

A Web Server Application for Intrusion Detection and Access Control Using Cronjob Scheduling Algorithm

Paul Ndudiri Ohia, Prof. P.O. Asagba

1, 2, Department of Computer Science, University of Port Harcourt Choba, Nigeria

ABSTRACT

Webserver application environment has become an attractive target for hackers due to high increase on deployment of web applications and adoption of cloud computing technologies. A webserver is a gateway through which the entire internet population connects to with the aim of sharing resources or providing online services. Webserver based applications are becoming the most common way to deliver and access online services, this trend also gained a higher increase especially during the covid-19 era. There is a high increase of intrusion on the cyber space since the web gained more concentration among others, the sophistication of attacks against these applications environment has grown even as technology evolves. To prevent common attacks of content corruption and data theft, integrity and privacy has become an increasing problem. While approaches to security have been improved upon in recent times, intrusion techniques have also become increasingly complex. In this research work an intrusion prevention and detection on webserver application environment using cronjob scheduling techniques has been developed. The new system adopted object oriented analysis and design methodology and was implemented using PHP7 server side scripting programming language, Laravel MVC framework and MYSQL relational database for the backend. The system has been developed, tested and was able detect and prevent intruders from gaining malicious access into the webserver application environment and also was able to send quick response message to the admin through sms messaging and emailing system. This quickly prompts the admin of the attempted intrusion and preventive measures are then taken to safeguard webserver. Hence this work will be beneficial to web developers and webserver administrators.

Keywords:- Web Server, Intrusion, Detection, Scheduling, Algorithm, Distributed Computing

1. INTRODUCTION

Since the arrival of cloud computing and other internet enabled technologies for the delivering of services globally, webserver applications environment is now used more than ever before. presently, we place more interest on web technology for everything we do from social contact to

the rendering of key services, this process now allows malevolent actors to invade into people's privacy, this now create risk in the internet space, a phenomenon known as cyber assaults. (Samuel et al., 2022). Security talks about the confidentiality, integrity, and availability of systems and while data confidentiality talks about the ability to ensure that information is private to only authorized parties which must be guided from unauthorized parties to have access to, integrity reflects the accuracy of information which necessitates technology and processes that prevent unauthorized parties from inappropriately modifying information and availability refers to the ability to ensure that information is available to authorized parties and protected from unauthorized users and availability refers to the ability to ensure that information is available to authorized parties and protected from unauthorized users also (Samuel et al 2022).

Since the Internet became an open system, web server applications environments have been increasingly popular for delivering important services on a global scale, among other things. In a study conducted by (Samuel et al.,

2022), it was discovered that webserver attacks cost enterprises more than 100 times more than malware, and 50 times more than viruses, worms, and trojans each year. Several intrusion detection systems, among others, have recently been created and assessed.

A webserver is a computer programme that fetches and serves web pages that are requested by a client. Web pages can be saved on the host computer's hard disc or on the hard drive of another computer.

The main purpose of network security and access control is to keep intruders at bay. However, the purpose of a Web site is to give the rest of the world regulated access to the network. The Apache web server is the most extensively used web server, with over 40 million web applications running on it, powering approximately 70% of all websites..(Victoria et al 2021)

The use of Internet services and webserver-hosted applications has increased rapidly across the world. However, this growth has also made webserver environments major targets for attackers who exploit weaknesses to gain access to sensitive information. Studies show that many cyber-attacks target web applications, making intrusion detection an important but still developing area of webserver security (Valeur et al., 2021). Webserver-based applications such as internet banking, online trading, blogs, and social networking platforms continue to expand, increasing the complexity

and security risks of webserver environments. Therefore, web applications should be carefully designed, tested, and supported with intrusion detection systems that can identify attacks and provide early warnings of suspicious activities (Valeur et al., 2021; Rajiv et al., 2022).

