

**VULNERABILITY ASSESSMENT ON BANGLADESHI PUBLIC DOMAIN BASED ON  
OWASP TOP VULNERABILITIES**

**BY**

**Md.Riazul Islam Gisun  
ID: 191-15-12772**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Computer Science and Engineering

Supervised By

**Touhid Bhuiyan**  
Professor & Head  
Department of CSE  
Daffodil International University

Co-Supervised By

**Md. Ismail Jabiullah**  
Professor  
Department of CSE  
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

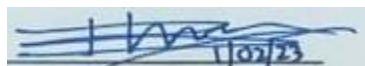
**DHAKA, BANGLADESH**

**JANUARY 2023**

## APPROVAL

This Project titled “**Vulnerability Assessment on Bangladeshi public domain based on OWASP top Vulnerabilities**”, submitted by Md.Riazul Islam Gisun ID No: 191-15-12772 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 29/01/2023.

### BOARD OF EXAMINERS



**Chairman**

---

**Dr. Touhid Bhuiyan**  
**Professor and Head**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



**Internal Examiner**

---

**Arif Mahmud**  
**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

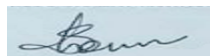


**Internal Examiner**

---

**Saiful Islam**  
**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



**External Examiner**

---

**Dr. Shamim H Ripon**  
**Professor**

Department of Computer Science and Engineering  
East West University

## DECLARATION

I hereby declare that, this project has been done by me under the supervision of **Touhid Bhuiyan, Professor & Head, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**



---

**Touhid Bhuiyan**  
Professor & Head  
Department of CSE  
Daffodil International University

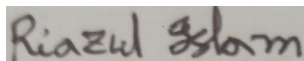
**Co-Supervised by:**



---

**Md. Ismail Jabiullah**  
Professor  
Department of CSE  
Daffodil International University

**Submitted by:**



---

**Md. Riazul Islam Gisun**  
ID: -191-15-12772  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to almighty Allah for his divine blessing makes me possible to complete the final year project/internship successfully.

I really grateful and wish my profound my indebtedness to **Touhid Bhuiyan, Professor & Head**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of “*Cyber Security*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Prof. Touhid Bhuiyan**, and Head, Department of CSE, for his kind help to finish my project and also to other faculty member and the staff of CSE department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

## **ABSTRACT**

In this modern world we can't think a single second without technology. Day by day technology is getting faster and websites are getting feature-full. But in most cases, everyone just forgot one and the most important thing, which is security. Security is the most important thing in this digital world. It's a basic human right. World needs to be concerned about it. In the same Bangladesh needs to focus more on web security. In developed countries they always focus on security first. But in our case, it's not a main concern. In recent years Bangladesh has suffered some massive cyber-attacks. But regardless to this massive blow's developers, companies & even government did not take any noticeable action about this issue. In this research I am going to investigate and study on OWASP top vulnerabilities in Bangladeshi public domain. And I will also describe some common mitigation & our current security level.

## TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Approval	i
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
<b>CHAPTER</b>	
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-2</b>
1.1 Introduction	1
1.2 Motivation	2
1.3 Expected outcome	2
<b>CHAPTER 2: BACKGROUND STUDY</b>	<b>3-14</b>
2.1 Introduction	3
2.2 Selected Vulnerability form OWASP TOP 10 Vulnerability	3
2.3 Classifications	4
2.4 Possible attacks	10
2.5 Literature review	12
2.6 Comparative Analysis and Summary	14
2.7 Scope of the Problem	14

2.8 Challenges	14
<b>CHAPTER 3: METHODOLOGY</b>	<b>15-25</b>
3.1 Introduction	15
3.2 Data collection method	15
3.3 Used tools for data collection	15
3.4 Mitigations	22
3.5 Flowcharts	23
<b>CHAPTER 4: RESULTS AND DISCUSSION</b>	<b>26-28</b>
4.1 Introduction	26
4.2 Assessment result	26
4.3 Discussion	28
<b>CHAPTER 5: IMPACT ON SOCIETY</b>	<b>29</b>
<b>CHAPTER 6: SUMMARY AND CONCLUSION</b>	<b>30-31</b>
6.1 Summary	30
6.2 Conclusions	30
6.3 Implication for Further Study	31
<b>REFERENCES</b>	<b>32-33</b>

<b>FIGURES</b>	<b>PAGE NO</b>
Figure 1: Reflected XSS	5
Figure 2: Stored XSS	6
Figure 3: DOM-based XSS	7
Figure 4: SQLi	8
Figure 5: Cryptographic Failure	9
Figure 6: Security misconfiguration	10
Figure 7: Default user interface of ACUNETIX	16
Figure 8: Directory discovered by ACUNETIX	17
Figure 9: Vulnerability list acunetix	18
Figure 10: Default scan of Nmap	19
Figure 11: AMASS	20
Figure 12: Th30n3p13c3 tool	21
Figure 13: Working process of Th30n3p13c	24
Figure 14: Data collection process	25
Figure 15: Assessment result on pie-chart	27
Figure 16: Acunetix result	27

## **LIST OF TABLES**

<b>TABLES</b>	<b>PAGE NO</b>
Table 1: OWASP TOP 10 Vulnerability	4



# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Nowadays websites have become more dynamic than ever before. It has various features and technique to become more interactive and useful to the user. HTTP is the most common protocol used by websites. Education, healthcare, news, financial transactions, and other services are supplied through web applications [1]. To become interactive and time efficient it uses various techniques like Cookies, sessions, etc. With the help of this feature, hackers can easily take over anyone's account, credit card information even entire web server. Hackers are types of guys who are very passionate about technology and they always try to discover new vulnerabilities and techniques to either exploit or test a website. Websites are getting more feature-full and interactive along with more vulnerabilities. There is an organization called OWASP that ranks these vulnerabilities and gives mitigations to encounter these vulnerabilities. The most common vulnerabilities are SQL injection, XSS and security misconfiguration. These vulnerabilities are swarming in the wild but developers and organizations still do not get any steps to fix them. In this paper I worked on OWASP top ranked vulnerabilities those are SQLi, XSS, security misconfiguration, and cryptographic failure. I used black box penetration testing as data collection.

## **1.2 Motivation**

Bangladeshi public site has more vulnerabilities than developed countries websites. The security flow in ours sites pulls us backward from development. Many people and organizations do not have any idea about these vulnerabilities and it's impacts. By awarding this issue we can make your cyberspace more secure, which will help our countries developed.

## **1.3 Expected outcome**

- 1) Current security level.
- 2) Find out the most common vulnerability.
- 4) Effects of vulnerabilities.
- 3) Common mitigation.

## CHAPTER 2

### BACKGROUND

#### 2.1 Introduction

This section describes the classifications of my selected vulnerabilities, possible attacks & some literature review.

#### 2.2 Selected Vulnerability form OWASP TOP 10 Vulnerability

OWASP ranked vulnerabilities based on their occurrence in 2021 [2]. Cryptographic failure ranked 2nd in OWASP 2021, moves up one spot to #2, formerly A3:2017-Sensitive Data Exposure, which was a symptom rather than a main cause. The emphasis of the new term continues to be implicitly on cryptographic weaknesses. This category frequently results in the exposing of sensitive data or system breach [2]. A drop to third place for injection, which was ranked third in OWASP 2021, occurs. The 33 CWEs mapped into this category had the second highest occurrences in applications with 274k occurrences, with a maximum incidence rate of 19% and an average incidence rate of 3.37%. 94% of the applications were checked for some type of injection. In this version, cross-site scripting has been added to this category [2]. Injection includes XSS and SQLi. OWASP ranked security misconfiguration as 5<sup>th</sup>. Moves up from #6 in the previous edition; 90% of applications were examined for misconfigurations of some kind, with an average incidence rate of 4.5% and more than 208k CWE incidences linked to this risk category. This category moving up is not unexpected given the increasing trend toward highly customizable software. This risk category now includes the previous A4:2017-XML External Entities (XXE) category [2].

