knowledge of how to conduct black box web application by using automated tools.

CHAPTER FIVE

TECHNICAL CONTROL

ASSESSMENT COMPARISON & DEVELOPMENT

As described in the holistic comparison the technical security assessment should have information gathering, vulnerability analysis, vulnerability validation, exploitation, privilege escalation, and should be applied on web application controls.

In this section the technical controls assessments methods that were developed by synthesizing the methodologies in to black box web application penetration testing methods; comparing the methodologies and hence identifying the strength and limitation of each the methods; then adding additional methods from already proven methods to minimize the limitations of the methods used in this research were used to develop each of the methods. A sample of the tables used in summarizing, comparing and developing the methodologies has been described in the criteria holistic comparison criteria.

The technical controls assessment are categorized as information gathering, input validation, authentication, authorization, business logic testing, session management, cryptography testing, and network security bypassing.

In comparing each of the techniques from the methodologies the factors taken were technology target coverage as described in the previous section, attack coverage (the maximum number of vulnerabilities that can be identified) and the effectiveness of the technique(if the technique described can assist to identify vulnerabilities).

5.1. Information gathering

This stage enables black box web application security testers to understand the target in detail. Black box web application testers can gather information about the target from search engines or by directly interacting with the targets. In this research the information gathering is not only about the web application but also about the infrastructure that host and protect it. Information about the web application and possible network security devices will be gathered at this stage.

Information Gathering Methods

a. Web application

1. Gather information about the target web application- at this stage gather sensitive data by first checking if the target has a web presence in the search engines as in black box web application the website address may not be provided by the client. The tester can also use the search engines to identify additional information about the target web application and the related users. The tester can use social networks, newsgroup, mailing lists, chart rooms, forum search, personal domain names, personal activities, archived information. Use advanced search engines to discover more information. The tools that can be used at this stage are Baidu, binsearch.info, Bing, Duch Duck Go, ixquick/Startpage, Google, Shodan, PunkSpider and google hacking database. The countermeasures that could be proposed include carefully consider the sensitivity of design and configuration information before it is posted online, periodically review the sensitivity of existing design and configuration information that is posted online. Testers can read additional references in PTES, OWASP's OTG-INFO-001, ISSAF's code 9.1.1, 9.1.8, 9.1.13, and 9.1.14.
2. Identify and understand the functionality of the web application. State what users can do with the web application. Intercept the requests using proxy and visit every link of the application. Understand the requests and response, the kinds of data being submitted and the way the client communicates with the server.
3. Once the target web presence is identified and some juicy information is gathered from search engines, the tester should identify the entry points and admin pages. This includes user input links, query string parameters, POST data, cookies, and

other HTTP headers. This can be done with the help of web crawlers or manually. Additional information can be obtained from OWASP OTG-INFO-006, OTG-CONFIG-005, and ISACA. The developer should restrict the entry points based on privilege.

4. As there are URL's that can't be accessed using standard ports, the tester should identify non standard URL and URL that have different ports through vulnerability scanners and port scanners such as nmap. If the URL are for restricted users implement an authentication. Additional information can be obtained at OWASP's OTG-INFO-004.
5. Identify the application parameters and the associated values. The application might also hide or disable some parameters and contents as a result the tester should identify them. Discover how the application handles for nonexistent content.
6. Identify the existence of hidden debug parameters such as debug=true. Additional information can be found at Hacking Exposed web application
7. The tester should search for sensitive information from meta data and comments. Additional information can be obtained at ISSAF's U.6.9 and U.6.10, PTES, and OWASP's OTG-INFO-005.
8. Web applications have common file installations such as readme, todo, changes and install.txt that can reveal sensitive information for the attacks hence those should be investigated by crawlers or manually
9. Similar to common files the tester should also identify if common directories are available in the application. Additional information is available at web application hackers handbook.
10. The file types supported by the application might reveal the technologies used by the application hence the tester should identify the type of files supported. The developer should restrict the upload of malicious files.
11. If the application permits authenticated users to access privileged information, the tester should identify the authentication technologies and locate all the authentication related functionality including login, registration, and account recovery. In addition to this, the authentication technologies used by the

application should be identified. Supportive tools include browser, proxies such as burpsuite. Testers can use ISACA and the web application hackers handbook.

12. The application might manage users using session management as a result the tester should understand how session management is handled by the application by identifying Session ID and cookies.
13. The assessor should identify the authorization levels and the user levels supported by the application.
14. Determine the type of encryption used by the application. Check if the parameters and their values are encrypted. Testers can read more at OWASP testing for weak cryptography and the web application hackers handbook.
15. Identify the technologies used by the web application including web presentation technologies, web application platform technologies, client side technologies and content management system technologies used by the web application
16. Fingerprint the methods used by the application as some of the methods are vulnerable. Additional information can be found at OTG-CONFIG-006.
17. Identify web presentation technologies such as HTML, Activex, applet, and javascript. Additional information can be found at Hacking Exposed web application
18. Identify web application platforms such as ASP.Net and java. Additional information can be found at Hacking Exposed web application
19. Identify content management systems such as Joomla, wordpress, kobo. Additional information can be found at Hacking Exposed web application
20. Fingerprint Query strings to gather information such as parameters, value of parameters, username, password, database queries. Additional information can be found at Hacking Exposed web application

b. Web server
    1. Identify the registrar information of the target web application including primary and secondary domain name server, contact person email address, phone number etc using tools such as whois, dig and nslookup. Additional information can be found at PTES and ISSAF 9.1.3

2. Perform zone transfer on primary and secondary ISP name server. In addition to this, perform reverse DNS lookup to determine all authoritative name servers using Dig, samspage, whois, dnsenum, nslookup and nmap. Disable reverse DNS at the server configuration to avoid the related vulnerabilities. Additional information can be found at ISSAF 9.1.23, 9.1.24, and OTG-INFO-004.
3. Check if the mail server and application server IP addresses in spam/attacker database lookup including spamhaus, spamcop, RBL, SANS ISC. Additional information can be found at ISSAF 9.1.3
4. Identify server types including web server, application server using banner grabbing and unexpected HTTP methods. Use tools such as Netcat, ttprint, httprecon, netcraft, Desenmascarame. Banner grabbing can be protected by implementing a false banner information. Additional references can be found OTG-INFO-002, U.6.1,Webapplication hacking exposed
5. Search systems/network survey sites- Gather Web-Server Information and find out what web server / operating system site is running, what is the IP Address and Net-block owner, the target uptime, operating system. Tools such as netcraft can be helpful. Change default banners provided by network services visible from the internet. Additional information can be found at ISSAF 9.1.10, 9.1.11.
6. Search for roboots.txt file to gather meta files for information leakage. Additional information can be found at OWASP OTG-INFO-003
7. Identify the administrator entry points used to manage the application. Administrators should avoid predictable admin web pages. Additional information can be found at OWASP OTG-INFO-005
8. Using port and service scanners identify the open ports and service running on the servers using tools such as nmap. Unnecessary ports should be closed.
9. Identify valid email accounts by connecting to the email server
10. Enumerate accounts including valid email accounts by connecting to the email server using tools such as netcat. Additional information can be found at

OWASP OTG-INFO-005, ISSAF 9.1.18, ISACA internet penetration testing. An intrusion detection systems might detect email gathering activities.

c. Database
    1. Depending on the database type use different tools to fingerprint the type of database. The tester might use sqlping, sqlat tools
    2. Fingerprint query strings to identify parameters, values of parameters, username , password and database queries

d. Proxies
    1. Detect proxies using TRACE request or CONNECT command

e. Load Balancers
    1. Detect the existence of load balancers using port scanners, timestamp analysis, comparing ETag and last modified, load balancer cookies and HTML source code. Additional information can be found at Hacking Exposed web application.

f. Firewall/edge router
    1. Using traceroute identify the existence of firewall. Notice that if icmp, udp and tcp packets are dropped then the request is being blocked by a firewall. Additional references can be found at ISSAF H.9.2
    2. Craft SYN packets using tools such as hping to identify packet based firewalls. Notice that if firewall exists icmp type 13 response will be obtained. Additional references can be found at ISSAF H.9.2
    3. Identify ports that are open, filtered, unfiltered by port scanners such as nmap, angryport. Additional references can be found at OWASP' OTG-INFO-004, ISSAF H.9.7, NIST, and PTES
    4. If the firewall exists use firewalk, hping or other tools to identify firewall rule set including open, closed, filtered ports. This can be mitigated by blocking the firewall to send out packets before the drop rule, and to send out icmp error messages. Additional references can be found at ISSAF H.11.1
    5. Identify the existence of web application firewall from cookies, HTTP response from crafted HTTP requests.

g. Honeypot

1. Detect honeypots from services running on the target server using vulnerability scanners. Or use tools such as send-safe honeypot, hunter, or hping

h. VPN
    1. Detect VPN by scanning for known ports or using tools such as ike-scan

5.2. Input validation testing

Users may enter malicious data, codes and files that enable them to attack the application. The application must implement proper validation mechanisms to mitigate such an attack. The vulnerabilities that can be obtained using this methodology include SQL injection, cross site scripting.

Input validation Testing Methods

1. From the information gathered at the information stage of this research review the identified and mapped user inputs.
2. Fuzz every parameter for input validation vulnerabilities including SQL injection, Cross site scripting. This can be done manually or using automated tools such as Burp intruder. Additional reference can be obtained from the web application hackers handbook.
3. Identify the parameters used in HTTP request and response using either crawlers or sending a request for each of the possible methods.
4. During SQL injection testing if the input attack strings generate any error messages, analyze the error messages and validate whether the application is vulnerable to SQL injection or not. If the error indicates the application is vulnerable to SQL injection then exploit the target manually or using automated tools. Test also specific SQL injection depending on the database type. The tester should read different references in how to conduct successful SQL injection but the web application hackers handbook and OWASP are a good starting point.
5. Identify parameters that are vulnerable to Reflected Cross site Scripting by inputting attack strings on each input. Identify any parameters values that have not modified the input attack strings. Using either automated tools or manually exploit vulnerable

parameters for reflected cross site scripting or permanent cross site scripting. Tools such as OWASP Xenotix XSS Exploit Framework, XSS-Proxy, and PHP Charset Encoder(PCE). Additional references can be found at OWASP's OTG-INPVAL-001.

6. Stored cross site scripting- Hijack users' sessions using the references in OWASP's OTG-INPVAL-002. Additional tools can be found at XSS Assistant, Backframe, BeEF, Hackvertor, and PHP Charset Encoder(PCE). Additional references can be found at OWASP's OTG-INPVAL-002.
7. Check for boundary testing by identifying the minimum and maximum limits and submitting data that is not in the limit as described in web application hackers handbook.
8. Test for injections such as SOAP, LDAP, ORM, XPATH, XML, SSL, IMAP/SMTP, code injection, command injection, buffer overflow, incubated vulnerability, HTTP Splitting, as described in OWASP's input validation testing from OTG-INPVAL-006 up to OTG-INPVAL-016 and the web application hackers handbook. The necessary tools and countermeasures are also described in OWASP input validation testing
9. Finally conduct a fuzzing testing to determine additional vulnerabilities that were not obtained using the above techniques

5.3. Authentication and administration testing

In this research web application authentication testing is finding the vulnerabilities of web application that starts from user registration attack up to when users enter credentials to the application. Various attacks can happen during user registration and authentication such as grabbing user credentials, registering using fake profile.

Authentication and administration testing methods

1. If the application uses username and password for authentication, collect possible usernames from error messages during input of invalid credentials and password reset; by trying to register for existing username; and use the possible usernames collected during information gathering. Additional references can be found at Web application hacking exposed, OTG-AUTHN-001,OTG-IDENT-004,OTG-IDENT-005.
2. Determine the implemented password policy in the application. Determine if the password policy has a minimum length limit; if it is mandatory to use a combination of

different characters set. Determining this will enable the tester to build the possible dictionary of password and the username collected during step 1. Additional references can be found from web application hackers handbook.