1.1 Overview of the Webserver

An Apache webserver is server-based software that receives requests from a client browser and delivers the requested web pages or resources. These web pages may be stored on the host computer or on another connected system (Abomhara et al., 2022). A webserver can also support web applications, storage, FTP, email, and database-driven services. A webserver responds to client requests either by sending the file linked to the requested URL or by generating a response through scripts that communicate with a database. In this process, access control is important because it determines which users, devices, or roles are allowed to view, modify, or execute specific web resources. Proper access control helps prevent unauthorized users from reaching protected pages, scripts, folders, and administrative functions (Aulds, 2020).

Webserver application security is a continuous risk management activity involving technology, policies, procedures, legal enforcement, and user awareness. It protects privacy, integrity, availability, authentication, and authorization within the webserver environment. A poorly configured webserver can expose sensitive resources, even when a firewall is present, especially if access rules are weak or protected directories are left open (Zhang and Yuan, 2022).

In an intranet environment, security becomes more complex because different users may require different access privileges. Apache remains one of the most widely used webservers, and understanding its security and access-control mechanisms can help improve the protection of web-based systems (Cova et al., 2021).

1.2 Types of Web Server

Different web servers are used to host and deliver web applications, each with its own features, supported platforms, performance level, and security controls. Understanding the major types of web servers is important because the choice of server can affect application availability, access control, intrusion detection, and overall web security.

1.2.1 Apache HTTP Server

The Apache Software Foundation has produced the world's most popular web server. The Apache web server is free software that may be installed on a wide range of operating systems, including Linux, Unix, Windows, FreeBSD, Mac OS X, and others. The Apache Web Server is used by about 60% of web server machines.

1.2.2 Internet Information Services

Microsoft's Internet Information Server (IIS) is a high-performance Web server. This web server is compatible with Windows NT/2000/2003 (and may be on upcoming new Windows version also). Because IIS is intimately connected with the operating system, it is relatively easy to administer. It is packaged with Windows NT/2000 and 2003.

1.2.3 lighttpd

Lighttpd, pronounced lightly, is another free web server included with the FreeBSD operating system. This open source web server is quick, safe, and uses very little CPU power. Lighttpd is also compatible with Windows, Mac OS X, Linux, and Solaris.

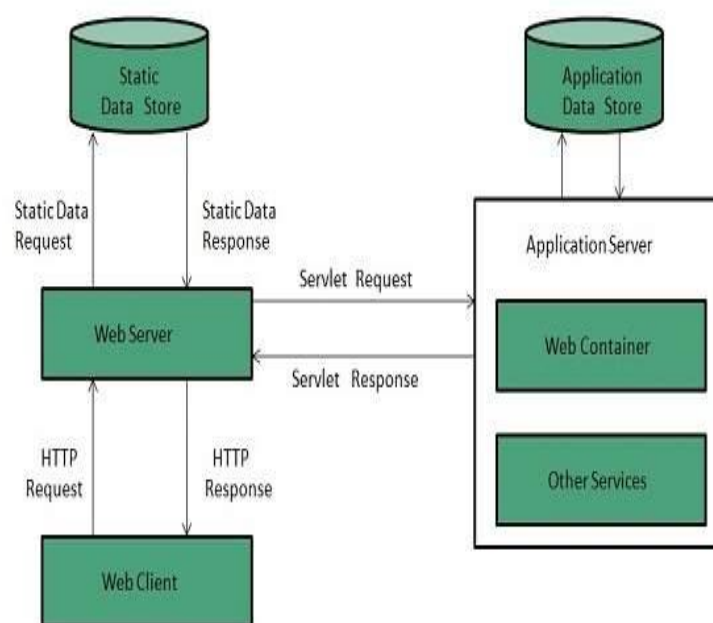


Figure 1: Web Server Architecture (Source: http://www.tutorialspoint.com/internet_technologies/web_servers.htm accessed 12-04-2021)

2. Web Servers Security Issues

For both Linux and FreeBSD, Apache is the most famous Web server. In view of its rules consistence, flexibility, dynamic shared articles, versatility, and programmability, it is the most

widely utilized Web Server on the Internet. (Rajiv et al. 2012). Web servers are an enticing objective for programmers, which is the reason security, is a particularly significant theme for administrators of both web associated and intranet-associated workers. The overall security difficulties of a web server are examined in this segment.

