International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 4
ISSN: 2088-5334

Asset Identification in Information Security Risk Assessment Using Process Mining

Edri Yunizal [a,b,*], Judhi Santoso [a], Kridanto Surendro [a]

[a] *School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia*
[b] *Information Management, Institut Agama Islam Negeri Batusangkar,Batusangkar, Indonesia*
*Corresponding author: *edriyunizal@iainbatusangkar.ac.id*

***Abstract*— Information security risk assessment (ISRA) currently has gaps in inadequate asset identification. This activity is still manual, depending on the approach adopted and used, thus leading to subjectivity and inaccuracies. Whereas incorrect identification will lead to inaccurate results. The need to consider the dependency of assets within ISRA, which is still not resolved by ISRA, complicates this. A process perspective that can view assets based on their role in organizational processes rather than physical connections should be able to bridge this gap. Unfortunately, Small and Medium Enterprises (SME) find it difficult to take advantage of this opportunity due to time and cost constraints. This research bridges this gap by providing a process-oriented perspective that uses process mining. It automates asset identification based on historically derived organizational workflows using Legacy Information Systems (LIS) triggers. For rigor and relevance, this research uses a series of design research evaluation stages: problem, design, construct, and usage. Problem evaluation is through the study of related literature. For design evaluation, it made comparisons with asset and process-oriented ISRA and preprocessing of process mining. The construct evaluation by testing the system before and after method implementation. It also considers the method's maximum capability. Meanwhile, usage evaluation through a case study on an inventory system. The contribution offered: (1) integrating process mining with ISRA, (2) making the process-aware LIS without disturbing the running process, (3) preparing an artifact to generate an event log using database trigger, and (4) automating ISRA's asset identification which also considers asset dependency.**

***Keywords*— Information security; risk assessment; asset identification; small and medium enterprise; process mining; event log.**

Manuscript received 4 Apr. 2021; revised 17 Oct. 2021; accepted 3 Dec. 2021. Date of publication 31 Aug. 2022.

I. INTRODUCTION

Information is increasingly crucial in developing business activities [1]–[3] that making the IS, and its processing tools comprise an essential part of the organization's survival. A momentary failure on IS can be a complete disaster for organizations [4]. Therefore, organizations need a secure IS [5], and to achieve it requires an ISRA. ISRA supports decision-makers in assessing and understanding the risks faced by their organization [6]. The continued growth of reports on information security crimes demonstrates this need for further research [7], [8]. ISRA should consider asset dependencies [9], [10]. Asset dependency is the failure of information assets in an organization that can affect other assets that depend on that asset and causes a greater system failure. For example, analyzing a high-security website risk on a low-security server should consider website dependency on the server. ISRA considers asset dependency as the best choice [11]. However, it still cannot be overcome by existing methods and tools [12].

The initial stage of ISRA is asset identification, a process to identify assets in the organization [13]–[15]. An asset is one of the main parameters used to calculate the security impact of threats [16]. Incorrect identification will lead to inaccurate ISRA results [13], [14]. Making this one of ISRA's crucial steps [17], [18]. This activity is still done manually, depending on the model adopted or proposed. Asset identification in its current form is inadequate for risk assessment [17]. We can see this through both ISRA perspectives, asset and process-oriented [12], [13]. First, the asset-oriented perspective is a commonly used approach because it is easy and has many supporting tools [12]. Second, in process-oriented perspective, a rapid development perspective, and its ability to ensure the effectiveness of ISRA [19].

Asset-driven perspectives address this problem in several ways. For example, Muller et al [9] ignore identification

because it emphasizes simulation for velocity measurement of the proposed method to address cycles on dependencies. Rahmad et al. [20], through an existing standard catalog combined with threat scenario data, have its dependencies mapped in a tree structure. Tatar and Karabacak [14] perform top-down identification, starting from the hardware, software using the hardware, and information processing. Breier and Schindler's [18] research is based on a simple organizational model; it arranged dependencies research based on a tree structure, where the building is at the highest level.

Meanwhile, the process-oriented identifies assets related to the interconnectedness of organizational processes. Like asset-oriented, some do not explain how to identify assets; for example, Loloei et al. [16] emphasize more on dependency valuation. Suh and Han [4] assigned line managers to compile an asset-function assignment table based on pre-defined business function boundaries; dependencies are mapped in an asset dependency diagram. Khanmohammadi and Houmb [21] emphasize collecting workflows and identifying the assets involved. Schmidt and Albayrak [22] identify assets and dependencies by requesting experts and asset managers. Shedden et al. [13] answered with document analysis, seeing work directly, and semi-structured interviews with organizational actors.

Process-oriented has advantages in terms of realistic resource values [12]. The drawback is that dependencies between resources will take time for companies with large amounts of assets [12]. It is also expensive because it requires expert and thorough knowledge of all business processes and entities [23]. Process mining should overcome this limitation. It is the most recent development in data science, with one of the main techniques being process discovery [24], [25]; it is a technique that aims to find a historical process model by analyzing historical data [26]. Process mining results in a more significant model than the ideal model [27]–[29]. By utilizing it, the organization will have the historical workflow of the system. Complete with identification of the assets involved and their dependencies.