3. Using the dictionary built in step 2 try a dictionary or hybrid attack to determine the possible username and password combination. If the dictionary or hybrid attack is not successful, the tester could test using brute force attack. Additional references can be found at NIST, and OTG-AUTHN-007. Tools such as hydra are helpful in credential testing. Additional information can be found at OTG-AUTHN-001.
4. Sniff for credentials transmitted from client to the server. If the password is encrypted, determine the encryption method and decrypt it using an appropriate decryption tool or manually.
5. Using the information gathered about user registration during the information gathering stage, try to register. Check if anyone can register for access; check if registered users need validation before they start to use it; if users can register multiple times; if users can register for different roles or permissions; try to forge the registration process. Additional information can be found at OTG-IDENT-002.
6. If application auto-generates credentials, test if the auto-generated credentials can be predicated by identifying for predictable sequence. Accordingly build the possible credentials. Additional information can be found at the web application hackers handbook.
7. If auto-generated credentials are transmitted through different means, grab credentials from unsafe distribution of credentials. If accounts are activated through an activation URL, predict the URL and activate for false credentials. The application owners should securely generate credentials and safely distribute them. Additional information can be found at the web application hackers handbook.
8. Enumerate user roles and generate a matrix of users roles with the permission assigned to them and try to bypass roles. Determine which roles are able to provision users and what sort of accounts they can provision. Application administrators should carefully consider the sensitivity of design and configuration information before it is posted online, and should periodically review the sensitivity of existing design and configuration

information that is posted online. Additional information can be found at OTG-IDENT-001.

9. Test account provisioning process- determine which roles are able to provision users and what sort of accounts they can provision. Additional information can be found at OTG-IDENT-002.
10. Determine if authenticated users become locked out, during inactive periods. Determine the minimum lockout period. The detail procedures can be found at OTG-AUTHN-003.
11. Bypass authentication schema by directly requesting pages that can only be accessed by authenticated users, parameter modification, session ID predication or if the application is vulnerable to SQL injection by pass it using SQL injection. The detail procedures can be found at OTG-AUTHN-004.
12. If the application has remember me for credentials, check whether it is vulnerable or not using OWASP OTG-AUTHN-005. Application developers should ensure that no credentials are stored in clear text or are easily retrievable in encoded or encrypted forms in cookies.
13. Test for browser cache weakness by checking the HTTP response parameters cache-control:no-cache, no-store, Expires:0, and Pragma: no-cache are set as described in OTG-AUTHN-006
14. Try to grab credentials of others during account recovery by guessing answers for recovery questions, by using the hint for account recovery. The detail techniques are described at OTG-AUTHN-008, OTG-AUTHN-009, the web application hackers handbook
15. Though an application might be strong through one authentication mechanism, it might be vulnerable through other authentication mechanism so test for each of the authentication mechanisms. The application developers should ensure a consistent authentication policy is applied across all channels so that they are equally secure. The detail techniques are described at OTG-AUTHN-010.

5.4. Authorization testing

Authorization testing includes identifying vulnerabilities related to managing users as per to their privilege levels

Authorization testing methods

1. Test for directory traversal in windows IIS to access unauthorized directories by using input vector enumeration and URL injection. More detailed techniques is available at OWASP's OTG-AUTHZ-001, ISSAF U.8.1
2. Test to access functions assigned to other roles including administrative functions. Escalate privilege from one role to another role. Administrative functions should be accessible by administrators and users should be assigned access based on their roles. More detailed techniques is available at OWASP's OTG-AUTHZ-002 and OTG-AUTHZ-003.
3. Bypass authorization and access resources directly by modifying the value of a parameter used to directly point to an object. Change the value of a parameter that is used to retrieve a database record, perform an operation in the system, retrieve a file system resource, and access application functionality. More detailed techniques is available at OWASP's OTG-AUTHZ-004.
4. Test directories which are not mapped including /tmp, /src, /abc, /xyz, /bkup using Nikto or other tool. More detailed techniques is available at ISSAF U.12.1.
5. Check if directory browsing is on using the detailed techniques as shown in ISSAF U.12.2.

5.5. Session Management testing

Session management testing is identifying vulnerabilities related to session and cookies. Vulnerabilities that can be identified in this testing include session fixation, abandoned session as described in the summary below

Session management testing Methods

1. Bypass session management schema by collecting cookies, analyzing sessions for predictability and randomness. If the session is not predictable, reverse engineer the cookies and perform brute force attack. Use tools such as ZAP, Burp Sequencer, Foundstone CookieDigger, YEHG's and JHijack. More detailed technique can be found at OTG-SESS-001, ISACA, and the web application hackers handbook
2. Test cookie attributes including Secure attribute, HttpOnly attribute, Domain attribute, path attribute and expires attribute using tools such as ZAP, and TamperIE. More detailed technique can be found at OTG-SESS-002
3. Test for session fixation by using session ID obtained during GET request in POST request to obtain the session of other users using tools sch JHijack, and webscrab. More detailed technique can be found at OTG-SESS-003
4. Test for exposed variables by testing for encryption and reuse of session token vulnerabilities; Proxies & Caching vulnerabilities; Testing for GET & POST vulnerabilities. More detailed technique can be found at OTG-SESS-003
5. Test for cross site request forgery to manipulate users to perform the activities the user wants using the techniques explained at OWASP's OTG-SESS-005. Additional tools such as WebScrab Spider, CSRF Tester, Cross Site Requester, Cross Frame Loader, Pinata-csrf-tool are helpful in conducting this test
6. Test if sessions are terminated after log out; grab the session ID and try to login to check if the server doesn't terminate the session; test for session timeout. Check mapping of tokens to sessions by loging twice for example using different browsers without loging out the previous session and determine if both sessions are active. More detailed technique can be found at OTG-SESS-006
7. Understand cookies and try to predict how cookies are generated. After understanding the cookie generation and predication, generate cookies and impersonate other users. More detailed technique can be found at ISSAF.
8. If the application uses HTTP cookies to transmit session tokens (or any other sensitive data), review the relevant Set-Cookieheaders, and check for any domain or path attributes used to control the scope of the cookies. Identify all the possible domain names and paths that will receive the cookies that the application issues. Establish whether any other web

applications are accessible via these domain names or paths that you may be able to leverage to capture the cookies issued to users of the target application. More detailed technique can be found at ISSAF.

5.6. Cryptography Testing

Users credentials might be grabbed while they are in transit and those credentials might be in clear text if the application does not implement a proper encryption technique.

Cryptography testing methods

1. Testing for weak SSL/TLS ciphers, insufficient transport layer protection- using OWASP's OTG-CRYPST-001 and tools such as Qualys and Nessus grab sensitive data on transit that are an encrypted, weak SSL/TLS ciphers/protocols/keys
2. Testing for padding Oracle- decrypt encrypted data without knowledge of the encryption key and used cipher by sending skillful manipulated cipher texts to the padding oracle and observing of the results returned by it. Supportive tools such as PadBuster, Python-paddingoracle, Poracle, and Padding Oracle Exploitation tool (POET) can be used in this test. More detailed testing can be found at OWASP's OTG-CRYPST-001.
3. Testing for sensitive data information sent via unencrypted channels- grab sensitive data that are unencrypted as described in OTG-CRYPST-003, and OTG-AUTHN-001 using tools such as curl.

5.7. Business Logic Testing

Many testing methodologies propose the use of automated tools to identify vulnerabilities however there are a large number of vulnerabilities that can't be identified using automated tools as the current automated tools can't understand the business logic behind. Most of the tests proposed at this control testing are dependent on the knowledge of the tester to understand the target.

Business Logic Testing Methods

1. Review the information gathered during the information gathering stage. The information necessary to conduct business logic testing include identification of entry points, the role of the users, the type of data to be processed at the entry point, the layers of processing, predictable or hidden functionality fields, processing time of critical functions, and types of files expected to be uploaded.
2. Insert logically invalid data into the application by trying to insert logically invalid data into the application/system; Perform front-end GUI Functional Valid testing on the application to ensure that the only "valid" values are accepted; Using an intercepting proxy observe the HTTP POST/GET looking for places that variables such as cost and quality are passed; Specifically, look for "hand-offs" between application/systems that may be possible injection of tamper points; and Once variables are found start interrogating the field with logically "invalid" data, such as social security. The detailed techniques are available at OWASP's OTG-BUSLOGIC-001
3. Bypass the frontend application to directly submit information for back end processing. The tests in this includes- From the information gathering stage look for guessable, predictable or hidden functionality of fields;Once found try to insert logically valid data into the application/system allowing the user go through the application/system against the normal business logic workflow; Using an intercepting proxy observe the HTTP POST/GET looking for some indication that values are incrementing at a regular interval or are easily guessable. The detailed techniques and related countermeasures are available at OWASP's OTG-BUSLOGIC-002
4. Insert invalid data to non editable or hidden parameters to bypass the application controls by intercepting the request using a proxy. The detailed techniques and related countermeasures are available at OWASP's OTG-BUSLOGIC-002
5. Send incomplete requests by intercepting the request in a proxy- For each parameter in turn, remove both the name and value of the parameter from the request. Monitor the application's responses for any divergence in its behavior and any error messages that shed light on the logic being performed. More detailed techniques are available at the web application hackers handbook

6. Understand what the application would do if the user does not submit the request in the expected time frame by Gathering information on the entry points that could be impacted by time Such as execution time or actions that help users predict a future outcome or allow one to circumvent any part of the business logic or workflow. The detailed techniques and related countermeasures are available at OWASP's OTG-BUSLOGIC-004
7. Test Number of times a function can be used by exploratory testing looking for functions or features in the application or system that should not be executed more than a single time or specified number of times during the business logic workflow; For each of the functions and features found that should only be executed a single time or specified number of times during the business logic workflow, develop abuse/misuse cases that may allow a user to execute more than the allowable number of times. The detailed techniques and related countermeasures are available at OWASP's OTG-BUSLOGIC-005
8. Testing for the circumvention of work flows- Misuse an application/system in a way that will allow to circumvent (not follow) the designed/intended workflow. The detailed techniques and related countermeasures are available at OWASP's OTG-BUSLOGIC-006
9. Testing defenses against application mis use- Test to determine whether there are application-layer defensive mechanisms in place to protect the application. The detailed techniques and related countermeasures are available at OWASP's OTG-BUSLOGIC-007 and Web security testing cookboox
10. Upload unexpected file types - test uploading of files performing some exploratory testing looking for file types that should be "unsupported" by the application/system; Try to upload these "unsupported" files and verify that it are properly rejected; If multiple files can be uploaded at once, there must be tests in place to verify that each file is properly evaluated. The detailed techniques and related countermeasures are available at OWASP's OTG-BUSLOGIC-008
11. Test upload of malicious files- upload malicious files by using the detailed techniques described in OWASP OTG-BUSLOGIC-008 and the related countermeasures are also described in OWASP.

5.8. Server testing

An attacker doesn't only target the application but also the servers hosting the application. An attack on the servers leads to compromising the application and the data hosted in the servers. Hence, the servers should be secured enough to protect the application. Note- ISSAF has provided the known vulnerabilities of servers but has not provided a method to test them so it was not written in here. It should also be noted that PTES, and NIST propose to scan for vulnerabilities of the target.

Server testing Methods

1. Identify known vulnerabilities of server- scan for known server vulnerabilities or search vulnerabilities in public vulnerability databases. The detailed techniques are available at OWASP's OTG-CONFIG-001, web application hacker handbook. Some of the helpful tools in vulnerability scanning are nessus, nexpose, and Qualys guard.
2. Identify configuration vulnerabilities- scan for configuration vulnerabilities using vulnerability scanners, vendor configuration advisory. The detailed techniques are available at web application hacker handbook
3. Test for known, old and unreferenced files- scan for known, and old files using tools such as nessus and nikto. The detailed techniques and countermeasures are available at OWASP's OTG-CONFIG-004.
4. Test for default credentials- bypass admin web pages using either dictionary, hybrid or brute force attack. The detailed techniques are available at web application hacker handbook.
5. Validate vulnerabilities- validate the vulnerabilities obtained in the above phases by either comparing the automated tools or manually validating them.
6. Exploit vulnerabilities- After the vulnerabilities are validated in the above phase, test for sample exploits using tools such as metasploit or manually.
7. Attack through another application hosted on the same server-Identify applications hosted on the same server; identify the vulnerabilities of the second application; attack the second application and control the server and then attack the primary application.

