Automated Testing: Testing Top 10 OWASP Vulnerabilities of Government Web Applications in Bangladesh

Azaz Ahamed
Computer Science & Engineering
Independent University, Bangladesh
2120637@iub.edu.bd

Nafiz Sadman
Silicon Orchard Ltd.
Bangladesh
nafiz@siliconorchard.com

Touseef Aziz Khan
Computer Science & Engineering
Independent University,
Bangladesh
2120638@iub.edu.bd

Mahfuz Ibne Hannan
Computer Science & Engineering
Independent University,
Bangladesh
2120635@iub.edu.bd

Farzana Sadia
Dept. of Software Engineering
Daffodil International University
sadia swe@diu.edu.bd

Mahady Hasan
Computer Science & Engineering
Independent University,
Bangladesh
mahady@iub.edu.bd

Presented By,

Azaz Ahamed
Co-author
Computer Science & Engineering
Independent University, Bangladesh
2120637@iub.edu.bd

***** Slide contains audio. Audio will play automatically on slide change.***

About Myself

Presenter, Co-author

Azaz Ahamed

- Senior Software Engineer at *Intelligent Machines Ltd*
- M. Sc. in Software Engineering from *Independent University, Bangladesh* 2021-Present
- B. Sc. in Computer Science from *Independent University, Bangladesh* 2015-2019

GitHub : github.com/ahamedzoha

Website : azazahamed.com

LinkedIn : linkedin.com/in/azazahamed

***** Slide contains audio. Audio will play automatically on slide change.***

Research Interests

- Software Engineering
- Project Management
- Software Quality Assurance and Testing
- Distributed Systems
- Web 3.0 / Blockchain
- Software Architecture
- Software Process Management

*** Slide contains audio. Audio will play automatically on slide change.*

Introduction

- Advancement of web technologies increased adoption
- Online web-based services are more convenient
- Government web applications contain sensitive citizen information
- Security prioritisation using automated vulnerability testing tools
- Selected sectors of Government web applications
- OWASP Top 10
- The tools *(Netsparker, BurpSuite and Zap)*

Slide contains audio. Audio will play automatically on slide change.

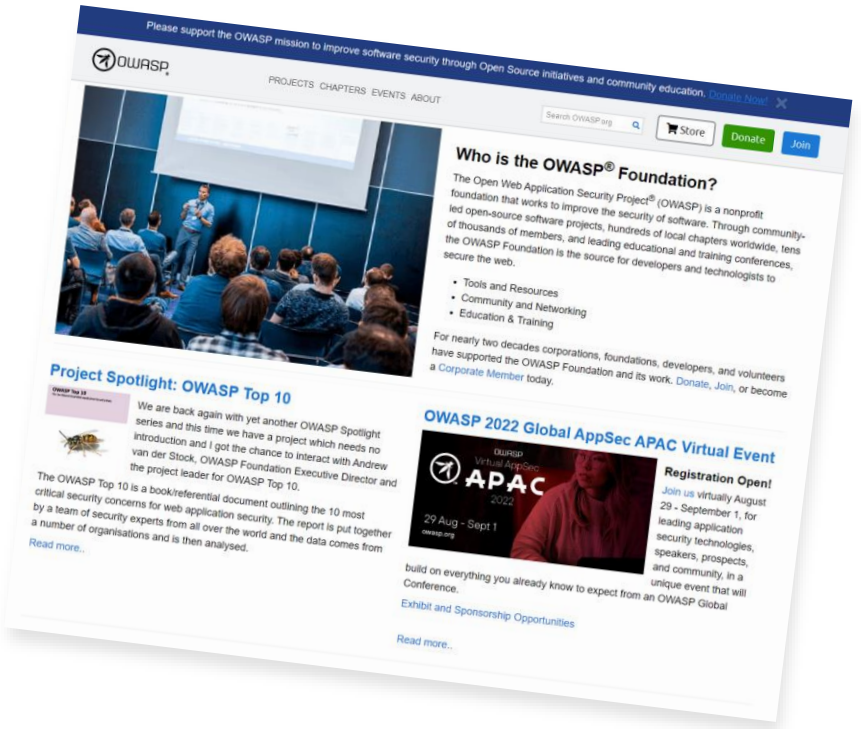
What We Intended

- Explore the three popular OWASP compliant testing tools
- Find vulnerabilities of live Bangladesh Government Web applications
- Understand the current status of Bangladesh's sensitive Government Websites
- Discover the threat detection and reporting capability of the tools used
- Hypothesize what the vulnerabilities tell us about these websites

