Digital Forensic InnoDB Database Engine for Employee Performance Appraisal Application

Daniel Yeri Kristiyanto[1,*], *Bambang* Suhartono[2] and *Agus* Wibowo[1]

[1]Department of Computer Systems, School of Electronics and Computer Science, Semarang - Indonesia
[2] Department of Electrical Engineering, School of Electronics and Computer Science, Semarang – Indonesia

Abstract. Data is something that can be manipulated by irresponsible people and causing fraud. The use of log files, data theft, unauthorized person, and infiltration are things that should be prevented before the problem occurs. The authenticity of application data that evaluates the performance of company employees is very crucial. This paper will explain how to maintain the company's big data as a valid standard service to assess employee performance database, based on employee performance of MariaDB or MySQL 5.0.11 with InnoDB storage engine. The authenticity of data is required for decent digital evidence when sabotage occurs. Digital forensic analysis carried out serves to reveal past activities, record time and be able to recover deleted data in the InnoDB storage engine table. A comprehensive examination is carried out by looking at the internal and external aspects of the Relational Database Management System (RDBMS). The result of this research is in the form of forensic tables in the InnoDB engine before and after sabotage occurs.

Keywords: **Data log; data analytics; artefact digital; sabotage; InnoDB; penetration test.**

1 Introduction

At present almost all companies have a database system [1]. Information is formed quickly and is a basic requirement of a company [2, 3]. Companies that already have a computer-based information system (CBIS) have stored their information into a database management system. This information is crucial and is used as the main tool to make decisions about the performance of employees in the company [3]. Results and information on forensic databases can be used for several reasons [4, 5]. First, find out how safe the data in the company is stored. The company wants to know whether certain privileges can be violated or even damaged related to the database they have. Second, the results of forensic databases can be used to prevent unwanted events. Early detection and analysis of attacks on the database must be able to be analyzed. Third, companies that have databases that contain employee data certainly concern the interest of many people's privacy, thus authentic verification of each employee's data is needed for a more specific purpose, namely being able to guarantee correct and fair data.

File systems are very important for data integrity, performance, and ease of administration. The integrity of data in a database is absolute because all forms of corrupt data must be prevented [5]. Database performance aspects are seen from the speed and stability of processing large companies' data as well as the ease of data administration that influences database performance. The file system in the employee appraisal research uses NTFS (New Technology File System). NTFS is very good for digital forensic analysis with extensive application support [5, 6].

Analysis of the data contained in a database, it requires deep knowledge of the data which is to know the data is formed, created and manipulated [7, 8]. The expectation of the results of the knowledge is a clear, consistent and explicit statement regarding the result of the analysis. The main objective of this research is to be able to analyze the MariaDB or MySQL database that uses the InnoDB engine with the main function to asses employee performance. MariaDB is a multiuser database that has one daemon, so it does not have wrapping as found in Apache + suexec/cgiwrap. Users who use the MariaDB database have access to all databases contained in the data directory, namely /mysql/data. MariaDB and MySQL already have excellent access privilege systems, but bugs in MariaDB code or even configuring the privilege system can potentially open a database containing important secrets that should have limited access. The in-file and out-file data in the MariaDB database are two things that can become security loopholes by replacing the database remotely.

This study explains the structure and architecture of a database that is owned by a company. The CBIS in the form of employee performance appraisers. Data that is owned by the company will be reconstructed by looking at the cluster file system. The results of digital forensic analysis are used to detect whether there is sabotage or not.

* Corresponding author: daniel.jerry182@gmail.com

The first discussion in this paper is about internal checks on MariaDB, settings used in MariaDB, access privileges, connection control checks, chroot checking, local in-file data load checking, and SSL-based connection checks. The second is a simple SQL-table internal explanation and looks for the relevance between keys, data types and other components of the database compiler. The third is the identification of acts of sabotage against the database.

2 Research Method

MariaDB is the most popular open source RDBMS server. MariaDB is a free version of MySQL, so both have the same database engine. MariaDB database security analysis used for corporate performance app assessment. It is analyzed from several aspects, which according to the author are crucial from a data security perspective.

2.1. External Investigation

2.1.1 Investigation the Installation of Database Engines

Installation of the MariaDB database engine in the employee performance app assessment in this study was analyzed by looking at the machines installed separately or not separately[5]. The potential for sabotage of engine installations is analyzed through various open ports including internet ports (80) and (3306). Furthermore, the investigation is intended to see the "Shell" configuration for each user whether the default or has a private configuration.

