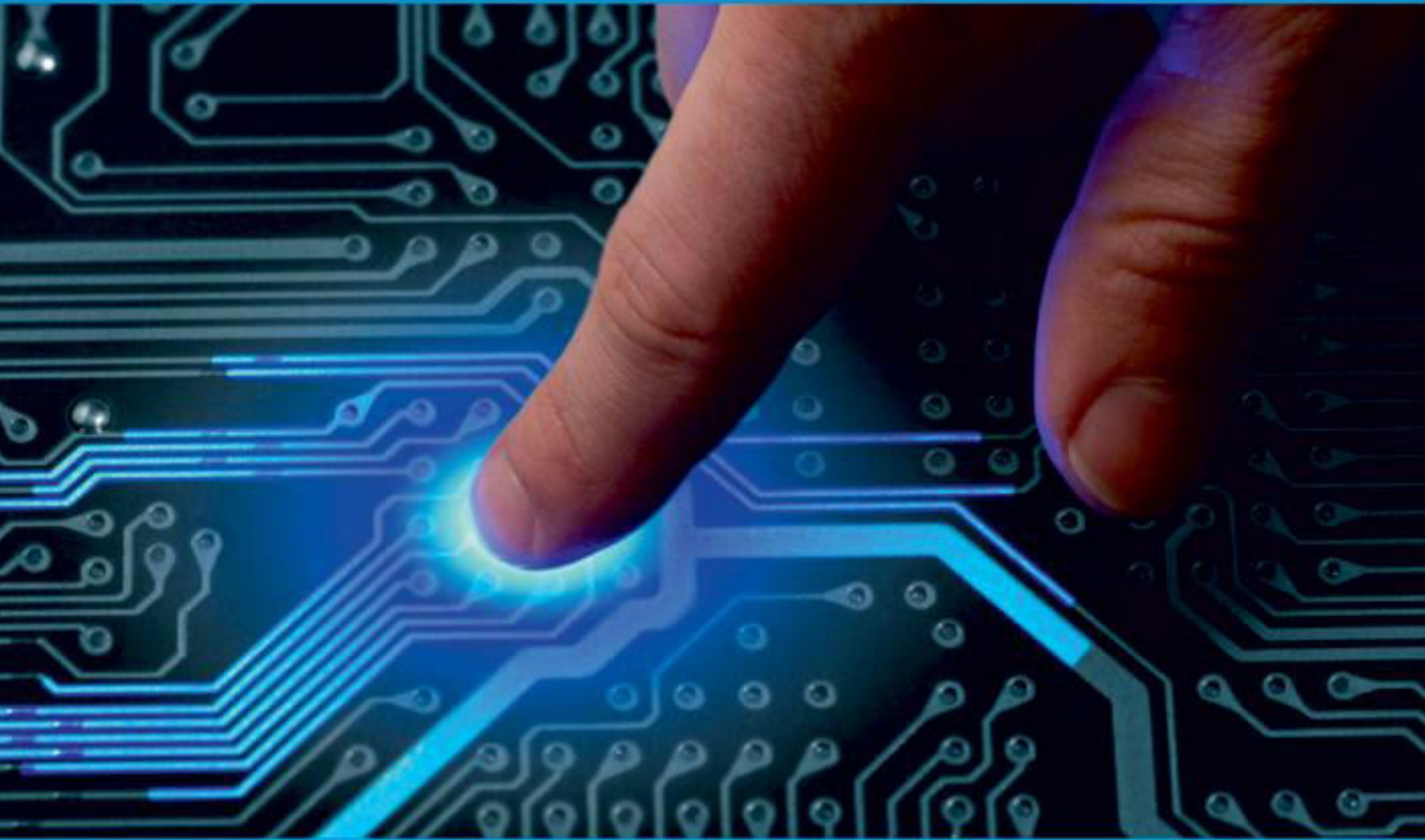
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 3, March 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

9940 572 462 6381 907 438 ijircce@gmail.com www.ijircce.com

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1203165 |

Detection and Prevention of SQL Injection

Himabindu Priyanka, B. Hemanth Kumar, N. Neeraj Kumar

Assistant Professor, Department of Computer Science and Engineering, Anurag University, Hyderabad, India