5.9. Bypassing Network Security Controls

Even though an application is vulnerable, infrastructure security controls might protect it. In addition to this the application may not even be accessible from the internet as it is limited inside a VPN hence there must be a way to bypass the VPN to attack the application.

Bypassing Network Security Controls Methods

a. Hacking VPN- if the application is behind a VPN external testers can't access the application as a result it is mandatory to bypass VPN. Use tools such as ikescan to identify vulnerabilities and exploit it.
b. Bypass IDS/IPS
    1. Denial of service attack on central logging- Identify the central log server and perform DoS attack. More detailed techniques are available at CEH V7 module 16.
    2. Obfuscating- bypass IDS/IPS by encoding the attack payload in a way the IDS can't detect and send the attack
    3. False positive generation- Generate large amount of false positive attacks to create a noise
    4. Session Splicing- Split data between several packets making sure that it won't be detected by an IDS
    5. Unicode Evasion- For each attack payload provide a unique identifier to multiple represent a character in the attack payload
c. Bypassing Firewall
    1. IP address spoofing- Spoof the IP address of the attacker as if it comes from internal zone to bypass firewall.
    2. ICMP tunneling- Bypass the firewall through ICMP tunneling by sending the attacker payload in ICMP echo packet
    3. ACK tunneling- Bypass the firewall through ACK tunneling by sending the attacker payload in ACK tunneling
    4. Obfuscating- Identify which characters/sequence are allowed by WAF obfuscate the payload and send the attack. If the attacker is not successful obfuscate through other mechanisms and try again

5. Port redirection- use port redirection tools to send the attack in the ports permitted by firewall

d. Bypassing Load balancer

If load balancer is detected it means there are redundant servers which may have their own name or IP address. So the attacker might attack one of the targets but still the other target work. So if for example the attacker wants to deface a website, he/she should deface both the applications hosted on different servers.

1. Multiple targets hosting the same application- if application is hosted in multiple servers try to test the domain name or the IP address if both of them are accessible remotely. More detailed techniques are available at SANS Web application penetration testing and ethical hacking reconnaissance and mapping module
2. Modify the configuration of load balancer- Identify vulnerabilities of the load balancer and attack it to modify the configuration to redirect the access to only specific target. More detailed techniques are available at SANS Web application penetration testing and ethical hacking reconnaissance and mapping module

CHAPTER SIX

SAMPLE TESTS

Using the methodologies developed in chapter 5, sample tests were conducted. Though each of the methods were tested against the six targets, this section will only provide a sample of the tests conducted during input validation testing and the vulnerability validation proposed at the assessment phase. The importance of this chapter is to enable the tester to understand how the methods developed in this research will be used to conduct the test. One of the criteria developed in this research is explainable. A methodology is explainable to a tester if it provides examples how to conduct a test. As a result the following sample tests were provided in this chapter.

In most of the samples below the request that should be delivered to the target to identify vulnerabilities is provided, the response of the request by the websites are also described and a sample of exploits are provided.

6.1. Sample Input validation Testing

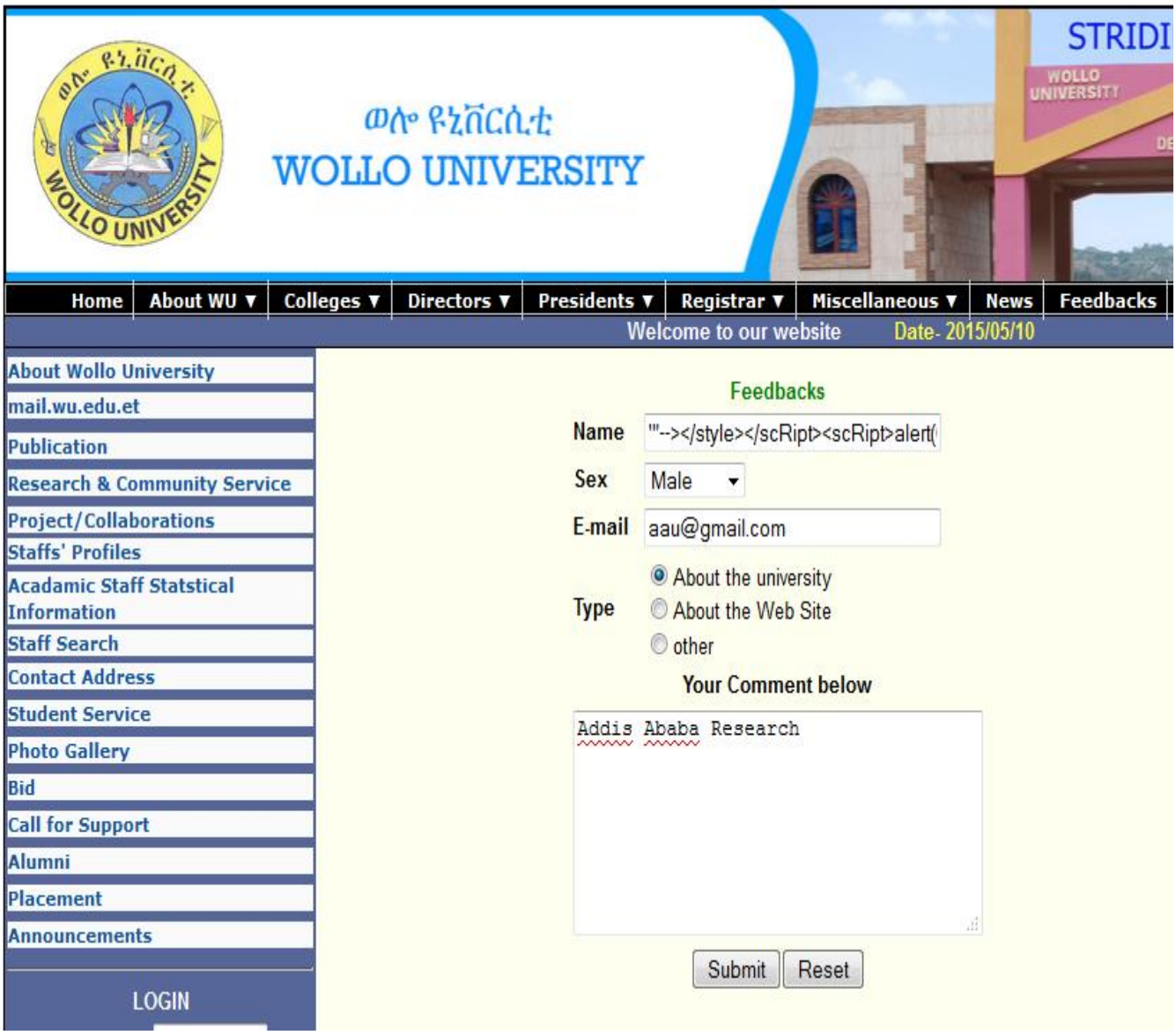
To determine which tests to conduct it is necessary to know the target types. From the information gathering stage it is known that the target database types are MySQL, Microsoft SQL. As a result the database tests will be based on those targets. In addition to that in none of the target technologies such as SOAP, LDAP and others were detected. Hence the test was limited only on the identified technologies.

1. Reflected Cross Site Scripting testing

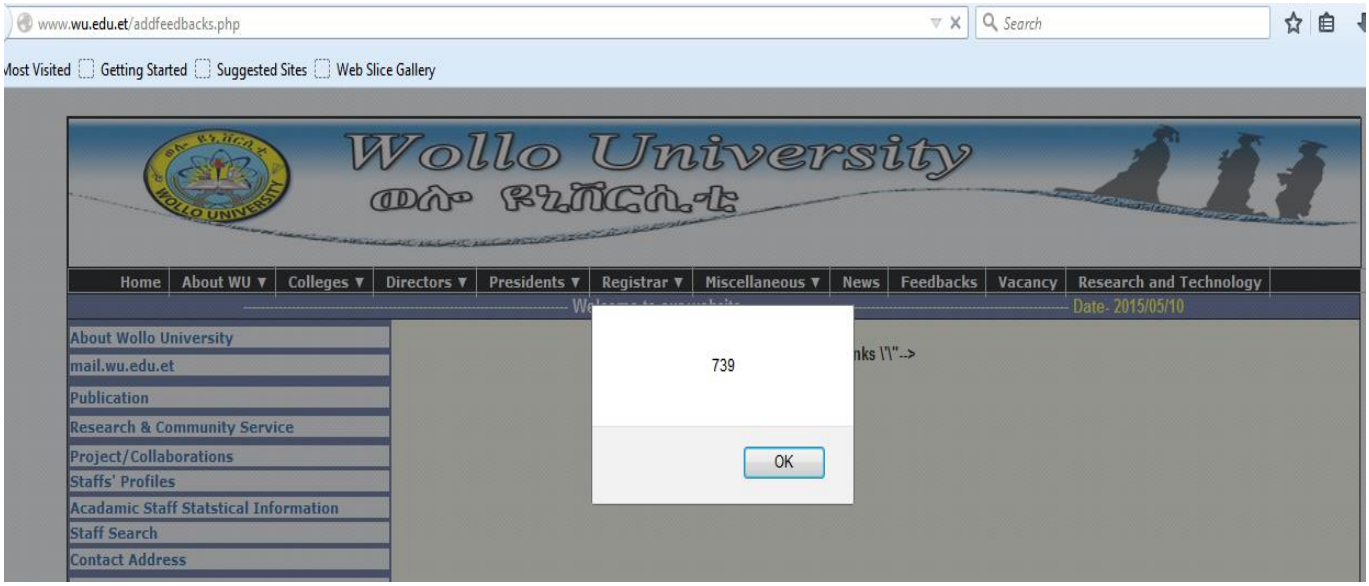
Targets vulnerable to Reflected cross site scripting can be obtained either manually or through automated tools like netsparker.

Wollo university is one of the targets vulnerable to reflected cross site scripting. One of the pages that are vulnerable to this attack is the feedback page with the “name” parameter.

To check this attack, let’s enter the script (**'"--></style></scRipt><scRipt>alert(0x0002E3)</scRipt>**) in the name parameter and the other values as shown in the following figure



When the feedback is submitted the website will reflect the request as shown in the following figure. As a result the website is vulnerable to Reflected Cross site scripting attack.



Similarly at http://estudent.ddu.edu.et/var/www/webhosts/estudent.ddu.edu.et/? nsextt the nsextt parameter is vulnerable to reflected Cross site scripting as shown in the following request and response.

