iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** V **Month of publication:** May 2021

DOI: https://doi.org/10.22214/ijraset.2021.34087

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

Bridging the Gap between Web Application Firewall and Web Applications

Salman Khan[1], Mohd Faisal Jamal[2], Mohd Ethesham Khan[3], Prof Sheena Mohammed[4]
[1, 2, 3]*UG Scholar, Dept of IT, ISL Engineering College, Bandlaguda, Hyderabad, Telangana, India*
[4]*HOD, Dept of IT, ISL Engineering College, Bandlaguda, Hyderabad, Telangana, India.*

Abstract: ***Web Application Firewalls (WAFs) are passed on to guarantee web applications and they offer all around security as long as they are planned viably. An issue rises when there is over-reliance on these gadgets. A misinformed impression that everything is great and great can be gained with the utilization of a WAF. In this paper, we give a chart of development filtering models and a couple suggestions to benefit the upside of web application firewall. To provide security for those applications we have to create separate software for network services users if they had upload .doc correct data then only they can access & convert to .pdf that file. If in that file any attacker key and web App Firewalls in document file that files are not allowed to convert to pdf file. For the security purpose that information will check to all network services. After that if the users recover the information form firewalls then only convert to pdf file. In our network services create different types of network services and everyone the information will passes peer to peer.***
Keywords: Web Application Firewall, shared data repository, static verification, run-time enforcement.

I. INTRODUCTION

Nowadays web systems are widespread and many companies are increasingly adding commercials to their business model to increase revenue. But web applications are often flawed, and these bugs are acceptable to attackers because of their high accessibility and potential profitability. Permission to make digital or hard copies of all or part of the work for personal or classroom use is provided free of charge unless copies are made or distributed for profit or commercial gain and that copies contain this notice and the full quotation on the front page. Alternative copying, republishing, sending to servers or redistributing lists, requires prior authorization and or funding.

There are a variety of countermeasures available and recently installed Web Application Firewalls (WAFs) added to the network infrastructure to combat traditional network failures. WAFs may, among other things, prevent the risk of broken access control such as a disability that results in forced browsing forcing the flow of a strict application. One of the problems with using WAFs in the use of a robust application is the fact that they tend to have a loose connection between their configuration and application implementation. That way, they can protect applications from common attacks, but there is no direct relationship with the bugs that live in the system they want to protect you from. Therefore, there is no guarantee that WAF's compulsory policy on incoming applications protects system-specific start-up bugs.

A major contribution of this paper is that it demonstrates that, with a combination of firm and powerful authentication, WAFs can officially verify the absence of certain types of misconduct in web applications. We have modeled the implementation of our approach to building an existing Java authentication tool, and have implemented our approach to J2EE-based web applications. In particular, we ensure that if the combination of web application and WAF policy exceeds our verification process, no customer / server communication will violate data dependence on shared session status between server third party items.

A. Proposed Methodology

Here we have to create any online account you have to register .while registering time for that particular user that user registration will be sending to database that data will stored in database. Then only they can login. After login admin will monitoring network services. Network service having different type of services and different users.

To provide security for those applications we have to create separate software for network services users if they had upload .doc correct data then only they can access & convert to .pdf that file. If in that file any attacker key and web App Firewalls in document file that files are not allowed to convert to pdf file.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