2.1.2 Access Privilege

Privileged access systems are very important for a digital forensic study [9]. Allegations of a case of sabotage will be directed into this passage. Employee appraisers in this study will analyze the access rights of each user, analyzing the user is very important to find sabotage agents, because sabotage is usually carried out by people who know for sure the information system used by the company. Ideally, for the security of shared hosting servers, each administrator is only permitted to access one database [10].

2.2 Internal Investigation

The internal investigation includes checking the database engine which is the default used by MariaDB [4]. MariaDB stores all information on each table into the .frm file. The filename is created automatically when a table is formed through the create_frm () function in the file /sql/table.cc. Each .frm file in the table by default has a limit of 4 GB if this limit is exceeded then the MariaDB engine will truncate to prevent overflow or error. Forensic data analysis of the application of the employee performance appraisal website in this study adopted an integrated digital forensic process model framework adapted to research.

The researcher acquired the NTFS file system where the MariaDB database was placed using digital forensic applications. Cluster acquisition in the database application performance appraisal employees have the main target, namely information from the tables in the database include general table information, primary keys, field definitions, data storage checks.

The internal investigation includes checking the database engine which is the default used by MariaDB. MariaDB stores all information on each table into the .frm file. The filename is created automatically when a table is formed through the create_frm () function in the file /sql/table.cc. Each .frm file in the table by default has a limit of is exceeded. MariaDB engine will truncate to prevent overflow or error. Forensic data analysis of the application of the employee performance appraisal website in this study adopted an integrated digital forensic process model framework adapted to research.

The researcher acquired the NTFS file system where the MariaDB database was placed using digital forensic applications. Cluster acquisition in the database application performance appraisal employees have the main target, namely information from the tables in the database include general table information, primary keys, field definitions, data storage checks.

2.3 Identification of Sabotage

Sabotage is an action that is planned, organized by a person or organization intentionally with a particular target area to cause damage, loss, original data changes, and destruction both physically and non-physically [11]. The database is a source of company activity. To identify database sabotage that passes through a network, the network traffic of the researcher reads the traffic pattern, if abnormal indications are seen on IP seen as private addresses using NAT, the IP does not appear frequently in the log, time-stamp is too close, source port and the sequence number that rises simultaneously is this guide used by researchers to carry out forensic analysis with suspected attacks on the MariaDB server.

Changes to database files are checked based on the connection between binary log files and the user record. This check is needed to prove the consistency of all tables in the database based on transaction ID in each record, if there is a change in data not within the reporting range, which should be based on log databases and network logs, it is matched with the company's routine reporting, and then acts of sabotage are ensured.

3 Result and Discussion

3.1. Knowledge Based Digital Forensic

Information gathering and observation are the first steps taken to complete digital forensic analysis of InnoDB database engine application to evaluate employee performance. The purpose of this study is to detect

sabotage of the database through the InnoDB engine that passes through computer networks.

Validation of the findings of forensic analysis was confirmed to the company and how to test and validate the act of sabotage. Correction and analysis results for digital forensic applications for evaluating employee performance under the supervision of digital forensic experts, when the InnoDB engine database is declared sabotage, the report is forwarded to the company for further legal action with evidence of digital forensic data.

3.2. Results of External Investigations

The installation of the MariaDB server in the employee performance app assessment in this study was not done separately, so the MariaDB daemon named mysqld is in the same engine. The potential for sabotage against the installation of this engine is through various open ports including internet ports (80). The shell for each user still uses the default, so the intruder can use this weakness to sabotage through shell access. Databases have great potential for sabotage by accessing database files and logs in the /mysql/data.

The results of the examination of privileged access were found that the app appraisers of employees in this study had different access but were still low regarding the understanding of data security threats, as evidenced by the privileged access “select” that is owned by each database admin without limiting privileges to file or process. Ideally for the security of shared hosting servers, each administrator is only permitted to access one database so that the findings of privileged access to the employee performance app are as follows: user: (‘localhost’, ‘user’, password (‘PASSWORD’), ‘N’, ‘N’, ‘N’, ‘N’, ‘N’, ‘N’, ‘N’, ‘N’, ‘N’). The table entry in the database has settings: db (‘localhost’, ’DATABASE’, ‘USER’, ‘Y’,’Y’, ‘Y’,’Y’, ‘Y’,’Y’, ‘Y’,’Y’, ‘Y’,’Y’). This arrangement can be used by intruders to sabotage. Users can view databases that exist on server machines because they have privileged grants and file privileges.