2.1 Integrity and Privacy

A web server is our computer's mother board, upon which other components are built, and it serves as a portal through which the entire internet population can peer in. Integrity and privacy are the major concerns in preventing common assaults of content tampering and data theft.

2.2 Common Gateway Interface (CGI) Script

CGI scripts are constant projects that work on Web servers. They deal with an assortment of client inputs and, thus, get input from internet browsers, access data sets, and return data to the customer program. CGI scripts are like little workers. Accordingly, a defective content could be an objective for an assault. In two ways, CGI projects can uncover security defects: They could show data from the host that could help assailants in breaking into the worker. The client sources of info could be adequately troublesome to establish executable orders, causing the host machine to endure hurt. (Benson et al. 2022).

2.3 Web-server Access Control

Access control in a web server environment refers to the process of determining who is allowed to access specific resources, pages, directories, files, scripts, or administrative functions on the server. It is not enough for a web server to simply receive and respond to user requests; the server must also verify whether the user, system, or application making the request has the proper permission to perform the requested action.

In the case of CGI scripts, access control is especially important because CGI scripts can execute programs on the web server and interact with files, databases, and system resources. If unauthorized users are allowed to execute poorly protected CGI scripts, they may gain access to sensitive information, modify server content, run malicious commands, or exploit weaknesses in the application. Therefore, access control helps ensure that only authenticated and authorized users can view, modify, or execute protected web server resources.

Effective web server access control can be implemented through user authentication, role-based permissions, IP

address restrictions, directory protection, file permission settings, and server configuration rules. For example, an administrator may allow public users to view ordinary web pages while restricting administrative pages, log files, upload folders, and CGI scripts to trusted users only. This reduces the risk of unauthorized access and supports intrusion detection by making suspicious access attempts easier to identify, record, and block.

2.4 Transmission of Data via TCP/IP

Security was not a priority when the TCP/IP protocol was created. Therefore, right when characterized chronicles are moved from the Web laborer to the program, or when the end-customer conveys private information back to the specialist through a wrap up structure, it is weak against network snooping. (Mohammed et al 2021)

3. Types of IDS Attacks

3.1 Phishing

Phishing is the act of sending messages that tends to come from solid sources in deals to get individual information or persuade customers to take action. It is a blend of social orchestrating and movement confusion. It might be a hurtful relationship with an email that taints your machine. It may really be a relationship with a dangerous site needed to trick you into presenting malware or uncovering fragile data. It expected to deceive you into introducing malware or uncovering delicate data (Enoch 2020).

3.2 Drive-by Attack

Drive-by download assaults are a typical way for malware to spread. Designers seek after frail objections and supplement a harmful substance into constrained by the designer so the HTTP or PHP code on one of the pages. This substance may obviously introduce malware on the PC of a guest to the site, or it might divert the difficulty to a site. (Gupta et al. 2022)

3.3 Password Assault

Getting passwords is a far reaching and astonishing assault approach since passwords are the most commonly utilized structure for affirming clients to a data system. Looking at a singular's workspace, "sniffing" the association relationship for decoded passwords, using social planning, acquiring induction to a mysterious key informational collection, or straight guessing are in general ways to deal with acquire permission to a singular's mysterious key. This may incite the usage of force. (Cova et al., 2021)

3.4 SQL Injection Assault

With data set driven sites, SQL infusion has turned into an inescapable issue. It happens when an evildoer utilizes the info information from the customer to worker to run a SQL inquiry on the data set. The SQL Injection assault targets Web pages that permit clients to type text into structure handles that are then used to inquiry information bases then, at that point, used to question data sets. Programmers can enter a satire SQL inquiry, which modifies the question's inclination. Thus, the questions can be utilized to get to the connected data set and adjust or erase its substance. (Kieyzun 2022)