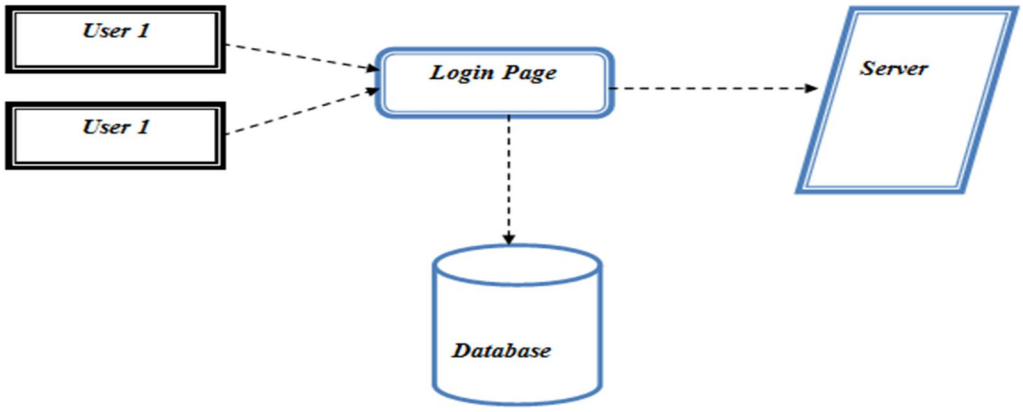


Fig 1. Proposed System

II. BACKGROUND

A. *Web Applications*

Web applications are server-side applications requested by small web clients (browsers), usually using Hypertext Transport Protocol (HTTP). The user can navigate through the web application by clicking the links or URLs in his browser, and is also able to set input parameters by filling out web forms. URLs in the client residential system generated by input parameters provided by the user. The result of the application (usually expressed in Hypertext Markup Language (HTML)) and then returned to the browser where you are given a continuous user communication.

B. *Servlet-based web Applications*

Java Servlet technology is part of J2EE specification and provides ways to extend web server performance and access to existing business applications. The J2EE web application is usually a collection of Java Servlets, which are distributed on a server-based web site such as Tomcat, JBoss or WebSphere. Java Servlets are functional web tier units and additional functions such as load balancing and security are added to the web server rather than to the apps themselves.

C. *Web Vulnerabilities and Web Application Firewalls (WAFs)*

Existing network security fails to effectively protect web applications from attackers. Network firefighters such as packaged filter filters usually operate on a network or transport layer (e.g. to provide access to a complete web application by enabling TCP port 80 traffic), and web applications are often attacked in the application layer. For example, attackers use design errors between application concepts and known vulnerabilities in HTTP protocol, browser or web server technology. Therefore, a network firewall only talks about network access control to control whether a web server can be accessed or not, regardless of the type of web requests and associated data sent to the server.

III. PROTOTYPE IMPLEMENTATION

In this section, we describe our model work and discuss how it can be used to protect Duke's Bookstore application.

A. *Server-side Specification and Verification*

In order to use existing verifiers to check if the implementation of a component adheres to its contract, the problem specific contracts are translated into the Java Modeling Language (JML) [18] which is a popular formal contract specification language for components written in Java.

The JML contract in listing 4 expresses interactions between actions and the shared data repository in terms of pre- and post-state of the repository. For read interactions, the component's contract indicates that the component requires that a non-null data item of the specified type can be read from the shared repository. For write interactions, this ensures pragma states which data items on the shared repository will be non-null and of the specified type after method execution. In Listing 4 for example, the JML contract of the do Get method of the Show Cart Servlet states that among others the shared data item messages will be a non-null Resource Bundle object before execution and after execution that the data item cart is ensured to be a non-null Shopping Cart. In addition, this modifies clause expresses the frame condition, i.e. what part of the session state a method is allowed to modify.

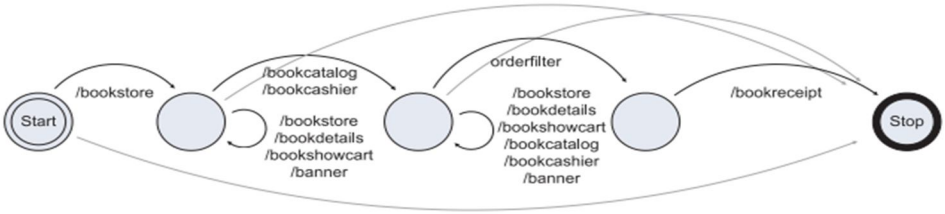


Fig 2. Prototype

B. Run-time Protocol Enforcement

As proof of concept, we installed a lightweight WAF in our web application container by installing the J2EE filter. Before a servlet is requested to use the application (Servlet Request, Servlet Response response) in the J2EE web application, a series of filters used are always applied to the application. At the time of distribution, our enforcement engine was loaded with object-based proofing of a labeled transformation program (Figure 4). In each user's case the current status is maintained, and in each incoming web application, the enforcement engine ensures that the change is approved and the current status is updated before the request is sent to the servlet. In the event of a violation, the connection strategy is considered, defining the action to be taken from restricting access to the developer's IP or disabling the user session to incorporate access infringement.