Unfortunately, there is still no ISRA research using process mining for asset identification. The primary reason is that it requires preprocessing activities. Not all systems support process mining; non-process-aware systems require these preprocessing activities. Preprocessing comprises two steps: defining the log architecture and completing the data architecture [30]. It aims to generate event logs not explicitly available on the non-process-aware legacy information system (LIS) [31], [32]. Event logs are a major component of the mining process [26], which comprise case id, task, originator, and timestamp. Preprocessing requires 60% of the effort in a process mining project; most manual, ad hoc, and domain-specific, time-consuming, and of inferior quality [31], [33].

Several studies proposed to ease preprocessing still cannot answer it. Calvanese et al. [32] carried out the process using an ontology approach. Jans et al. [30] proposed procedures that involve project objectives, key processes, essential and relation tables, document instances, instance levels, activities, attributes, and activity attributes. Andrews et al. [31] researched by generating logs that emphasize database relationships and assess data quality. van der Aalst [26] researched the concept of modeling databases as classes and objects and built event models based on database changes. de Murillas [34] proposed the creation of object-centric data to address one-to-many and many-to-many relationships in databases with correlations based on objects. Meanwhile, Pérez-Castillo et al. [35] carried out similar research through statistical analysis and source code modification in LIS. All this solution requires a correct system design, both system and database. Meanwhile, approximately 99% of organizations are SMEs, with a market share of 80% to 90% [36]. Limited resources on SME [3], [37], [38] making it focus more on the current and rapid solutions needed, avoiding complex stages, sometimes neglecting database design, object, and ontology [3]. This constraint also forces the development of their system to be achievable while running.

However, organizations already have a LIS with their relational database [39]. A relational database should be able to provide all the components needed to build an event log. It can do by optimizing the use of triggers, a database feature that has received less attention in information. Therefore, this study proposes a method for using process mining to identify data security assets using real data from relational databases. The application of this method is without disturbing the running system. There are four resulting contributions associated with preliminary studies. First, integrate process mining with ISRA. Second, make the current system process aware without disturbing the running process. Third, prepare the Automatic Trigger (Oger), an artifact to generate an ISRA asset identification dataset based on real data. And last, automate ISRA's asset identification, which considers asset dependencies. Proving the relevance of research using the design research methodology by Hevner et al. [40], with evaluation using Sonnenberg and Brocke [41], and a case study based on Brereton et al. [42].

We structure the remaining sections of this research as follows. The material and method that contains the development of design objectives using related literature, which is then followed by the proposed method, provides the steps for adding and accumulating change logs, followed by generating event logs. Section Result and discussion include evaluating problems, design, construct, and usage of the results, then discussion of the result. Finally, the conclusion section explains the research achievement.

II. Materials and Method

Information security needs redundant and reciprocally reinforcing proof. A criminal activity like larceny might illustrate it. Proof involving videos, fingerprints, and also the thief's schedule. However, all of its redundancy is still needed owing to their ability to strengthen each other. An associated example in information security is collecting and combining data before and after manipulation, as well as data actual. It becomes the basis for naming the proposed method Mining All Manipulation (AMin). AMin builds using design research; the first stage is problem identification in the previous section. Then, this section completes it by determining the design objectives. The problem identification results become the basis for method design and development.

A. Design objective

Design objectives are a feature available in previous studies as a requirement in the proposed method; it is all

shown in Table I and comprises 9 features, DO1 to DO9. We based DO1 on ISRA's considering asset dependency. DO2 shows the adoption of the ISRA's process-oriented. DO3 will help overcome the conceptual model problem, which is constrained by the subjectivity and implementation stages. The need for DO4 is because of data and process mining tools that require flat data. It will later integrate the advancement of the network sector with information systems, bringing us to DO5. The methods that use process mining must provide a basic component of process mining, an event log; this is the basis of DO6. Because of the shortcomings of the syntax-centric approach [43] and the benefits of a data-centric approach [19], [44], the method of proposition employs a data-centric approach, making it a DO7. DO8 based on most organizations that will use SMEs, the current system should not be disturbed. Finally, DO9 emphasizes that event logs can be made dynamic, and methods provide the capability to change them if needed.

TABLE I
DESIGN OBJECTIVES

Design objective		Ref.
DO1	Use of a dataset that takes into account asset dependencies	[9], [14], [18], [20], [45], [46]
DO2	Use of a dataset that takes into account the involvement of assets in organizational processes	[4], [13], [21]
DO3	Real field data needs	[31], [34], [35]
DO4	The method generates a flat dataset.	[4], [9], [13], [14], [18]–[21], [26], [30]–[32], [34], [45]
DO5	The use of datasets that can later be integrated with network datasets	[26], [30]–[32], [34]
DO6	The method produces a dataset as an event log.	[26], [30]–[32], [34], [35]
DO7	Data sources are data-centric	[31], [34]
DO8	Solutions that do not interfere with the running system	[36]
DO9	Customizable event log	[31], [32]

Design objective references include cross-domains: information security with asset dependencies, in both process and asset perspective, and preprocessing in process mining. DO1 is based on ISRA research with asset dependencies, specific to process-oriented is DO2. DO3, DO4, DO6, DO7, and DO9 for process mining. Meanwhile, integration of ISRA and process mining builds DO5.