<b>S/NO</b>	<b>OWASP TOP 10 - 2021</b>
1	<b>A01:2021-Broken Access Control</b>
2	<b>A02:2021-Cryptographic Failures</b>
3	<b>A03:2021-Injection</b>
4	<b>A04:2021-Insecure Design</b>
5	<b>A05:2021-Security Misconfiguration</b>
6	<b>A06:2021-Vulnerable and Outdated Components</b>
7	<b>A07:2021-Identification and Authentication Failures</b>
8	<b>A08:2021-Software and Data Integrity Failures</b>
9	<b>A09:2021-Security Logging and Monitoring Failures</b>
10	<b>A10:2021-Server-Side Request Forgery</b>

**Table 1.** 2021 - OWASP TOP 10 Vulnerability

## **2.3 Classifications**

In this section includes the description of vulnerability and their types.

### **2.3.1 XSS**

Cross-site scripting, commonly referred to as XSS, is a vulnerability in web security that enables an attacker to alter the way a vulnerable application is interacted with by users. This vulnerability is what allows the same origin policy, which separates different websites from one another, to be implemented. XSS weaknesses often permit the attacker to impersonate the victim user and

perform any actions they are authorized to do, as well as gain access to any of the user's information. If the targeted user has privileged access within the application, the attacker may have complete control over all its features and data. The three primary types of XSS attacks those are given below. [3]

### A. Reflected XSS

Reflected The most fundamental type of cross-site scripting is XSS. When an application obtains information from an HTTP request and unsafely combines that information into the immediate reply, then it arises. [3]

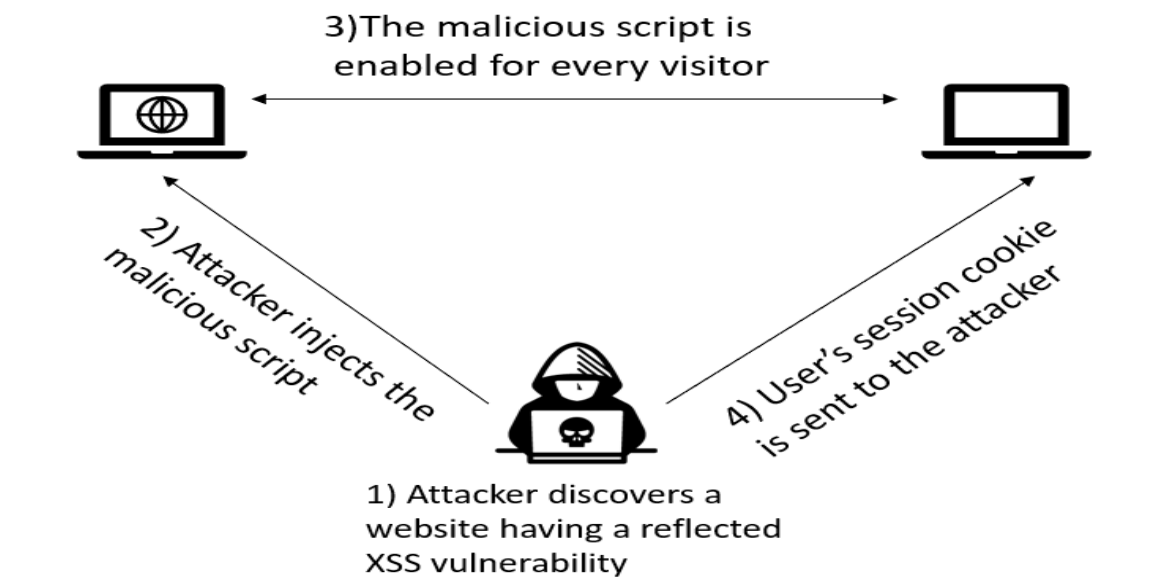


Figure 1. Reflected XSS

## A. Stored XSS

Stored XSS, often referred to as persistent or second-order XSS, occurs when an application acquires information from an unreliable source and includes that information in later HTTP replies in a risky way. The application may receive the relevant information via HTTP requests, such as comments on a blog post, chat room user names, or contact details or a customer purchase. Other times, the data may originate from unreliable sources, such as a webmail software that displays SMTP-received messages, a marketing tool that displays updates from social media, or a network monitoring program that displays packet data from network traffic. [3]

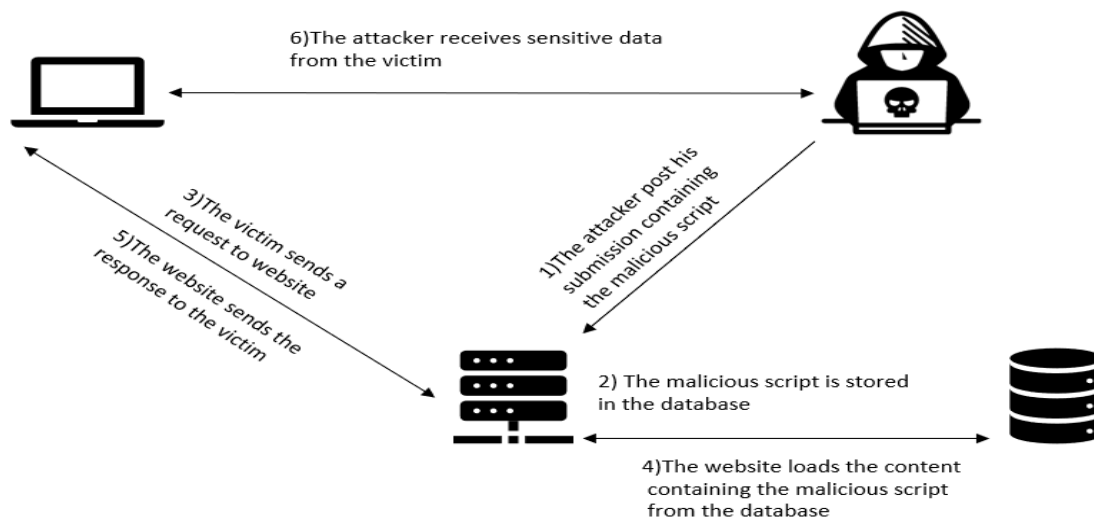
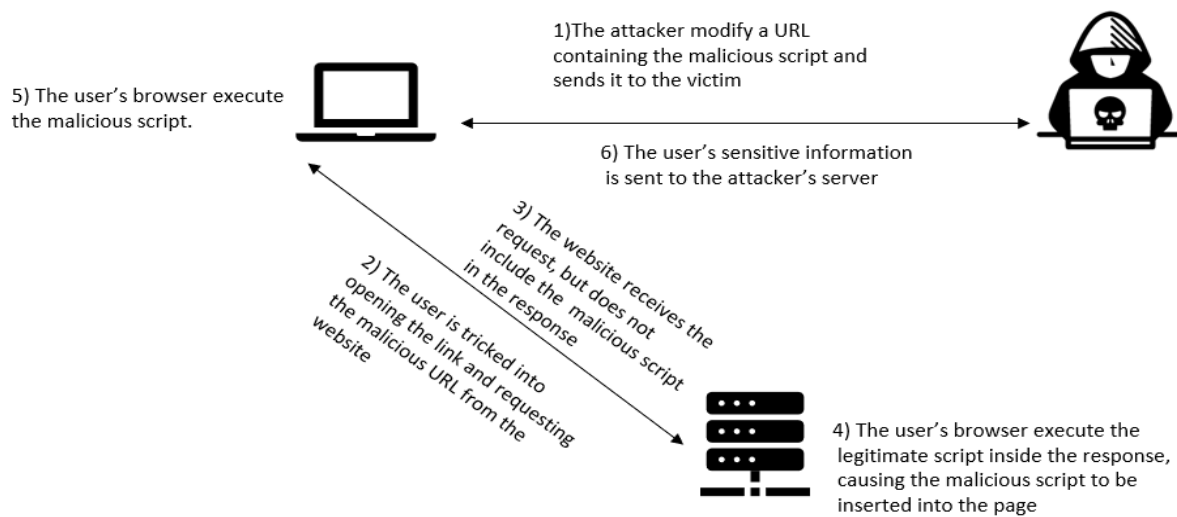