***** Slide contains audio. Audio will play automatically on slide change.***

What is OWASP®

- A nonprofit, community-driven project
- Studies vulnerabilities in web applications
- Standardizes vulnerability identification, causes and mitigation



*** Slide contains audio. Audio will play automatically on slide change.*

OWASP Top 10 (2017)

Vulnerabilities	Description	Denotations
A1:2017	Injection	A1
A2:2017	Broken Authentication	A2
A3:2017	Sensitive Data Exposure	A3
A4:2017	XML External Entities (XXE)	A4
A5:2017	Broken Access Control	A5
A6:2017	Security Misconfiguration	A6
A7:2017	Cross-Site Scripting (XSS)	A7
A8:2017	Insecure Deserialization	A8
A9:2017	Using Components with Known Vulnerabilities	A9
A10:2017	Insufficient Logging & Monitoring	A10

OWASP®

*** Slide contains audio. Audio will play automatically on slide change.*

Research Methodology

Our research methodology workflow:

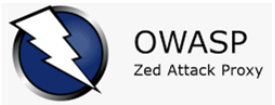
1. Tool Discovery
2. Target Application
3. Scanning
4. Reporting
5. Result Analysis

*** Slide contains audio. Audio will play automatically on slide change.*

Tool Discovery

Testing tools discovered

1. Netsparker
2. BurpSuite
3. Zap



F. Ö. Sönmez and B. G. Kiliç, "Holistic web application security visualization for multi-project and multi-phase dynamic application security test results," *IEEE Access*, vol. 9, pp. 25 858–25 884, 2021.

***** Slide contains audio. Audio will play automatically on slide change.***

Target Applications by Sector

Sectors of Government web application targeted

1. Services
2. Transportation
3. Welfare
4. Healthcare
5. Telecommunications

*** Slide contains audio. Audio will play automatically on slide change.*

Vulnerability Scanning

The following data was collected from every test run:

- Number of runs (number of test runs on the same Website using the same tool).
- Time taken to complete the test (in minutes).
- Counts of vulnerabilities found for severities: Low, Medium, and High.
- OWASP Vulnerability category (e.g. *A1:2017 Injection*, *A2:2017 Broken Authentication*)
- Total number of vulnerabilities.

***** Slide contains audio. Audio will play automatically on slide change.***

Reporting

We have classified severity based on RF (Risk Factor) scores in the following manner:

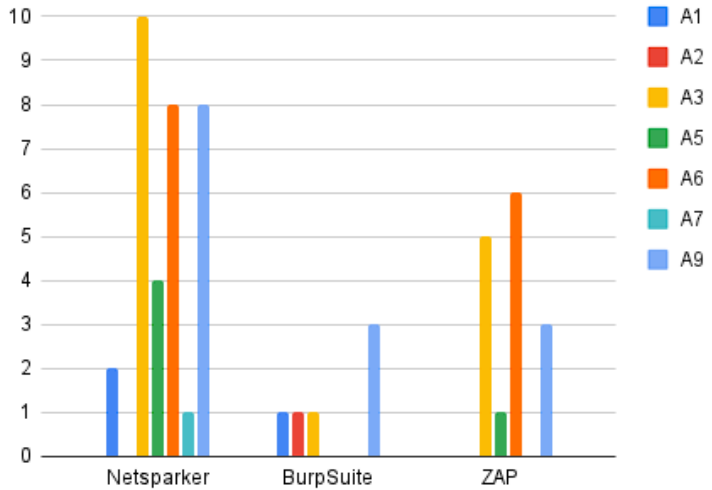
- **LOW** if 4 ≤ RF < 5
- **MEDIUM** if 5 ≤ RF < 7
- **HIGH** if RF ≥ 7

