A Survey of SQL Injection Attacks on E-commerce

Elif Ak, Gülçin Baykal, Ece Naz Sefercioğlu
Computer Engineering
Istanbul Technical University
Istanbul, Turkey
akeli@itu.edu.tr, baykalg@itu.edu.tr, sefercioglu@itu.edu.tr

***Abstract*—E-commerce is a place where people can buy products and services over the Internet. Since they share their private information such as their names and their credit card's number, its security is a major responsibility to obtain and to maintain. One of the attack types that can be used to steal information from the database, to steal money and to manipulate the prices of the products is SQL Injection attack. Therefore, this paper covers the differences of e-commerce from other web sites, e-commerce vulnerabilities that enable SQL Injection attacks, SQL Injection attack types and previously proposed SQL Injection prevention methods.**

Keywords—sql injection; e-commerce; security; prevention;

I. Introduction

Nowadays, Internet is a place where every individual have access. Since it is a common place for all people to access, the rise of e-commerce is inevitable. However, security is an issue, which is hard to maintain because of intense data transactions.

Since e-commerce is a very popular subject of the Internet that also improves people's life quality, its security is also an important concern. That is why, we chose to study on the area of e-commerce security. Each day, websites are under attack of more than 229.000 attacks, according to Symantec's Internet Security Threat Report [1]. E-commerce websites are also affected by these attacks and their security protocols should be strong enough to defeat these attacks. If they are not strong enough, e-commerce can be a hazardous environment for people to buy products.

55 percent of all web application attacks that occurred in the third quarter of 2016 include SQL Injection (SQLI) [1]. SQL Injection is a technique to attack applications that uses database systems, as is evident from its name. All e-commerce websites use database systems to store their customers', sellers' and products' information. With respect to the facts that are mentioned, SQL Injection attacks should be defeated for the sake of secure shopping over the Internet. Having many advanced search options to go through products as well as registration process provides input areas open to exploit. Even the payment process is exposed. Outcome of a successful attack may leak user name and hashed password. With a long list and technique such as brute force, passwords can be broken and the attacker can do personated shopping. In another point of the case, attacker simply acquires payment info of the user and uses it on other web sites. Personal information theft problem is also valid for e commerce websites. That is why SQLI attacks should be examined in detail especially on e-commerce web sites

In this paper, we present e-commerce vulnerabilities, type of SQL injection attacks and prevention/detection techniques as well as SQLI Tools.

II. E-commerce and SQL Injection

E-commerce is the sale of products and services over the Internet. People can buy any products that can come to mind and also they can sell their own products. On the other hand, people use their credit cards' information to buy products and if the e-commerce website has vulnerabilities that cyber criminals can easily exploit, this hijack can put the customers into dangerous situations. The one way to use vulnerabilities of e-commerce web sites is SQL injection attacks. SQL Injection Attacks (SQLIA) are the attacks that try to exploit database management vulnerabilities of websites by collecting, modifying and removing data. Vulnerability may arise from simple miss out on protective coding against injections to general features of the used system. Websites that is not secure against SQLIAs, feature input fields that paves the way to the database of the system through queries and leave it defenseless for harmful and strategically written SQL queries of the attacker.

In the result of a successful SQLIA, attacker can take many actions that risk the security of both the system and the users. Injecting malware to database is one of the most dangerous consequences for the system. Another threat is data loss and even destruction of the whole database. Modification of already existed data could also cause problems on both sides. In the target of these attacks, user information data may be disclosed. Taking into account the value of data, many impactful personal info of users may be leaked. A share of personal address or phone number could cause problems to experience on the first hand.

In order to highlights the importance of SQLI attacks, which target to e-commerce web sites, differences between e-commerce sites and other web sites should be examined in detail beforehand.

A. E-commerces and Other Web Sites

E-commerce is a kind of website that products are on sale and customers can make purchases on. As for every websites genre, there are some categories to look for in process of developing one. Song and Zahedi [2] present these categories for e-commerce under six titles.

Exhibition in which, exposure of products to the user in done ways of announcements, recommendations, and promotions. On delivery category, it is seen that the category is about providing the user some services on their purchase process. These services include fulfillment of customer rights and provided security. Impact category gives power to the user to share their thoughts or experience with the product on comments or rating as well as conducting searches on products a sharing the results to users, enabling other users to benefit from this information. Self-potency category is divided into 3 sub-topics that include individualization of user experience, practical use of the website and efficient information sharing. Lastly, category source simplification is done through improving product exposure, enabling custom-made products and providing different payment and shipping choices.