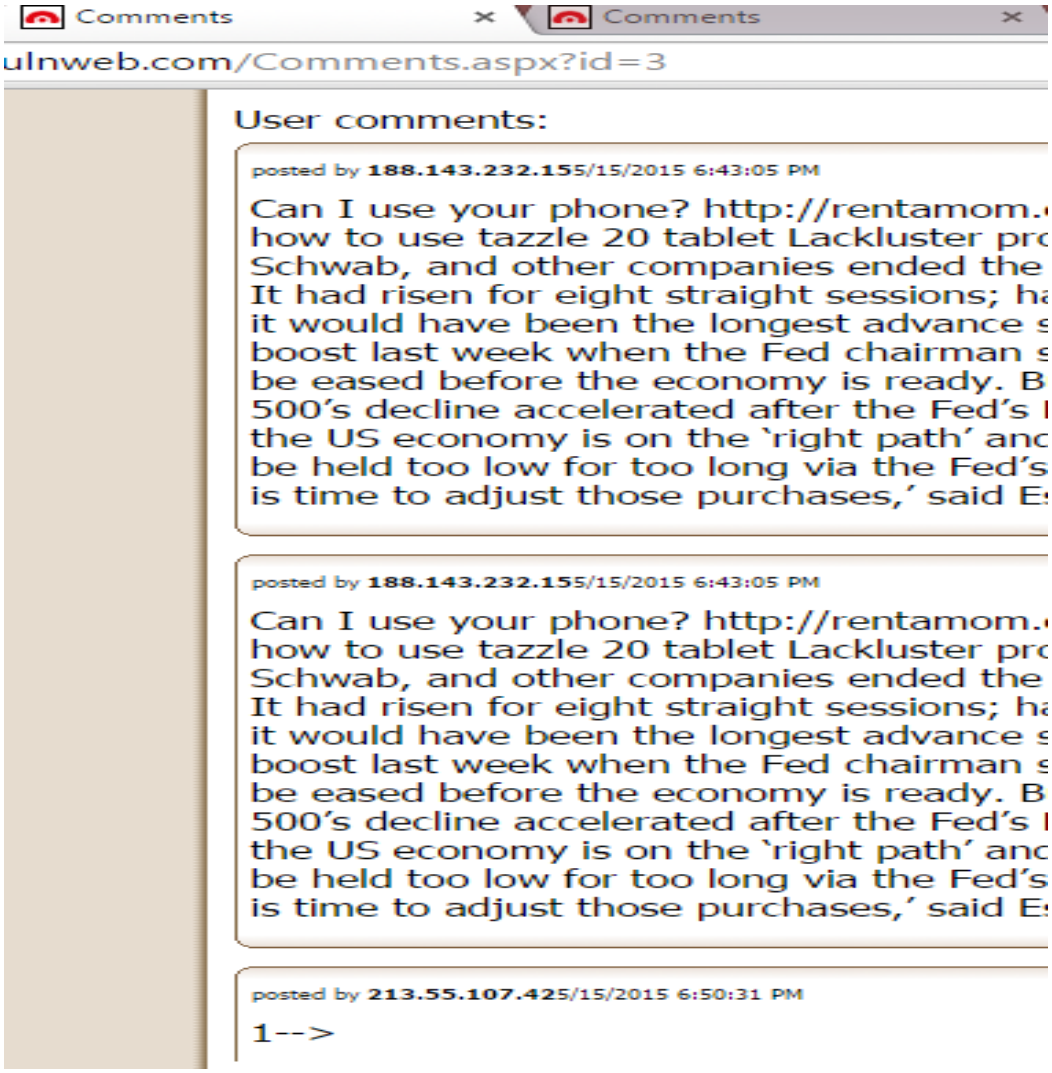
```
Request
GET /var/www/webhosts/estudent.ddu.edu.et/?nsextt='%22--
%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Ealert(0x0005C6)%3C/scRipt%3E  HTTP/1.1
Cache-Control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; Netsparker)
Accept-Language: en-us,en;q=0.5
Host: estudent.ddu.edu.et
Cookie: PHPSESSID=0mmpjfctsh5la5cj3oref0q6f0
Accept-Encoding: gzip, deflate

Response
...
run()
#7 {main} </pre>

<h3>Request Parameters:</h3>
<pre>array (
'controller' => 'var',
'action' => 'www',
'webhosts' => 'estudent.ddu.edu.et',
'module' => 'default',
'nsextt' => '"--></style></scRipt><scRipt>alert(0x0005C6)</scRipt>',
) </pre>

</body>
</html>
</div>
</div><!-- end of second row for the content area-->

<!-- footer area starts here -->
<div class="row" style="width: 100%;padding-bottom: 0px;">
```

2. Testing for stored cross site scripting

Using a vulnerability scanner acublog has been identified as vulnerable to stored(permanent) cross site scripting.

The vulnerable link is at http://testaspnet.vulnweb.com/Comments.aspx?id=3

When a user inserts a script at the comments and submits it the following figure will be displayed. As shown in the figure the comment “1-->” was inserted by an Ethiopian public address. -



After the script is inserted when ever a new user comes the following output is displayed. As shown in the figure, for every user that comes a malicious form might be displayed for example asking the user to enter the username and password. This way every user account might be in the hand of an attacker

3. Testing for HTTP verb Tampering

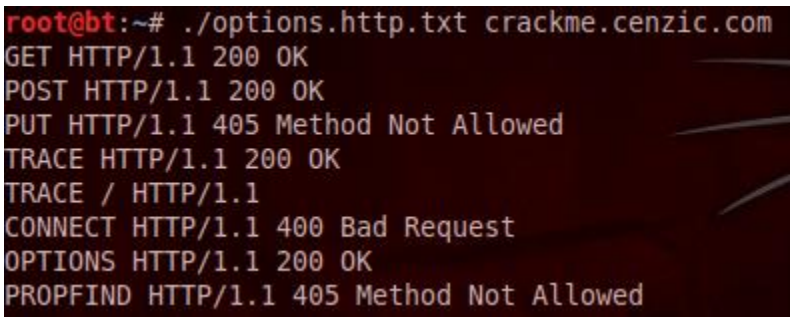
The following script from was used in this test which was copied from https://pentestlab.wordpress.com/2012/12/20/http-methods-identification/

```
#!/bin/bash

for webservmethod in GET POST PUT TRACE CONNECT OPTIONS PROPFIND;

do
printf "$webservmethod " ;
printf "$webservmethod / HTTP/1.1\nHost: $1\n\n" | nc -q 1 $1 80 | grep "HTTP/1.1"

done
```

Using the script the targets was tested and accordingly the following are the sample tests of Assosa University and acublog

As can be seen in the figure below Assosa university website not only allows GET and POST but also PUT, TRACE, OPTIONS, and PROPFIND as a result it is vulnerable

From the result shown below for acublog in addition to GET and POST methods OPTIONS is also used as a result the website is vulnerable



4. HTTP Parameter Pollution

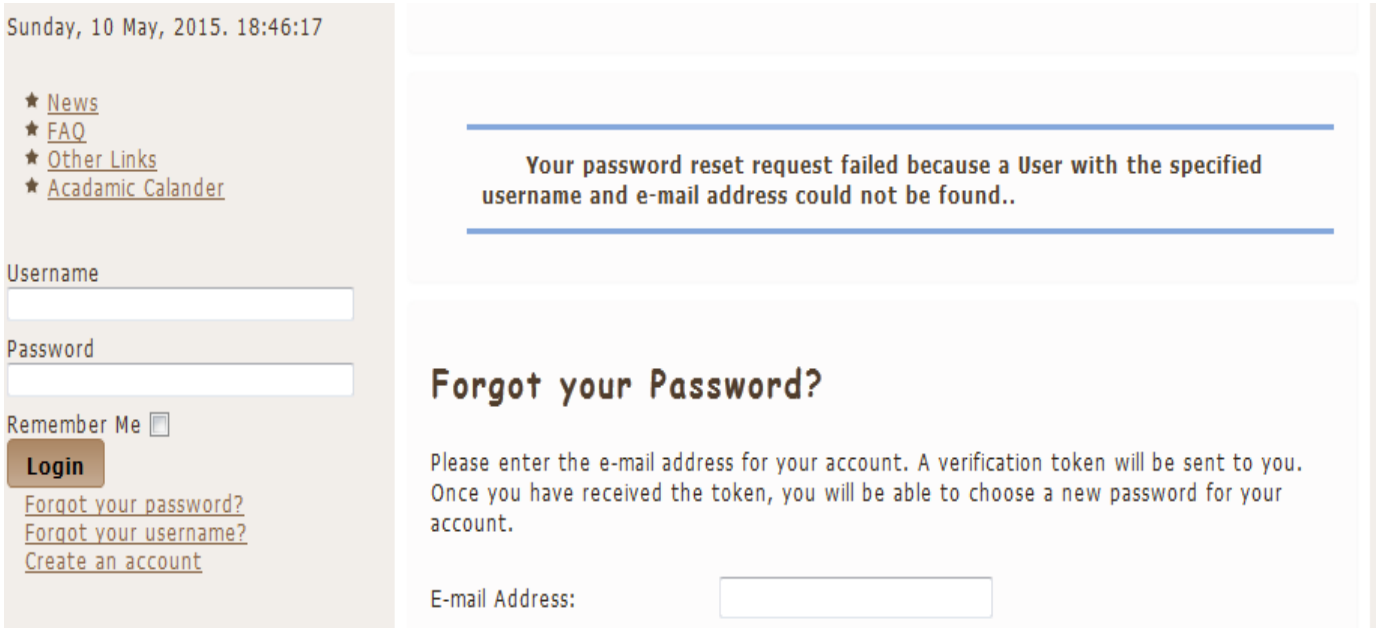
The following are some of the results conducted at this phase.

Let’s test Assosa Universities forgot password.

There is no user registered with an email address alishkonjo@gmail.com in the website so when the following request is sent, the website should state that there is no user with such an email address.

```
POST /index.php?option=com_user&task=requestreset&lang=en HTTP/1.1
Host: www.asu.edu.et
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:37.0) Gecko/20100101 Firefox/37.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.asu.edu.et/index.php?option=com_user&view=reset&lang=en
Cookie: cfacd9bcee952c148abb822d201ee73c=g5ag0hgq2gl46m8fvke3m79ob4; jfcookie[lang]=en
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 63

email=alishkonjo%40gmail.com&b126156102eda9b1c8b3945c5e34746c=1
```

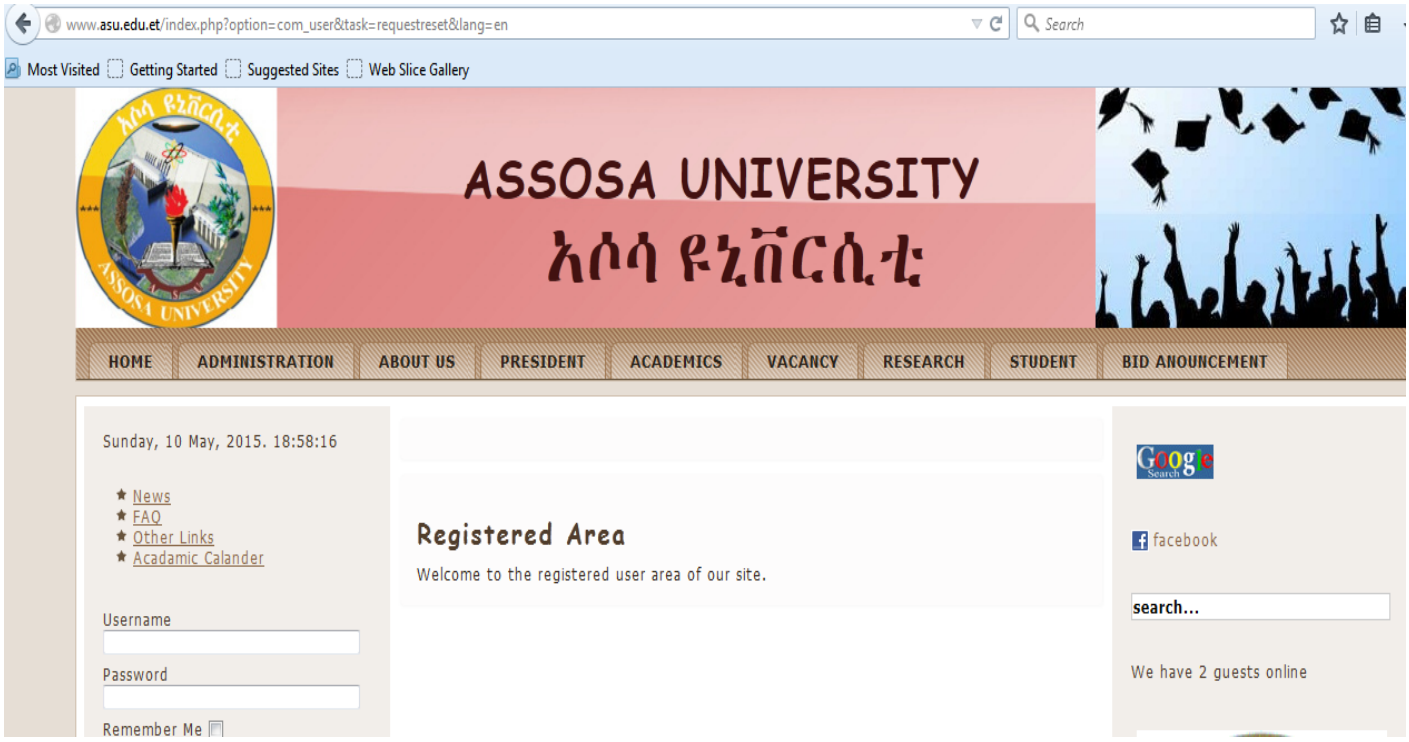
As shown in the following figure the website provided a response as expected.