RISK	Threat Agents	Attack Vectors: Exploitability	Security Weakness: Prevalence	Security Weakness: Detectability	Impacts: Technical	Impacts: Business	Score
A1:2017-Injection	App Specific	EASY 3	COMMON 2	EASY 3	SEVERE 3	App Specific	8.0
A2:2017-Authentication	App Specific	EASY 3	COMMON 2	AVERAGE 2	SEVERE 3	App Specific	7.0
A3:2017-Sens. Data Exposure	App Specific	AVERAGE 2	WIDESPREAD 3	AVERAGE 2	SEVERE 3	App Specific	7.0
A4:2017-XML External Entity (XXE)	App Specific	AVERAGE 2	COMMON 2	EASY 3	SEVERE 3	App Specific	7.0
A5:2017-Broken Access Control	App Specific	AVERAGE 2	COMMON 2	AVERAGE 2	SEVERE 3	App Specific	6.0
A6:2017-Security Misconfiguration	App Specific	EASY 3	WIDESPREAD 3	EASY 3	MODERATE 2	App Specific	6.0
A7:2017-Cross-Site Scripting (XSS)	App Specific	EASY 3	WIDESPREAD 3	EASY 3	MODERATE 2	App Specific	6.0
A8:2017-Insecure Deserialization	App Specific	DIFFICULT 1	COMMON 2	AVERAGE 2	SEVERE 3	App Specific	5.0
A9:2017-Vulnerable Components	App Specific	AVERAGE 2	WIDESPREAD 3	AVERAGE 2	MODERATE 2	App Specific	4.7
A10:2017-Insufficient Logging&Monitoring	App Specific	AVERAGE 2	WIDESPREAD 3	DIFFICULT 1	MODERATE 2	App Specific	4.0

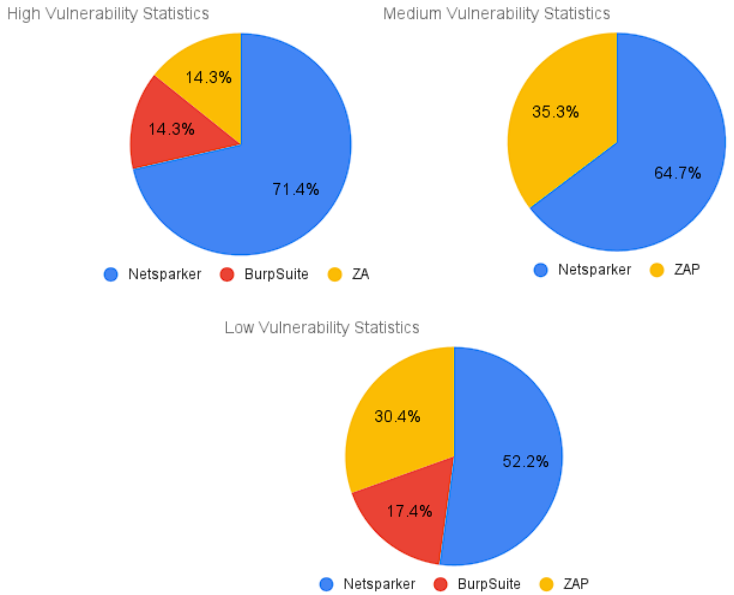
https://owasp.org/www-project-top-ten/2017/Details_About_Risk_Factors

*** Slide contains audio. Audio will play automatically on slide change.*

Result Analysis



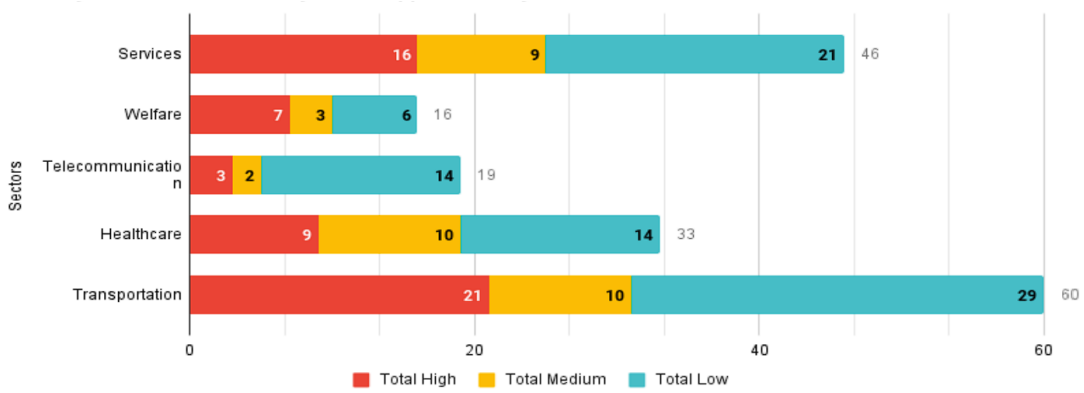
*Vulnerabilities **count** across all tested web applications from the tools (Netsparker, BurpSuite, Zap)*