Figure 2. Stored XSS

## B. DOM-based XSS

An occurrence of DOM-based XSS, or DOM XSS, happens when a program has JavaScript on the client side that mishandles untrusted data by writing it back into the Document Object Model in an unsafe way. [3]



**Figure 3.** DOM-based XSS

### 2.3.2 SQLi

SQL Injection (SQLi), a type of injection attack, allows malicious SQL instructions to be executed. A database server that is in front of a web application is managed via these commands. Attackers can circumvent application security measures by using SQL Injection weaknesses. The full content of a SQL database can be retrieved by getting past authentication and authorization of a web page or online application. Additionally, they can use SQL Injection to add, modify, and remove records from the database. Any website or web application that utilizes a SQL database, such as MySQL, Oracle, SQL Server, or others, may be vulnerable to SQL Injection. [4]

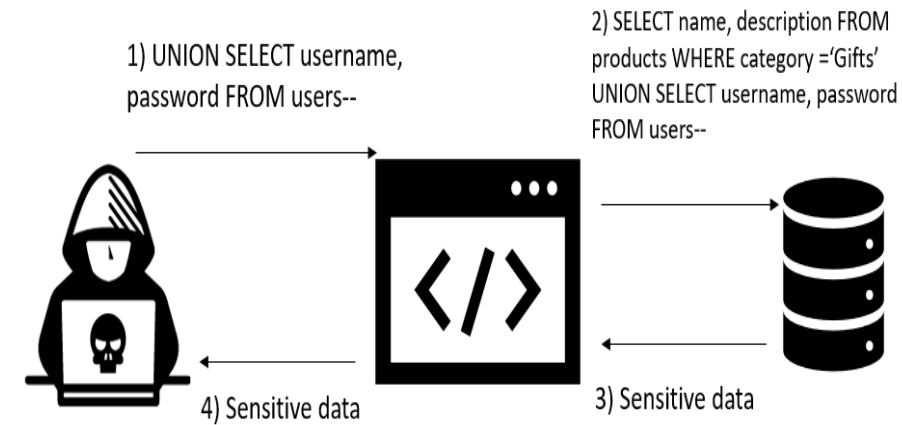
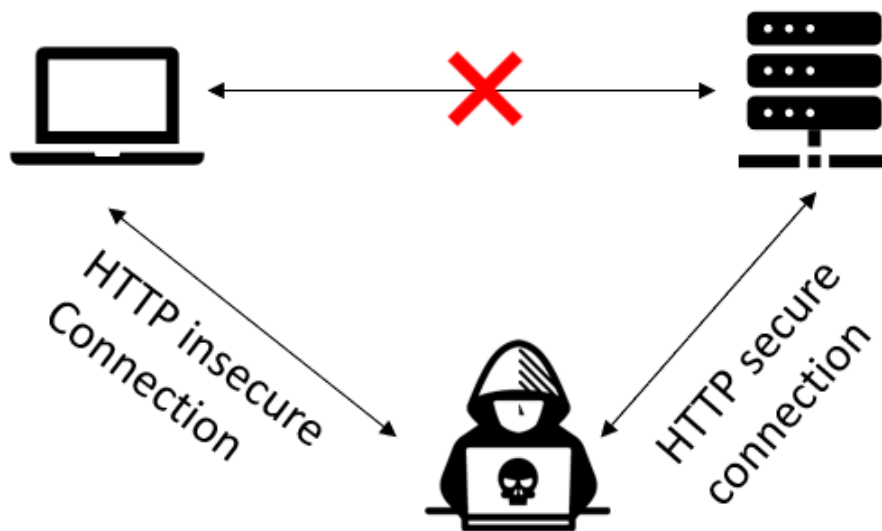


Figure 4. SQLi

### 2.3.3 Cryptographic Failure

A cryptographic failure, a major vulnerability in web application security, exposes private application data to a broken or absent cryptographic system. Trade secrets, credit card information, email addresses, patient medical records, passwords, and other sensitive user data are some examples. [5]



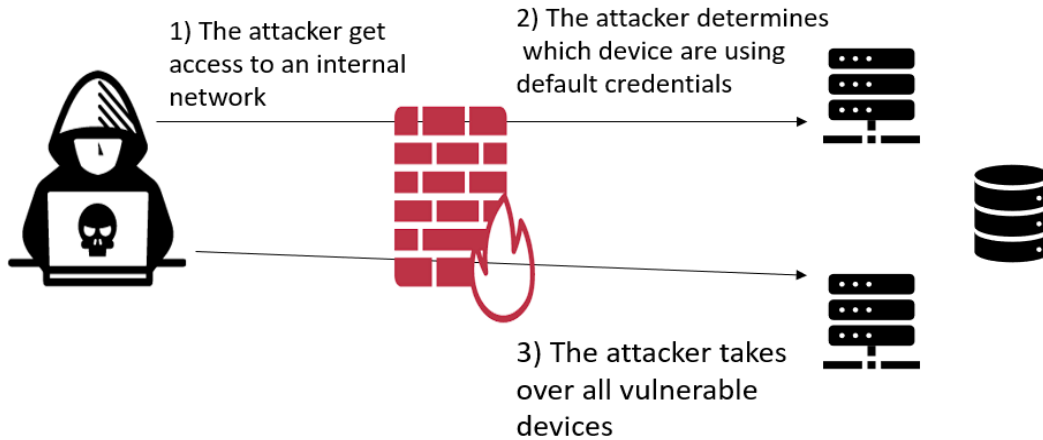


---

Figure 5. Cryptographic Failure

#### iv. Security misconfiguration

Security Misconfiguration happens when security settings are made, used, and kept as defaults. For effective security, the application, web server, database server, and platform must all have secure configurations that have been established and put into place. The software's update status is also very important. [6]



**Figure 6.** Security misconfiguration

## 2.4 Possible attacks

In this section I will describe about some common attack related to my selected vulnerabilities.