In a need to differentiate e-commerce websites to other websites, some characteristics can be presented. First of all, target of e-commerce is selling products, normal websites are content sharing oriented with no anticipation to sell any products. That is why E commerce is a need to expose products to customers as much as they can. To exposure the products there is a need for specialized search engines and detailed definitions and photographs for each product. Ensuring users to buy products can be achieved by providing trust regarding both of the product and the purchase operation. Trust for the product is acquired by the ratings and comments of other users and the trust for purchase operation is provided by informing the user the security policies used. In the light of this information regarding e-commerce websites, in table 2.1 differences between e-commerce websites and normal websites are shown briefly.

	E-commerce Websites	**Other Websites**
Aim	Selling products	Sharing content
Approach	Detailed description, feedback sharing for each product	Information distribution
Security	Should be guaranteed	Should be provided
Search capability	Rich	Simple
Feedback	On product and order	On shared information

Table 2.1 Differences between e-commerce and other web sites

B. SQLI Vulnerabilities of E-Commerce

As all web applications have vulnerabilities to SQLI attacks, e-commerce does not differ from them. Of course, these general vulnerabilities can be filtered for e-commerce. First of all, e-commerce requires many inputs from the user through the communication with her/him. Three main communication ways are the input fields of the forms in registration phase, login credential requirement in login phase and lastly search phase to find products [3]. Surely input fields of comment sections and forums can be included to input fields that are especially open to SQLI. Input fields are not the only places that Injection of threatening SQL script is done. URL fields of e-commerce are also threatened by the injection. With smart inspection, attackers can exploit values in the URL [4]. As a result of a successful SQLI attack, the attacker can reach classified information of the system, manipulate user profiles and remove important information from the system [5]. Money transactions may be changed by changing the amount or source to attacker's account.

Beside user actions and search capabilities, payment plays a big role in the operations of the e-commerce. If the application does shipping, leading to more vulnerable input areas and that is generally allowed to make modifications to the address in the process of payment. Address information is not the only part of the payment that is vulnerable to SQLI attacks, choice of payment method also exposures the database. Main payment methods in e-commerce can be listed as follows, using cards, bank transfer, cash, crypto-currency and direct to carrier [6]. From the listed methods most extensively input area user is payment by cards. Thorough the information taking phase, the site is vulnerable to possible SQLI attacks from the user. That may end up in accessing a part of or a whole of other users' credit card information, assuming they were saved in a separate database and could not be reached from the other operations on e-commerce. Other methods listed do not cause many threats in regarding the payment process, as they generally do not require any input at the moment of order. Moreover, gift coupons also cause vulnerability. If no scan is done for the input, the attacker may reach more prepared codes as well as other information. Inevitably, throughout the mentioned applications, there is a possible vulnerability for URL sided SQLI attacks.

E-Commerce Payment Method	SQLI Vulnerability
Credit/Bank Card	URL, credit card information fields
Bank Transfer	URL
Cash	URL
Crypto-currency	URL
Direct to Carrier	URL
Gift Coupons	URL, coupon input field

Table 2.2 SQLI vulnerabilities of e-commerce payment methods

What is more, the attacker can commit identity theft so that he/she can use the information on other platforms and cause harm the victimized user. If the administrative role is taken, the database of the e-commerce is all vulnerable to actions like data disclosure, destruction, or locking [7].

SQLI injection can exploit back-end mistakes as well. Such as programming that leaves the system easy to be exploited and lack of SQL server certificates that leave authentication unverified by right owners [8]. Moreover, user accounts should be examined regularly so that an attempt to steal an inactive/lost user's profile can be spotted.

In an example scenario, if a user added a harmful SQL code next to his/her search prompt and there is no countermeasure for SQLI attacks on the e-commerce system, a simple user could reach delicate data of the database, moreover may manipulate or delete it permanently [9].

In the lights of above information, what security titles [10] are affected by successful SQLI attacks on e-commerce will be commented on.

- Data manipulation harms protection of data
- With thoughtful scripts, attacker could reach server side and do harmful activities there
- The data that is transferred can have harmful modification that may affect users or the web application

- Suppliers of attacked e-commerce will also be affected as their data may be stolen or the transaction values would be modified
- Mechanism of protecting the security of the e-commerce is proven to be unsuccessful and study is needed on new countermeasures
- Reputation of the e-commerce will be damaged and their future works will be affected
- Web application would be under of denial of service with the loss or modification of important data

C. *SQLI Attacks on E-Commerce*

There exist some differences between SQLI attacks on e-commerce and other web applications. Even though some similarities on motivation can be inspected such as user data manipulation destruction or theft, one of the most striking differences between the motivations to SQLI attacks is transaction manipulation and accessing regarding the transaction.