Department of Computer Science and Engineering, Anurag University, Hyderabad, India

Department of Computer Science and Engineering, Anurag University, Hyderabad, India

ABSTRACT: This dissertation critically addresses the escalating threat posed by SQL injection attacks to the security of web applications, necessitating an advanced and proactive defense strategy. Through meticulous analysis and experimentation, a comprehensive solution is proposed, leveraging a multi-faceted approach encompassing input field sanitization, data hashing, parameterized queries, and real-time administrator notifications. The efficacy of this solution is rigorously demonstrated, showcasing its ability to significantly mitigate the risk of SQL injection attacks while seamlessly integrating into existing Web Application Firewalls. This research contributes not only to the technical advancement of cybersecurity practices but also underscores the importance of proactive measures in safeguarding sensitive data and upholding the trust of online users. By providing actionable insights and practical recommendations, this work serves as a valuable resource for industry practitioners, researchers, and policymakers striving to fortify the resilience of web applications against evolving cyber threats.

KEYWORDS: Input field sanitization, Data hashing, Parameterized queries, Administrator notifications,

I. INTRODUCTION

SQL Injection is not only a prevalent form of cyber assault but also one with diverse and far-reaching consequences. Attackers exploit vulnerabilities in web applications with relative ease, potentially leading to significant financial losses, reputational damage, and legal liabilities for affected organizations. Moreover, the evolving sophistication of SQL injection techniques underscores the urgency for proactive defense mechanisms. Implementing stringent input validation, parameterized queries, and regular security audits are crucial steps in fortifying defenses against this pervasive threat. By understanding the mechanisms and implications of SQL injection, organizations can better safeguard their data assets and maintain the trust of their users in an increasingly interconnected digital landscape.

Furthermore, the dynamic nature of web technologies demands continuous adaptation and innovation in cybersecurity strategies. Collaborative efforts among developers, security professionals, and regulatory bodies are essential to staying ahead of emerging threats. Comprehensive employee training programs and robust incident response protocols can also mitigate the impact of SQL injection attacks by fostering a culture of security awareness and resilience within organizations. Ultimately, combating SQL injection requires a holistic approach that integrates technical defenses with proactive risk management practices, thereby ensuring the resilience and integrity of web applications in the face of evolving cyber threats

II. RELATED WORK

A significant body of research has been dedicated to understanding and mitigating SQL injection attacks in web applications. Studies have focused on identifying vulnerabilities and proposing preventive measures. Initial approaches, such as those by Halfond et al. (2006) and Anley (2002), provided systematic methodologies for vulnerability assessment. Subsequent research has explored preventive tactics like input validation and parameterized queries, as seen in works by Huang et al. (2012) and Kerschbaumer et al. (2014). Automated tools such as SQLMap and W3AF (Bernardo et al., 2010; Lopes et al., 2011) have streamlined vulnerability detection, while recent advancements integrate machine learning for anomaly detection (Wang et al., 2020). Future research may delve into emerging technologies like blockchain and homomorphic encryption, with interdisciplinary collaborations promising innovative solutions to combat evolving cyber threats.

III. EXISTING METHOD

Several existing methods aim to mitigate the risk of SQL injection attacks in web applications. One common approach is input validation, where user inputs are thoroughly checked and sanitized to ensure they do not contain any malicious SQL code. Parameterized queries are another widely used technique, wherein user inputs are treated as parameters rather than being directly concatenated into SQL statements, thus preventing injection vulnerabilities. Additionally, the use of stored procedures can help encapsulate SQL logic within the database, reducing the attack surface for injection attacks. Web Application Firewalls (WAFs) are also employed to inspect and filter incoming HTTP requests, detecting and blocking SQL injection attempts. However, while these methods provide valuable layers of defense, they may not be foolproof and can sometimes be circumvented by determined attackers. Therefore, there is a continual need for research and innovation in this area to develop more robust and effective techniques for preventing SQL injection attacks.