### 2.4.1 Attack scenario with security misconfiguration

When directory listing is not disabled on the server, an attacker can easily discover this and list all the folders on the server. They can then locate and download the Java classes, decompile and reverse-engineer the code to inspect the source code. Through this process, the attacker may uncover a significant vulnerability in the program's access control. [6]

### 2.4.2 Attack scenario with Cryptographic failure

A website provides poor encryption or doesn't utilize or enforce TLS on all of its pages. A hacker keeps an eye on network activity (for instance, on a wireless network that is not secure),

downgrades connections from HTTPS to HTTP, intercepts requests, and takes the user's session cookie. The attacker then uses this cookie again to hijack the user's (authenticated) session in order to access or change the user's personal information. As an alternative to the aforementioned, they might change any transferred data, such as the beneficiary of a money transfer. [5]

### **2.4.3 Attack scenario with SQLi**

Think about a shopping app that shows items in several categories. When the user clicks on the Price category, their browser requests the below URL. [4]

URL: <https://insecure-website.com/products?category=Price>. [4]

This causes the application to make an SQL query to retrieve details of the relevant products from the database: `SELECT * FROM products WHERE category = 'Price' AND released = 1`. [4]

### **2.4.4 Attack scenario with XSS**

An attacker can simply steal a cookie from a user who has been authorized if the application doesn't check the input data. All the attacker has to do is to place the following code in any posted input (ie: message boards, private messages, user profiles). [7]

Malicious code:

```
<script>alert(document.cookie)</script>
```

## 2.5 Literature review

In 2016, Delwar Alam et al. presented vulnerability assessment in XSS and CSRF on 500 web sites, they found 335 were vulnerable with both attacks. Their data collection method was black box testing. In their figure 8 they showed the percentage of XSS and CSRF vulnerability. The statistics showed 75% of the 335 web applications were vulnerable to CSRF attack and 65% were vulnerable to XSS attack. And within this about 40% were vulnerable to both XSS and CSRF attacks. [8]

In 2020, Md. Abdur Rahman et al. Showed that across a range of 1 to 12 web applications and 5 different types of vulnerabilities, the total number of vulnerabilities detected ranged from 21 to 54. Specifically, vulnerabilities in CSS, HTML, BL, EE, and DL were recorded as 2228, 1524, 764, 1708, and 206 respectively. The research showed that CSS-type vulnerabilities were the most prevalent among all the types identified. Additionally, the study revealed that the number of vulnerabilities detected per web application varied, with the lowest recorded number being 3 and the highest being 2208. [9]

In 2015, Delwar Alam et al. conducted a study on the security of 900 web applications in the .bd public domain. They discovered that 600 of these web applications had vulnerabilities, with 510 being vulnerable to SQL injection via GET requests and 90 being vulnerable to SQL injection via POST requests. [10]

In 2019, Md. Sadek Ferdous et al. performed vulnerability measurements. In their figure 5 showed 36% of the websites were secure and the remaining 64% of the websites were run with multiple vulnerabilities. Where SQL injection, XSS, and TLS vulnerability were common. They found 56% of the websites were vulnerable to SQL Injection while 53% of the websites were vulnerable to XSS. They also found that 21% of all websites were vulnerable to BAS and 15% of all websites were found to be vulnerable to CSRF. And The vulnerability percentage for BAS and CSRF was low. The main reason for that the majority of the government websites did not

have essential authentication mechanisms in the website. For that reason, CSRF and BAS applied to about 50% of the website. [11]

In 2015, Md Alamgir Kabir et al. evaluated 600 web applications of the .bd domain. Among these websites, 400 websites were found vulnerable and 60 web applications were vulnerable to post-based SQLi. They found the key reason for that. The reason was the language and the platform of the development. In the post-based vulnerable dataset, 49% of the vulnerable web applications were developed using PHP version 5 or higher. About 46% were built using PHP version 4.4.9. And in their findings, less than 2% of websites were built with Joomla version 2.5 and 3% were built with ASP.net version 4 or lower. In their dataset of post-based SQLi vulnerable web applications, 53.33% of the applications were built using PHP version 4.9 and less. And the rest of the 43.33% were built using PHP version 5. [12]

In 2017, Moniruz Zaman et al. examined 1298 web applications and found that 538 of them were using the SSL/TLS protocol for security. The remaining 753 did not have any security protocol or were using other types of security. As illustrated in their Figure 6, 42% of the web servers employed the HTTPS protocol. Among the 538 SSL/TLS web applications, 156 were using older versions of SSL/TLS and 382 were using updated versions, as depicted in their Figure 7. These versions were determined by whether they were published before or after the discovery of the "Heartbleed" vulnerability. [13]

In 2015, Delwar Alam et al. Evaluated 359 educational websites in Bangladesh. Among these websites, 309 were found vulnerable to various types of SQLi attacks. They used a manual black box testing approach for data collection. [14]

In 2015, Tanjila Farah et al. conducted a study on SQL injection vulnerabilities in Bangladesh. They found that many web applications in the country lacked adequate security measures. The study was based on manual penetration testing of 108 financial web applications. According to the statistics presented in their Fig 11, 60% of the web applications in the dataset had no or inadequate security systems, making them vulnerable to basic SQL injection attacks. 30% of the

web applications had MOD security IDS that were also susceptible to SQL injection attacks, and 10% of the vulnerable web applications had forbidden 404 BAD request securities. [15]

## **2.6 Comparative Analysis and Summary**

Bangladeshi public domain is suffering from cyber-attack for a long time. The reason is not the websites but also the mentality of the society. Most of them thinks this is not an issue at all. By this analysis we can re-open their eyes. In my research I have used data using automation. I used ACUNETIX as main tool alongside with NMAP and AMASS, I also make my own script to collect data. In this assessment I almost scan all of the subdomain from my selected domain list. I worked with four different kind of vulnerability in my assessment. Those vulnerabilities are top listed in OWASP top ten vulnerability table 2021.

## **2.7 Scope of the Problem**

There is difficulty in every work, so I am not an exception I have faced lots of difficulty to collected data. And make a workable tool to collect data. In my data collection process ACUNETIX consume a large amount of time. And it's needs to run nonstop and with fast internet. So, in this area I have suffered a lot of problem.

## **2.8 Challenges**

There is no such task can be completed without challenges. We can avoid challenges; we have to perform our best to get rid of them. And I did in my assessment. I ran my tools nonstop to collect data. There were lots of bugs in my script I get ride all off them. And make a workable tool to scan website.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction**

In this section I will talk about my data collection method, tool, step to collect data and flow charts and mitigations.

#### **3.2 Data collection method**

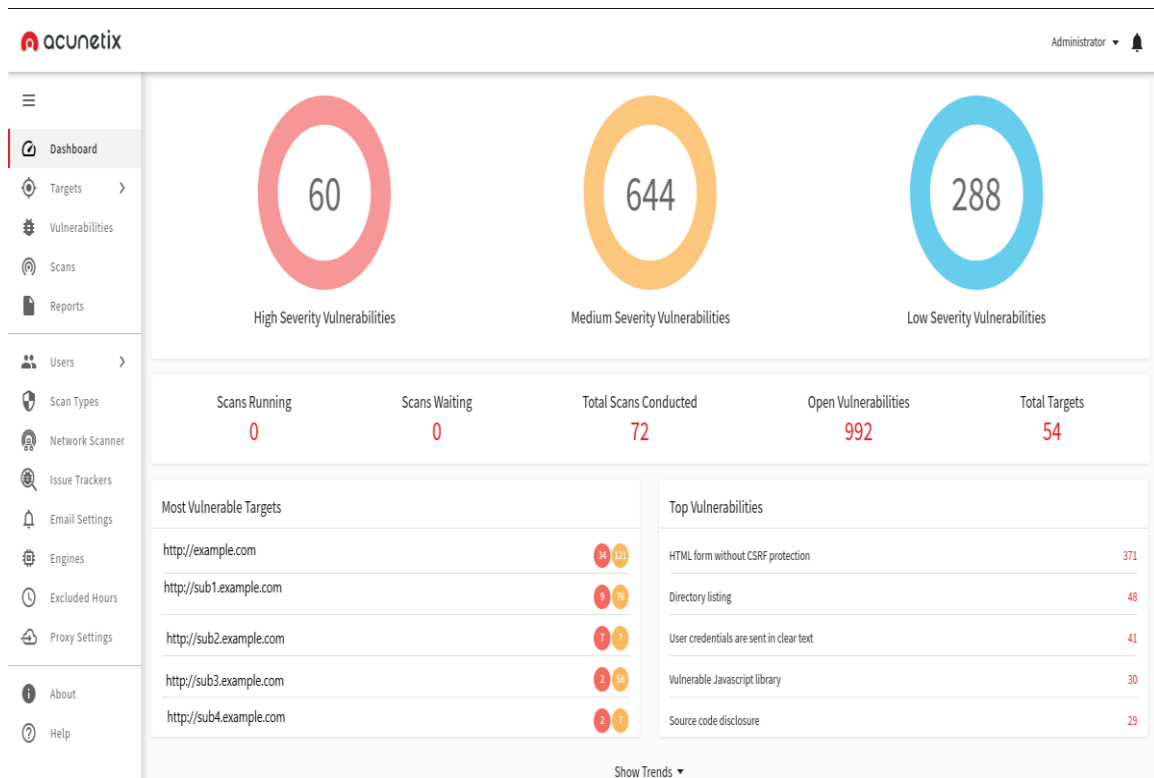
I have collected most of my data in an automated manner by using acunetix. I have also used some manual approach to collect data from a site. Such as ports scan and subdomain discovery.