In the process of applying SQLI attack, applications share vulnerabilities on input and URL fields. On the other hand, e-commerce wants the user to find the products buy effectively and that is why they serve excessive search options and fields. These presented input fields increase the possibility of security leak and the success possibility of SQLI on e-commerce.

After the successful SQLI, sensitive data of the web applications left open to be exploited. Additionally, for e-commerce, attackers may steal payment information of account or credit card and use or sell them accordingly. Moreover, in the process of the transaction, by making manipulation to purchase amount they commit unjust enrichment, living the e-commerce in a state of impoverishment.

SQL Injection attacks are the most known attacks that attackers use to exploit especially payment system’s vulnerabilities. According to Ponemon reports, forty-nine percentage of respondents said that own companies face to SQLI attacks and result to data exploitation [11]. For example, in 2015, Magento’s e-commerce platform was under SQL Injection attack [12]. With the attack, attackers inserted a new admin user to the database system. “vpwq” and “defaultmanager” were the user names used in the exploit. This attack has shown that, because of the vulnerabilities that led to this attack, system was available for the manipulation of products’ prices. After the incident, Magento released a new updated version.

Another possible example of SQL Injection attack over an e-commerce platform is the one that happened recently, in April 2017. Reports from Rapid7, CERT/CC and SwissCERT warned BPC Banking Technologies of Switzerland about the SQL Injection vulnerabilities that were present in their SmartVisa suite of e-commerce and financial software product [13].

III. SQL Injection Attack Methods

Many methods can be used to apply SQL injection. Attack can be carried out by handwritten queries as well as mass query input enabling, SQL injection vulnerability detecting programs like *sqlmap*. On a tutorial, carried out by Charania and Vyas, sqlmap is proved to be a prosperous tool to detect and invade websites with SQL injection vulnerabilities [14].

A. *Type of Attacks*

In order to understand in better way, SQL Injection attacks can be classified in three categories as: Tautology-based Attacks, Availability Disruption Attacks and Piggy-backed Queries.

1) *Tautology-based Attacks*

In this type of attacks, malicious user injects SQL statement, which always produce *true* condition. This type of conditional tricks may cause authorization problems. For example, most of e-commercial web applications uses authentication process in order to eliminate unauthorized person. In generally, this is simple form page with two input area: user name and password. Following example shows that usage of tautology-based attacks to break login page.

```
<form id="form1" method="post" runat="server">
      Username: <input name="usrName" id="usrName" type="text" />
       <br />
      Password: <input name="pswd" id="pswd" type="password" />
       <br />
      <input type="button" id="btnLogin" value="Login" runat="server"/>
</form>
```

In the basic login page, desired inputs from user are user name and password as string format. Moreover, the server-side, verification system checks the whether such user in exist in database. In order to apply verification, SQL command runs and server sends to regarding database table, for example `UserAccounts` table. In normal case, SQL Command can be produces using string format. For example, regarding C# code should be like this,

```
string query = String.Format("SELECT ObjectId FROM UserAccounts Where UserName = {0} AND Password = {1}", usrName.Value, pswd.Value);
```

Above C# code produces desired SQL script and checks the `ObjectId` is exist or not. Given that if the malicious user enters a password, which includes meaningful SQL keywords, code is queried on web server, which is vulnerable to SQL injection. Consider the following SQL script,

```
SELECT ObjectId FROM UserAccounts
Where UserName = 'admin' AND
      Password = 'anything' OR 1=1
```

As a result, hacker gains entry clearance to the web application using tautologies. A. Tautology-based Attacks technic can be used such authentication cases, which can produce grand access issues for e-commerce web applications and break two basic security principles: *confidentiality* and *authenticity.*

2) Denial of Service Attacks

In this type of attacks, hacker targets the denial of database server to destroy or damage the information, which means disrupting of another two basic security principles: *integrity* and *availability.*

SQL scripts has keywords to capable of destroying database tables, rows in the table (information) and even database itself.