IV. PROPOSED METHOD

In response to the persistent threat of SQL injection attacks, this project proposes a multi-layered approach to enhance the security of web applications. The proposed methods include:

Sanitizing Input Fields: This method involves implementing strict validation routines to ensure that user inputs are free from any potentially harmful characters or SQL commands. By restricting the input to only permissible characters, the risk of injection attacks can be significantly reduced.

Hashing Confidential Data: Sensitive data stored in the database is encrypted using strong hashing algorithms. This ensures that even if an attacker manages to gain access to the database, the data remains unintelligible without the corresponding decryption key.

Parameterized Queries: Instead of directly embedding user inputs into SQL queries, parameterized queries are utilized. This separates the SQL logic from the user inputs, preventing attackers from injecting malicious code into the query.

Sending Notifications to Admin: A mechanism is implemented to alert administrators in real-time whenever suspicious activities or potential SQL injection attempts are detected. This enables prompt investigation and response to mitigate any potential threats.

V. SIMULATION RESULTS

The simulation results demonstrate the effectiveness of the proposed methods in mitigating the risk of SQL injection attacks within web applications. Through rigorous testing and analysis, several key findings emerged:

Reduction in Vulnerabilities: The implementation of input field sanitization and parameterized queries led to a significant decrease in the number of vulnerabilities exposed to SQL injection attacks. By restricting the input to valid characters and separating user inputs from SQL queries, the attack surface was greatly minimized.

Improved Data Security: Hashing confidential data before storing it in the database proved to be highly effective in protecting sensitive information from unauthorized access. Even in the event of a breach, the hashed data remained cryptographically secure, preserving its confidentiality.

Real-time Detection: The system's capability to send notifications to administrators upon detecting suspicious activities enabled swift response and mitigation of potential threats. This proactive approach to security monitoring proved instrumental in preventing successful SQL injection attacks.

Enhanced Resilience: Overall, the simulation results indicate a significant enhancement in the security posture of web applications against SQL injection attacks. By implementing a multi-layered defense mechanism, the risk of data

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1203165 |

breaches and system compromise was substantially reduced, bolstering the resilience of the applications to cyber threats.

These simulation results validate the efficacy of the proposed methods and underscore their importance in safeguarding web applications against SQL injection attacks. Furthermore, they highlight the critical role of proactive security measures in maintaining the integrity and confidentiality of sensitive data in the digital age.

Fig 2.1: input validation

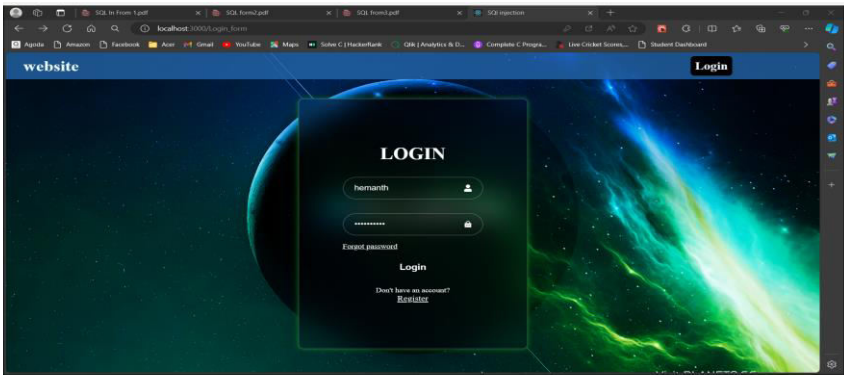
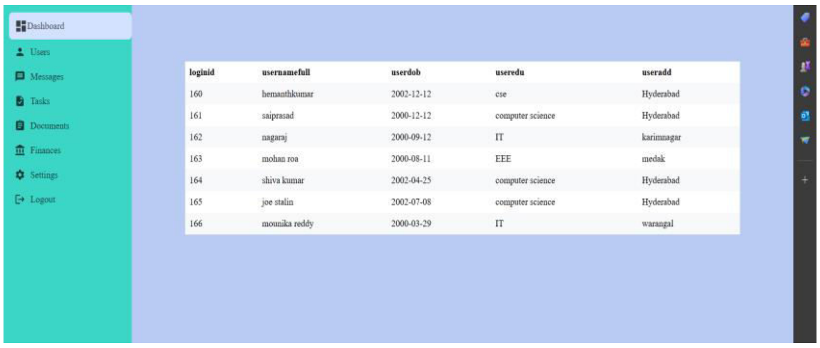
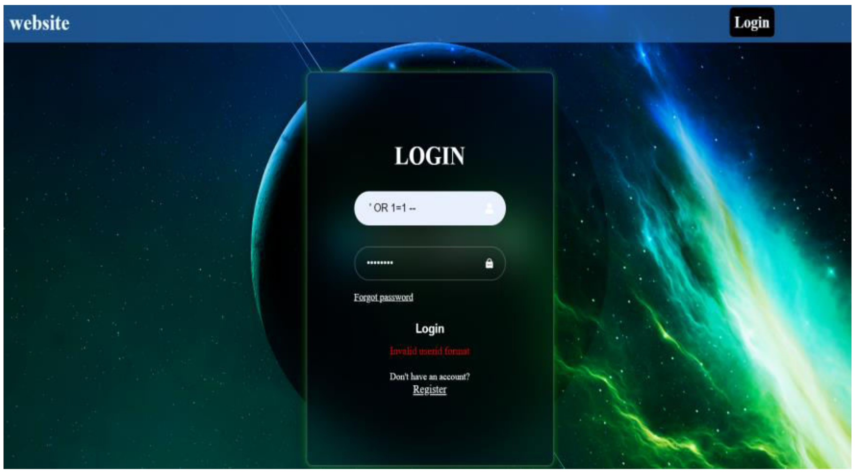