#### **3.3 Used tools for data collection**

For my data collection purpose, I used ACUNETIX as my main tool. Alongside with it I also used NMAP, AMASS and our own command line tool.

##### **3.3.1 ACUNETIX**

Acunetix is a tool that automates the process of evaluating the security of web applications, it scans websites for vulnerabilities such as SQL Injection and Cross-Site Scripting, and other potential weaknesses. [16]



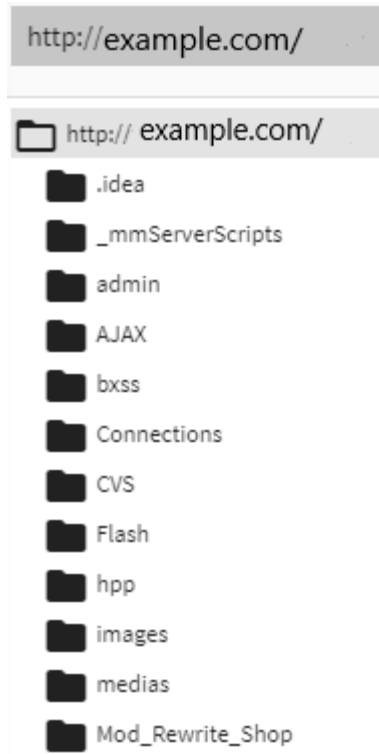
**Figure 7.** Default user interface of ACUNETIX

### 3.3.1.1 How ACUNETIX Works

ACUNETIX Work flow is given below:

1. ACUNETIX Deep-scan examines every link on the website, including links created dynamically using JavaScript, links identified in robots.txt, and links detected in sitemap.xml (if available). ACUNETIX will use the generated site map to perform specialized tests against each section of the site. [16]





**Figure 8.** directory discovered by ACUNETIX

2. When Acunetix AcuSensor technology is enabled, the sensor retrieves a list of all files present in the web application directory and adds any files not detected by the crawler to the crawler output. Such files are usually not accessible from the web server or linked from the web site, so they are not detected by the crawler. Acunetix AcuSensor analyzes files that are not accessible over the Internet. Such file like web.config. [16]
3. Following crawling, the scanner automatically launches a number of vulnerability tests on each page it locates, effectively playing the role of a hacker. Acunetix examines every page for areas where you may submit information and tests all possible combinations of inputs. The automated scanning phase is at this point. AcuSensor technology will launch a number of extra website vulnerability assessments once it is enabled. [16]

The screenshot shows the 'Vulnerabilities' section of the Acunetix interface. At the top, there are buttons for 'WAF Export', 'Generate Report', 'Mark as', 'Retest', and 'Send To Issue Tracker'. Below these is a filter bar with 'Status: Open' and a search icon. The main area contains a table of vulnerabilities.

<input type="checkbox"/>	Severity ↓	Vulnerability	URL	Parameter	Status	Confidence %	Last Seen
<input type="checkbox"/>	High	Blind SQL Injection	http://example.com/		Open	95	
<input type="checkbox"/>	High	Blind SQL Injection	http://example.com/		Open	95	
<input type="checkbox"/>	High	Cross site scripting	http://example.com/		Open	95	
<input type="checkbox"/>	High	Cross site scripting	http://example.com/		Open	95	
<input type="checkbox"/>	High	Cross site scripting	http://example.com/		Open	95	
<input type="checkbox"/>	High	Cross site scripting	http://example.com		Open	100	

**Figure 9.** vulnerability list acunetix

4. The Scan Results display the vulnerabilities found. Each vulnerability warning includes details on the issue, including the POST data used, the item that is impacted, the server's HTTP response, and more. [16]
5. Details such as the source code line number, stack trace, or impacted SQL query that led to the vulnerability are shown if AcuSensor Technology is used. There are also suggestions for fixing the vulnerability. [16]
6. Executive Summary reports, Developer reports, and different compliance reports, such as OWASP 2017 or ISO 270001, may all be created on successful scans. [16]

### 3.3.2 NMAP

Nmap is an open source and free tool used by security professionals and network engineers for network discovery and security auditing.

```
└─$ nmap example.com
Starting Nmap 7.92 ( https://nmap.org )
Nmap scan report for example.com
Host is up (0.038s latency).
Other addresses for example.com (not scanned):
rDNS record for
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
```

Figure 10. default scan of Nmap

### 3.3.3 AMASS

A free tool used for DNS discovery, subdomain discovery and many more. By this command line tool anyone can perform a scan to collect websites subdomain and other valuable information.

```

└─$ amass
tools

.+++;.
+W@#@#@#@#@#  S+W@#  oBWB:  .+++  +W@#@#@#@#@#  oW@#@W#+
S@#+  .o@##.  .@#@o@W.o@#@o  :@#@S@W8o  .@#  .:oW+  .@#+@+@S#S
+@S  S@S  #@8  +@W@S8@+  :@W.  +@8  +@:  .@8  .@8
@#@  @#@  @#@  @#@  WW  .@W  W@+  .@W.  .@#  .@#
WW  S@o  S@:  o@+  o@+  #@.  S@o  +W@#+.  +W@8:
#@  :@W  S@+  S@+  @8  :@o  o@o  oW@#@W+  oW@8
o@+  @@S  S@+  S@+  #@  S@.  .W@W  .+@@S  o@W.
WW  +@W@8.  S@+  :S  o@+  #@  :@W@S@S  S@:  .:  :@o
:@W:  o@#  +W@  S@+  :W:  +@W@o++o@W.  S@S  S@#o+S@W.  #@:  o@+
:W@W@W@W@W@W@8  +  :S@W@#@#@#@  S@W  .o#@@W@S.  :W@W@W@W@#@S
+o@S@S@S+.  +o@o@.

v3.19.3
OWASP Amass Project - @owaspamass
In-depth Attack Surface Mapping and Asset Discovery

Usage: amass intel|enum|viz|track|db [options]

-h Show the program usage message
-help Show the program usage message
-version Print the version number of this Amass binary

Subcommands:

    amass intel - Discover targets for enumerations
    amass enum - Perform enumerations and network mapping
    amass viz - Visualize enumeration results
    amass track - Track differences between enumerations
    amass db - Manipulate the Amass graph database

The user's guide can be found here:
https://github.com/OWASP/Amass/blob/master/doc/user_guide.md

An example configuration file can be found here:
https://github.com/OWASP/Amass/blob/master/examples/config.ini

The Amass tutorial can be found here:
https://github.com/OWASP/Amass/blob/master/doc/tutorial.md