3.3 Results of Internal Investigations

Internal database investigation MariaDB requires knowledge to interpret every bit contained in each table in the database [4, 8]. Digital forensic analysis is not possible without a good understanding of the properties of the MariaDB table [12]. Internal checks on the .frm file table files that are created automatically when a table is formed through the create_frm () function in the /sql/table.cc file through a series of processes. The first process is the cluster acquisition process in the NTFS file system where the MariaDB database is located. The internal inspection file system is an in-depth examination to find log files and digital artefacts. Both will be used as evidence for acts of sabotage database files.

Process of Acquiring the MariaDB database on the NTFS file system will produce an image file. The images file is used as evidence and digital forensic based analysis [4, 5, 13]. The acquisition of image files called bitstream is implemented to maintain valid evidence according to digital forensic provisions [14]. The bitstream technique copies the original bit-by-bit files that are in the system file in the form of binary numbers. The main purpose of bitstream techniques in the digital analysis of employee performance applications is to find hidden files, temporary files, files before overwriting and defragmented files [13].

Fig. 1. The Process of Acquisition of MariaDB in the NTFS File System

This study uses computer specifications: Acer Veriton S680G, Processor: Intel® Core™ i5 CPU 650@3,20 GHz (4 CPUs), RAM: 8192 MB, Operating System: Windows 10 Enterprise 64 bit (10.0, Build 16299), Direct X 12. Tool digital forensic: Stellar Log Analyzer for MySQL, FtK (Forensic toolkit) Imager. The website specifications tested are: web based on PHP 5 with the Model View Controller (MVC) framework Codeigniter. The database specifications tested are MySQL database mysqlnd 5.0.11-dev – 20120503 with the InnoDB database engine.

This Study examines ten (10) main tables contained in the application database assessing employee performance. Digital forensic analysis is done by combining the results of the analysis using forensic software and manual analysis to comprehensively examine the contents of a table. The examination of each table is done by checking the .frm file that is owned. The file has a binary identity that is very important to be examined more deeply about the potential for sabotage that occurs. So that digital forensic analysis of applications to assess employee performance are shown in Fig 2.

```
000 FE 01 0A 0C 12 00 56 00-01 00 0A 0C 00 00 CB 05 þ·····V·····Ë·
010 52 03 00 00 00 00 00 00-00 00 00 02 50 00 09 00 R···········P···
020 00 05 00 00 00 00 08 00-00 00 00 00 00 00 00 CB ···············Ë
030 05 00 00 20 87 01 00 10-00 00 00 00 00 00 00 00 ··· ············
040 00 10 F8 2F 76 DD 5A 9D-11 E8 AF 15 10 78 D2 EE ··ø/vÝZ··è¯··xÒî
050 8B 5B 83 09 00 00 03 03-00 00 17 00 00 00 04 00 ·[··············
060 01 00 00 00 01 80 01 00-00 1B 00 04 00 28 00 17 ·············(··
070 00 01 00 00 00 09 80 F5-02 00 00 40 17 00 28 00 ·····õ···@··(·
080 0A 00 01 00 00 00 02 80-05 00 00 00 40 0A 00 FF ·····€·····@··ÿ
090 50 52 49 4D 41 52 59 FF-75 73 65 72 6E 61 6D 65 PRIMARYÿusername
0a0 FF 6E 69 6B FF 00 00 00-00 00 00 00 00 00 00 00 ÿnikÿ···········
```

Fig. 2. Hexadecimal Structure of Pengguna.frmTables

Based on the hexadecimal structure in the .frm file the application table needs an in-depth examination so that the act of sabotage can be known with evidence that is following the digital forensic standards. Examination of these tables has the main purpose of finding table manipulation activities in the application database evaluating employee performance. Analysis of .frm table applications is as follows:

Table. 1. Explanation of Pengguna.frm Tables

Offset	Length	Value	Meaning	Cursor Pos
0x00	1	FE	fixed value	0
0x01	1	01	fixed value	1
0x02	1	0A	FRM_VERSION (/include/mysql_version.h + 3 + test(create_info->varchar)	2
0x03	1	0C	Database type (sql/handler.h Z. 258-279) z.B.: DB_TYPE_MYISAM: 9, DB_TYPE_INNODB: 12	3
0x04	1	12	unknown	4
0x05	2	00 56	unknown or	5
0x07	2	00 01	IO_SIZE (4096) Definition in include/my_global.h	7
0x09	2	00 74	unknown	9
0x0A	4	06 00 00 F9	keylength (IO_SIZE+key_length+reclength+ create_info->extra_size). Table 3 shows exact meaning of the key length (key_length)	11
0x0E	2	01 12	length of the temporary key, based on key_length	15
0x10	2	02 00	length of the record	17
0x12	4	00 00 00 00	create_info->max_rows (definition of the create_info structure HA_CREATE_INFO in	19
0x16	4	00 00 00 00	create_info->min_rows	23
0x1A	1	00	unused or padding / alignment	26
0x1B	1	02	fixed value (use long pack-fields)	27
0x1C	2	21 00	key_info_length	28
0x1E	2	09 00	create_info->table_options	30
0x20	1	00	fixed value	32
0x21	1	05	fixed value (frm version number: 5)	33
0x22	4	00 00 00 00	create_info->avg_row_length	34
0x26	1	08	create_info-	38
0x27	1	00	create_info-	39
0x28	1	00	create_info->row_type	40
0x29	6	00 00 00 00 00 00	RAID Support	41
0x2F	4	00 00 F9 01	key_length	45
0x33	4	00 00 22 87	MARIADB_VERSION_ID (only saved to prevent a warning, because of unaligned key_length of 3 bytes)	49
0x37	4	01 00 10 00	create_info->extra_size (length of extra data sequence)	53
0x3B	2	00 00	extra_rec_buf_length (reservation)	57
0x3D	1	00	default_part_db_type (reservation)	59
0x3E	2	00 00	create_info->key_block_size	62

Each table analyzed is intended to examine security violations in the form of sabotage. Data in the table that changes before the time of reports collection is considered as an act of infiltration or sabotage of the application database assessing employee performance. An explanation of the structure of the table is very necessary for forensic purposes because it is impossible for digital forensics to be done without knowing about the structure of the table.

The xamination of each table is continued by looking at each structure and data type used. The data type used in application tables evaluates employee performance using integers, varchar, and enum. A table that has a technical primary key will have a hexadecimal

arrangement that is almost similar to each other, especially if the table has many fields [4]. Digital forensic analysis is carried out to ensure by examining tables that have a primary key. Each table arranged in this study has a different structure but is fixed since the application is used by the company so it is very necessary to look at the entire structure of tables and databases carefully, so that intrusion detection can be done.

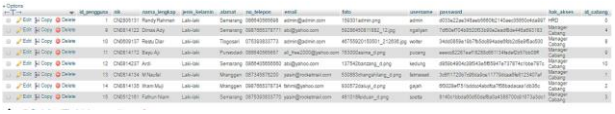

Fig. 3. Record of Pengguna Tables

The table structure in the performance app is accessible through Cpanel owned by MariaDB. Digital forensic identification is done is to make sure the table structure does not change according to the default application, to be able to check the composition of the field; a hexadecimal check is used to obtain clear results.

Based on hexadecimal checks to each table represented by Fig.4 can be seen that, each table has a primary key, in the picture of primary key structure starts from position 0x00, cursor pos = 0, cluster = 10884595, log section = 87076760 to 0x80, cursor pos = 144, cluster = 10884595, log section = 87076760. In general, the log section range for frm table Employee is 87076760-87076764. Each table examined in the employee performance app appraisers has a unique record in the form of cursor posts, clusters and log sections that are different but are constant when inputted by each branch manager.

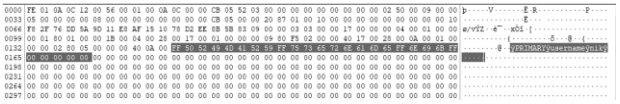

Fig. 4. Hexadecimal PrimaryKey Investigation of Pengguna

Furthermore, information about hexadecimal structures in employee performance appraisal application tables is used as a database comparison table that has been sabotaged. The information needed after the table key is the header file. The header file has important information, checksum, and offsets [4]. The header file has a key header that is intended to check the data storage allocation around the data file. Checking header files containing key header is done using the perfect copy technique. Perfect copy is a term in digital forensic science to save digital evidence. In the analysis the comparative data has a key header like the following:

Table. 2. Key Definition of Pengguna

Offset	Len	Value	Meaning		Log Sec
0x04	2	0C 12	KEY HE	key->flags ^ HA_NOSAME dupp key and pack fields	3
0x06	2	00 56		key->key_length length of the key	5
0x08	1	00	ADER	key->key_parts sum or number of key parts	7
0x09	1	01		key->algorithm key algorithm	8
0x0A	2	00 8E		key->block_size	9
0x0B	3	09 00 00		unused (padding, alignment)	11
0x0E	2	E2 03	KEY PARTS	key_part->fieldnr+1+FIELD_NAME_USE D	47
0x80	2	52 49		offset (key_part->offset+data_offset+	128
0x83	1	4D		fixed value (sorting order)	130
0x84	2	41 52		key_part->key_type	131
0x86	1	59		key_part->length length of the key parts in bytes	133
0x87	1	FF		unused (padding, alignment)	134
0x88	1	6E	separator NAMES_SEP_CHAR (before name of key)		135
0x89	7	69 6B FF 00 00 00 00	name of key (PRIMARY)		136
0x1F	1	00	separator NAMES_SEP_CHAR (after name of key)		142

3.4 Penetration Sabotage InnoDB

The hardware in this study consisted of 1 Dell PowerEdge T-30 Server, TP-Link TL-SG1024D, a Cloud Core Router CCR1009-7G-1C-PC Router, and 4 Acer PC Clients as in Table 3. The software in this study used the MariaDB server, Windows Server 2003, Windows 7 Ultimate, Metasploid, IDA Pro as shown in table 4. The hardware and software in this study are used to simulate the threat of attacks into the database engine. Sabotage is done using exploits. In Fig.5, the topology describing sabotage can be done by planting an exploid on the database server. Node 4 with IP 192.168.2.62 is used as a sabotage actor by exploiting the database via network media to the server.

Penetrating tests are carried out using exploits. The way it works is to exploit weaknesses in the network system through open ports to attack user tables. Intruders can scan all ports, especially those that go to the database where the employee data is stored [15]. Furthermore, scanning on the network is to find the host and service for all computers on a network. The network scan results in identifying internal checks can be used by

intruders to sabotage the InnoDB database engine. Status host a network can be tracked using NMAP commands via the -sP (IP Range) parameter, then the intruder will infiltrate through the company's LAN network by seeing hosts that are "UP" [16]. After the IP server was found, an intruder who wanted to sabotage port scans in more detail by changing the parameters in NMAP, the -A-T4 (IP target server parameter), as shown in Fig.6. Penetration is done successfully obtaining complete information about the server where the employee assessment database is located. This information contains a port, protocol, status, database server information, the programming language version, and fingerprint service. The logical topology penetrating test can be seen in Fig.5.

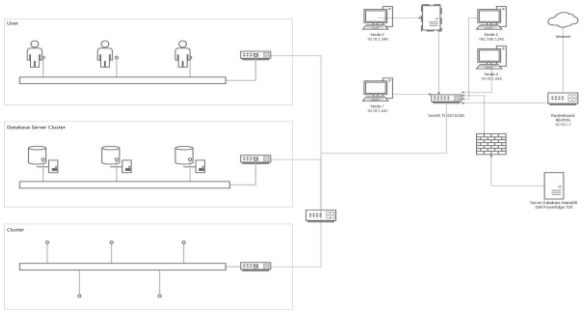

Fig. 5. The Logical Topology by Penetrating Test

```
C:\Users\daniel>nmap -sP 10.10.1.1-100
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-10 12:12 SE Asia Standard Time
Nmap scan report for 10.10.1.1
Host is up (0.0020s latency).
MAC Address: D4:CA:6D:D9:83:F5 (Routerboard.com)
Nmap scan report for 10.10.1.2
Host is up (0.0010s latency).
MAC Address: 84:16:F9:A9:19:18 (Tp-link Technologies)
Nmap scan report for 10.10.1.16
Host is up (0.044s latency).
MAC Address: 30:B5:C2:B8:B7:61 (Tp-link Technologies)
Nmap scan report for 10.10.1.26
Host is up (0.093s latency).
MAC Address: 8C:C5:E1:45:C2:9D (ShenZhen Konka Telecommunication Technology)
Nmap scan report for 10.10.1.79
Host is up (0.0020s latency).
MAC Address: 00:21:97:66:98:10 (Elitegroup Computer Systems)
Nmap done: 100 IP addresses (5 hosts up) scanned in 8.36 seconds
```

Fig. 6. Scan Host on the Network

Information about the MariaDB server is known as a port and database version, and then sabotage is done using Metasploit. Metasploit was chosen because of its effective ability to enter the server security system and firewall. The trial is done by releasing the AUX module on Metasploit, the purpose of using the AUX module is to find out the database version on the target server, then penetrate the MariaDB login and username, by dumping the username and password module used is PAYLOAD. In the next step, the researcher sets the username and password on the target server, then resumes sending the EXPLOITS module.

Intruders do sabotage to have a target username and password manager or super user [15, 17]. Table "pengguna", table "karyawanterbaik_periode", table "nilaikaryawan", table "karyawan" is the main target in a database sabotage mechanism for employee app assessment, because in the table the location of all data managers is located. Forensic analysis of sabotage crime using exploid will be directed to the table assuming with the Metasploit technique then he will pretend to be the super user manager or admin. Intruders will target the username and password as the manager. The prediction will lead to the login module on MariaDB, which contains the accounts, Guest and Root. Metasploit has a module that can enumerate the database, and dump the hash username and password [15]. Sabotage is not done by defacing or injecting a database or cracking, but pretending to be one of the top management decision makers of a company. The intruder has the target to master the password and username. The username and password always have a hash and potentially be attacked by has dumping through the auxiliary module on metasploit. The workings of the Metasploit and module are shown in Fig.7, and the Metasploit command line is shown in Fig.8

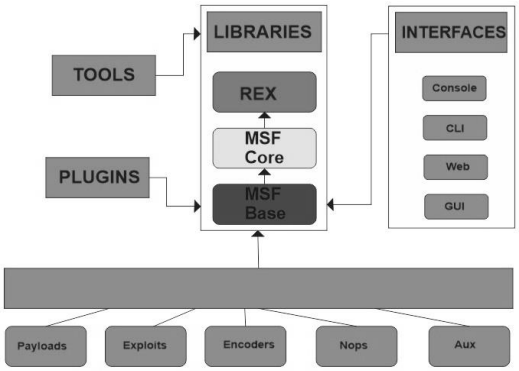

Fig. 7. Metasploit Framework

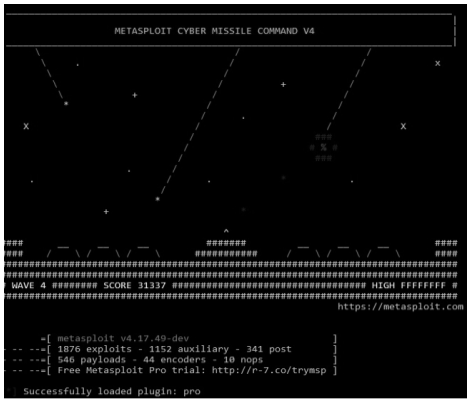

Fig. 8. Metasploit Command Line

3.5 Result of Sabotage Database Engine Analysis

Based on the database environment settings used in the employee performance app assessment, penetrating the test on the MariaDB database can be done using exploits. The first stage in testing is to examine the internal and external database environments that are used for employee performance appraisers, including the table structure in the database. The second stage validates security threats through potential users. Testing and validating processes are carried out using forensic tools and database penetration tools. The database penetration tool is used to sabotage the best employee performance achievement results, so that the employee score is changed which leads to the award received by each

employee. Metasploit is a database penetration tool that is compatible with the sabotage mechanism. The sabotage mechanism is made as if the final result of determining the best employee of the company is "best employee id=17" whereas the results before sabotage "best employee id=56", then sabotage is made as if the decision was approved by the manager area with NIK = CN2805131, so the selection decision id = 17 is valid. Sabotage uses database queries with the UPDATE command in the "nilai". "id_karyawan=56" is the target of sabotage. The "id_nilaikaryawan" field has several values based on the valuation variables that apply to the company. The assessment variables are attendance, late, report, recap, print log, and stock opname". Before the act of sabotage was carried out "id_karyawan=56" namely "Monica" has the highest value so that "Monica" is the employee with the best value of all branches, when penetration is done "Monica" is replaced by "id_karyawan=17" with the name "Lestari" as seen in Fig.9 and Fig.10.

```
MariaDB [spkcampusnet]> select * from nilaikaryawan where id_karyawan = '56';
+------------------+------------+-------------+-------------+-------+
| id_nilaikaryawan | id_periode | id_karyawan | id_kriteria | nilai |
+------------------+------------+-------------+-------------+-------+
|              223 |          3 |          56 |           1 |     5 |
|              224 |          3 |          56 |           2 |     1 |
|              225 |          3 |          56 |           3 |     5 |
|              226 |          3 |          56 |           4 |     5 |
|              227 |          3 |          56 |           5 |     1 |
|              228 |          3 |          56 |           6 |     1 |
+------------------+------------+-------------+-------------+-------+
6 rows in set (0.00 sec)

MariaDB [spkcampusnet]> update nilaikaryawan set nilai='2' where id_nilaikaryawan='223';
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [spkcampusnet]> update nilaikaryawan set nilai='2' where id_nilaikaryawan='225';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Fig. 9. Table Record before Sabotage

```
MariaDB [spkcampusnet]> update nilaikaryawan set nilai='2' where id_nilaikaryawan='223';
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [spkcampusnet]> update nilaikaryawan set nilai='2' where id_nilaikaryawan='225';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [spkcampusnet]> update nilaikaryawan set nilai='2' where id_nilaikaryawan='226';
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [spkcampusnet]> select * from nilaikaryawan where id_karyawan = '56';
+------------------+------------+-------------+-------------+-------+
| id_nilaikaryawan | id_periode | id_karyawan | id_kriteria | nilai |
+------------------+------------+-------------+-------------+-------+
|              223 |          3 |          56 |           1 |     2 |
|              224 |          3 |          56 |           2 |     1 |
|              225 |          3 |          56 |           3 |     2 |
|              226 |          3 |          56 |           4 |     2 |
|              227 |          3 |          56 |           5 |     1 |
|              228 |          3 |          56 |           6 |     1 |
+------------------+------------+-------------+-------------+-------+
6 rows in set (0.00 sec)
```

Fig. 10. Table Record after Sabotage

The sabotage was continued by convincing the area manager with "NIK=CN2805131" to print the results. The next target is a table that stores the final assessment data. The target table is "arsip_laporan" which has a "nama_file" field that stores Portable Document File (.PDF) files, which contains the final assessment of the results of the application, complete with the date. The date of being targeted is replaced because the date session is based on current computer time, so to launch the sabotage action, the "tgl_buat" field is made exactly the same as before sabotage occurred so that the report printed on the PDF file will be the same as the following picture Fig.11 and Fig. 12.

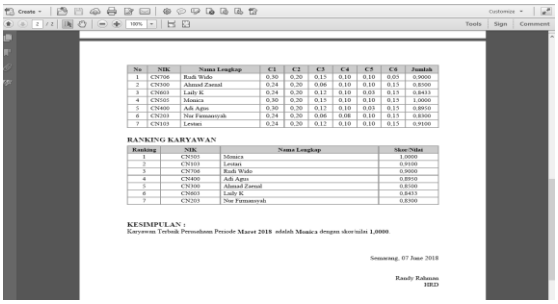

Fig. 11. PDF Files before Sabotage

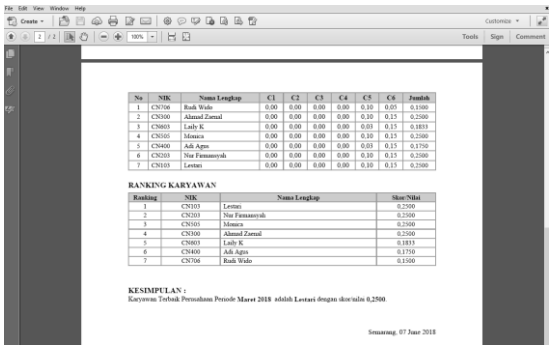

Fig. 12. PDF Files after Sabotage

3.6 Validation of the Sabotage Forensic Database Engine

Forensic results prove that the database engine used is InnoDB, which is shown in the type column. The InnoDB engine supports transactions such as commit, rollback and crash recovery. InnoDB stores a "Null" value into a place called "Placeholder" when a primary key has been defined. Offset 260 on cursor heading 616-621 has hexadecimal 49 6E 6E 6F 44 42. Six strings of hexadecimal values refer to the database engine, so it can be said that 49 6E 6E 6F 44 42 is an interpretation of the InnoDB database engine. All tables in the database evaluating the performance of employees of PT. Campus Media consistently has the same hexadecimal number.

```
260 00 00 00 00 00 00 06 00-49 6E 6E 6F 44 42 00 00  ........InnoDB..
270 00 00 00 00 B1 01 B1 01-00 00 00 00 00 00 00 00  ....±.±.........
```

Fig. 13. Hexadecimal InnoDB Engine

Forensic results of all table structures have not changed and remain in the structure as explained in table 1, table 2 and fig 2-4 and all tables in the database show hexadecimal structures from the "Pengguna" table. At offset 010 it can be seen that the table has 3 key attributes. Hexadecimal if there is a change in the primary key in the form of a key addition or key reduction, then the offset will change according to the number of primary keys. Three fields of three parts that become primary key markers are: : 28 00 17 00 01 00 00 00 09 80 F5 02, 28 00 0A 00 01 00 00 00 02 80 05 00, and 0A 00 FF 50 52 49 4D 41 52 59. The first byte of each hexadecimal block indicates the name of the field that can be seen at offset 010, but the results are inversely proportional to the records stored in the table. Some results were successfully sabotaged as seen in

Fig.9, Fig.10, Fig.11, and Fig.12. Digital artifacts were found to look for transactions that occurred, namely checking Post and Get.

The examination of the GET and POST methods has digital evidence in the form of digital artifacts that indicate unnatural queries in the table. Examination carried out on the Apache webserver by analyzing the log recorded between the website and the database the results obtained are as illustrated in Fig.14, the data in the table has been changed.

Fig. 14. Digital Artefact on the Application

4 Conclusion

The company data was successfully reconstructed through a file cluster system so that acts of sabotage can be known. Internal and external audits successfully examine sabotage actions that take advantage of the weaknesses of the privilege system, connection control, chroot, local in-file, and SSL based connections. The tables in the database of employee performance appraisers can be explained through the relevance of primary keys, data types, and other database compiler components. Sabotage is a planned crime that is very difficult to detect because it is able to exploit the weaknesses of standard security features. Furthermore, sabotage is not easily detected using ordinary checks on the table structure in the database, because the structure of each table targeted for sabotage does not change from the original, so digital forensic checks are needed. Log files and digital artefacts can be found when comparing the final report and checking the Get and Post methods on a web server using digital forensic tools. Based on penetrating digital forensic tests and analysis, employee performance appraisers can be exposed to sabotage attacks, so companies need to take steps to pay attention to a better level of security, and not only trust standard features. This study succeeded in answering potential security threats against the database in employee performance appraisers through a penetrating test.

References

1. F.S. Rosa, R.J. Lunkes, M.M.. Brizzolla, *Data on the environmental sustainability index of large Brazilian companies,* Data Br., no. March 103819, (2019)
2. I. Ruthven, *The language of information need: Differentiating conscious and formalized information needs*, Inf. Process. Manag. **56** no. 1, 77–90 (2019)
3. T. Jiang, Q. Guo, Y. Xu, S. Fu, *A diary study of information encountering triggered by visual stimuli on micro-blogging services*, Inf. Process. Manag. **56** no. 1, 29–42 (2019)
4. J. Wagner, A. Rasin, J. Grier, *Database forensic analysis through internal structure carving*, Digit. Investig. **14**, S106–S115 (2015)
5. G. Horsman, *Tool testing and reliability issues in the field of digital forensics*, Digit. Investig. **28,** 163–175 (2019)
6. R. Nordvik, F. Toolan, S. Axelsson, *Using the object ID index as an investigative approach for NTFS file systems*, Digit. Investig **28,** S30–S39, (2019)
7. E. Casey, *Interrelations between digital investigation and forensic science*, Digit. Investig. **28,** A1–A2 (2019)
8. N. Al-Mutawa, J. Bryce, V.N.L. Franqueira, A. Marrington, J.C. Read, *Behavioural Digital Forensics Model: Embedding Behavioural Evidence Analysis into the Investigation of Digital Crimes*, Digit. Investig. **28**, 70–82, (2019)
9. D. Fleurbaaij, M. Scanlon, *Privileged Data within Digital Evidence*, Comput. Sci. Crytography Secur. (2017)
10. F.S. Roozbahani, R. Azad, *Security Solutions against Computer Networks Threats*, Int. J. Adv. Netw. Appl. **2581,** 2576–2581 (2015)
11. M. Mufadhol, G. Aryotejo, D.Y. Kristiyanto, *Rule Based Reasoning Method for Safety Room by Means of Temperature Sensor and Motion Detector*, Adv. Sci. Lett. **23,** 2481–2483 (2017)
12. P. Pinoli, S. Ceri, D. Martinenghi, L. Nanni, *Metadata management for scientific databases*, Inf. Syst.**81,** 1–20 (2019)
13. N. Kishore, B. Kapoor, *Faster file imaging framework for digital forensics*, Procedia Comput. Sci. **49,** no. 1, 74–81 (2015)
14. X. Lin, J.H. Li, S.L. Wang, A.W.C. Liew, F. Cheng, X.S. Huang, *Recent Advances in Passive Digital Image Security Forensics: A Brief Review*, Engineering **4**, no. 1, 29–39 (2018)
15. T. Okamoto, *SecondDEP: Resilient computing that prevents shellcode execution in cyber-attacks*, Procedia Comput. Sci. **60,** no. 1, 691–699 (2015)
16. L. Révay, *From malware testing to virtualization*, Procedia Comput. Sci. **150,** 751–756 (2019)
17. G. Aryotejo, D.Y. Kristiyanto, Mufadhol, *Hybrid cloud: Bridging of private and public cloud computing*, J. Phys. Conf. Ser. **1025,** no. 1 (2018)