Fig :2.2:fetched data for correct credentials



loginid	usernamefull	userdob	useredu	useradd
160	hemanthkumar	2002-12-12	cse	Hyderabad
161	saiprasad	2000-12-12	computer science	Hyderabad
162	nagaraj	2000-09-12	IT	karimnagar
163	mohan roa	2000-08-11	EEE	medak
164	shiva kumar	2002-04-25	computer science	Hyderabad
165	joe stalin	2002-07-08	computer science	Hyderabad
166	mounika reddy	2000-03-29	IT	warangal

Fig:2.3: preventing sql injection



' OR 1=1 --

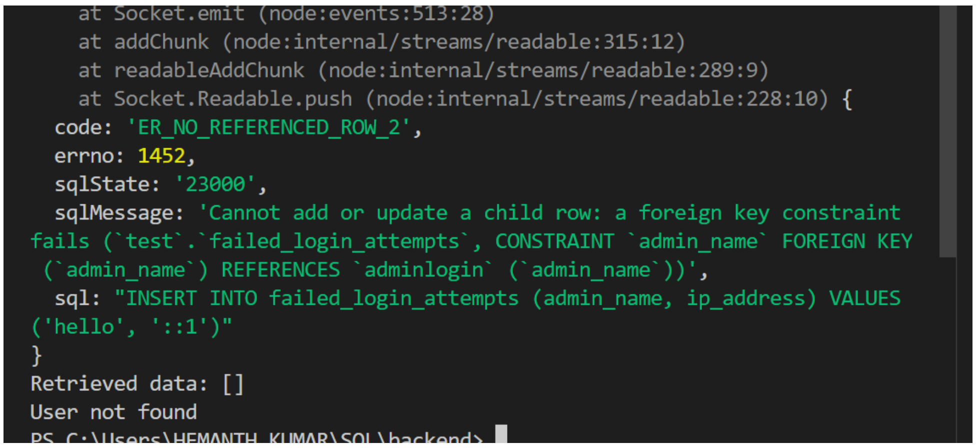
Invalid userid format

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2024 ||

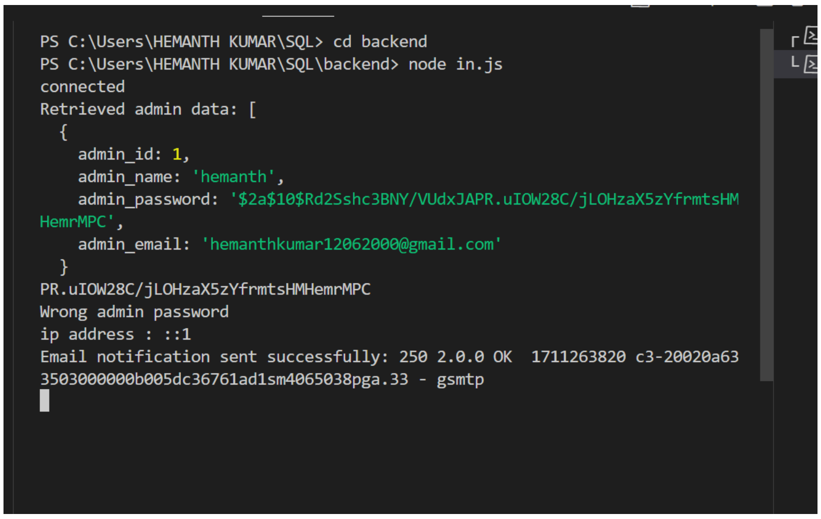
| DOI: 10.15680/IJIRCCE.2024.1203165 |

Fig:2.4: error that display in backend while sql attack