IV. OUTPUT SCREEN

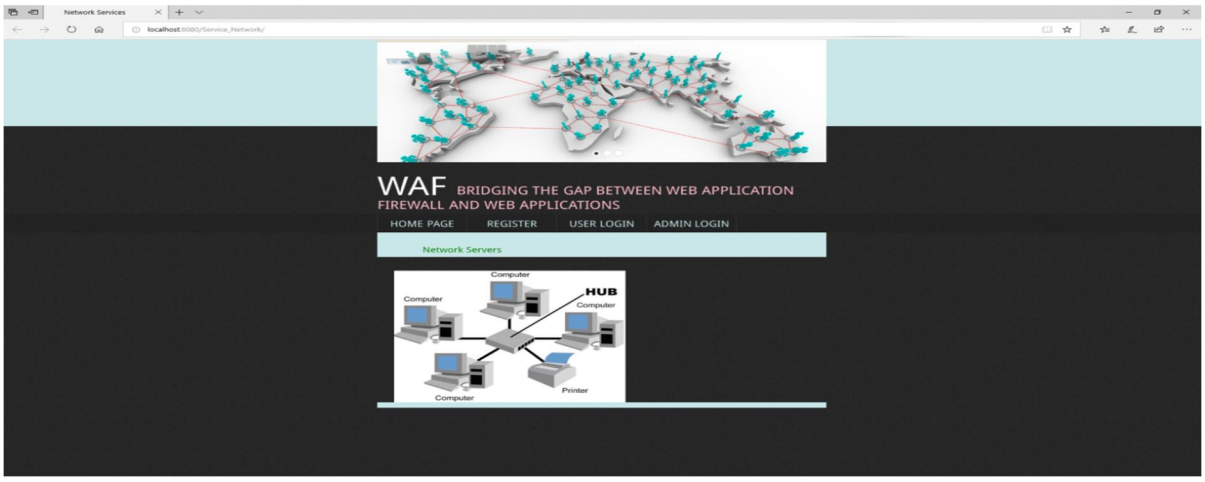


Fig 3. Home Page

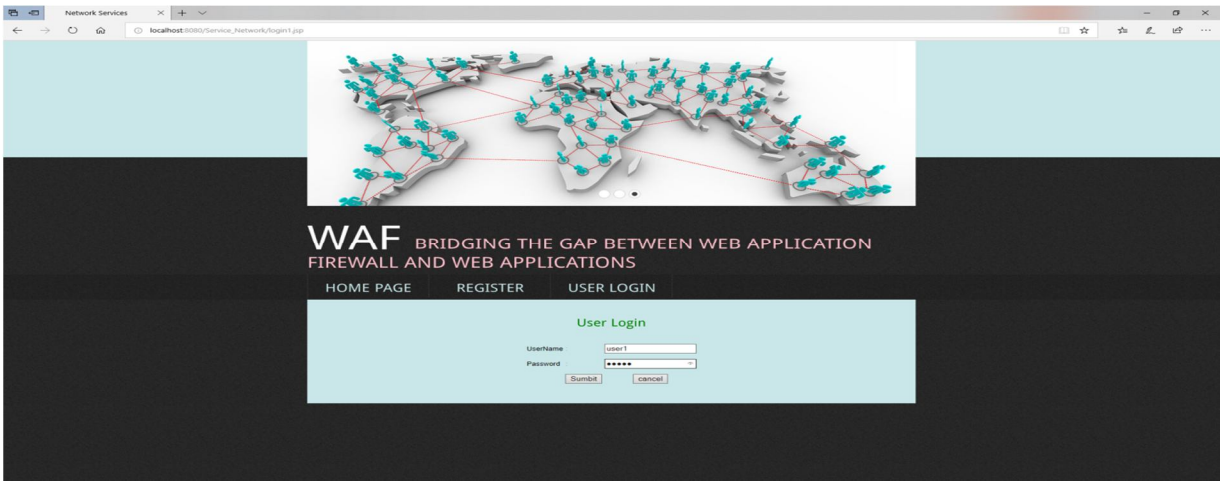


Fig 4. Login page

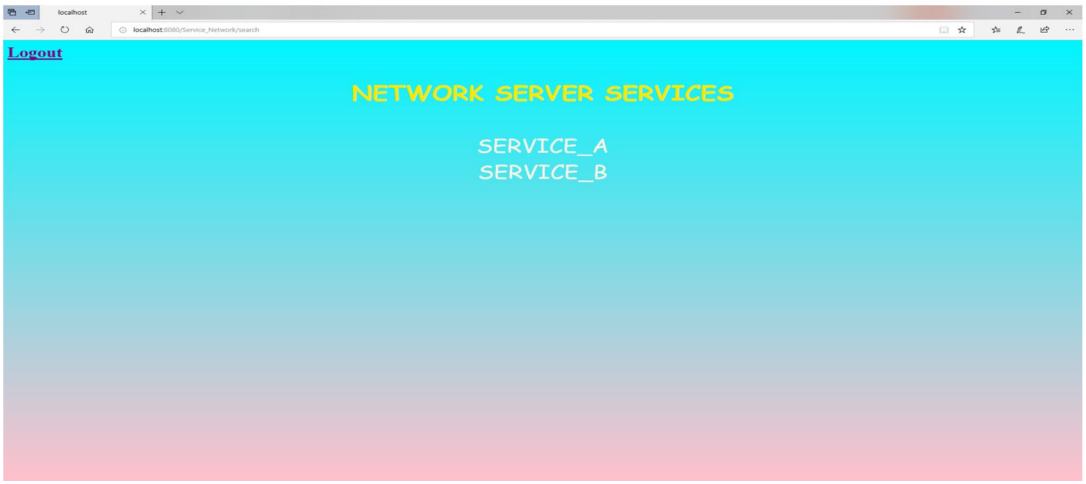


Fig 5. Network service page

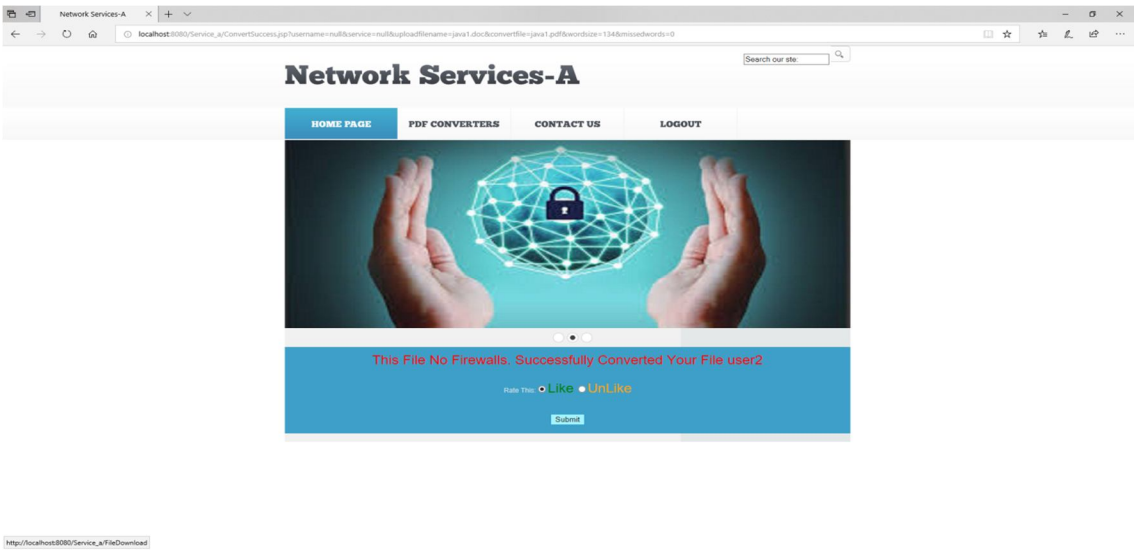


Fig 6. Effective File Converted Without Harmful Data

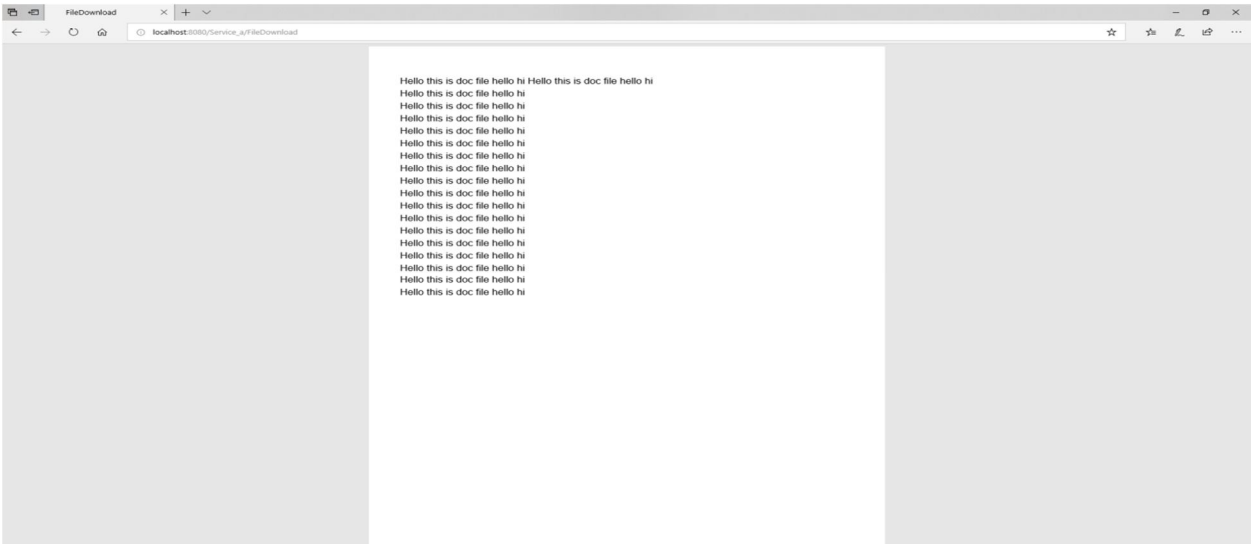


Fig 7. Successful Converted Pdf File

V. CONCLUSION

This paper focuses on closing the gap between WAFs that force the flow of solid application, as well as other bugs related to the implementation of this type of firewalls that try to protect themselves. We have shown that with a combination of static and dynamic verification, WAFs can officially verify the absence of certain types of misconduct in web applications. In particular, we have ensured that if a combination of web application and WAF policy exceeds our verification process, no customer / server communication will violate data dependence on shared session status between server third party items. Although there are still some limitations with our proposed solution (as discussed in section 6.2), the tests performed show that using the available authentication tools to improve the security of the web application looks promising.