B. AMin

Fig. 1 shows how AMin works. It uses a trigger that records all data manipulation language (DML). AMin powered by the Automatic trigger (Oger) artifact, currently at version 1.0.3.5. Built with Visual Studio Community 2017 and uses the MySQL database version 5.0.51b-community-nt-log. Oger has three features, adding change log, accumulating change log, and generating event log. The adding change log feature generating a LIS into LIS + change log without disturbing the running system. Accumulating change log feature accumulates change logs periodically (monthly). Furthermore, generating event log compiles a customized event log from the accumulated change log using the user's SQL parameters.

1) Adding change logs: Each table in the database undergoes two types of commands through the CreateTrigger function, (1) provision of change log tables and (2) provision of triggers to store change logs, as shown in Algorithm 1. The first command provides three container change log tables. While the second adds a trigger after the DML to populate all three container tables. _Insert and _Delete container tables contain all the columns from the source table. Current-user (cur_usr) column for the user that is running commands. While the current-timestamp (cur_tm) column for the runtime commands. _Update contains all the columns from before and current data (cur_data) columns for storing original data. There are two ways to use Algorithm 1, execute and SQL Script. Execute is the easiest way to carry out SQL commands directly to the database. While SQL script provides an option to update trigger commands manually. Update manual is to solve IS that involves triggers during the transaction.

2) Accumulating change logs: The goal of this stage is to avoid data accumulation, version changes, and slow access. Algorithm 2 shows the process to accumulated LIS + change logs monthly in the database and container tables. PrepareAccumulatedDB creates a list of change log tables. This function checks each table per month. Then, change logs moved from the current IS to a table in the accumulated database. Oger renames each table with a structure different from the latest structure to remain available. The renaming goal is to accommodate the difficulty of changing software versions [31], [47]. According to the last structure, Oger carries new data accumulation out on a created table.

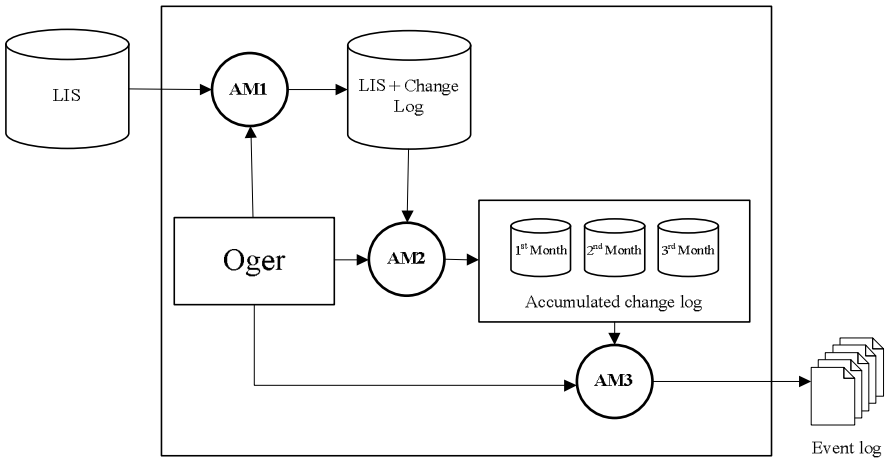

Fig. 1 How the proposed methods work

Algorithm 1. Adding change log

```
      Input: _Constring, _DBName
      Output: _ListOfCommand()
 1:   _ListOfTable()=GetListOfTable(_Constring)
 2:   _row=-1
 3:   For i=0 To _ListOfTable.count-1
 4:    _row=_row+1
 5:
      _ListOfCommand(_row)=CreateTrigger(_ListOfTable(i),_Insert
      )
 6:   _row=_row+1
 7:   _ListOfCommand(_row)=CreateTrigger(_ListOfTable(i),_Updat
      e)
 8:   _row=_row+1
 9:   _ListOfCommand(_row)=CreateTrigger(_ListOfTable(i),_Delete
      )
10:   Next
11:   For i=0 To _ListOfTable.count
12:    _row=_row+1
13:
      _ListOfCommand(_row)=CreateTrigger(_ListOfTable(i),_TrigIn
      sert)
14:   _row=_row+1
15:
      _ListOfCommand(_row)=CreateTrigger(_ListOfTable(i),_TrigU
      pdate)
16:   _row=_row+1
17:
      _ListOfCommand(_row)=CreateTrigger(_ListOfTable(i),_TrigD
      elete)
18:   Next
19:   Return _ListOfCommand()
```

Algorithm 2. Accumulating change log

```
      Input: _Constring, _DBName
      Output: -
 1:   _ListOfChangeLogTable()=PrepareAccumulatedDB(_Constring
      , _DBName)
 2:   For i=0 To _ListOfChangeLogTable.Count-1
 3:          _rs=SelectGroupMonthYearFromTable(_ListOfCha
      ngeLogTable(i))
 4:          Do While _rs.Read
 5:
             InsertIntoDBAccumulated(_rs(month),_rs(year),
       ListOfChangeLogTable(i))
 6:
             DeleteFromDBName(_rs(month),_rs(year),
      _ListOfChangeLogTable(i))
 7:          Loop
 8:   Next
```

Algorithm 3. Generating event log

```
      Input: _Constring, _DBName, DBEventLog
      Output: -
 1:   _ListOfAccumulatedChangeLogDB()=GetListOfAccumu
      latedChangeLogDB()
 2:   CreateTableEventLog(EventLogTemplate)
 3:   For i=0 To _ ListOfChangeLog DB.Count-1
 4:          InsertFromAccumulatedToEventLog(EventLo
      gTemplate)
 5:   Next
```

3) Generating event logs: The last stage is generating the event log from the accumulated change logs. According to Jans et al [30], the event log contains events related to business processes. It contains such as what, when, and who conducted the process. Existing change logs can provide these three requirements. Thus, overcoming this requires providing a database with three event log entities. These entities comprise tbeventlog, tbeventlog_table, and tbeventlog_branch. Fig. 2 shows the required ERD database design.