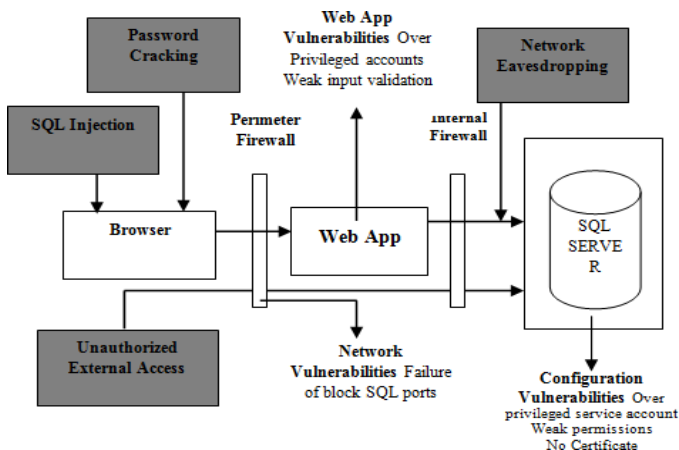


Figure 2.2: SQL Injection Attack

(Source: *A Complete Strategy for Web Application Security*,
(Humayun et al. 2020))

4. Intrusion Detection System

According to (Taha et al. 2021), an intrusion detection system (IDS) is a device or software programme that monitors network or system operations and detects any hostile activity. There are various parts to IDS:

Sensors that generate security events

A console to display events and alarms as well as control the sensors.

A central engine that stores the events logged by the sensors in a database and utilizes a set of rules to generate alarms based on the events.

IDS can be classed as Host-based Intrusion Detection System or Network-based Intrusion Detection System, depending on the stated objective or information source.

4.1 Host-based Intrusion Detection System

The earliest type of intrusion detection programming was built for the host-based intrusion detection system, with the original target system being the mainframe PC, where outside cooperation was uncommon.

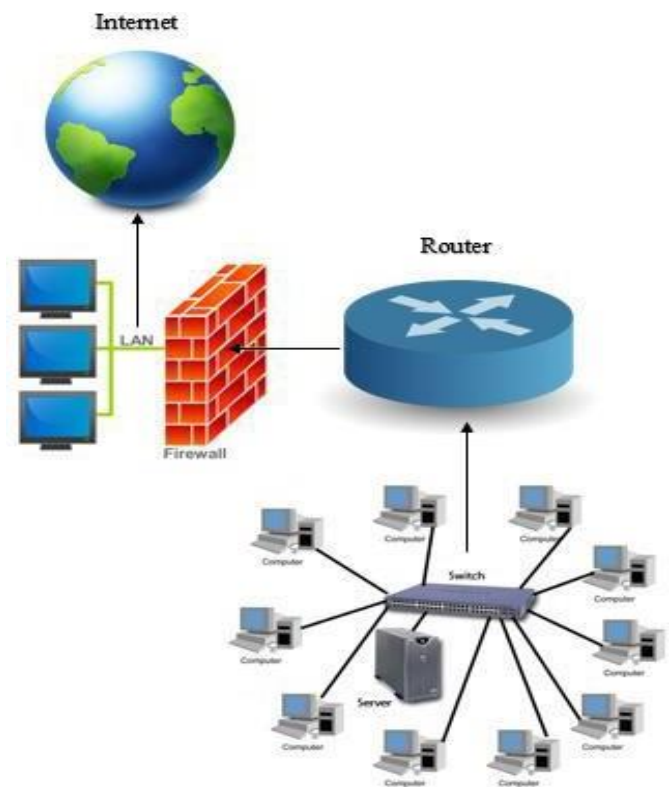


Figure 3: Host-Based Intrusion Detection System

5. Methodology

A software development methodology or system design methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system. The methodology will be adopted in the analysis and design in this study is Dynamic Systems Development Model (DSDM) Methodology. The Dynamic Systems Development Model was developed in the U.K in the mid-1990. It is the evolution of rapid application development (RAD) practices. DSDM boasts the best-supported training and documentation of any of the agile software development techniques.