However, let's change the request by appending another value ("delete") for the "task" parameter as shown in the request below

```
POST /index.php?option=com_user&task=requestreset&task=delete&lang=en HTTP/1.1
Host: www.asu.edu.et
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:37.0) Gecko/20100101 Firefox/37.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.asu.edu.et/index.php?option=com_user&view=reset&lang=en
Cookie: cfacd9bcee952c148abb822d201ee73c=g5ag0hgq2gl46m8fvke3m79ob4; jfcookie[lang]=en
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 63

email=alishkonjo%40gmail.com&b126156102eda9b1c8b3945c5e34746c=1
```

The website should have rejected the request or should have provided the previous error message that there is no such a user registered in the website. However, the website allowed the user to enter into the registered area as shown in the following figure. As a result the website is vulnerable to HTTP Parameter manipulation.

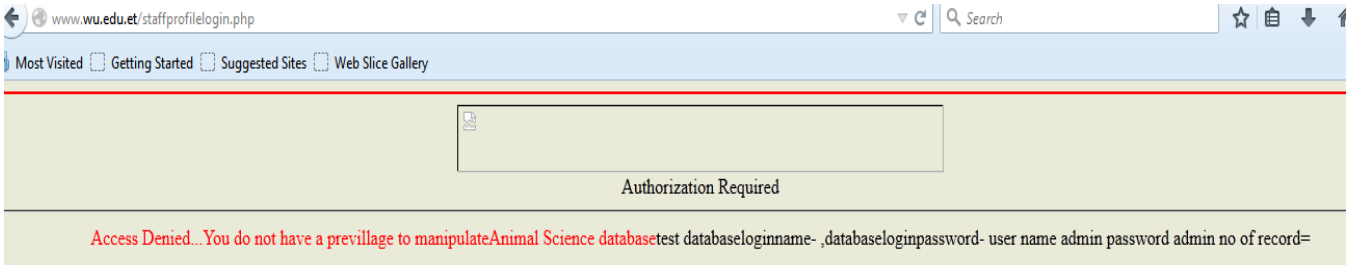


Contrary to Assosa university, Wollo University is not vulnerable to HTTP Parameter pollution.

As shown in the following request the user requested to login with admin as a username and password

```
POST /staffprofilelogin.php HTTP/1.1
Host: www.wu.edu.et
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:37.0) Gecko/20100101 Firefox/37.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.wu.edu.et/login
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 145

loginname=admin&loginpassword=admin&faculty=Agriculture&department=Animal+Science&attemptprofile=%0D%0A%09%09%091%09%0D%0A%09%09%09&action=S
```

The requests response is as shown in the following figure



By repeating one of the parameters of the request as shown in the following request the same response error message as the above one was generated hence it is not vulnerable to HTTP Parameter Pollution.

```
POST /staffprofilelogin.php HTTP/1.1
Host: www.wu.edu.et
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:37.0) Gecko/20100101 Firefox/37.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.wu.edu.et/login
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 145

loginname=admin&loginpassword=admin&department=Forestry&faculty=Agriculture&department=Animal+Science&attemptprofile=%0D%0A%09%09%091%09%0D%0A%09%09%09&action=Submit
```

5. SQL Injection

In OWASP it is recommended that the tester should first identify the vulnerable parameter of the target. This can be done manually or by using vulnerability scanner. The vulnerability identified by vulnerability scanners should be confirmed manually so that the tester avoids false positive results.

In this research ZAP, Netsparker, Qualysguard, Nexpose and other vulnerability scanners were used to identify such a vulnerability.

The vulnerability scanners have identified SQL injection vulnerability in Diredawa University, acublog, and cenzic. The following examples will be for diredawa university and acublog.

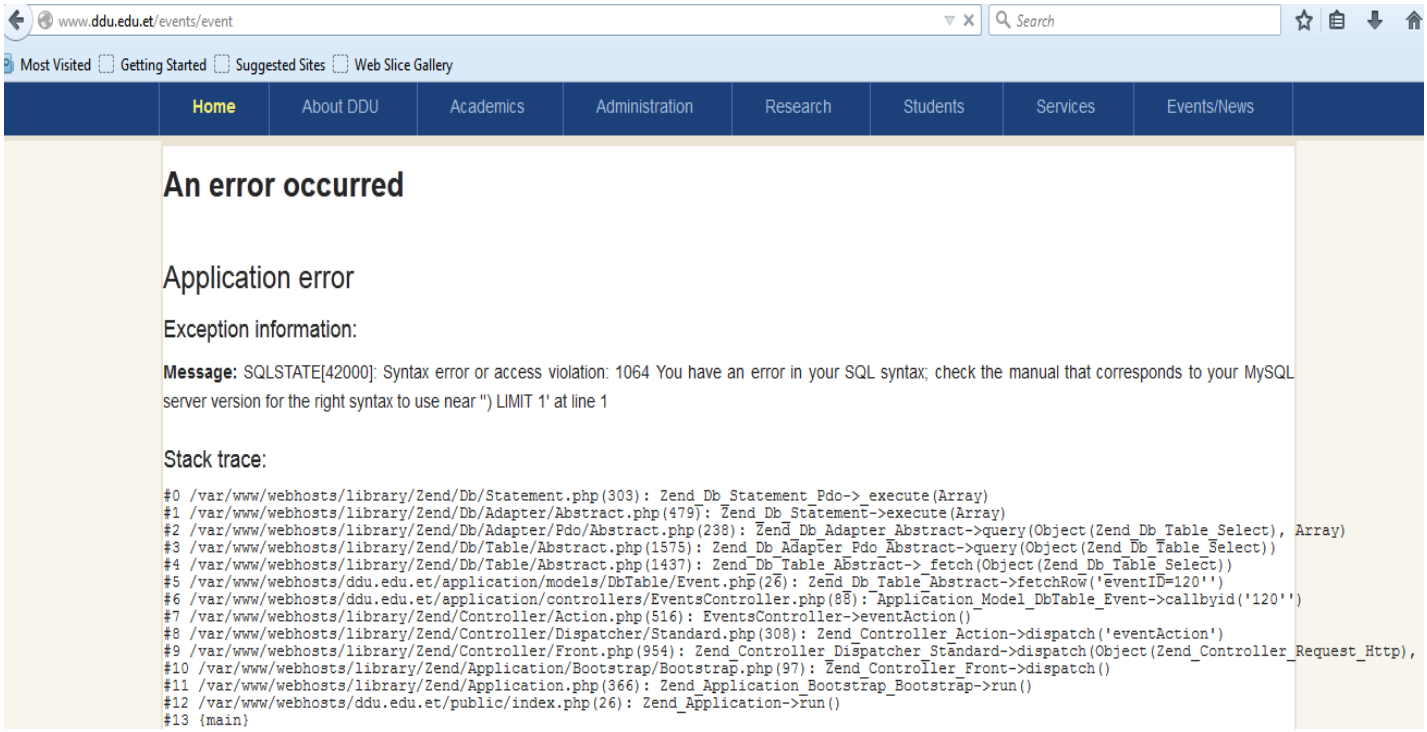
Diredawa University

Diredawa university website is vulnerable to SQL injection, Blind SQL injection and Boolean SQL injection. Let's see each of them

SQL injection vulnerability in Diredawa university

The "id" parameter in the GET request of http://www.ddu.edu.et/events/event?id is vulnerable to SQL injection. By injecting at the id parameter as shown in the following request an SQL error message is displayed. As a result the website is vulnerable to SQL injection.

```
GET /events/event?id=20' HTTP/1.1
Host: www.ddu.edu.et
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:37.0) Gecko/20100101 Firefox/37.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: PHPSESSID=didath7d911cud3ul7i0c2b302
Connection: keep-alive
Cache-Control: max-age=0
```



Using the identified vulnerabilities the following outputs were exploited from diredawa university website. As can be seen in the tables below, the table names and columns are identified. In addition to this, an example of the values of the columns of some of the tables is displayed below.

DDU_website	
Table Name	**Columns**
adminstaff	SID fname lname position unit telephone mobile email emailAllias fax website skype photo
article	articleID title body imagePath isFeatured catagoryID unitID access approvedByID postedDate status
category	categoryID categoryName description parentCategoryID updatedDate status
event	eventID title eventImage startDate endDate eventLocation body eventArticleID categoryID status approvedByID postedDate access
slider	sliderID imagePath articleID eventID createdDate status sliderCaption
sqlmapoutput	data
staff	SID fname lname rank school department telephone mobile email emailAllias fax website specialization researchInterest program publications photo
unit	unitID unitName
user	userID fullName username password role createdDate status
view_allarticles	imagePath postedDate articleID title body categoryName access status isFeatured

view_catunitarticles	articleID title body postedDate categoryName unitName
view_uniteven t	

Table: user

Total Rows: 2

userID	fullName	username	password	role	createdDate	status
2	Yared	yared	ddac418a1be76098d01107464026f65d2a3192bf1		2014-06-21 18:30:18	0
3	Karyos	karyos	db88f704304504fe9b76b6cb91e22d9213a0ec9b	0	2015-02-08 20:53:22	0

Data Base: information_schema

Table: USER_PRIVILEGES

Total Rows: 141

Data Base: dormitaryassignment

Table: user_account

Total Rows: 1

AccountID	EmployeeID	UserName	Password	Status	role
1	1	sysadmin	9a3e61b6bcc8abec08f195526c3132d5a4a98cc01		Administrator

When the password is decrypted using decrypting tool the password for sysadmin user with administrator role is “444” as shown in the following figure

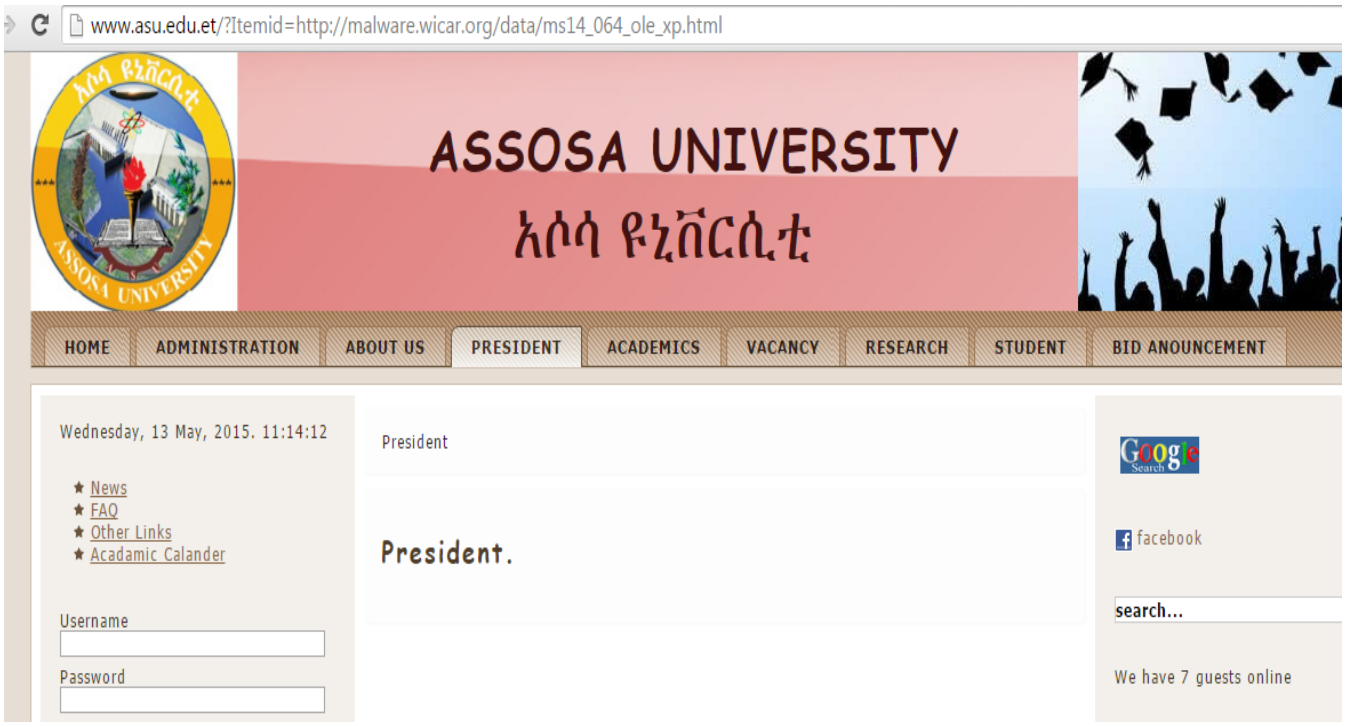
Code Injection

This test was conducted manually by injecting malicious URL that is not harmful but should not be accepted by the websites as a parameter.