```

Figure 11. AMASS

### 3.3.4 Th30n3p13c3

It's a python script we made. It's a pretty straight forward tool to perform vulnerability scan for basic level vulnerability assessment.

```
#####
# 1) Port scan. #
# 2) Cryptographic failure. #
# 3) Security misconfiguration. #
# 4) Directory traversal. #
# 5) Reflected xss. #
# 6) Exit #
# MADE BY R1@2 #
#####

Enter the number: █
```

**Figure 12.** Th30n3p13c3 tool

### 3.3.4.1 Features

- 1) Port scan: It can discover unnecessary ports used in a web application.
- 2) Cryptographic failure: By automatically checking the SSL certificate, it can determine a web app has a cryptographic failure or not.
- 3) Security misconfiguration: By brute forcing in the URL it can find out default pages and admin pages on a web site.
- 4) Directory traversal: In the same manner as security misconfiguration this feature can discover directory in a website.
- 5) Reflected XSS: By automatically search a string on a search bar if present, can find out a possibility of a reflected XSS attack.

### **3.4 Step to collect data**

- 1) Perform a mass scan to discover all the subdomains of a web site.
- 2) After determining the target, we used Nmap to scan all ports. By port scanning we listed a web page is vulnerable to security misconfiguration or not.
- 3) Then we put subdomain and the main domain in acunetix to perform full scan.
- 4) After that we have also used our tool to scan for possible basic level vulnerability.

### **3.5 Mitigations**

In this part we are going to discuss some mitigations for our selected vulnerability.

#### **i. Mitigation for SQLi**

Only parametrized queries with prepared statements and input validation can effectively prevent SQL Injection attacks. Never let the application code directly use the input. Every input should be sanitized by the developer, not just web form inputs like login forms. Remove all single quotes and other possibly dangerous code elements. It's a good idea to turn off the display of database errors on your live website. By taking advantage of database flaws, SQL Injection may be utilized to learn more about your database. [4]

#### **ii. Mitigation for XSS**

Most effective solution for XSS to Sanitize user input. The user input always has to be checked. The output should encoded prevent potentially malicious user-provided data. User input data Should be limited. And the use of CSP can be pretty handy to prevent XSS attacks. [17]

### **iii. Mitigation for Cryptographic failures**

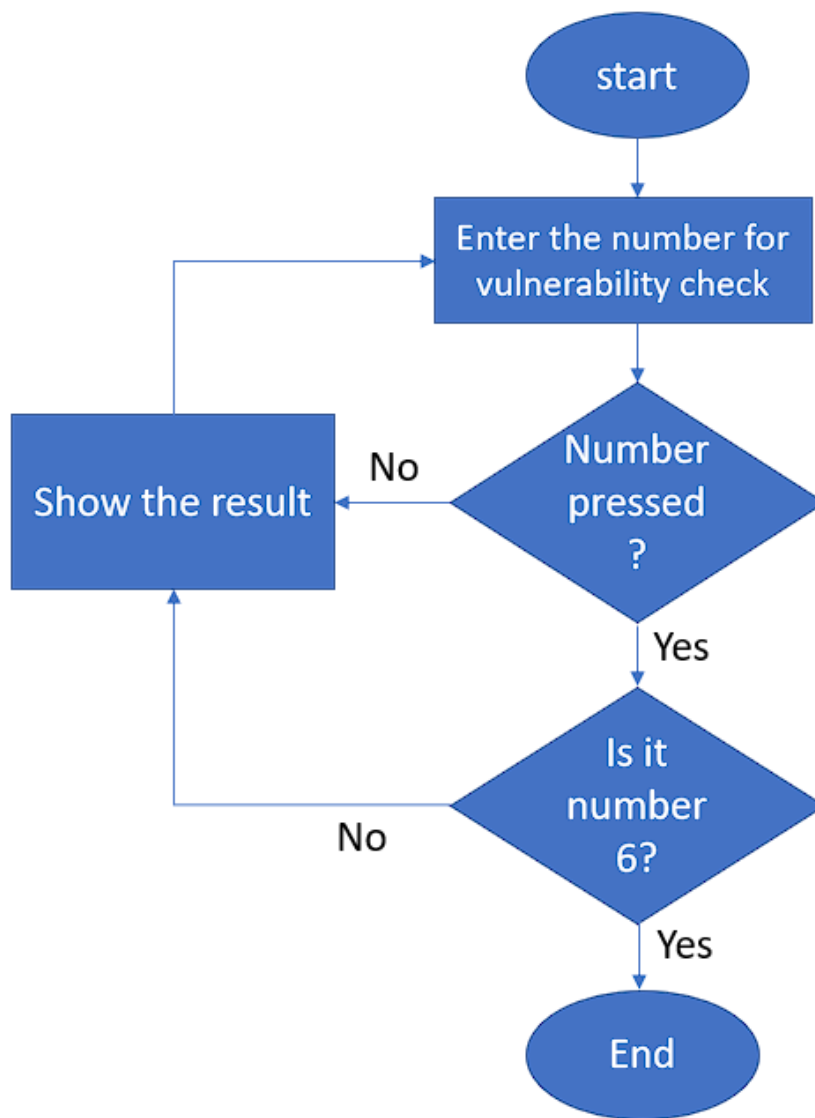
It is advised that all encryption keys be generated using cryptography. Byte arrays should be used to store them. Passwords in plain text should always be encrypted with these keys or converted to cipher text. Only a reliable encryption technique or algorithm should be used. For extra security, sensitive data might be encrypted with long salts. [18]

### **iv. Mitigation for security misconfiguration**

There is some common thing to prevent this vulnerability, those are disabling default accounts, encrypting data and Enforcing strong access controls. By ensuring these things we can decrease the risk of security misconfiguration. [19]