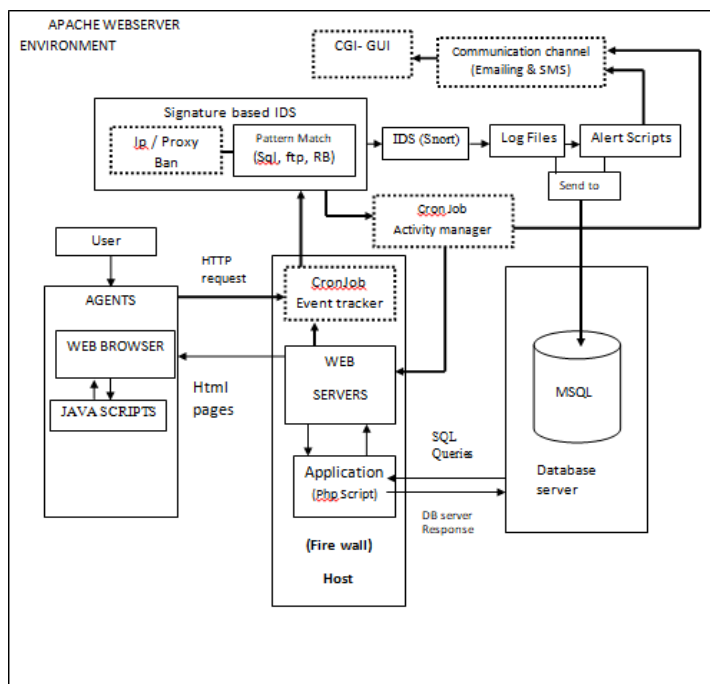


Figure 4: Architecture of the Proposed System

The Cronjob scheduling technique: is used to track every event that occurs on the server in order to trigger the IDS system to start working with the goal of keeping the webservice stable in order to improve resource availability and minimize quota utilization as seen in the previous system. The cronjob event tracker receives all the https requests, analyses them, and then passes them to the

IDS scripts for further analysis. All actions in the system are managed by the cronjob activity manager.

Signature-based method: compares given network traffic and log data to existing attack patterns in order to identify possible known attacks.

The ip / proxy ban component: was added to existing IDS scripts to provide an intrusion prevention mechanism on the server by capturing the user's IP and determining if the request is a proxy (bot request), etc. If a proxy is discovered at that stage, the request will be denied and the IP will be banned from further accessing the system.

The CGI graphical user interface: is a key component that bridges the gap between the log file and the web server administrator, given how time-consuming viewing and reading log files on the server may be. It presents the data entered into the database in a more comprehensive and visually appealing format for analysis and decision-making.

Communication channel components: deliver instant alerts to the webserver administrator through email and SMS, calling for preemptive action to be taken if necessary.

6. Results and Discussion



Figure 5: Home Page Design Layout

Figure 5 shows the proposed system landing page design interface which is usually launched immediately the program runs or initiated.

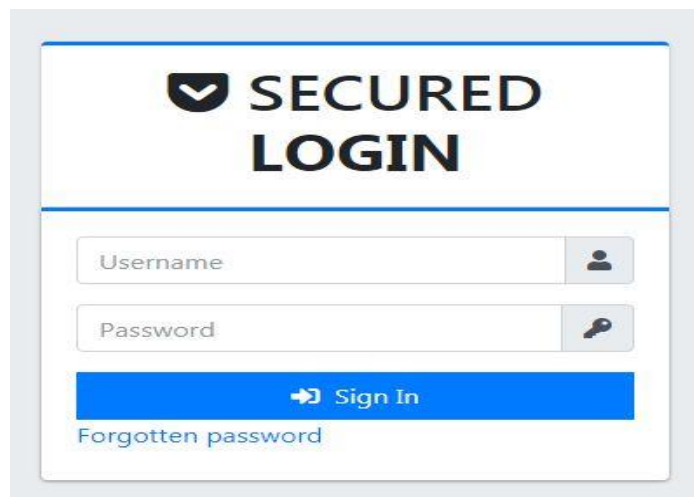


Figure 6: Login Page Design Interface

Figure 6 shows the user login design interface through which valid users can gain access to the system.