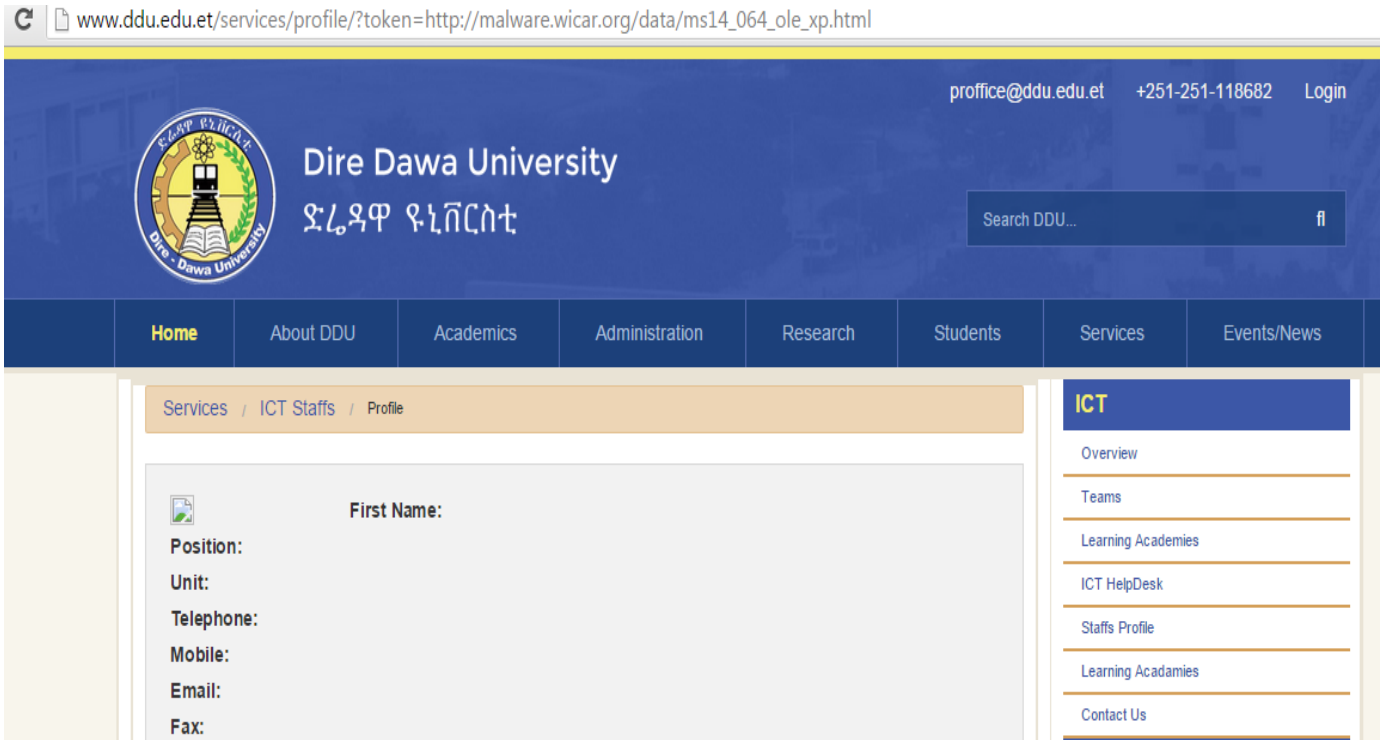
The malicious URL is http://malware.wicar.org/data/ms14_064_ole_xp.html

Assosa University

The university website accepts the malicious URL and executes it as shown in the figure below. As a result the website is vulnerable to code injection.



Similarly dire dawa university is also vulnerable to this attack as shown below



6. Fuzzing

An attacker can try to attack the username and password manually or using automated tools to grab the credentials. However, there are additional entry points that accept input from a user. Those additional entry points may not have a validation in place to control an unknown and invalid inputs. As a result, the application or the server hosting it might crash becoming exposed and under the control of attackers. It is for this scenario fuzzers are needed. Using fuzzers black box testers can include arbitrary inputs to crash either the application or the infrastructure.

In this research however fuzzing was not conducted as it might crash the target of tests and they may not have allocated the resource to backup their systems in time.

Blackbox testers however, can use tools such as OWASP ZAP, Simple Fuzzer(SFUZZ), Brutefoce Exploit Detector (BED).

For example if the tester is going to conduct http fuzzing on diredawa university website the following command could be executed.

```
root@bt:/pentest/fuzzers/sfuzz# ./sfuzz -TO -f sfuzz-sample/basic.http -S www.dd
u.edu.et -p 80
```

6.2. Sample vulnerability Validation

In the assessment phase of this research it is proposed that the outputs of automated tools should be compared to avoid false positive and false negative vulerabilities. This sample test will show the vulnerability validation of an automated scanning on Assosa university using Qualysguard, nessus and nexpose vulnerability scanners. First the vulnerability obtained by scanning will be organized in table form that contains the vulnerability name and its CVE ID. Then an analysis will be performed and the vulnerabilities of the university will be identified.

1. Vulnerabilities of assosa univeristy website obtained by QualysGuard

No.	**Vulnerability**	**CVE ID**
1.	EOL/Obsolete Software: PHP 5.3.x Detected	-----
2.	EOL/Obsolete Software: Joomla! 3.1.x Detected (port 80/tcp)	----
3.	Joomla! com_contact, com_weblinks, and com_newsfeeds Multiple Cross-Site Scripting Vulnerabilities (port 80/tcp)	CVE-2013-5576
4.	Joomla! weblinks-categories SQL Injection Vulnerability (port 80/tcp)	---
5.	Joomla! Cross-Site Scripting and Unauthorized GMail Login Vulnerabilities (port 80/tcp)	---
6.	Web Server Uses Plain-Text Form Based Authentication (port 80/tcp)	---
7.	HTTP TRACE / TRACK Methods Enabled (port 80/tcp)	CVE-2004-2320, CVE-2010-0386,

		CVE-2003-1567
8.	AutoComplete Attribute Not Disabled for Password in Form Based Authentication (port 80/tcp)	
9.	Web Server HTTP Trace/Track Method Support Cross-Site Tracing Vulnerability (port 80/tcp)	CVE-2004-2320, CVE-2007-3008
10.	Expose_php Set to On in php.ini (port 80/tcp)	

2. Vulnerabilities of assosa univeristy website obtained by Nexpose

   Multiple PHP and HTTPD vulnerabilities were identified using Nexpose in Assosa univerity but only a sample of them are shown in the following table

No.	**Vulnerability**	**CVE ID**
1.	PHP Vulnerability: CVE-2012-2376 (php-cve-2012-2376)	CVE-2012-2376
2.	PHP Vulnerability: CVE-2012-2688 (php-cve-2012-2688)	CVE-2012-2688
3.	PHP Vulnerability: CVE-2012-2386 (php-cve-2012-2386)	CVE-2012-2386
4.	PHP Vulnerability: CVE-2013-1635 (php-cve-2013-1635)	CVE-2013-1635
5.	Apache HTTPD: insecure LD_LIBRARY_PATH handling (CVE-2012-0883) (apache-httpd-cve-2012-0883)	cve-2012-0883
6.	Apache HTTPD: mod_status buffer overflow (CVE-2014-0226) (apache-httpd-cve-2014-0226)	cve-2014-0226
7.	Joomla!: [20130801] - Core - Unauthorised Uploads (CVE-2013-5576) (joomla-20130801-core-unauthorised-uploads)	CVE-2013-5576
8.	HTTP TRACE Method Enabled (http-trace-method-enabled)	

9.	Apache HTTPD: HTTP Trailers processing bypass (CVE-	CVE-2013-5704
10.	Apache HTTPD: XSS in mod_proxy_balancer (CVE-2012-4558) (apache-httpd-cve-2012-4558)	CVE-2012-4558
11.	Joomla!: [20131101] Core XSS Vulnerability (joomla-20131101-core-xss)	
12.	PHP Fixed possible attack in SSL sockets with SSL 3.0 / TLS 1.0 (php-cve-2011-3389)	cve-2011-3389

3. Vulnerabilities of assosa univeristy website obtained by Nessus

No.	**Vulnerability**	**CVE ID**
1.	PHP unsupported version detection	
2.	PHP 5.3.x < 5.3.15 Multiple Vulnerabilities	CVE-2012-3365
3.	Apache 2.2 < 2.2.28 Multiple Vulnerabilities	CVE-2013-5704
4.	PHP 5.3.x < 5.3.27 Multiple Vulnerabilities	CVE-2013-4113 / Bug #65236
5.	PHP 5.3.x < 5.3.29 Multiple Vulnerabilities	CVE-2013-6712, CVE-2014-0207, CVE-2014-0237, CVE-2014-0237, CVE-2014-0238, CVE-2014-3515, CVE-2014-398, CVE-2014-4049

6.	PHP 5.3.x < 5.3.26 Multiple Vulnerabilities	
7.	PHP 5.3.x < 5.3.14 Multiple Vulnerabilities	CVE-2012-2386, CVE-2012-2143, CVE-2012-3450, CVE-2012-6113
8.	HTTP TRACE / TRACK Methods Allowed	CVE-2003-1567, CVE-2004-2320, CVE-2010-0386
9.	PHP expose_php Information Disclosure	
10.	Apache 2.2 < 2.2.27 Multiple Vulnerabilities	CVE-2013-6438, CVE-2014-0098
11.	Apache 2.2 < 2.2.23 Multiple Vulnerabilities	CVE-2012-0883, CVE-2012-2687
12.	Apache 2.2 < 2.2.25 Multiple Vulnerabilities	CVE-2013-1862, CVE-2013-1896
13.	Apache 2.2 < 2.2.24 Multiple XSS Vulnerabilities	CVE-2012-3499, CVE-2012-4558
14.	PHP 5.3.x < 5.3.28 Multiple OpenSSL Vulnerabilities	CVE-2013-4073, CVE-2013-6420
15.	PHP 5.3.x < 5.3.23 Information Disclosure	CVE-2013-1643
16.	PHP 5.3.x < 5.3.22 Multiple Vulnerabilities	CVE-2013-1635, CVE-2013-1643
17.	Web Server Uses Plain Text Authentication Forms	
18.	Web Server Allows Password Auto-Completion	
19.	Web Server robots.txt Information Disclosure	

Vulnerability Comparison of the three vulnerability scanner on Assosa university

Most of the vulnerabilities identified by the three vulnerability scanners are common except their naming of the vulnerabilities. Some of the vulnerabilities identified by only one of the vulnerability scanners are also the vulnerabilities that were obtained in the previous stages of this

research. In addition to this, some of the vulnerabilities such as cross site scripting were exploited in the input validation testing.