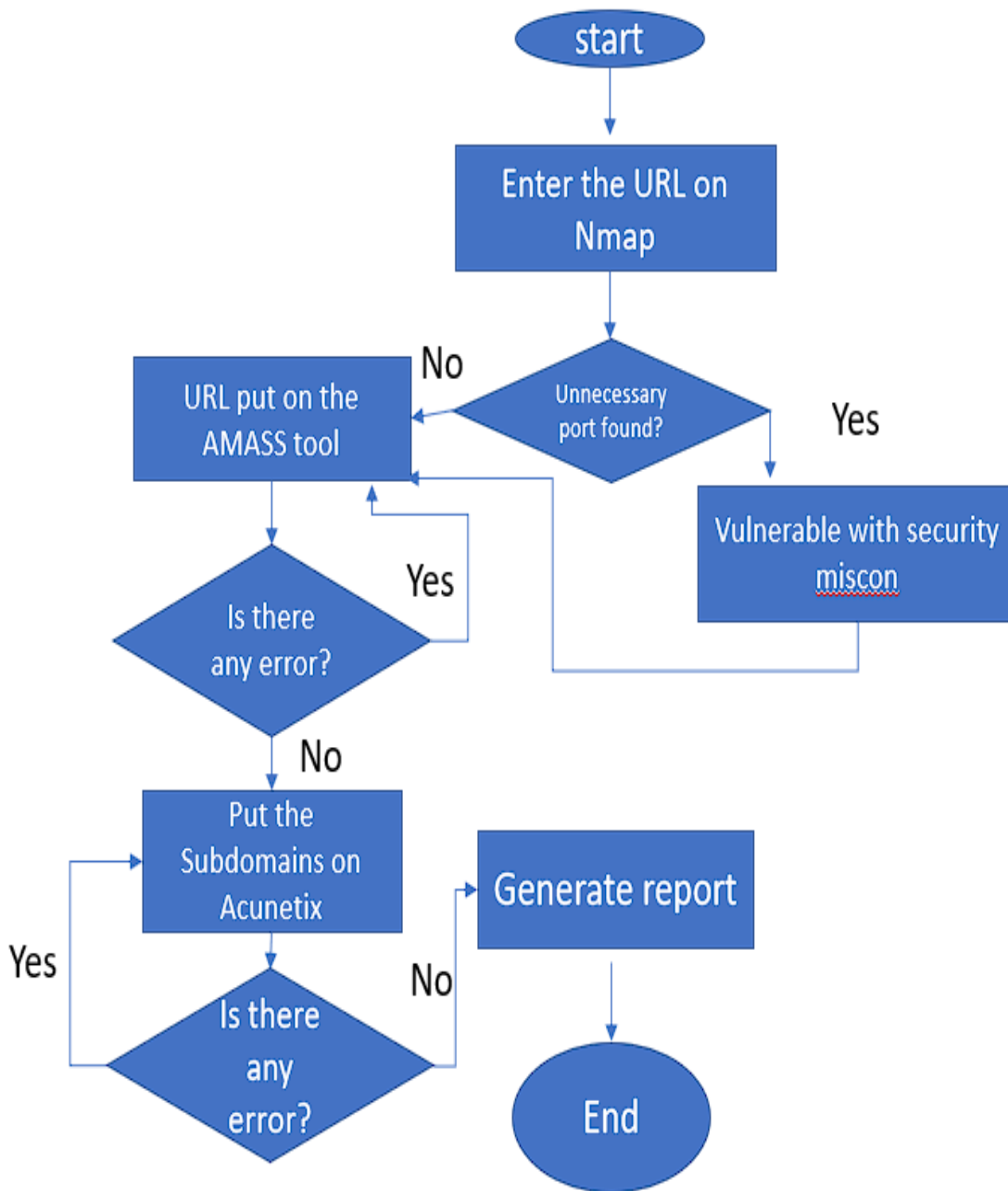
## **3.6 Flowcharts**

In this section we are going to show some flowcharts based on our work.



**Figure 13.** working process of Th30n3p13c





**Figure 14.** Data collection process

## **CHAPTER 4**

### **RESULTS AND DISCUSSION**

#### **4.1 Introduction**

To collect data, I used Acunetix as my main tool, I also used Nmap, amass & my own tool for that purpose. I scanned almost all submain from my selected domain. My whole process was automated and manual both. By this analysis developer will get some idea about the current situation of the Bangladeshi cyber space.

Form my analytical research, we also get a better idea about which vulnerability are present in our websites most. And I can also say that from my research, subdomains are more vulnerable form main domain.

#### **4.2 Assessment result**

From my ten websites, we can see that from figure 15 below, ten out of ten websites have some kind of cryptographic failure and sensitive data exposure both. In the second position security misconfiguration comes in. where almost seven websites are vulnerable to it. And after that XSS and SQLi comes. two of my tested websites were found to be vulnerable to it. And in the end, my last vulnerability SQLi comes. Where I found one website is vulnerable to SQLi.

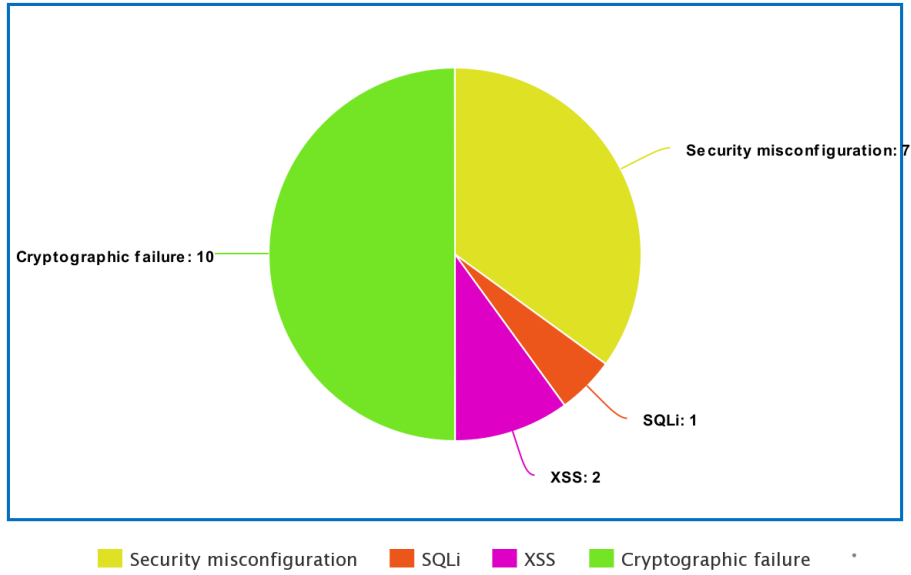


Figure 15. assessment result on pie-chart

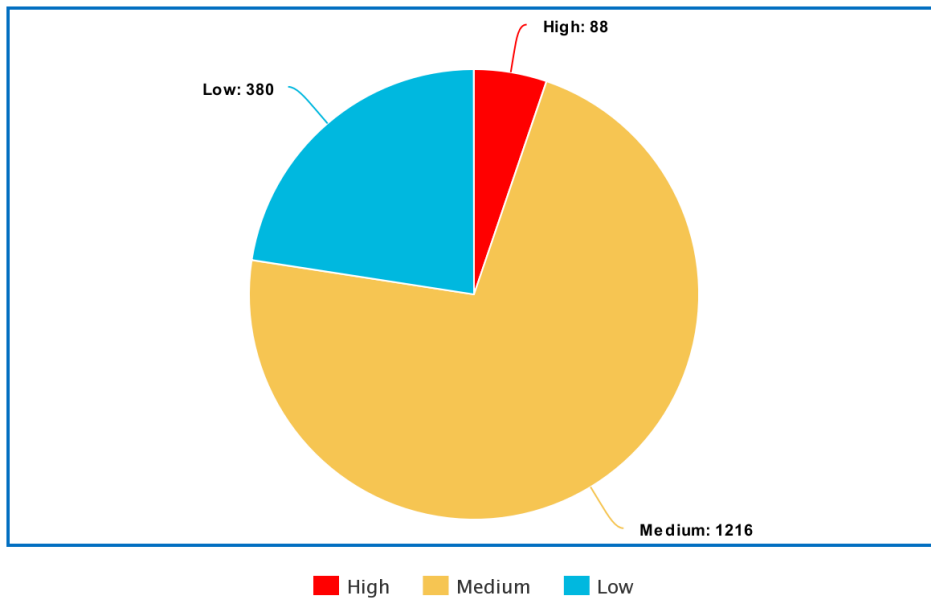


Figure 16. Acunetix result

We can see that from acunetix result in my assessment the major portion of vulnerability is Medium category vulnerability. I found that almost 1216 websites are vulnerable with medium category issue, 380 websites are vulnerable with Low category vulnerability and 88 websites are vulnerable to High category vulnerability.

### **4.3 Discussion**

In my finding I have found that subdomains are more vulnerable than main domain and the reason behind that is people give their effort in subdomain lot less than main domain. We also see that from figure 16 in my analysis most of the vulnerability was Medium category vulnerability and those vulnerability can be pretty much dangers as High-level vulnerability if anyone exploit that with good technique.