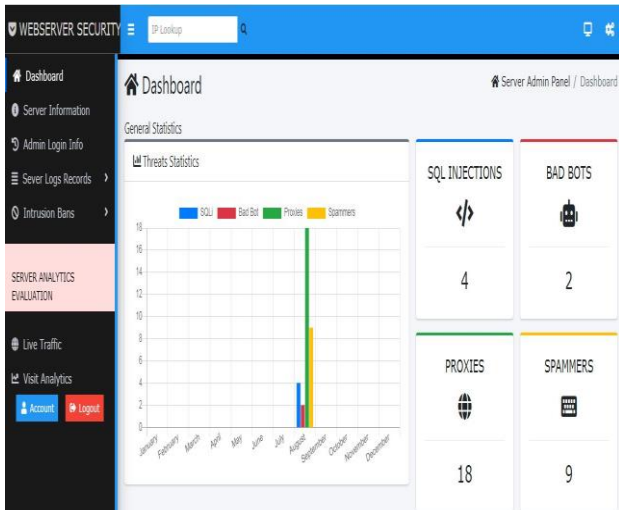


Figure 7: User Dashboard Interface

Figure 7 shows the user dashboard interface through which user can navigate to other modules; it also shows some historical data on it for a quick preview.

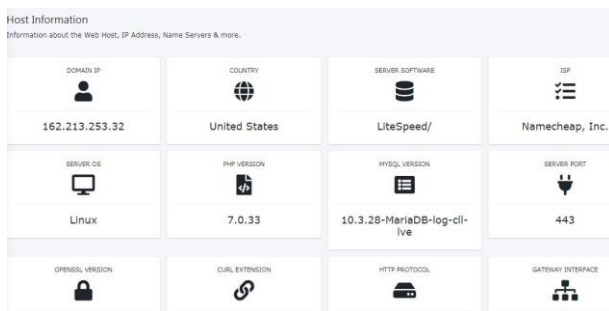


Figure 8: Host Information GUI

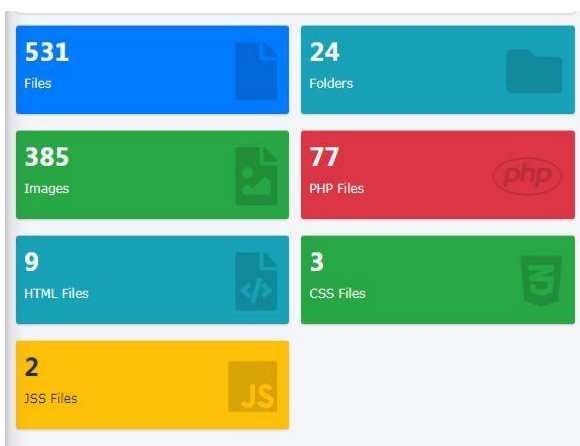


Figure 9: Server Storage Quota GUI

Figure 9 shows the server storage quota, hence it contained the server storage space and usage statistics such as available storage space and used storage space.