*Vulnerabilities by **severity** reported by testing tools (Netsparker, BurpSuite, Zap)*

*** Slide contains audio. Audio will play automatically on slide change.*

Result Analysis



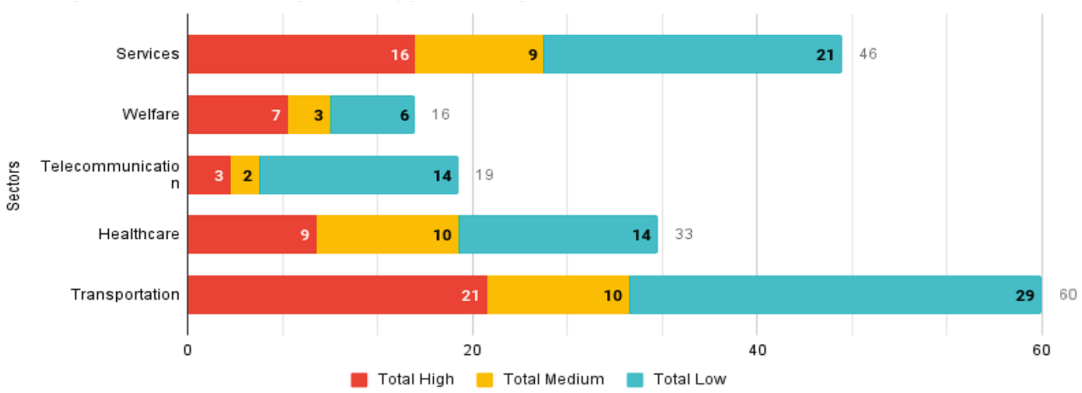
Severity** of OWASP vulnerability and **count** found in web applications by **sector

Vulnerabilities	Description	Denotations
A1:2017	Injection	A1
A2:2017	Broken Authentication	A2
A3:2017	Sensitive Data Exposure	A3
A4:2017	XML External Entities (XXE)	A4
A5:2017	Broken Access Control	A5
A6:2017	Security Misconfiguration	A6
A7:2017	Cross-Site Scripting (XSS)	A7
A8:2017	Insecure Deserialization	A8
A9:2017	Using Components with Known Vulnerabilities	A9
A10:2017	Insufficient Logging & Monitoring	A10

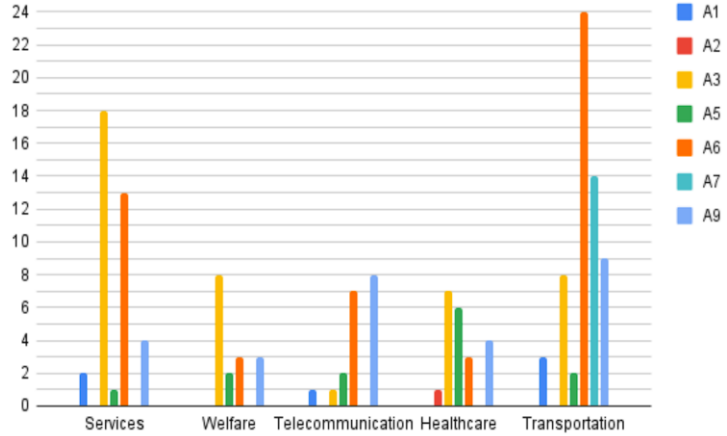
OWASP Top 10 vulnerabilities classes

*** Slide contains audio. Audio will play automatically on slide change.*

Result Analysis



Severity** of OWASP vulnerability and **count** found in web applications by **sector



***OWASP Top 10 vulnerabilities count** found across web applications from individual sectors*

*** Slide contains audio. Audio will play automatically on slide change.*

Final Thoughts

Government Web applications in Bangladesh suffer from important security oversights.

Most of the vulnerabilities arise from common software development pitfalls such as:

- Not writing maintainable code
- Not writing reusable code
- Not writing unit or integration tests
- Not maintaining up-to-date documentation of the project
- Not updating Software packages used in software development

Lack of maturity of technology in Bangladesh

*** Slide contains audio. Audio will play automatically on slide change.*

Thank You

We appreciate your valuable time

For any questions please contact,

mahady@iub.edu.bd