VI. FUTURE WORK

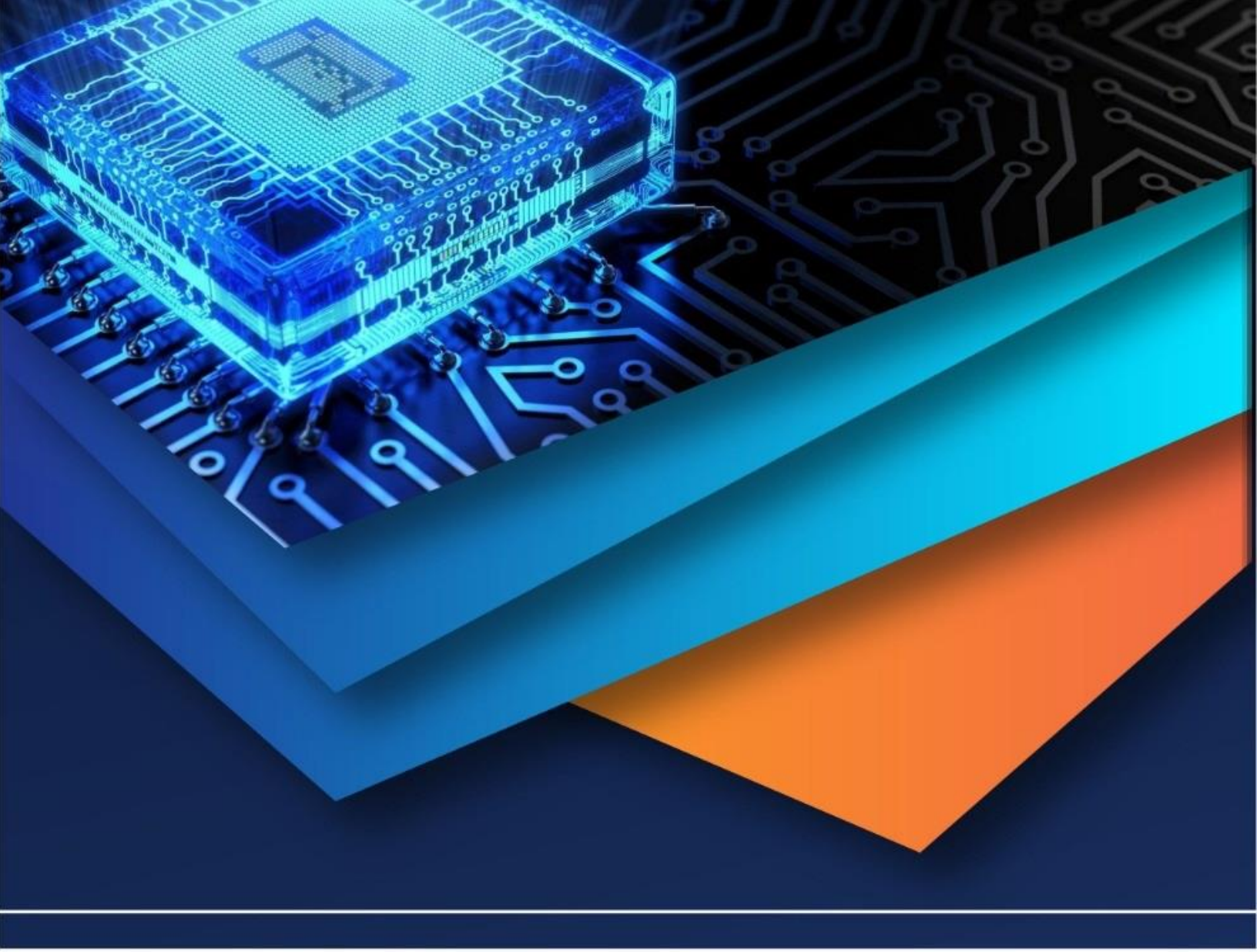
Furthermore, Configuration of Web Application Firewalls provides result auto correction to automatically correct compromised results to improve the result quality. We have implemented Firewalls and tested it on a commercial data stream processing platform running inside a production virtualized network security infrastructure. Our experimental results show that Firewalls rules model can achieve higher pinpointing accuracy than existing alternative schemes. WAF is light-weight, which imposes low performance impact to the data processing services running inside the network security service infrastructure.

VII. ACKNOWLEDGEMENTS

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all those who have helped to make this work a great success.
Thanks to Mrs. Sheena Mohammed is the Head of the Department of Information Technology, for her moral support throughout our project.
Thanks to Mr. Dr. Prem Chander for his important suggestions and the guidance he gave us during this project.
Thanks to Mrs. Hadiya Sameen for his important suggestions and the guidance he gave us during this project.
We would like to thank the teaching and non-teaching staff of IT Department for sharing their knowledge.

REFERENCES

[1] Tekerek, C. Gemsy, O. Bay, "The Development of a Hybrid Web Application Firewall to Prevent Web Based Attacks," Proceedings Of 8th IEEE International Conference on Application of Information and Communication Technologies (AICT), Oct 2014.
[2] A. Razzaq, A. Hoor, S. Shahbaz, M. Masood, "Critical Analysis on the Web Application Firewall Solutions," Proceedings of IEEE Eleventh International Symposium on Decentralized Systems (ISADS), March 2013.
[3] J. Beechey, Web Application Firewalls: Defense in Depth for Your Web Infrastructure, 2009, Taken from https://www.sans.edu/student-files/projects/200904_01.doc
[4] M.Gendron, Introduces First Network Adaptive Web Application Firewall. 2006, taken from http://investors.imperva.com/phoenix.zhtml?c=247116&p=irol -newsArticle&ID=1596618
[5] M. Heiderich. E. Nava, D. Lindsay, Web application obfuscation, Elsevier, 2011, taken from https://doc.lagout.org/security/Web%20Application%20Obfuscation/Web%20Application%20Obfuscation.pdf



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY