Network Forensics for Detecting SQL Injection Attacks Using NIST Method

Arif Roid Caesarano[1], Imam Riadi[2]
[1]*Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[2]*Department of Information System, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
(arif1300018111@webmail.uad.ac.id, imam.riadi@is.uad.ac.id)

ABSTRACT

SQL Injection is a technique to exploit web applications that use the database as data storage. By being able to influence what will be forwarded to the database, the attacker can exploit the syntax and capabilities of SQL, as well as the power and flexibility to support database operation functions and available system functionality to the database. The purpose of this study is that Snort IDS that can detect SQL Injection attacks produces logs that can provide information about attackers and attack notifications in real time using email. The subjects in this study are building a webserver network system using Snort IDS to detect SQL Injection attacks. The method used is NIST 800-30, where there are 9 important stages in risk assessment. Data collection methods in this study are observation and literature study. The research stage is the stage of doing a case simulation to try to implement Snort in detecting intrusions or attacks, where there are 5 stages of research namely vulnerability testing, attack scenario, snort configuration, data collection, and analysis stage. The results of this study are a webserver system development using Snort IDS for SQL Injection attack detection systems and real time attack notifications using email.

KEYWORDS

Network, Forensics, SQL, Injection, NIST.

1 INTRODUCTION

The security of Internet-based information systems in today's global era is a must to pay more attention, because the public and global Internet networks are basically unsafe. When data is sent from one computer to another on the Internet, it passes through a number of other computers which means it will allow the user to take over one or more computers. Unless a computer is locked in a room with limited access and the computer is not connected out of the room, it will be safe. Internet security breaches happen almost every day around the world.

Computer networks connected to the internet provide a lot of convenience in accessing information from around the world. However, network connections with the Internet actually increase the possibility of interference with system security. Computer network security is very important to maintain the validity and integrity of data and ensure the availability of services for its users.

The other side, internet does not always give the promised that can provide a variety of information that exists in any part of the world, because the various crimes that exist in real life was more common in the internet world. Crime on the internet is popular with the name cybercrime. Today, the internet has become part of our daily lives as one of the means of communication in business as well as for the private. But behind that there are many holes weakness of the system on the internet that can be exploited by the cracker for not good purposes, such as mail bombs, randomize home page, data theft, password or credit card number.

Computer networks connected to the internet provide a lot of convenience in accessing information from around the world. However, network connections with the Internet actually increase the possibility of interference with system security. A computer becomes easily accessible and at risk to be infiltrated by parties who want to access the computer. Consequently computer systems are at risk to threats or attacks. It is very dangerous for corporate computer systems that contain confidential data and may only be accessed by certain people only. Forms of threat that may occur is tapping or theft of confidential data. Therefore the computer network system must be equipped with a system that can detect intrusion or intrusion. The system is known as Intrusion Detection System (IDS)

Computer network security is very important to maintain the validity and integrity of data and ensure the availability of services for its users. In order for the computer network system is not disturbed even to be damaged by intruder attacks, it is necessary network security system that can cope and prevent intruder attacks.

Intrusion Detection System (IDS) is an attempt to identify intruders entering the system without authorization or a legitimate user but misuses the privilege of system resources. Intrusion Detection System is implemented with the application of sniffing process, observation of data traffic, and traffic log analysis. Thus, an administrator can make decisions based on observed traffic to determine the network security settings they manage.

SQL Injection is a technique of exploiting web applications that use database as data storage. By being able to influence what will be forwarded to the database, an attacker can take advantage of SQL syntax and capabilities, as well as the power and flexibility to support database operation functions and system functionality available to the database.

Detect of SQL Injection attacks on the web server as done with forensic evidence in the forensic process model approach, is forensic methods for collecting information, checking, analyzing, and reporting. Methods that can be used to perform information security risk management such as Octave, NIST SP 800-30 and ISO 27001. NIST SP800-30 Has 9 steps to perform risk analysis is system characterization, threat identification, vulnerability identification, control analysis, trend analysis, impact, risk determination, control recommendations and documentation.