For example, consider a search panel with multiple search criteria and given that, one of them is user name. Following SQL query gets the users, which include user name *John:*

```
SELECT * FROM UserAccounts
Where UserName = 'John'
```

If malicious user insert SQL keywords to input in the search screen form area instead of just writing *John,* new SQL query can take forms as following;

```
SELECT * FROM UserAccounts
Where UserName = 'John'; Truncate Table UserAccounts;
```

In this case, user writes to user name search input area `'John'; Truncate Table UserAccounts;` so that `UserAccounts` table is destroyed and all rows in the table are deleted. Other keywords to damage availability are `Delete` and `Drop,` delete row/rows with specified condition if any, drop table from database respectively.

3) Piggy-backed Queries

As the name implies, in such SQLI attacks means another query attachment to existing query to obtain additional data or damage the database as mentioned previous part. In order to that, delimiter character '**;**' is used to add harmful query right after original query. This type of attacks may break both *availability* and *confidentiality.*

SQLI Attack Methods\ Objectives of Computer Security	Confidentiality	Integrity	Availability
Tautology-based Attacks	Attacker can see the data inside the database	Attacker can modify, cause damage to or destroy the data inside the database	X
Denial of Service Attacks	X	Attacker can cause damage to or destroy the data inside the database	Attacker puts the database in a non-reachable state.
Piggy-backed Queries	Attacker can see the data inside the database	Attacker can modify, cause damage to or destroy the data inside the database	Attacker puts the database in a non-reachable state.

Table 3.1 SQLI attack methods' possible effects on objectives of computer security

B. *SQLI Tools*

After described security weaknesses, there are many atomized malwares to defend against SQLI. A user that wants to choose their SQL injection program, as Shar and Tan commented, he/she has some other choices apart from sqlmap [15]. A brief reminder for sqlmap, it is a python based console program run on terminal with predefined commands. Using specific commands user can detect the vulnerabilities as well as infiltrate the database system. One of the options can be chosen is SecuBat that comes with UI in reverse of sqlmap. It go through the html side of the page and searches for input fields, then implements exploiting query injections. To verify the attack is successful, after the query post it monitors and analyses the response from the server if it matches with the response waited from the implemented attack. Another SQL injection tool is Nikto2 that also runs on terminal. The tool scans the target website and returns the requested type of vulnerabilities as a report. One feature of the program that can be called a flaw that it is not stealthy, websites with security measures may detect they are being scanned. The reason is simply nikto2 was developed for monitoring purposes on mind, not usage for stealthy actives. From the mentioned tools, Nikto2 is more open to detection then other as it did not have any prevention to be found out. As having a user interface, SecuBat seems more user friendly, at least to beginners. For the SQL injection power sqlmap prove itself exterior as it provides more features on authentication, control, control and coverage. Moreover, it supports input vectors on cookie and header files of to be exploited web site.

	sqlmap	Secubat	Nikto2
Technology	Python	.Net	Perl
UI	No	Yes	No
Reporting	Yes	No	Yes
Usage	Simple	Simple	Simple
Input Vector	Get, Post, Cookie, Header	Get, Post, Cookie	Get, Post
Configuration	Complex	Simple	Medium

Table 3.2 Basic information of SQLI tools[16,17,18]

IV. SQL Injection Prevention Methods

In generally, SQLI defense methods are classified as coding, detection and run-time prevention. In coding method, more work load burden the developers because developers either manually should eliminate the escaping characters and special SQL keywords such as `Drop`, `Truncate` etc. or dynamically uses stored procedures as inter layer between user interface and database. As mentioned by Shar and Tan in their article, most developers tend to use stored procedures to prevent SQLI; however, still there is possibility to become vulnerability web application [15]. Another method is detection involve testing step of the software. It can be self-written test cases or be tool. For example, sqlmap is open source SQLI attack and detection tool. Last way to defend against SQLI is run-time prevention, which is again hard-coded run time based module. On the other hand, Microsoft state that Entity Framework is design to prevent SQLI attacks with parameterized quires without not combining user inputs directly SQL queries using string concatenation [19]. Following prevention techniques are listed by type and origins:

In [20], authors presented a new approach to protect Web applications against SQL Injection, which is also applicable for e-commerce. Positive tainting and syntax-aware evaluation are the bases of this approach and they implemented their techniques in a WASP tool. According to their test results, WASP did not produce any false positives while it was able to stop all of the otherwise successful attacks.

In [21], authors presented a technique, which can be considered as an extended version of WASP, called R-WASP. This technique focuses on real time web application SQL Injection attacks and since e-commerce is also a real time system, it is more suitable for e-commerce to protect it from SQL Injection attacks.

In [22], a prevention tool called SecuriFly is implemented for Java. It can be used for e-commerce websites that are written with Java. This tool tries to sanitize query strings, which are generated by tainted input, but it is not usable to stop injection attacks that insert SQL commands in numeric fields.