I) The three vulnerability scanners identified that the web application uses < PHP 5.3 and this version is obsolete and hence for every vulnerability that is idendtified the vendor will be exposed to attack. Due to this obsolete version the vulnerability scanners identified mulitiple vulenrabilities

II) The content management system used by the application is joomla 3.1. This CMS is also obsolete as identified by the vulenrability scanners

III) Cross site scripting- the three vulnerability scanners have also detected that the application is vulnerable to cross sites scripting

IV) Apach 2.2 < multiple vulnerability- the three vulnerability scanners also identified that due to vulnerable Apach the website is vulnerable to multiple vulnerabilities

V) Autocomplete attribute not disabled for password- this vulnerability was detected by Qualysguard and Nessus and was confirmed

VI) HTTP TRACE/TRACK Method- it was detected by the three vulnerability scanners. This was also detected at the input validation testing

VII) robots.txt- though only Nessus detected the vulnerability, this was identified during the information gathering stage

VIII) Plain text authentication- Both Qualysguard and Nessus detected this vulnerability, and this vulnerability was also identified during the information gathering stage

CHAPTER SEVEN

SAMPLE VULNERABILITIES

Sample input validation vulnerabilities were identified using the tests conducted in chapter six. This chapter will provide a summarized sample vulnerabilitie that were obtained during information gathering and input validation testing.

The sample vulnerabilities were organized in table form where

- Target is one of the six website targets,
- Vulnerable- to state whether the target is vulnerable or not
- Vulnerability information/link- details about the identified vulnerability or the link where the vulnerability was identified
- Paramater-the vulnerable parameter

This chapter will help the tester how to organize the vulnerabilities obtained during testing. Since risk calculation was out of the scope of this research, the summary of vulnerabilities would not have a risk level. Besides to this, the tester can select the countermeasure proposed under each test methods in chapter five.

1.1. Vulnerabilities of input validation

1. Vulnerable Methods

An attacker can gain information about the target using unnecessary methods

Target	Vulnerable	Identified methods
DDU	Yes	GET, POST, OPTIONS
ASU	Yes	GET,POST, PUT, TRACE, OPTIONS, and PROPFIND
WU	Yes	GET, POST, OPTIONS
HU	Yes	GET, POST, OPTIONS
Cenzic	Yes	GET,POST, TRACE, OPTIONS
acublog	Yes	GET,POST,OPTIONS

2. Code injection vulnerability

If website accepts codes, it might interpret/execute it which can lead to exploit the website. This can be mitigated by validating user inputs.

Target	Vulnerable	Vulnerability information
DDU	Yes	Injected codes are not validated
ASU	Yes	
WU	No	
HU	No	
Cenzic	No	
acublog	No	

3. Reflected Cross Site Scripting

Target	Vulnerable	Vulnerable Links	Vulnerable Parameters
DDU	Yes	http://www.ddu.edu.et/events/event?id=	id
		http://www.ddu.edu.et/announcements/announcement?id=	id
		http://www.ddu.edu.et/news/article?id=	id
		http://www.ddu.edu.et/authentication/login	username
ASU	No		
WU	Yes	http://www.wu.edu.et/addfeedbacks.php	name
		http://www.wu.edu.et/staffprofilelogin.php	loginname
		http://www.wu.edu.et/staffprofilelogin.php	loginpassword
		http://www.wu.edu.et/index.php	Username
		http://www.wu.edu.et/staffprofilelogin.php	department
HU	No		
Cenzic	Yes	http://crackme.cenzic.com/Kelev/view/updateloanrequest.php	txtAnnualIncome
		http://crackme.cenzic.com/Kelev/register/register.php	UserId
		http://crackme.cenzic.com/Kelev/php/login.php	hLoginType
		http://crackme.cenzic.com/Kelev/view/updateloanrequest.php	txtFirstName
		http://crackme.cenzic.com/Kelev/php/login.php	hUserType
acublog	Yes	http://testaspnet.vulnweb.com/ReadNews.aspx?id=3&NewsAd=	NewsAd
		http://testaspnet.vulnweb.com/Comments.aspx?id=3	tbComment

4. Stored/Permanent Cross Site Scription

Target	Vulnerable	Vulnerable Links	Vulnerable Parameters
DDU	No		
ASU	No		
WU	No		
HU	No		
Cenzic	No		
acublog	Yes	http://testaspnet.vulnweb.com/Comments.aspx?id=3	tbCommen

5. SQL Injection

Target	Vulnerable	Vulnerable Links	Vulnerable Parameters
DDU	Yes	http://www.ddu.edu.et/events/event?id	id
		http://www.ddu.edu.et/authentication/login	username
		http://www.ddu.edu.et/announcements/announcement?id=	id
		http://www.ddu.edu.et/events/event?id=	id
		http://www.ddu.edu.et/news/article?id=	id
ASU	No		
WU	No		
HU	No		
Cenzic	Yes	http://crackme.cenzic.com/Kelev/view/updateloanrequest.php	txtAnnualIncome
acublog	Yes	http://testaspnet.vulnweb.com/Comments.aspx?id=3	tbComment
		http://testaspnet.vulnweb.com/ReadNews.aspx?id	id

		http://testaspnet.vulnweb.com/login.aspx	tbUsername

12.1. Summary of Vulnerabilities

The Ethiopian University websites are vulnreble to critical vulnerabilities such as SQL Injection, Cross Site Scripting, and other attacks. Especially Dire Dawa Univesity and Hawassa university can be exploited and students grade information can easily be modified. Unnecessary information can be posted on the Ethiopian university websites. As can be seen each of the websites have not considered most of the security controls that a web application needs to have. The Ethiopian university websites should assess themselves to find more vulnerabilities and implement an appropirate controls.

Universities such as Assosa university are using an obsolete systems in which the vendor will no more release any patches for any identified vulnerabilities as has been described in the sample vulnerabilities chapter. If those universities continue to use such systems they will be exposed to more attacks and critical information of the target would be exposed.

University administrators are providing critical inforamtion such as credentials to users as has been posted in their facebook page of dire dawa and Hawassa university. The university administrators should be trained not to disclose such information.

If further assessments are conducted on the other university websites including Addis Ababa university, it would be highly probable to obtain critical vulnerabilities as has been seen in the four universities test in this research. So universities should take the intiative to test their websites before being attacked.

CHAPTER EIGHT

DISCUSSION

From similar researches conducted on web application penetration testing methodology, it had been understood that methodologies are mandatory in penetration testing. However, there are a large number of methodologies that are confusing the testers as there are repetitions in some of the testing methodologies, there are testing methods that are not covered by the five testing methodologies, and additional limitations of the most widely used testing methodologies were also identified.

This discussion is organized by analyzing if the objectives of the research have been met and by discussing the measure findings of this research.

8.1. Achievement of Research Objectives

This subsection will summarize if each of the specific objectives specified in the objective of the research are met during conducting the research.

8.1.1. Identifying web application technologies and the methodologies used to test them

This research was conducted by first identifying the targets in black box web application penetration testing. The web application technologies were categorized into web presentation, web application platforms, content management systems, scripting languages, HTTP requests and responses, browser extensions, client side controls.

However, additional controls were identified for a black box web application penetration tester. If the web application is inside a VPN, external penetration testers can't access the web application hence their first target is attacking the VPN. It is after attacking the VPN, either the penetration testers or attackers can start attacking the web application or identifying related vulnerabilities. However, the additional network security controls such as firewall, IPS/IDS, load balancer, honeypot, and any other controls, might defend the application by not allowing to identify vulnerabilities or exploit the web application. As a result, the black box web application penetration tester should bypass those controls.

Additional targets that process the web application request and response were also identified. Those targets are authentication server, web/application/database server, database technologies, other backend technologies and remote management technologies.

In none of the testing methodologies compared in this research recommend as step to bypass the network security controls if the target of test is web application. In addition to this, none of the testing methodologies have any technique to bypass network security controls though ISSAF, NIST, PTES and ISACA have different techniques to assess the vulnerabilities and exploit those network security controls.

The previous researches conducted on the area identified that OWASP, ISSAF, OSTMM, and NIST SP 800-115 are the most common methodologies used in testing. However, PTES and ISACA have been included in this research as they contain detailed testing methodologies for web application penetration testing. In addition to this, OSTMM has no web application penetration methodologies and hence has not been included in this research.

In conclusion fourteen targets were identified that web application penetration tester need to either access or bypass them. OWASP, ISSAF, PTES, ISACA IS audit guideline, and NIST SP 800-115 are the five testing methodologies that have methodologies for any one of the fourteen targets identified in this research.

8.1.2. Developing a set of criteria to compare current and future web application testing methodologies

By analyzing the set of comparison criteria used by previous researches and understanding the requirements of black box web application penetration testing a new set of nine comparison criteria was developed. The nine comparison criteria developed are holistic process, target coverage, repeatable, explainable, attack coverage, manual and automated testing, increasing knowledge of the tester, checklist and futuristic.

a) Synthesizing the current testing methodologies to be used for black box web application penetration testing methodologies

The five testing methodologies are not specific to the fourteen targets identified in this research hence they were first synthesized to those targets. In addition to this, the

testing methodologies were not also specific to black box web application testing hence they were synthesized accordingly.

8.1.3. Comparing, and developing black box web application penetration testing methodologies

By tailoring prince2 to black box web application penetration testing, the penetration testing was categorized into pre-assessment, assessment and post- assessment phases. The pre-assessment phase was developed to be used before the technical assessment was used and contains the components determining the scope of the test, developing project management, tester preparation and project kick off. The post-assessment phase contains reporting (countermeasure selection and risk calculation), presentation and fellow up.

The assessment phase contains information gathering and mapping, vulnerability identification, vulnerability validation, privilege escalation and quality assurance to be applied on the fourteen controls identified in this research.

The methodologies compared in this research proposed the outputs of information gathering to be the inputs of vulnerability identification. However, vulnerabilities can be identified at the information gathering stage and the target might even be attacked at this stage hence the outputs of information gathering can be the inputs of vulnerability identification, vulnerability validation or privilege escalation. The vulnerability identification can be done manually, using automated tools and through research.

The vulnerability validation can be done by removing false positive vulnerabilities by the knowledge of the tester about the targets, and comparing the outputs of tools and finally sample of vulnerabilities are exploited. The vulnerability validation output can be the input of privilege escalation. Testers can enumerate further during vulnerability identification, validation or privilege escalation hence the outputs of those phases can be the input of information gathering and mapping.

There might be fresh and experienced testers in one team. The fresh testers need the support of experienced testers and hence there must be a quality assurance phase during the assessment. The quality of tests at the information gathering, and vulnerability identification should then be checked.

The following table summarizes the comparison and development of methodologies for technical controls. The comparison contains the techniques used in information gathering, vulnerability identification and validation, exploitation and privilege escalation.

No.	target	Number of techniques the methodology has				
		OWASP	**NIST**	**PTES**	**ISSAF**	**ISACA**
1.	Information gathering					
a.	Web Application (17)	11	0	5	6	5
b.	Web Server (10)	5	1	3	8	4
c.	Database(1)	0	0	0	1	0
d.	Proxies(0)	0	0	0	0	0
e.	Load balancer (1)	0	0	0	0	0
f.	Firewall/edge router (5)	2	4	4	4	1
g.	Honeypot(1)	0	0	0	0	0
h.	VPN(1)	0	0	1	1	0
2.	Input Validation (19)	17	0	1	5	4
3.	Authentication and administration (16)	12	2	2	1	0
4.	Authorization(6)	4	0	0	3	0
5.	Session Management testing (8)	6	0	0	1	1
6.	Cryptography Testing(3)	3	0	0	1	1
7.	Business Logic testing (11)	10	0	0	0	0
8.	Bypassing VPN(1)	0	0	0	0	0
9.	Bypassing IDS/IPS(5)	0	0	0	0	0
10.	Bypassing Firewall(5)	0	0	0	1	0
11.	Bypassing Load balancer (2)	0	0	0	0	0
12.	Server testing (5)	2	1	1	1	1

Table 9: Technical Methods comparison summary

1. **Information gathering technical methods summary**

From the above table it can be concluded that none of the testing methodologies have the complete technique. Though OWASP can help to gather information of web application information and web server more than the other methodologies, it is weak in gathering information about the network infrastructure.