NIST SP 800-30 has been shown to contribute more like providing a consistent and comprehensive information security insight for policy makers, structured resource modeling, information security insights acceptable to multiple risk takers, easy identification of threats, decision makers do not hesitate to take risks because every risk has been properly investigated. NIST SP 800-30 best of 3 methods for risk analysis is Mehari, Magerit and Microsoft's Security Management Guide especially when performing risk analysis, NIST SP 800-30 provides control recommendations.

2 BASIC THEORY

2.1 Network Forensics

Network forensics is defined as the capture, recording, and analysis of network events to determine the source of security attacks or other incident problems. In other words, network forensics involves retrieving, recording and analyzing network traffic. Network data comes from existing network security equipment such as Firewall or intrusion detection systems, checked for attack characterization, and investigated to be traced back to the attacker.[1]

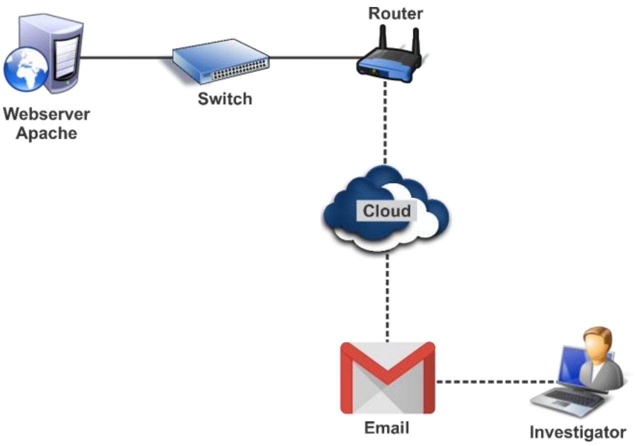


Figure 1. network forensics flow

Digital evidence can be collected from various sources depending on the needs and changes in the investigation. Figure 1 shows Collection evidence in this study used recordings of traffic from web server apache. Digital evidence can be collected at the server level, proxy level or some other source. For example the level of digital proof server can be collected from web server logs that store browsing activities that are frequently visited.[1]

2.2 Intrusion Detection System (IDS)

Intrusion Detection System (IDS) is a software or hardware application that can detect suspicious activity in a system or network. IDS can inspect inbound and outbound traffic in a system or network, perform analysis and look for evidence of intrusion experiments (intrusions).[2]

2.3 Snort IDS

Snort IDS is an open source IDS that is de facto the industry standard IDS. Snort can be downloaded at www.Snort.org. Snort can be implemented in a multi-platform network. Snort is one of the software to detect system instruments, able to analyze real-time traffic and logging IP Address, able to analyze port and detect any kind of intrusion or attack from outside like buffer overflows, stealth scan, CGI attacks, SMPS probes, OS fingerprinting. [3]
The Snort Configuration The steps in performing the Snort configuration are as follows:[6]

1. Setting the variable for the network to be detected.
2. Configuring dynamic loaded libraries.
3. Configure preprocessors.
4. Configure the output plugins.
5. Add other configuration runtime.
6. Customize the rule or rule to be added.

2.4 Structured Query Language (SQL)

Structured Query Language (SQL) is basically a textual language that allows interaction with database servers. SQL commands like INSERT, RETRIEVE, UPDATE, and DELETE are used to perform operations on the database. The programmer uses this command to manipulate the data in the database server.

2.5 SQL Injection

SQL Injection is defined as a technique that exploits an Unvalidated input vulnerability and injects SQL commands through a web application run in a back-end database. Based on the definition, it can be said that SQL Injection attack is very dangerous because the attacker who has successfully entered the system database can manipulate the existing data in the system database. Improper manipulation of data by an attacker can cause harm to the owner of an injected website. Leakage of data and information is fatal. Such data may be misused by irresponsible parties.[4]

Based on the definition, it can be said that SQL Injection attack is very dangerous because the attacker who has successfully entered the system database can manipulate the existing data in the system database. Improper manipulation of data by an attacker can cause harm to the owner of an injected website. Leakage of data and information is fatal. Such data may be misused by irresponsible parties.

3 Methodology and Experimental Setup

3.1 System Architecture

Website with a local server with http:// 192.168.232.1/SQLInjection/ address used as an example of SQL Injection attack, the website has a basic PHP script that loopholes against SQL injection attacks and uses mysql Database.[5] There is a login menu on the home page using a web application in the form of input to be able to access information in the website. after user login there are 4 menu that is menu send message, view message, edit profile. In send message menu user can send message to other user and can see the message in view message menu, user can edit profile information of each user in edit profile menu, user can edit firstname, surname, and email. web server using windows OS 10, web application technology PHP 5.6.31, apache 2.4.26 and back end MySQL DBMS 5.[6]

Snort IDS generally runs in Linux operating system. on the windows operating system must be configured in advance in order to run properly. This forensic research network uses a localhost server. This server uses a static IP that is 192.168.232.1. The initial configuration stage is the installation process of Snort IDS and then enter the default rule of snort to detect SQL Injection attacks.[7]

3.2 NIST Method

NIST 800-30 is a standard document developed by the National Institute of Standards and Technology which is a continuation of the legal responsibilities under the Computer Security Act Act of 1987 and the Information Technology Management Reform Act of 1996. NIST 800-30 there are two important stages of risk assessment and risk mitigation.[8]
Stages of risk assessment based on NIST 800-30 are:

1. System Characterization
   At this stage, the boundaries of IT systems must be identified, including resources and information.
2. Threat Identification
   Consideration of the possibility of emerging threats such as sources, potential vulnerabilities and existing controls.
3. Vulnerability Identification
   Identification of vulnerabilities is used for the development of vulnerability lists of systems that can be utilized later.
4. Control Analysis
   An analysis of the controls that have been implemented or planned for implementation by the organization to minimize or eliminate the likelihood of development from threats.
5. Likelihood Determination
   The process of ranking against potential vulnerabilities can be carried out in the environment of vulnerability. Factors to be considered are threats (source and ability), nature of vulnerabilities and the existence and effectiveness of controls when applied.
6. Impact Analysis
   This stage is used to determine the negative impact resulting from the successful application of vulnerability.
7. Risk Determination
   Risk level assessment of IT systems is done in this step.
8. Control Recommendations
   This stage assesses which controls can reduce or eliminate identified risks. the recommended control should be able to reduce the risk level on the IT system and data, to an acceptable level of risk.
9. Results Documentation
   At this stage, the development of the risk assessment report (source of threat, vulnerability, assessed risk and recommended control) is carried out.

3.3 *SQL Injection Attack Scenario*

Scenario of SQL Injection attack was conducted to test whether the configuration Intrusion Detection System (IDS) Snort on the web server has been successfully installed. The simulation was performed using SQLMAP used to test Intrusion Detection System (IDS) Snort to detect attack of SQL Injection as shown in Figure 2.[9]

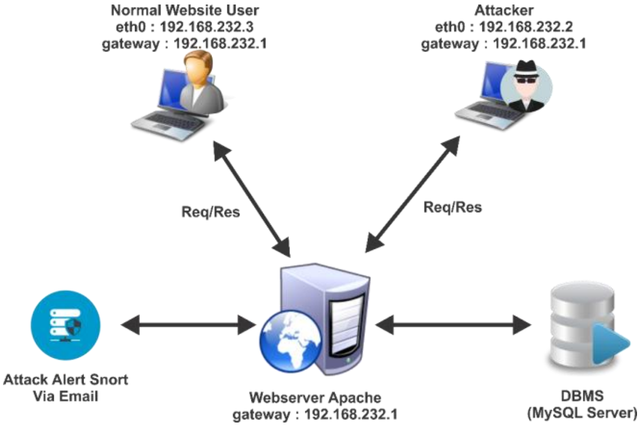
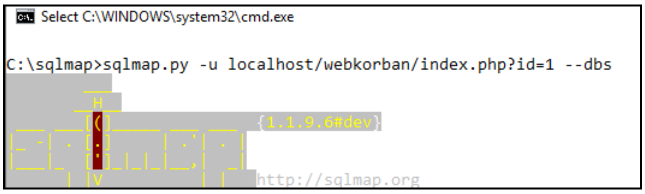


Figure 2. SQL Injection Attack Experiment Scenarios

Phase 1 : information gathering

This tool has a command line interface and with a series of commands one can retrieve the data from the database and take over the web server of the application. The command shown in the Figure 3 is used to retrieve data like the database names, the server it is running on, the operating system etc.[9]

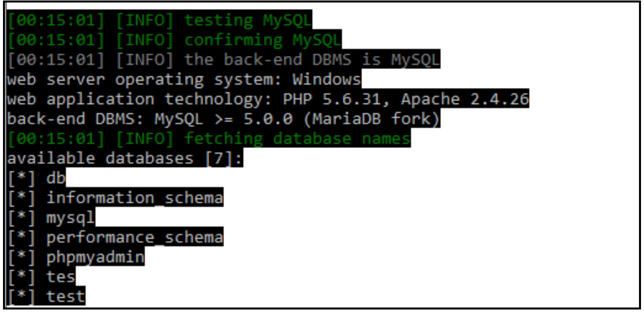

Select C:\WINDOWS\system32\cmd.exe

```
C:\sqlmap>sqlmap.py -u localhost/webkorban/index.php?id=1 --dbs
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.1.9.6#dev}
|_ -| . [.]     | .'| . |
|___|_  [:]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org
```

Figure 3. command used to gather information

Phase 2. Database system disclosure

In order to carry out an exploit on a web application, gathering the information about the server is said to be crucial and plays an important role in exploiting the victim's data.[10] The command to gain information about the database server, the command that is executed is `sqlmap.py –u localhost/webkorban/ index.php?id=1 --dbs`. This command returns the information like the back-end DBMS, the server running on the web application, etc.



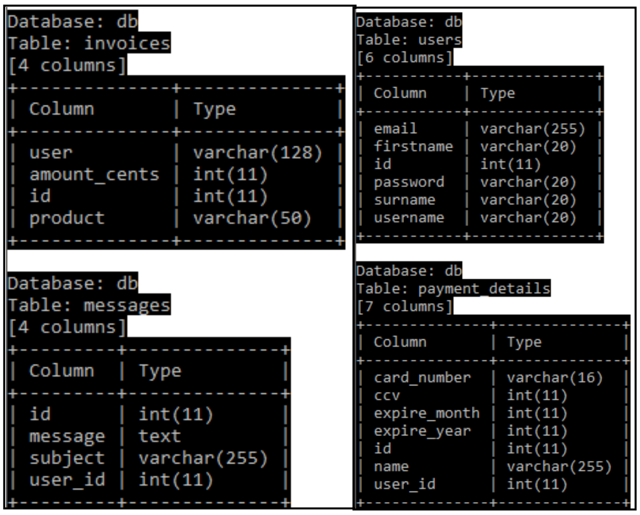
```
[00:15:01] [INFO] testing MySQL
[00:15:01] [INFO] confirming MySQL
[00:15:01] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.6.31, Apache 2.4.26
back-end DBMS: MySQL >= 5.0.0 (MariaDB fork)
[00:15:01] [INFO] fetching database names
available databases [7]:
[*] db
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] tes
[*] test
```

Figure 4. retrieving server information and return database

Here the sqlmap directs a series of payloads into the input fields and finds the compatible payload that escape clause the input parameter and fetches data as shown in Figure 4.

Phase 3. Exploiting tables

Database that would contain crucial information like credit card details, social security numbers, usernames, passwords etc. Here in this case we discovered 7 databases, so we had to screen all the databases and we did the scanning in the sequential manner.[11] The first database here was db. Hence the database can be skimmed for tables. Figure 5 is result for the command to exploit tables in a database on the sqlmap tool is `Sqlmap.py -u localhost/webkorban/index.php?id=1 -D db --tables`.



```
Database: db
Table: invoices
[4 columns]
+--------------+--------------+
| Column       | Type         |
+--------------+--------------+
| user         | varchar(128) |
| amount_cents | int(11)      |
| id           | int(11)      |
| product      | varchar(50)  |
+--------------+--------------+
```

```
Database: db
Table: users
[6 columns]
+-----------+--------------+
| Column    | Type         |
+-----------+--------------+
| email     | varchar(255) |
| firstname | varchar(20)  |
| id        | int(11)      |
| password  | varchar(20)  |
| surname   | varchar(20)  |
| username  | varchar(20)  |
+-----------+--------------+
```

```
Database: db
Table: messages
[4 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| id      | int(11)      |
| message | text         |
| subject | varchar(255) |
| user_id | int(11)      |
+---------+--------------+
```

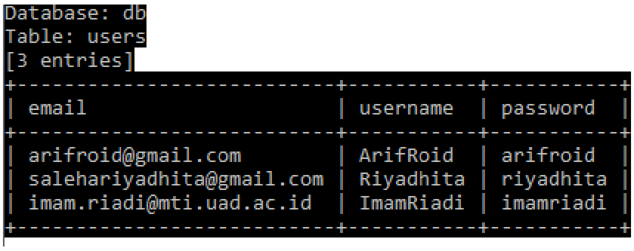
```
Database: db
Table: payment_details
[7 columns]
+--------------+--------------+
| Column       | Type         |
+--------------+--------------+
| card_number  | varchar(16)  |
| ccv          | int(11)      |
| expire_month | int(11)      |
| expire_year  | int(11)      |
| id           | int(11)      |
| name         | varchar(255) |
| user_id      | int(11)      |
+--------------+--------------+
```

Figure 5. displaying table and column name

Phase 4. Dumping column value

The attacker would retrieve value for personal information. The user table has the id, password and username values of the application. The command to dump column values is `Sqlmap.py -u localhost/webkorban/index.php?id=1 -D db -T user -C email,username,password -dump`. This command dumps the column values in the respective table and displays it as shown in Figure 6. In the same fashion, the passwords are retrieved as well and Then the access to the website is granted with the admin credentials.[12]

Figure 6 shows the experiment result from dumping user table that contains email, username and password.



```
Database: db
Table: users
[3 entries]
+--------------------------+-----------+-----------+
| email                    | username  | password  |
+--------------------------+-----------+-----------+
| arifroid@gmail.com       | ArifRoid  | arifroid  |
| salehariyadhita@gmail.com | Riyadhita | riyadhita |
| imam.riadi@mti.uad.ac.id | ImamRiadi | imamriadi |
+--------------------------+-----------+-----------+
```

Figure 6. dumped email, username, and password.

3.4 Intrusion Detection System (IDS) Snort Configuration

Configuration phase Intrusion Detection System (IDS) Snort performed to detect any demand (request) data, either by request or attack. after configuring snort, then the next rule configuration in accordance with the rules that have been owned by the snort to detect attack of SQL Injection as shown in Figure 7.[13]

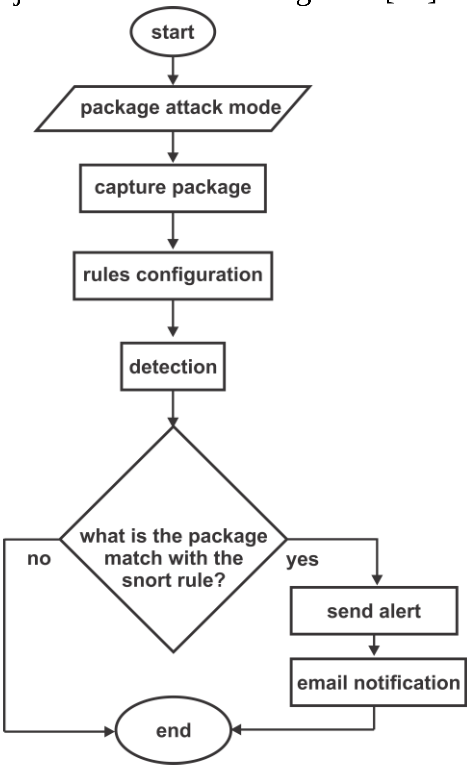


Figure 7. flowchart the proposed architecture

Figure 7 shows the detection system workflow on snort and warning in the form of email notification on a network that is attacked by SQL Injection attack. This attack detection stage consists of several processes:[14]

a. System is given SQL Injection attack packets.

b. The SQL Injection attack package is detected by the IDS snort system and processed by the detection engine.

c. The attack packet is checked and matched by the snort rule whether it includes SQL Injection attacks or just normal traffic.

d. The attack packet matches the rule, the packet will be detected as an attack, otherwise it is just normal traffic.

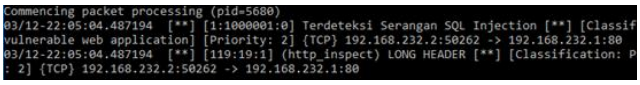
e. Data packets will be detected as attack and stored traffic data as logs.

f. then processed by Folder Watchdog to be forwarded as notification via email.

The experiments of the proposed work has been described in this section. We have created a web server whose data is secured with ASCII.[15] For that, id and password are stored in ASCII also. In web server SNORT is implemented which has a rule to alert about SQL Injection from any IP. Program list below shows the rules to detect the incoming SQL injection attack.

```
Alert   tcp   any   any   ->   any
$HTTP_PORTS     (msg:     "Terdeteksi
Serangan SQL Injection";
flow:to_server,established;
content:"User-Agent|3A|Sqlmap";
fast_pattern:only;      http_header;
metadata:servicehttp;
reference:url,Sqlmap.sourceforge.ne
t;classtype:web-application-
activity; sid:1000001;)
```

Figure 9 show the log generated for SQL Injection by SNORT.



```
Commencing packet processing (pid=5680)
03/12-22:05:04.487194  [**] [1:1000001:0] Terdeteksi Serangan SQL Injection [**] [Classif
vulnerable web application] [Priority: 2] {TCP} 192.168.232.2:50262 -> 192.168.232.1:80
03/12-22:05:04.487194  [**] [119:19:1] (http_inspect) LONG HEADER [**] [Classification: P
: 2] {TCP} 192.168.232.2:50262 -> 192.168.232.1:80
```

Figure 9. SQL injection logged by SNORT rule

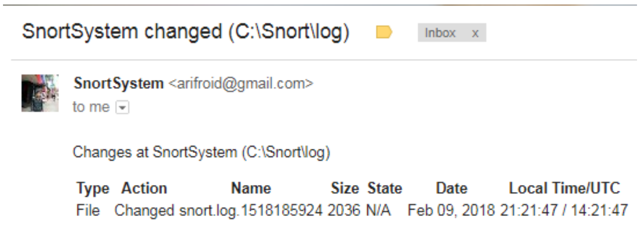
When the attacker enters the SQL Injection and it is encountered then the attacker is being transferred to log file. Now in the log file, the log of the attacker is traced and folder watchdog 2 triggered to sent mail to administrator to alert them.[16]

3.4 Data Collection

Collection of evidence in this study using the recording of IDS traffic. The reconstruction process starts after the internet-connected web server sends a notification via email using Watchdog Folder 2 which has supervised the log folder, the snort log changes because the Snort IDS captures the traffic that is considered a predefined rule. log files appear in the Snort folder directory in c: \ snort \ log. The database is connected and connected to the internet. Then the switch is connected to the server IDS (Intrusion Detection System) Snort, so that if there is anomaly traffic that will be directly detected by the snort and the notification appears. The Watch Dog folder serves as the notification sender when there is a log change to the folder that the Snort log is being monitored.