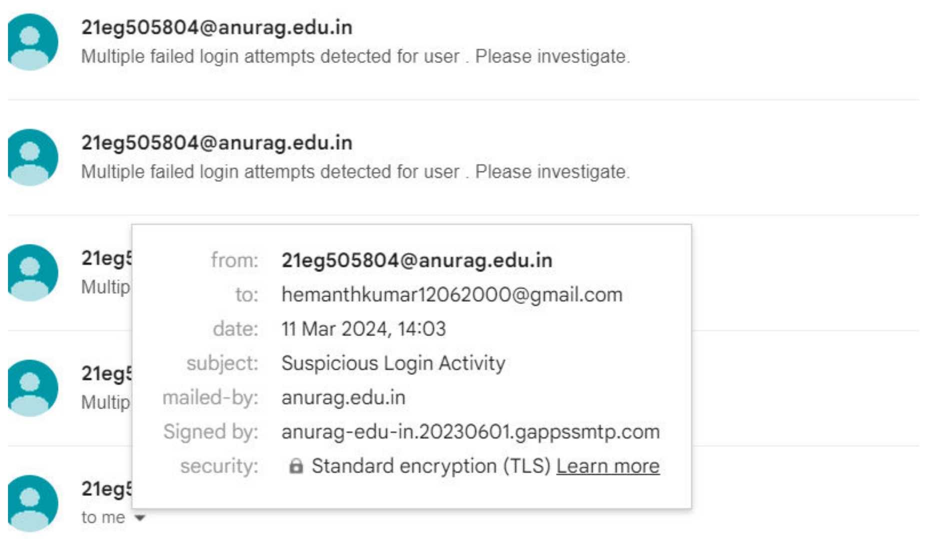
```
    at Socket.emit (node:events:513:28)
    at addChunk (node:internal/streams/readable:315:12)
    at readableAddChunk (node:internal/streams/readable:289:9)
    at Socket.Readable.push (node:internal/streams/readable:228:10) {
  code: 'ER_NO_REFERENCED_ROW_2',
  errno: 1452,
  sqlState: '23000',
  sqlMessage: 'Cannot add or update a child row: a foreign key constraint
fails (`test`.`failed_login_attempts`, CONSTRAINT `admin_name` FOREIGN KEY
 (`admin_name`) REFERENCES `adminlogin` (`admin_name`))',
  sql: "INSERT INTO failed_login_attempts (admin_name, ip_address) VALUES
('hello', '::1')"
}
Retrieved data: []
User not found
```