In [23], authors presented a method that dynamically extracts programmer intended SQL queries for automatic prevention, called CANDID.

In [24], authors discussed prepared statements. Since some of the SQL Injection attacks consist of special characters in the queries, with the help of Prepared Statement interface, queries that consist the special characters are eliminated from execution in databases that uses JDBC for database connectivity.

In [25], authors suggested a technique that can prevent tautologies and union attacks for e-commerce whose databases in PHP, called PDO Parameterized Queries.

In [26], authors suggested a model based approach for query validation in run time. If the generated query in run time is malicious, it is not matched with the static query modal. Therefore, it will be rejected. E-commerce that are written with .NET can use this technique since .NET have tools for implementation of this technique.

In [27], authors proposed a manual approach that programmer writes code in a way that any malicious input cannot be inserted or code is reviewed to prevent SQL Injection attacks on the database.

In [28], authors suggested a prototype called SQL Injection Protector for Authentication (SQLIPA) to evaluate a hash mechanism for user name and password to improve authentication performance. Since e-commerce stores users' information, this technique is applicable for e-commerce.

SQLI Prevention Approaches	Description	Special Target Platform
Stored Procedures[15]	Using input as variable inside SQL procedures.	None
Entity Framework[19]	Parametrized queries	.NET
WASP tool[20]	Positive tainting and syntax-aware evaluation	None
R-WASP tool[21]	Emphasizes on real time for WASP	None
SecuriFly[22]	Sanitizes query strings, which are generated by tainted input	Java
CANDID[23]	Dynamically extracts programmer intended SQL queries	None
Prepared Statement[24]	Eliminates queries that consist the special characters	JDBC
PDO Parameterized Queries[25]	Prevents tautologies and union attacks	PHP
Modal Based[26]	Malicious query is not matched with the static query modal	.NET
Manual[27]	Malicious input cannot be inserted with written code	None
SQL Injection Protector for Authentication[28]	Evaluates a hash mechanism for user name and password	None

Table 4.1 Prevention techniques of SQLI attacks

V. CONCLUSION AND FUTURE WORK

E-commerce is an important platform to inspect security vulnerabilities, as it has both valuable data on users and products and Money transaction systems. SQLI attacks are one of the threats that threaten these components. SQLI attack methods should be investigated and countermeasures should be taken. Moreover, tools that enable an intense number of simultaneous SQLI attacks should be taken into consideration. To fight SQLI attacks regarding e-commerce, first, one must understand what e-commerce is and how it differs from other web applications. So that the possible exposures can be spotted and protective actions are applied. Finally, a survey of prevention of SQLI attacks is presented to enlighten the area to secure e-commerce system before the occurrence of the attack. Our contribution is providing this survey of prevention techniques of SQLI attacks on e-commerce to aid people who work in this area so that they can benefit from past works.

Investigating our paper brightens up the vulnerability and prevention techniques regarding SQLI attacks on e-commerce. As a future work, what actions must be taken after being attacked with SQLI can be researched and that actions can be classified under specific topics in security.

VI. REFERENCES

[1] H. Huang, Z. Zhang, H. Cheng and S. Shieh, "Web Application Security: Threats, Countermeasures, and Pitfalls", *Computer*, vol. 50, no. 6, pp. 81-85, 2017.

[2] J. Song and F. Zahedi, “Web Design in E-Commerce: A Theory and Empirical Analysis”, *International Conference on Information Systems (ICIS)*, pp. 205-220, 2001.

[3] R. Thiyab, M. Ali and F. Abdulqader. “The impact of SQL injection attacks on the security of databases in Zulikha”, *Proceedings of the 6th International Conference of Computing & Informatics*, pp. 323-331, 2017.

[4] T. Tilahun, “Designing Application Based Intrusion Detection Mechanism for E-Commerce Systems Using Supervised Data Mining Technique”, *2014 NIT-MTMI International Conference on Emerging Paradigms and Practices in Global Technology, Management & Business Issues*, vol. 11, pp. 504-509, 2014.

[5] P. Suchánek,: “E-commerce Systems and E-shop Web Sites Security*”.The Internet, Competitiveness and Organisational Security in Knowledge Society*, 2009.

[6] THE PAYPERS, "Ecommerce Payment Methods Report 2016", 2017.

[7] K. Mlelwa and Z. Yonah, “Challenges That Restrict The Efficiencies Of Security Frameworks In E-Commerce: A Review”, *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 3, 2017.