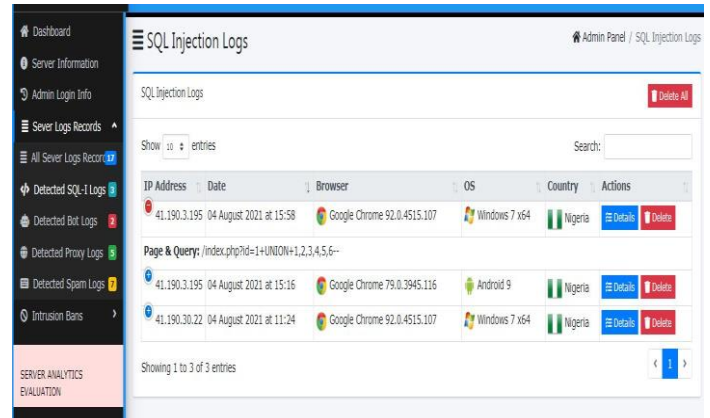


Figure 10: Detected Logs records

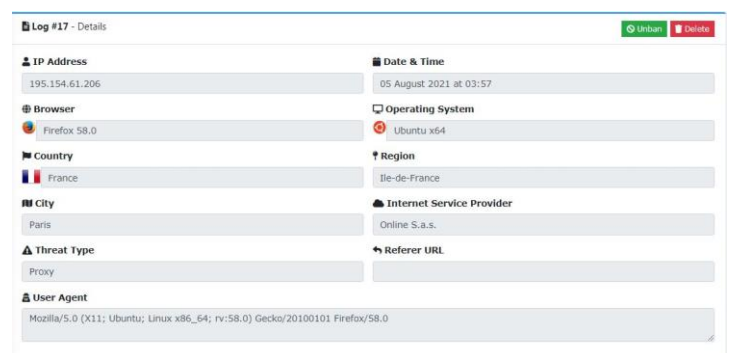


Figure 11. Intrusion Information

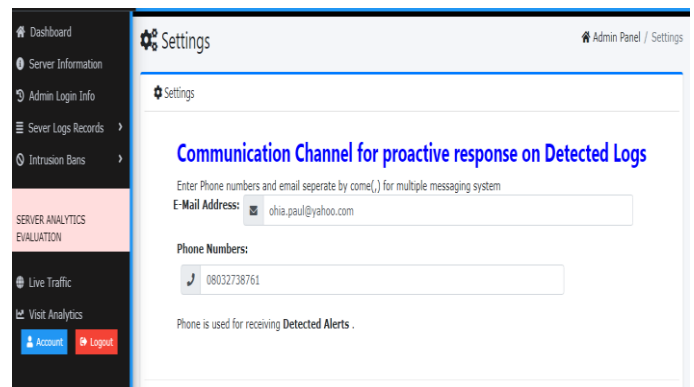


Figure 12: Communication Channel

6.1 Results Discussion

Webserver intrusion detection and prevention remains an important area of interest among programmers and researchers because of the rapid adoption of cloud computing and new web technologies. As more applications are hosted online, attackers continue to use different tools and techniques to gain unauthorized access to webserver environments. Even when a web application is properly designed, the entire system can still be compromised if the webserver environment and its access-control mechanisms are weak.

The result of this study is an improved webserver application intrusion detection and prevention system that uses a cronjob scheduling algorithm. The system is designed to detect and prevent attacks such as SQL injection, proxy abuse, bots, spam, brute-force attempts, and FTP backlogging. It also supports access control by helping to restrict suspicious users, block unauthorized requests, and prevent attackers from reaching protected server resources.

In many existing intrusion detection systems, IDS programs run continuously in the background, which can slow down server performance and increase resource usage, especially in shared hosting environments where storage quota and server limits are critical. Continuous background checks may also create conditions that worsen denial-of-service risks. To address this, the proposed system uses cronjob scheduling to activate intrusion detection modules only when an event or action occurs on the server. This reduces unnecessary resource consumption while ensuring that server activities are captured and controlled by the cronjob manager.

The developed system also includes a communication channel that sends SMS and email alerts to the administrator when attacks or suspicious access attempts are detected. This enables proactive response and strengthens webserver security by combining intrusion detection, prevention, access control, and real-time notification.