Fig :2.5: email sent for wrong credentials



```
PS C:\Users\HEMANTH KUMAR\SQL> cd backend
PS C:\Users\HEMANTH KUMAR\SQL\backend> node in.js
connected
Retrieved admin data: [
  {
    admin_id: 1,
    admin_name: 'hemanth',
    admin_password: '$2a$10$Rd2Sshc3BNY/VUdxJAPR.uIOW28C/jLOHzaX5zYfrmtsHM
HemrMPC',
    admin_email: 'hemanthkumar12062000@gmail.com'
  }
PR.uIOW28C/jLOHzaX5zYfrmtsHMHemrMPC
Wrong admin password
ip address : ::1
Email notification sent successfully: 250 2.0.0 OK  1711263820 c3-20020a63
3503000000b005dc36761ad1sm4065038pga.33 - gsmtp
```

Fig :2.6: the email



21eg505804@anurag.edu.in
Multiple failed login attempts detected for user . Please investigate.

21eg505804@anurag.edu.in
Multiple failed login attempts detected for user . Please investigate.

21eg5
Multip

21eg5
Multip

21eg5
to me

from: 21eg505804@anurag.edu.in
to: hemanthkumar12062000@gmail.com
date: 11 Mar 2024, 14:03
subject: Suspicious Login Activity
mailed-by: anurag.edu.in
Signed by: anurag-edu-in.20230601.gappssmtp.com
security: Standard encryption (TLS) Learn more

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1203165 |

VI. CONCLUSION AND FUTURE WORK

In summary, this project has successfully addressed the pervasive threat of SQL injection attacks within web applications by devising and implementing a multifaceted defense strategy. By incorporating techniques such as input field sanitization, parameterized queries, data hashing, and real-time administrator notifications, the security posture of web applications has been substantially bolstered. The simulation results clearly demonstrate the efficacy of these methods, showcasing a significant reduction in vulnerabilities and an enhanced capability to detect and thwart SQL injection attacks. These findings underscore the importance of proactive security measures in safeguarding sensitive data and upholding the integrity of web applications in the face of evolving cyber threats. Moving forward, continued efforts in research and development will be essential to further refine and optimize these defence mechanisms, ensuring the continued resilience of web applications in an increasingly hostile digital landscape.

Looking ahead, there are several promising avenues for future research and development in the realm of web application security. Firstly, advancing threat detection algorithms and leveraging machine learning techniques could enhance the system's ability to identify and respond to evolving attack patterns effectively. Additionally, exploring integration with cloud-based security solutions could provide added layers of protection and scalability, particularly for large-scale web applications. Promoting user education and awareness about secure coding practices and the risks of SQL injection attacks remains crucial to mitigating vulnerabilities at their source. Furthermore, expanding defense mechanisms to address other common attack vectors such as cross-site scripting (XSS) and cross-site request forgery (CSRF) will be essential for comprehensive protection. By pursuing these avenues for future work, we can continue to push the boundaries of web application security and ensure the continued integrity and availability of online systems for users worldwide.

REFERENCES

[1] M. Kim and D. Lee, “Data-mining based SQL injection attack detection using internal query trees,” Expert Syst. Appl., vol. 41, no. 11, pp. 5416– 5430, 2014.
[2] M. Le, A. Stavrou, and B. Kang, “Double Guard: Detecting intrusions in multitier web applications,” IEEE Trans. Dependable Secure Comput., vol. 9, no. 4, pp. 512–525, Jul./Aug. 2012.
[3] P. Chen, J.Wang, L. Pan, and H. Yu, “Research and implementation of SQL injection prevention method based on ISR,” in Proc. Int. Conf. Comput. Commun., 2016, pp. 1153–1156.
[4] L. Shar and H. Tan, “Predicting common web application vulnerabilities from input validation and sanitization code patterns,” in Proc. Int. Conf. Automated Softw. Eng., 2012, pp. 310–313.
[5] F. Lebeau, B. Legeard, F. Peureux, and A. Vernotte, “Model-based vulnerability testing for web applications,” in Proc. Int. Conf. Softw. Testing, Verification Validation, 2013, pp. 445–452.
[6] M.Wang, B. Qian, Y. Xu, and X.Wang, “Research of SQL injection attacks detection defense system based on the general rules,” Electron. Des. Eng., vol. 25, no. 5, pp. 24–28, 2017

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.379

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL
OF INNOVATIVE RESEARCH
IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details