[8] P. S. Lokhande, B. B. Meshram, “E-Commerce Applications: Vulnerabilities, Attacks and Countermeasures”, *International journal of Advanced Research in Computer Engineering and Technology*, vol. 2, pp. 499-509, 2013.

[9] C. Guynes, Y. Wu and J. Windsor, "E-Commerce/Network Security Considerations", *International Journal of Management & Information Systems (IJMIS)*, vol. 15, no. 2, p. 1, 2011.

[10] A. Abdo and A. Belbel, "Security and Reliability", Researchgate, 2017. [Online]. Available: https://www.researchgate.net/publication/319154153_Security_and_Reliability. [Accessed: 21- Nov- 2017]. Ponemon Institute (2014). *The SQL Injection Threat Study*.

[11] J. Kirk, "Hackers exploit Magento e-commerce vulnerability", *ITworld*, 2017. [Online]. Available:

https://www.itworld.com/article/2914515/hackers-exploit-magento-ecommerce-vulnerability.html. [Accessed: 06- Nov- 2017].

[12] M. Mimoso and T. Spring, "Vendor BPC Silent on Patching SQL Injection in SmartVista Ecommerce Software", *Threatpost | The first stop for security news*, 2017. [Online]. Available: https://threatpost.com/vendor-bpc-banking-silent-on-patching-sql-injection-in-smartvista-ecommerce-software/128386/. [Accessed: 06- Nov- 2017].

[13] S. Charania and V. Vyas, "SQL Injection Attack: Detection and Prevention", *International Research Journal of Engineering and Technology*, vol. 3, no. 4, pp. 1496-1501, 2016.

[14] L. Shar and H. Tan, "Defeating SQL Injection", *Computer*, vol. 46, no. 3, pp. 69-77, 2013.

[15] "Security Considerations (Entity Framework)", *Docs.microsoft.com*, 2017. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/security-considerations. [Accessed: 18- Oct- 2017].

[16] "Detailed Web Application Scanner Information - sqlmap - WAVSEP Benchmark 2014/2016", Sectoolmarket.com, 2017. [Online]. Available: http://www.sectoolmarket.com/web-application-scanners/39.html. [Accessed: 30- Nov- 2017].

[17] "Detailed Web Application Scanner Information - Secubat - WAVSEP Benchmark 2014/2016", Sectoolmarket.com, 2017. [Online]. Available: http://www.sectoolmarket.com/web-application-scanners/20.html. [Accessed: 30- Nov- 2017].

[18] "Nikto v2.1.5 - The Manual", Cirt.net, 2017. [Online]. Available: https://cirt.net/nikto2-docs/. [Accessed: 30- Nov- 2017].

[19] W. Halfond, A. Orso and P. Manolios, "WASP: Protecting Web Applications Using Positive Tainting and Syntax-Aware Evaluation", *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 65-81, 2008.

[20] M. H. Alattar and S. P. Medhane, “R-WASP: Real Time-Web Application SQL Injection Detector and Preventer”, *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 5, pp. 327–330, 2013.

[21] M. Martin, B. Livshits and M. Lam, "Finding application errors and security flaws using PQL", *ACM SIGPLAN Notices*, vol. 40, no. 10, p. 365, 2005.

[22] P. Bisht, P. Madhusudan and V. Venkatakrishnan, "CANDID", *ACM Transactions on Information and System Security*, vol. 13, no. 2, pp. 1-39, 2010.

[23] S. Thomas and L. Williams, “Using Automated Fix Generation to Secure SQL Statements”, in *Proceedings of the Third International Workshop on Software Engineering for Secure Systems (SESS '07)*, p. 9, 2007.

[24] M. Sendiang, A. Polii and J. Mappadang, “Minimization of SQL Injection in Scheduling Application Development”, in *International Conference on Knowledge Creation and Intelligent Computing (KCIC)*, IEEE, Indonesia, 2016.

[25] S. Jain and A. R. Pais, " Model Based Approach to Prevent SQL Injection Attacks on .NET Applications", *International Journal of Computer Science & Informatics*, vol. 1, no. 2, 2011.

[26] M. Junjin, “An Approach for SQL Injection Vulnerability Detection” *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations*, IEEE Computer Society, Las Vegas, pp. 1411-1414, 2009.

[27] S. Ali, SK. Shahzad and H. Javed, “SQLIPA: An Authentication Mechanism against SQL Injection”, *European Journal of Scientific Research*, vol. 38, no.4, pp. 604-611, 2009