7. Webserver security is based on confidentiality, integrity, availability, authentication, and proper access control. A poorly secured webserver application environment can expose organizational data and allow unauthorized users to access protected resources, even when the web application itself appears secure. As the use of internet and web-based applications continues to increase, abnormal and malicious activities are also recorded daily, making intrusion detection and prevention essential in webserver environments.

This research presented and implemented an intrusion detection system for a webserver application environment using a cronjob scheduling technique. The system improves the detection of different categories of

webserver intrusions and supports prevention through known pattern-matching methods. Access control is also strengthened by helping to restrict suspicious users, block unauthorized requests, and protect sensitive server resources.

The performance of the developed system was evaluated by deploying it on a Namecheap live shared hosting webserver to obtain log datasets, which showed reasonable detection results. The system can therefore be deployed in different webserver application development environments to support intrusion detection, prevention, access control, and proactive server security management.

REFERENCES

- [1]. Abomhara, M.& Koien, G.M. (2022). Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. *Internet of things and cyber security* 4(1), 65–88.
- [2]. Aulds, C (2020). *Linux Apache Web Server Administration*, latest Edition. Sybex Inc. 1(1)2
- [3]. Benson, V. & McAlaney, J. (2022). Frumkin, L.A.: Emerging threats for the human element and
- [4]. Counter measures in current cyber security landscape. *Psychological and Behavioral*
- [5]. Examinations in Cyber Security, 1(2) 266–271.
- [6]. Cova, M. Balzarotti, D., Felmetsger, V & Vigna, G. (2021). Swaddler: An Approach for the Anomaly-Based Detection of State Violations in Web Applications. *Proc. Int'l Symp. Recent Advances in Intrusion Detection* 1(7) 2-15.
- [7]. Enoch, S.Y (2020). A systematic evaluation of cyber security metrics for dynamic networks.
- [8]. Computer Networking 144, 216–229. <https://doi.org/10.17492/computology.v4i1.2406>
- [9]. Gupta, Shashank, & Gupta, (2022). XSS-SAFE: A server-side approach to detect and mitigate crosssite scripting (XSS) attacks in JavaScript code. *Arabian Journal for Science and Engineering* 41(3) 897–920.
- [10]. Kieyzun, A. (2022). Automatic creation of SQL injection and crosssite scripting attacks. In: *Proceedings of the 31st International Conference on Software Engineering*. IEEE Computer Society. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 19(5), 01-04.
- [11]. Mohammed A., Ambusaidi, Xiangjian He, Priyadarsi Nanda & Zhiyuan Tan, (2021). Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. *IEEE transactions on Computers*, 2(1) 2986 - 2998.

- [11]. Rajiv. A, A.Prashanthi, Ch. Bharadwaja (2022).Web Server Security evaluation and analysis.
- [12]. International Journal of Computer Science and Mobile Computing, 1(2)23-32.
- [13]. Samuel, K. O., & Osman, W. R.(2022) Cyber Terrorism Attack of the Contemporary Information Technology Age: Issues, Consequences and Panacea. International Journal of Computer Science and Mobile Computing, 3(5),1082-1090.
- [14]. Taha, A.F.; et al. (2021) Risk mitigation for dynamic state estimation against cyber attacks and unknown inputs. IEEE Trans. Smart Grid 9(2), 886–899
- [15]. Valeur. D, Vigna. G, Balzarotti, F, Robertson, C. Kruegel, &Kirda E, (2021). Reducing Errors in the Anomaly-Based Detection of Web-Based Attacks through the Combined Analysis of Web Requests and SQL Que-ries. Journal of Computer Security, 17, (3)305-329.
- [16]. Victoria Felmetzger, Ludovico Cavedon, Christopher Kruegel,& Gio-vanni Vigna,(2021). Towards Automated detection of logic vulnerabilities in web applications, International Journal of Advanced Research in Computer Science and Software Engineering 1(4)55-63.
- [17]. Zhang &Yuan, (2022).Proposed a Study of database intrusion detection based on improved association rule algorithm, International Journal of Advanced Research in Computer Science and Software Engineering 1(2)23-25