2. **Input validation testing comparison summary**

From the 18 techniques OWASP contains almost all techniques necessary in conducting input validation. PTES might incorporate those tests using vulnerability scanners but there is a possibility that those vulnerabilities might not be obtained using automated tools. As shown in the table above ISSAF and ISACA also contain some of the input validation techniques. However, ISACA only describes what to test but not how to test.

3. **Authentication and administration comparison summary**

Out of the 16 black box testing techniques selected in this research for authentication, OWASP has 12 techniques but the other methodologies has almost none. Though again OWASP is better in authentication testing, it is not complete.

4. **Summary of authorization testing Comparison**

OWASP only lacks two additional testing techniques to conduct authorization testing. Contrary to this, NIST, PTES and ISACA don't have a direct way to conduct authorization testing.

5. **Summary of session management comparison**

Though OWASP has most of the testing techniques necessary for session management it has missed one of the critical vulnerabilities as reported by OWASP itself. Cookie poisoning is one of the most common attacks now days.

6. **Cryptography Testing Comparison Summary**

All of the testing techniques used for cryptography testing are from OWASP testing guide though ISSAF and ISACA have a testing requirement for sensitive credentials transmitted over an encrypted channel. The transmission of credentials over an encrypted channel was covered during authentication testing phase.

7. **Business Logic Comparison Summary**

In the previous controls testing methodologies NIST, PTES, ISSAF and ISACA have one or more testing method but in the business logic testing, none of the four testing methodology have a testing method. OWASP excels in the business logic testing compared to other methodology as it has most of the testing methods.

8. Server Comparison Summary

Though OWASP did not take server testing as one control assessment, its configuration testing however contain some server testing techniques. Most of the testing methodologies proposed in server testing use vulnerability scanners and exploitation tools. So it is possible the vulnerability described in server testing might be detected by any of the methdologies though they don't have specific techniques as described above.

9. Summary of Bypassing Network Security Controls

In the previous controls assessment comparisons OWASP covered most of the assessment methods but in bypassing Network security controls the testing guide had nothing useful for the tester. So it is mandatory for the tester to look for other references to bypass network security controls.

8.1.4. Identifying to what type of attack are the four university websites vulnerable

The four Ethiopian universities are vulnerable to the top vulnerabilities as summarized in the table below

No.	Vulnerabilities	DDU	ASU	WU	HU Registrar
1.	SQL injection	Yes	No	No	Yes
2.	CSRF	Yes	Yes	Yes	No
3.	Cross site scripting	Yes	No	Yes	No
4.	Sensitive data exposure	Yes	Yes	Yes	Yes
5.	Broken authentication	Yes	Yes	Yes	Yes

	and session management				
6.	Insecure direct object references	Yes	Yes	No	No
7.	Security mis configurations	Yes	Yes	Yes	Yes
8.	Unvalidated redirects and forwards	Yes	Yes	Yes	Yes
9.	Using known vulnerable components	Yes	Yes	Yes	Yes

Table 10: Top vulnerabilities of four Ethiopian Universities

8.2. Findings of the research

The measure findings of this research are

1. New direction of testing- the methodologies used in this research to develop new methodology considered web application and the other network security controls as separate entities however a black box web application security tester can't test for example for web application target that is inside a VPN from an area that is not a member of the VPN. Hence, the black box web application security tester should find a way to bypass network security controls before a test is conducted on the target.
2. Comparison Criteria- Though the previous researches conducted developed criteria, some of them don't align with black box web application testing hence they were customized and new criteria were added based on the test requirements. Those criteria can be used for new and future testing methodologies comparison and new methodology development
3. New Assessment phase model- this research divided the black box web application penetration into three phases. Though value was added at the pre-assessment and post assessment, more changes and new direction was set at the assessment phase. Additional steps such as quality assurance were added. Enumerating further was modified to take the outputs of vulnerability identification, validation and escalation to information gathering.

The information gathering and mapping output were fed not only to vulnerability identification but also to vulnerability validation and privilege escalation. The vulnerability identification proposed manual, automated and conducting research. This will enable to avoid the limitation of automated tools and new vulnerabilities will be identified using the outputs of the research

4. Complete testing Methodology development- some of the methodologies that are compared in this research focused more on the pre assessment and post assessment phases and the rest focused on the technical assessment. However, in this research the methodology developed has all the phases required for a penetration tester.
5. Additional methods- after synthesizing the existing methodologies into black box web application testing, it was identified that there are some controls that needed additional methods. Hence, based on the fourteen targets new methods were incorporated
6. Identifying the four Ethiopian universities critical vulnerabilities- though the main target of this research wasn't to identify vulnerabilities of Ethiopian universities, it was identified that the Ethiopian universities are vulnerable to the most critical vulnerabilities. Hence the universities should not waste their time to mitigate those vulnerabilities

CHAPTER NINE

CONCULSION

In this research five testing methodologies were compared for black box web application security testing. None of them were specific to black box web application security testing. In addition to this NIST, PTES, ISACA and ISSAF also include other targets such as network devices, social engineering etc. Hence it was necessary first to synthesize those testing methodologies to be used for black box web application testing.

Criteria were developed to compare the existing testing methodologies and based on those criteria the strength and limitations of each testing methodology were identified. The technical controls were categorized in to 8 categories and 116 methods. Among the 116 OWASP has 72 methods, ISSAF has 32 methods, PTES has 17 methods, ISACA has 17 methods, and NIST has only 8 methods. ISSAF, PTES and NIST provide more assistance to the tester at the Pre and Post assessment phases. None of the testing methodologies were strong enough at each of the comparison points so it was necessary to develop new methodology based on those criteria.

The new testing methodology developed in this research has Pre-Assessment, Assessment, and Post Assessment phases. In each phase a testing method or helpful guides were identified and incorporated to be used by black box web application security tester.

Using the developed black box web application security assessment, black box web security assessor are expected to conduct efficient and complete black box web application security assessment.

Almost all of the Ethiopian University websites tested in this research are vulnerable to critical vulnerabilities such as SQL injection, Permanent Cross Site Scripting and code injection. If the universities did not secure the websites it is highly likely they will be attacked and a chaos may occur as a result.

CHAPTER TEN

FUTURE WORK & RECOMMENDATION

10.1. Future work

The research conducted in this research is wide in which each testing method can be a point of research by itself. There are large number of research that could be conducted in this area in the future and some of them are described below

1. White Box Web application Security Testing Methodology Development- the methods and techniques used to develop the black box web application testing methodology in this research can also be applied to white box testing by widening the scope of this research. From the research conducted in this research, it can be concluded that the existing testing methodology doesn't cover many important test phases and techniques hence the same can also apply for white box testing
2. Black and White Box testing methodology of other technology- in this research it was possible to select methods to bypass network security controls so that the application would be exposed to an attack. However, the detailed network infrastructure security assessment has not been covered in this research hence a similar methodology should be developed for other target such as network infrastructure
3. Social Engineering- it is currently one of the most successful attacks in the world as it uses the weakest link to exploit targets. The current methodology developed in this area should be investigated further and methodology should be developed that can cover most of the attack techniques
4. Risk Calculation- one of the limitation of this research's testing methodology is not having risk calculation as it can be by itself one research. So it would be mandatory to incorporate risk calculation.
5. Knowledge and Skill required by testers- organizations have difficulty of identifying the knowledge and skill for their security assessors. Researchers should conduct to set the basic, advanced and expert level knowledge and skill to conduct an effective security assessment by a qualified assessor.

10.2. Recommendation

The methods identified for non technical controls should be tailored as per the agreement between the tester and the client as some of them can be merged or used as they are depending on the skill of the tester in managing and conducting tests. The detail agreement between the tester and the client can be customized as per the test type and the laws that govern the country where the test is conducted.

Not all of the technical assessment methods proposed in this research needs to be tested for every target as some of them mayn't be applicable for the target under test. The tester should understand the target in detail at the information gathering stage. The information gathered at the information gathering and mapping stage will enable the tester to select specific tests as per the target of tests.

The methods developed in this research by identifying the strength and weakness of the five methodologies provides a reference for each of them. The testers should read more in detail the references as examples and detail techniques are expressed in them.

Testers should not take the outputs of tools as final output and provide a report to the client from the direct outputs of the tools because most of the tools identify vulnerabilities from the banners the targets provide. The banners of the targets can be forged hence the outputs of vulnerability scanners might be false positive.

It is highly recommended if multiple testers conduct the target as false positive might be reported and false negative vulnerabilities might be not identified by a single user. The quality assurance process recommended in this research will be helpful to avoid false positive and false negative information at each phase of the tests.

As risk level is one of the criteria that testers recommend the priority that should be followed in implementing the countermeasures for the vulnerabilities identified black box web application penetration testers should choose the right technique in calculating risks.

The final output of penetration testing is a report of the vulnerabilities identified, sample exploits and the possible countermeasures. If the vulnerabilities identified are not mitigated as per their risk level the value of conducting the penetration testing would become zero. Hence, if

clients want their targets to be tested, they must be ready to implement the countermeasures if the risks are not acceptable. Penetration testers should also fellow up the implementation of unacceptable risks after the tests are finalized. It is also recommended to retest the target after the client implements the countermeasures.

The methods proposed in this research can be used for compliance testing. The tester can select the methods of this research after understanding the standard the target plans to comply with. In addition to this, Ethiopia might build its own standard for measuring the security level of web application using this research as an input.

References

[1] Xiaowei Li and Yuan Xue. A survey on Web application security. Department of Electrical Engineering and computer science, Vanderbilt Unviersity.2010

[2]OWASP; OWASP Testing Guide V4.0(17th September, 2014).

available at https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

[3]OISSG; Information Systems Security Assessment Framework; V0.2.1, April 30,2006

[4] Gaithersburg: [NIST Computer Security Division], 2008. Available: http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

[5]ISACA. IS Auditing Procedure Security Assessment- Penetration testing and Vulnerability analysis document P8

[6]PTES. Penetration Testing Execution Standard (16th August 2014)

Available at: http://www.pentest-standard.org/index.php/Main_Page

[7]Najwa Haya. E-learning Stakeholders information security vulnerability Model. Cranefield university, School of Applied Sciences, March 2012

[8]Stanislav Zitta, Josef Horalek, OndrejMarik, SonaNreadove. Conducting effective penetration testing.Faculty of electrical engineering and informatics, university of Pardubice, Czech Republic.

[9]Byeong-Ho Kang. About effective penetration testing methodology.Journal of Security Engineering, 2008.

[10]FakhreldinT.Alssir and Moataz Ahmed. Web Security Testing Approaches: Comparison Framework. Proceeding of the 2011 2nd international congress on computer applications and computational science.

[11]PouryaNikfard, Suhaimi bin Ibrahim, Mohammed Hossein. A comparative Evaluation of approaches for web application testing. The proceeding of International Conference on Soft Computing and Software Engineering, San Francisco State University, CA, U.S.A, March 2013

[12]White Hat Security. 2014 Website Security Statistics

[13] Imperva. Web application attack report, 2012

[14]Russel Miller. The changing face of cyber attacks: understanding and preventing both external and insider breaches. Whitepaper, 2014

[15]Gencer Erdogen. Security Testing of Web Based Applications. Norwegian University of Science and Technology, July 2009

[16] Joel Scambray, Vincent Liu, Caleb Sima. Hacking Exposed Web Applications, Third Edition. McGraw Hill Professional, Oct 15, 201

[17]ISECOMM. 2010. Open Source Security Testing Methodology Manual (OSSTMM). [ONLINE] Available at: http://www.isecom.org/mirror/OSSTMM.3.pdf.

[18] Dafydd Stuttard & Marcus Pinto. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. September 2011, Wiley

Declaration

I declare that the thesis is my original work and has not been presented for a degree in any other university and all sources of materials used have been duly acknowledged.

Name	**Signature**
Gebrekidan Gebremedhin	

This thesis has been submitted for examination with my approval as university advisor.

Name	**Signature**
Workshet Lamenew	