3.5 Mitigation

Mitigation is a series of efforts to reduce risk. After the Snort log files are recorded, the log file will be taken and analyzed using Wireshark to have this forensic evidence. When detected, the Snort rules will give a warning message in the alerts as shown in Figure 9. And also, folder watchdog to alert administrator via email as shown in Figure 10.[17]



SnortSystem changed (C:\Snort\log)

SnortSystem <arifroid@gmail.com>
to me

Changes at SnortSystem (C:\Snort\log)

Type	Action	Name	Size	State	Date	Local Time/UTC
File	Changed	snort.log.1518185924	2036	N/A	Feb 09, 2018	21:21:47 / 14:21:47

Figure 10. alert using email

Email notification system is used To send alerts in the form of email that there is an attack into the web server. Email is the fastest and most appropriate medium used for alerts. From the simulation of attacks that have been done into the web server network, it will bring up a log file followed by an email notification. Set the recipient's email address as well as view and set the email notification log to be sent to the recipient's email address. An example of email notification from log snort can be seen in Figure 10. The image explains that there has been a log change on the Snort IDS with the name snort.log.1518185924 on February 29, 2018 at 21:21. The email is sent instantly in the event of SQL Injection attack on the web network. the speed of internet connection can be a factor of email speed sent by the watch dog folder residing in the system.

The analysis continued with statistics module endpoint in Wireshark used to collect attack packets contained in log files Intrusion Detection System (IDS) Snort during the attack simulation.[18]

4 Result Analysis

4.1 User Acceptance Test (UAT)

User Acceptance Test is a user testing process that is intended to produce a document that serves as proof that the software has been developed acceptable by the user, if the test can be considered to meet the needs of the user.

Testing by performing a SQL Injection attack simulation process before and after the mitigation process on the website provided by the researcher. There are 10 simple questionnaire lists on the system, 10 questions and results from testing before and after the mitigation. Assessment categories used for the questionnaire include:

a. Disagree (TS)
b. Less Agree (KS)
c. Agree (S)
d. Strongly Agree (SS)

There are 10 simple questionnaire lists on the system, 10 questions and results from testing before and after the mitigation

a. The Snort IDS system can run when a program call is made.
b. Snort IDS System can monitor traffic from IP Address under supervision.
c. Snort IDS system can detect SQL Injection attacks.
d. Snort IDS system can bring up logs when SQL Injection attacks occur.
e. Snort IDS system can display real time alerts on system commands when SQL Injection attacks occur.
f. Log IP Address can be a source of information from the attacker.
g. Log port source can be a source of information from the attacker.
h. Snort IDS system can protect from SQL Injection attacks.
i. Snort IDS system can block the attacker's IP Address.
j. Snort IDS system can send email alerts to managers email (Administrator).

After the above questionnaire is given to participants before and after mitigation, then the questionnaire data is processed to get the result of User Acceptance Test. For user acceptance test data can be seen in the attachment. From the assessment results of User Acceptance Test testing can be drawn conclusions are:

a. The system user who has chosen Disagree (TS) before mitigation gets 35% and after mitigation gets 0%.
b. System users who have selected Less Agree (KS) before mitigation get 48% and after mitigation get 10%.
c. System users who have selected Agreement (S) before mitigation score 17% and after mitigation score of 44%.
d. System users who have chosen Strongly Agree (SS) before mitigation get 0% and after mitigation score of 46%.

5 Conclusion

The conclusions that have been obtained during the research process in detecting SQL Injection attacks on the web server concludes that the implementation of Snort's Intrusion Detection System (IDS) on the web server can be used to help provide SQL Injection attack detection information by utilizing special SQL Injection rules that are implemented by the Intrusion Detection System (IDS) Snort. Log files are taken from snort to analyze illegal action activities that occur in the web server environment based on log file analysis. Based on the implementation of the Intrusion Detection System (IDS) snort to detect SQL Injection attacks in this study can be used properly and

REFERENCES

1. Mualfah, D., & Riadi, I. (2017). Network Forensics For Detecting Flooding Attack On Web Server, *15*(2), 326–332.
2. Riadi, I., Istiyanto, J., & Ashari, a. (2014). Log Analysis Techniques using Clustering in Network Forensics. *International Journal of Computer Science*, *10*(7).
3. Jannah, M., Hustinawati, & Wildani, R. (2012). Implementation of Intrusion System Snort in Computer Network Laboratory. *UG Jurnal*, *6*(5), 1–4.
4. (EC-Council), I. C. of E.-C. C. (2012). *SQL Injection. Certified Ethical Hacker V8.00*.
5. Dahlan, M., Latubessy, A., Nurkamid, M., & Anggraini, L. H. (2015). Testing And Analysis Website Security Against SQL Injection Attack (Case Study: UMK Website), *7*(1), 13–19.
6. Gudipati, V. K., Venna, T., Subburaj, S., & Abuzaghleh, O. (2017). Advanced automated SQL

injection attacks and defensive mechanisms. *2016 Annual Connecticut Conference on Industrial Electronics, Technology and Automation, CT-IETA 2016*. https://doi.org/10.1109/CT-IETA.2016.7868248

7. Harjono, & Wicaksono, A. P. (2014). Intrusion Detection System with Snort, *III*(1), 31–34.

8. Syafitri, W. (2016). Information Security Risk Assessment Using the NIST Method 800-30 (Case Study: Academic Information System of XYZ University). *CoreIT*, *2*(2), 8–13.

9. Ping, C., Jinshuang, W., Lin, P., & Han, Y. (2016). Research and Implementation of SQL Injection Prevention Method Based on ISR, 1153–1156.

10. Sonewar, P. P. A., & Thosar, P. S. D. (2015). Detection of SQL Injection and XSS Vulnerability in Web Application, (3), 16–21.

11. UtpalUpadhyay, & GirishKhilari. (2016). SQL Injection Avoidance for Protected Database with ASCII using SNORT and Honeypot, (978), 596–599.

12. Riadi, I., Muhammad, A. W., & Sunardi. (2017). Neural network-based ddos detection regarding hidden layer variation. *Journal of Theoretical and Applied Information Technology*, *95*(15), 3684–3691.

13. Mazdadi, M. I., Riadi, I., & Luthfi, A. (2017). Live Forensics on RouterOS using API Services to Investigate Network Attacks. *International Journal of Computer Science and Information Security (IJCSIS)*, *15*(2), 406–410.

14. Putri, R. U., & Istiyanto, J. E. (2012). Forensic Analysis Network Case Studies SQL Injection Attack on Gadjah Mada University Server. *Indonesian Journal of Computing and Cybernetics Systems*, *6*(2). Retrieved from http://journal.ugm.ac.id/index.php/ijccs/article/view/2157.

15. Syaimi, A., Utami, P., Lidyawati, L., & Ramadhan, Z. (2013). Design and Performance Analysis of Network Infiltration Prevention System Using Snort IDS and Honeyd.. *Jurnal Reka Elkomika ©TeknikElektro | Itenas Jurnal Online Institut Teknologi Nasional Jurnal Reka Elkomika*, *1*(4), 2337–439.

16. Voitovych O.P., O.S., Y., & L.M., K. (2016). SQL Injection Prevention Cheat Sheet. *2016*, 1. Retrieved from https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet.

17. Editya, A. S., Sumpeno, S., & Pratomo, I. (2017). Performance of IEEE 802 . 14 . 5 and ZigBee protocol on realtime monitoring augmented reality based wireless sensor network system, *3*(2), 90–97.

18. Zulkifli, M. A., & Dahlan, U. A. (2018). Live Forensics Method for Analysis Denial of Service (DOS) Attack on Routerboard, *180*(35), 23–30.