## **CHAPTER 5**

### **IMPACT ON SOCIETY**

Web vulnerability is not a new thing. Because of fast development of technology people sometime forget about to fix the previous one. We are human kind it's our nature to forget things. By utilizing this weakness bad people in other word black hat hackers always try to get access to private network and system. They not also get access but also destroy and steal data form it. Whole world is suffering from this kind of issue for a long time and they will if they do not take a step-in right time. Any company or organization can be destroyed in one night by utilizing vulnerability and information. In our country most of the company does not take it seriously. But compare with other countries they come up with some fast solutions for that, those are own cyber security team, private penetration tester hiring and bug bounty. Bugs or issues can't be solved from the zero day of the application. So, the technology or software should be in a continues process for make it better and better. There is a good chance that if we do not take any step to solve it issues or society will suffer the most. Peoples trust will be gone from our websites. So, we should make cyber security and penetration testing our first priority.

## **CHAPTER 6**

### **SUMMARY AND CONCLUSION**

#### **6.1 Summary**

Our websites are suffering from cyber-attacks for a long time. In 2016 Bangladesh bank suffered the most dangerous attack ever. Bangladesh bank lost \$951 million dollars. Attack will never stop until we defend our self. For defending purpose, we need to be more careful about little things. In my research I showed the vulnerability level in our sites. And I also give some common mitigation for this vulnerability. I have faced the most difficult situation in data collection. I performed my scan continually to get good reports. My data collection mainly based on automation. That's why I have faced some less issues in data collection compare to fully manual approach.

#### **6.2 Conclusions**

Nowadays our country focused on modern technology. As a developing country government already working with some solution for cybersecurity and information security. Our countries ICT department spreading the awareness of cyber security. The result show that we have medium level vulnerability and low-level vulnerability more than high level vulnerability. Low level vulnerability and high-level vulnerability can be pretty much dangerous too. So, we should not overlook these categories.

### **6.3 Implication for Further Study**

In this research I worked with very little amount of data for analysis. And my analysis was mostly based on automation. Our script is based on python and it's a basic level command line tool. I am working on it. In the future I will like to work with large amount of data. And I want to collect data in manual approach. I also like to make a for interactive tool in the future which will be more user friendly and accurate in data collection and vulnerability assessment.

## REFERENCES

- [1] M. Mirjalili, A. Nowroozi, and M. Alidoosti, "A survey on web penetration test Security Control via Ensuring Immunity View project A survey on web penetration test," 2014. [Online]. Available: <https://www.researchgate.net/publication/270523617>
- [2] "Welcome to the OWASP Top 10 - 2021." <https://owasp.org/Top10/> (accessed Oct. 20, 2021).
- [3] "Cross-site scripting." <https://portswigger.net/web-security/cross-site-scripting> (accessed Apr. 10, 2021).
- [4] "What is SQL Injection (SQLi) and How to Prevent It." <https://www.acunetix.com/websecurity/sql-injection/> (accessed Jun. 05, 2022).
- [5] "OWASP Top 10 Cryptographic Failures A02 – Explained." <https://crashtest-security.com/owasp-cryptographic-failures/> (accessed Sep. 12, 2022).
- [6] "Security Misconfiguration." [https://www.tutorialspoint.com/security\\_testing/testing\\_security\\_misconfiguration.htm](https://www.tutorialspoint.com/security_testing/testing_security_misconfiguration.htm) (accessed Jul. 19, 2022).
- [7] "Cross Site Scripting (XSS)." <https://owasp.org/www-community/attacks/xss/> (accessed Nov. 01, 2022).
- [8] T. Farah, M. Shojol, M. Hassan, and D. Alam, "Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF," in *2016 6th International Conference on Digital Information and Communication Technology and Its Applications, DICTAP 2016*, Aug. 2016, pp. 74–78. doi: 10.1109/DICTAP.2016.7544004.
- [9] M. Abdur Rahman, M. Amjad, B. Ahmed, and M. Saeed Siddik, "Analyzing web application vulnerabilities: An empirical study on e-commerce sector in Bangladesh," in *ACM International Conference Proceeding Series*, Jan. 2020. doi: 10.1145/3377049.3377107.
- [10] D. Alam, M. A. Kabir, T. Bhuiyan, and T. Farah, "A Case Study of SQL Injection Vulnerabilities Assessment of. bd Domain Web Applications," in *Proceedings - 4th International Conference on Cyber Security, Cyber Warfare, and Digital Forensics, CyberSec 2015*, Jun. 2016, pp. 73–77. doi: 10.1109/CyberSec.2015.23.
- [11] M. Moniruzzaman, F. Chowdhury, and M. S. Ferdous, "Measuring Vulnerabilities of Bangladeshi Websites," in *2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019*, Apr. 2019. doi: 10.1109/ECACE.2019.8679426.
- [12] D. ALAM, M. ALAMGIR, and T. FARAH, "Exploring the SQL injection vulnerabilities of bd domain web applications," Oct. 2015, pp. 110–114. doi: 10.15224/978-1-63248-064-4-23.
- [13] M. Zaman, D. Alam, T. Bhuiyan, and T. Farah, "A Study of the Effects of Heartbleed Vulnerability in Bangladesh," 2017. [Online]. Available: <https://www.researchgate.net/publication/316990137>
- [14] D. Alam, T. Bhuiyan, M. A. Kabir, and T. Farah, "SQLi vulnerabilty in education sector websites of Bangladesh," in *2015 2nd International Conference on Information Security and Cyber Forensics, InfoSec 2015*, Mar. 2016, pp. 152–157. doi: 10.1109/InfoSec.2015.7435521.



- [15] T. Farah, D. Alam, M. A. Kabir, and T. Bhuiyan, “SQLi penetration testing of financial Web applications: Investigation of Bangladesh region,” in *2015 World Congress on Internet Security, WorldCIS 2015*, Dec. 2015, pp. 146–151. doi: 10.1109/WorldCIS.2015.7359432.
- [16] “Introduction to Acunetix.” <https://www.acunetix.com/support/docs/introduction/#:~:text=Acunetix%20is%20an%20automated%20web,scripting%20and%20other%20exploitable%20vulnerabilities> (accessed Oct. 28, 2021).
- [17] “Cross-Site Scripting (XSS) Explanation and Prevention.” <https://www.rapid7.com/fundamentals/cross-site-scripting/> (accessed Feb. 16, 2022).
- [18] “Introduction to Cryptographic Failures.” <https://www.softwaresecured.com/introduction-to-cryptographic-failures/> (accessed Jun. 13, 2022).
- [19] “A guide to preventing common security misconfigurations.” <https://resources.infosecinstitute.com/topic/guide-preventing-common-security-misconfigurations/> (accessed Oct. 24, 2022).

## Plagiarism report

Submission date: 22-Jan-2023 12:41PM (UTC-0500)

Submission ID: 1996984271

File name: 191-15-12772.pdf (774.67K)

Word count: 4507

Character count: 23701

### VULNERABILITY ASSESSMENT ON BANGLADESHI PUBLIC DOMAIN BASED ON OWASP TOP VULNERABILITIES

#### ORIGINALITY REPORT

24%

SIMILARITY INDEX

15%

INTERNET SOURCES

7%

PUBLICATIONS

16%

STUDENT PAPERS

#### PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	www.acunetix.com Internet Source	2%
3	www.seekdl.org Internet Source	2%
4	www.researchgate.net Internet Source	2%
5	Submitted to University of Glamorgan Student Paper	1%
6	Submitted to Liverpool John Moores University Student Paper	1%
7	Submitted to Purdue University Student Paper	1%
8	nozerobit.gitbook.io Internet Source	1%