Generating the event log takes four steps, creating an event log template (GEL1), determining the DB change log (GEL2), filling in the event log template per table (GEL3), and generating the event log (GEL4). For example, in GEL1, the user needs an event log with two columns, id, and name. Each column uses one of three supporting data types, text, number, and datetime. The three data types used must in the largest size to support customization. It must support either a regular column, or a combination of multi-column, multi-table, or implicit code. GEL2 aims to accommodate GEL1 in the organization that has IS with over one branch.

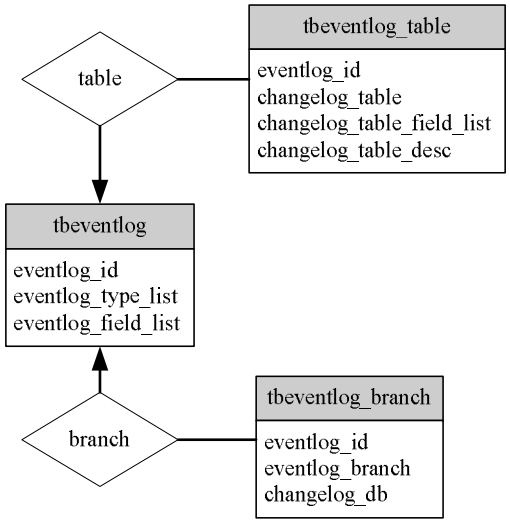

Fig. 2 Oger E-R diagram

In GEL3, it defines the tables involved for each event log created. Definition using SQL's *CONCAT* and *AS* keywords. SQL is a natural choice for many users [34]. It allows customization according to the GEL1 template and the GEL2 branch. For the GEL3 example, there is a template table comprising columns *event_id* and *event_name*, with data derived from *tbitem* (*item_id*, *item_name*). SQL commands will make it possible through command *item_id AS event_id* and *item_name AS event_name*.

The latest, GEL4, starts with getting a list of the accumulated change log databases available, using it to create event log tables. Tables on GEL4 then filled with all data based on GEL3 customizations, as shown in Algorithm 3.

III. Results and Discussion

This is the last stage of design research: results. And supported the findings, it conferred on a discussion of previous analysis contributions.

A. Results

We compiled the results based on an evaluation research design which comprises an evaluation of problems, design, construct, and usage [41].

1) Problems: Evaluation of problems aims to get observations of problems, problems statement, existing solutions, research needs, and design objectives. We have

presented this in the introduction and material and method section.

2) Designs: this evaluation aims to position the planned technique supported by existing solutions (see TABLE II). AMin is positioned at ISRA and process mining with design objectives on TABLE I as its comparison. In ISRA's analysis, there's no automation in asset identification. The planned technique so much outperforms each of them, using automation and process mining (DO3, DO5 to DO9). Meanwhile, with process mining, AMin excels in preprocessing while not perturbing the running system (DO8), including generating customized event logs (DO9).

TABLE II
DESIGN EVALUATION

Research	Design objective								
	1	2	3	4	5	6	7	8	9
ISRA asset-oriented									
[9], [14], [18], [20], [45]	✓	×	×	✓	×	×	×	×	×
ISRA process-oriented									
[4], [13], [21], [22]	✓	✓	×	✓	×	×	×	×	×
Preprocessing process mining									
[31]	✓	✓	✓	✓	✓	✓	✓	×	✓
[32]	✓	✓	×	✓	✓	✓	×	×	✓
[30]	✓	✓	×	✓	✓	✓	×	×	×
[34]	✓	✓	✓	✓	✓	✓	✓	×	×
[26]	✓	✓	×	✓	✓	✓	×	×	×
[35]	✓	✓	✓	×	×	✓	×	×	×
Proposed method (AMin)	✓	✓	✓	✓	✓	✓	✓	✓	✓

3) Constructs: Evaluation of constructs is artifact validation through artificial settings. Thus, the authors conducted a benchmark test using the IS-X. It is a database prototype of a simple sales system. Comprise 4 tables, tbitem, tbconsumer, tbsale, and tbsalesman. Benchmark using Intel Core i7 4702MQ CPU @ 2.20GHz 2.19 GHz with 16 GB memory. The operating system used is Windows 8.1 64-Bit, with database MySQL 5.0.51b-community-nt-log database using MyISAM table type.

TABLE III
MAXIMUM LIMIT LIS+CHANGE LOG (IN SECONDS)

Data	Client						
	5	10	15	20	25	30	35
1,000	4.0	6.6	12.4	11.5	16.9	18.2	23.0
2,000	7.0	11.4	21.4	24.9	30.4	43.7	658.7
3,000	9.8	20.0	26.7	38.8	52.9	60.2	727.6
4,000	13.0	18.8	40.8	49.6	64.8	82.3	1,157.4
5,000	16.0	32.8	47.6	64.0	73.4	101.5	1,539.5
6,000	19.4	36.7	58.9	73.8	93.5	122.5	1,760.2
7,000	21.8	36.0	66.0	84.8	117.2	143.2	1,826.8
8,000	25.8	42.8	76.0	99.1	141.1	107.9	2,161.1
9,000	29.0	56.9	80.4	111.5	146.1	139.1	2,856.1
10,000	33.0	63.0	94.9	127.3	145.2	140.6	3,545.5

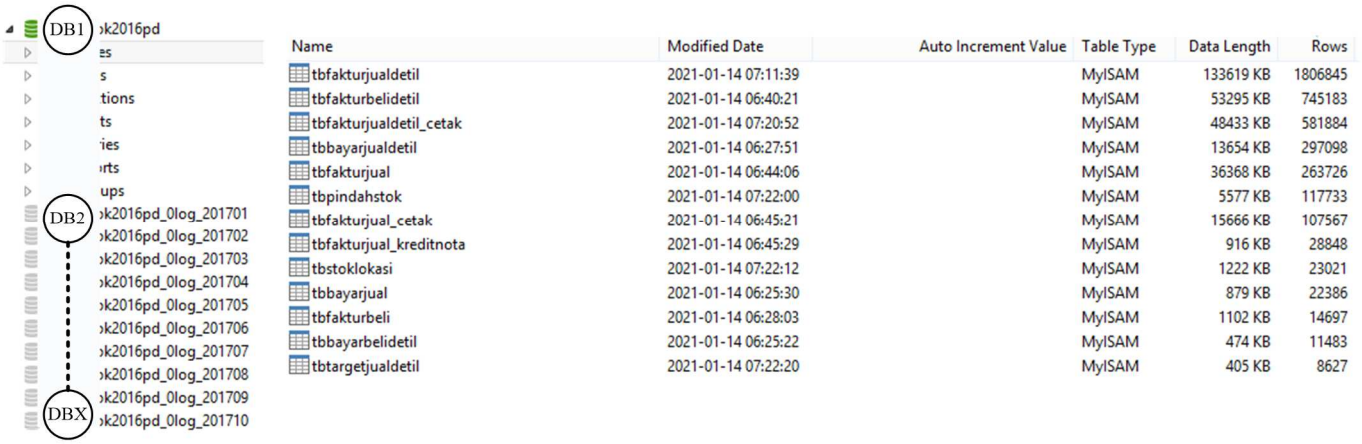

Fig. 3 IS-D database at the Accumulating change log stage

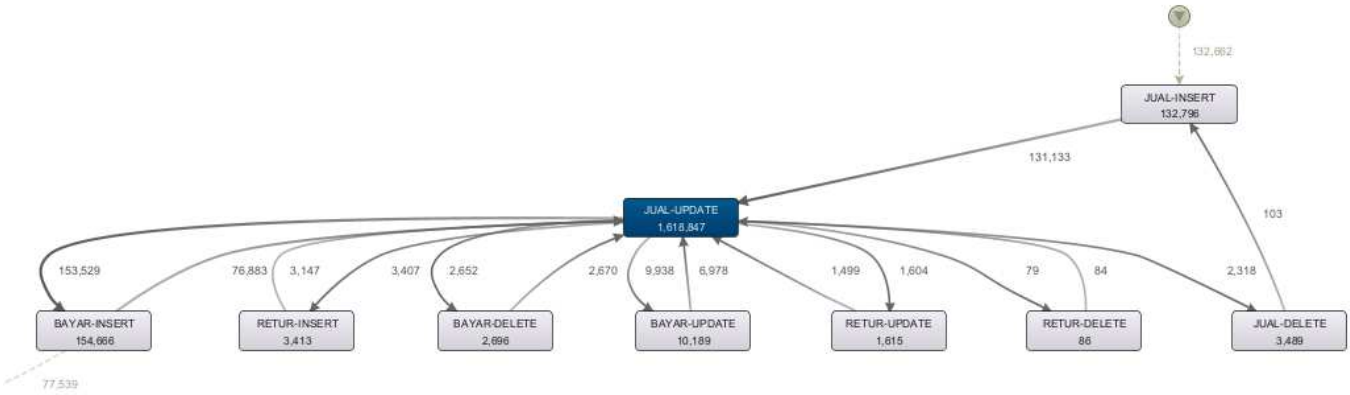

Fig. 4 IS-D's sales-to-pay workflow 1

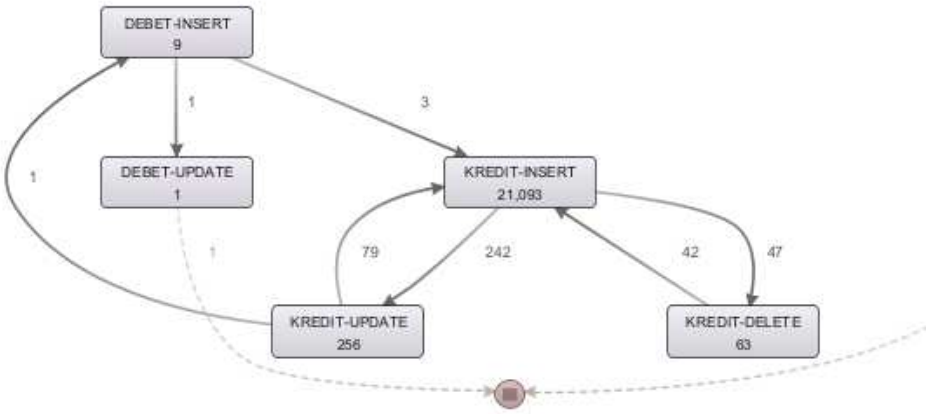

Fig. 5 IS-D's sales-to-pay workflow 2

The construct evaluation simulates DML on IS-X 100 times with a mix of data and clients ranging from 50 to 500, with a comparison of DML INSERT (75%), UPDATE (25%), and DELETE (10%) data amounts. According to the test results, each client takes an average of 1.56 seconds for IS-X + change log and 1.58 seconds for IS-X + change log. It continued the simulation to determine the maximum capability of the IS-X + change log. Data ranges from 1,000 to 2,000 and up to 10,000, with clients ranging from 5 to 35. As shown in TABLE III, constraints appeared on 10 clients with 10,000 data where the time required is greater than 1 minute (63.00 seconds). Significant changes occurred in 35 clients with 2,000 data points, which took about 11 minutes (658.77 seconds). Evaluation of constructs shows AMin can be used without disrupting the running system, which is especially important for SMEs.

4) Usages: The last evaluation is the evaluation of usages; it is an evaluation of implementing the AMin method through a case study. It employs Oger artifacts on a non-process-aware LIS. The case study included seven criteria, six LIS candidates, and five procedures based on Brereton et al. [42]. We discuss the details separately because it is outside the area of focus.

The selected LIS candidate is IS-D based on the candidates, procedures, and criteria. It is an outsourced distributed inventory system. The latest version is 3.5.1.3 dated October 9, 2020. This IS using MySQL 5.0.51b-community-nt-log and 5.5.57-log databases with 99 tables. Built with Visual Studio Community 2017, comprises one solution with two projects. Each project consists of 41,705 and 241,762 lines of code (LOC). It serves 1 server and 24 clients. The implementation results up to the accumulating change log stage; it shown in Fig. 3. DB1 shows the IS-D + change log database. While DB2 to DBX illustrates the monthly accumulated change log.

The authors and IS-D developers discussed which process to use to generate the event log, and sales-to-pay was chosen. The IS-D developer then creates a manual workflow for the sales-to-pay process, including five computerized processes. First, the salesperson makes a sale to customers by creating invoices. Second, the warehouse receives returns from customers with Create return invoices. Third, accounting adds credit notes to invoices. Fourth, accounting adds a debit note to the invoice. Finally, accounting receives and makes payments from customers with invoices as proof. The process involves 5 tables: tbfakturjual (JUAL), tbfakturjual_retur (RETURN), tbbayarjualdetil (BAYAR), tbfakturjual_debetnota (DEBIT), and tbfakturjual_kreditnota (KREDIT).

Oger generates an event log, which is then processed using a process mining application. The application used is Disco with an academic license version 2.10.1. Disco's Overview shows 1,949,219 incidents with 166,831 cases and 14 activities. The data is from June 19, 2017, 10:43:07 to October 9 2020, 17:49:29. Data mean and median case durations of 4.9 and 26.9 days respectively. This data generates sales-to-pay workflow in IS-D as shown in Fig. 4 and Fig. 5.

B. Discussion

Shedden et al. [13] have described the importance of ISRA's asset identification. It is just that the solutions involve a fairly complex qualitative process. Abdulrazzaq and Wei [48] propose simplification by utilizing Grassmarlin, Nmap, and ISF software. Unfortunately, this approach emphasizes more the physical relationship of assets; it cannot map the involvement of assets to organizational processes. Adesemowo [17] then concludes that it is still not sufficient for the current state of ISRA. We prove AMin has overcome this gap by integrating ISRA with process mining. As a result, asset identification is seen physically and involves the processes that use them. The only resources needed are LIS with its relational database, which is something most organizations already have.

The relational database has a trigger feature, which allows method implementation to be done automatically. According to research carried out by Pérez-Castillo et al. [35], there is a significant increase in the time needed compared to manipulating the application source code through statistical analysis. The time required for IS-D of 283,467 LOCs is 1,588,488 milliseconds [35]. Meanwhile, using evaluation of usages on IS-D with 99 tables, Oger only takes 112,000 milliseconds, showing an increase of 1,418% without disturbing the running system.

A relational database allows the proposed method to make asset identification based on real data. AMin giving it an event log feature, a dataset that contains assets and its process involvement. This dataset can connect assets to the dataset of supporting hardware, such as modems and routers. It can link again this to the network dataset to show asset dependency using research [49]. The event log dataset also allows the

creation of user profiles. It can use to identify employee theft, such as false returns [50]. Based on evaluation of usages, with 84 returns events from 1,949,219 total events on IS-D. It shows that these false returns are not possible on IS-D.

Usages evaluation shows how data from the event log identifies the computerized process of IS-D. Fig. 4 and Fig. 5 show 14 activities identified from what should be 15. These activities show five computerized processes from manual workflows. The only activity missing is DEBET-DELETE, as it was never conducted. These address the difficulty in collecting data on Shedden et al. [13], which required 31 sessions with 11 levels in the organization and took 20.1 hours.

The event log can also show how to determine asset dependency using data only. The IS-D event log can show 68 computers involved as clients using Disco. Only 25 computers were active until September 2020. The computer is dominated by 192.168.1.69, which is the IP address of the IS-D server. This will avoid subjectivity and confidentiality, such as interviews with system owners or domain experts on ISRA [4], [19].

IV. Conclusion

In conclusion, this research proposes a method for automatic asset identification for ISRA using process mining. It comprises adding and accumulating change logs, then generating them into customizable event logs. The weakness of the existing solution has requirements that are difficult to apply to a system that must remain running overcome by using Oger artifact. Increasing implementation possibility in the organization that has limited resources. Gaps filled by identifying the asset and current system's workflow using only data from LIS's relational database. Utilization data avoids the need for documentation and a manual interview with the system owner or domain expert. Dataset also can determine the asset dependency in the workflow.

The research shows that there is a major improvement for ISRA process-oriented and process mining. Automatically asset identification which additionally considers its dependency builds the chance ISRA's being widely used by the non-process-aware organization. It'll later turn out a positive chain impact. Beginning with the increasing possibility of overcoming ISRA's asset dependency constraints using data science. It additionally can be a start development of the ISRA dataset. Meanwhile, in process mining, the proposed method shows that it can make the existing system process-aware with a customized event log without disturbing the running system. Finally, ISRA process-oriented and process mining are a potent combination that promises easier comparison and can be performed quantitatively to remove subjectivity.

Acknowledgment

The authors are grateful to the IAIN Batusangkar for their financial support. The authors are also grateful to the Fluxicon team for providing academic license and support for the Disco software.

References

[1] P. Tubío Figueira, C. López Bravo, and J. L. Rivas López, "Improving information security risk analysis by including threat-occurrence predictive models," *Comput. Secur.*, vol. 88, p. 101609, 2020.

[2] D. G. Rosado, J. Moreno, L. E. Sánchez, A. Santos-Olmo, M. A. Serrano, and E. Fernández-Medina, "MARISMA-BiDa pattern: Integrated risk analysis for big data," *Comput. Secur.*, vol. 102, 2021.

[3] A. Ključnikov, L. Mura, and D. Sklenár, "Information security management in SMEs: Factors of success," *Entrep. Sustain. Issues*, vol. 6, no. 4, pp. 2081–2094, 2019.

[4] B. Suh and I. Han, "The IS risk analysis based on a business model," *Inf. Manag.*, vol. 41, no. 2, pp. 149–158, 2003.

[5] Y. Wang, M. Zhao, Y. Hu, Y. Gao, and X. Cui, "Secure computation protocols under asymmetric scenarios in enterprise information system," *Enterp. Inf. Syst.*, vol. 15, no. 4, pp. 492–512, 2021.

[6] C. Schmitz and S. Pape, "LiSRA: Lightweight Security Risk Assessment for decision support in information security," *Comput. Secur.*, vol. 90, 2020.

[7] N. S. Safa *et al.*, "Deterrence and prevention-based model to mitigate information security insider threats in organisations," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 587–597, 2019.

[8] P. J. Steinbart, R. L. Raschke, G. Gal, and W. N. Dilla, "The influence of a good relationship between the internal audit and information security functions on information security outcomes," *Accounting, Organ. Soc.*, vol. 71, pp. 15–29, 2018.

[9] S. Muller, C. Harpes, Y. Le Traon, S. Gombault, and J. M. Bonnin, "Efficiently computing the likelihoods of cyclically interdependent risk scenarios," *Comput. Secur.*, vol. 64, pp. 59–68, 2017.

[10] D. Gritzalis, G. Stergiopoulos, V. Kouktzoglou, and M. Theocharidou, "A process-based dependency risk analysis methodology for critical infrastructures," *Int. J. Crit. Infrastructures*, vol. 13, no. 2/3, p. 184, 2017.

[11] Y. Y. Haimes, "Risk Modeling of Interdependent Complex Systems of Systems: Theory and Practice," *Risk Anal.*, vol. 38, no. 1, pp. 84–98, Jan. 2018.

[12] A. Shameli-Sendi, R. Aghababaei-Barzegar, and M. Cheriet, "Taxonomy of information security risk assessment (ISRA)," *Comput. Secur.*, vol. 57, pp. 14–30, 2016.

[13] P. Shedden, A. Ahmad, W. Smith, H. Tscherning, and R. Scheepers, "Asset identification in information security risk assessment: A business practice approach," *Commun. Assoc. Inf. Syst.*, vol. 39, no. 1, pp. 297–320, 2016.

[14] Ü. Tatar and B. Karabacak, "An hierarchical asset valuation method for information security risk analysis," in *International Conference on Information Society, i-Society 2012*, 2012, pp. 286–291.

[15] C. Joshi and U. K. Singh, "Information security risks management framework – A step towards mitigating security risks in university network," *J. Inf. Secur. Appl.*, vol. 35, pp. 128–137, 2017.

[16] I. Loloei, H. R. Shahriari, and A. Sadeghi, "A model for asset valuation in security risk analysis regarding assets' dependencies," in *ICEE 2012 - 20th Iranian Conference on Electrical Engineering*, 2012, pp. 763–768.

[17] A. K. Adesemowo, "Towards a conceptual definition for IT assets through interrogating their nature and epistemic uncertainty," *Computers and Security*, vol. 105. 2021.

[18] J. Breier and F. Schindler, "Assets dependencies model in information security risk management," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8407 LNCS, pp. 405–412.

[19] A. Shameli-Sendi, "An efficient security data-driven approach for implementing risk assessment," *J. Inf. Secur. Appl.*, vol. 54, 2020.

[20] B. Rahmad, S. H. Supangkat, J. Sembiring, and K. Surendro, "Modeling asset dependency for security risk analysis using threat-scenario dependency," *Int. J. Comput. Sci. Inf. Secur.*, vol. 10, no. 4, pp. 103–111, 2012.

[21] K. Khanmohammadi and S. H. Houmb, "Business process-based information security risk assessment," in *Proceedings - 2010 4th International Conference on Network and System Security, NSS 2010*, 2010, pp. 199–206.

[22] S. Schmidt and S. Albayrak, "A quantitative framework for dependency-aware organizational IT Risk Management," in *Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications, ISDA'10*, 2010, pp. 1207–1212.

[23] V. Agrawal, "A Comparative Study on Information Security Risk Analysis Methods," *J. Comput.*, pp. 57–67, 2017.

[24] E. G. L. de Murillas, H. A. Reijers, and W. M. P. van der Aalst, "Connecting databases with process mining: a meta model and toolset," *Softw. Syst. Model.*, vol. 18, no. 2, pp. 1209–1247, Apr. 2019.

[25] A. Augusto *et al.*, "Automated Discovery of Process Models from Event Logs: Review and Benchmark," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4. pp. 686–705, 2019.

[26] W. M. P. van der Aalst, "Extracting Event Data from Databases to Unleash Process Mining," in *BPM-Driving innovation in a digital world*, Springer, 2015, pp. 105–128.
[27] C. dos S. Garcia *et al.*, "Process mining techniques and applications – A systematic mapping study," *Expert Syst. Appl.*, vol. 133, pp. 260–295, 2019.
[28] L. Lan, Y. Liu, and W. Feng Lu, "Learning from the Past: Uncovering Design Process Models Using an Enriched Process Mining," *J. Mech. Des.*, vol. 140, no. 4, 2018.
[29] J. Maeyens, A. Vorstermans, and M. Verbeke, "Process mining on machine event logs for profiling abnormal behaviour and root cause analysis," *Ann. des Telecommun. Telecommun.*, vol. 75, no. 9–10, pp. 563–572, 2020.
[30] M. Jans, P. Soffer, and T. Jouck, "Building a valuable event log for process mining: an experimental exploration of a guided process," *Enterp. Inf. Syst.*, vol. 13, no. 5, pp. 601–630, 2019.
[31] R. Andrews, C. G. J. van Dun, M. T. Wynn, W. Kratsch, M. K. E. Röglinger, and A. H. M. ter Hofstede, "Quality-informed semi-automated event log generation for process mining," *Decis. Support Syst.*, vol. 132, 2020.
[32] D. Calvanese, M. Montali, A. Syamsiyah, and W. M. P. van der Aalst, "Ontology-driven extraction of event logs from relational databases," in *Lecture Notes in Business Information Processing*, 2016, vol. 256, pp. 140–153.
[33] A. P. Kurniati, E. Rojas, D. Hogg, G. Hall, and O. A. Johnson, "The assessment of data quality issues for process mining in healthcare using Medical Information Mart for Intensive Care III, a freely available e-health record database," *Health Informatics J.*, vol. 25, no. 4, pp. 1878–1893, 2019.
[34] G. Li, E. G. L. de Murillas, R. M. de Carvalho, and W. M. P. van der Aalst, "Extracting object-centric event logs to support process mining on databases," in *Lecture Notes in Business Information Processing*, 2018, vol. 317, pp. 182–199.
[35] R. Pérez-Castillo, B. Weber, J. Pinggera, S. Zugal, I. G. R. de Guzmán, and M. Piattini, "Generating event logs from non-process-aware systems enabling business process mining," *Enterp. Inf. Syst.*, vol. 5, no. 3, pp. 301–335, 2011.
[36] Y. Barlette, K. Gundolf, and A. Jaouen, "CEOs' information security behavior in SMEs: Does ownership matter?," *Systèmes d'information Manag.*, vol. 22, no. 3, p. 7, 2017.
[37] S. Kabanda, M. Tanner, and C. Kent, "Exploring SME cybersecurity practices in developing countries," *J. Organ. Comput. Electron. Commer.*, vol. 28, no. 3, pp. 269–282, 2018.
[38] T. Woschke, H. Haase, and J. Kratzer, "Resource scarcity in SMEs: effects on incremental and radical innovations," *Manag. Res. Rev.*, vol. 40, no. 2, pp. 195–217, 2017.
[39] M. Jans, "Auditor choices during event log building for process mining," *J. Emerg. Technol. Account.*, vol. 16, no. 2, pp. 59–67, 2019.
[40] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q. Manag. Inf. Syst.*, vol. 28, no. 1, pp. 75–105, 2004.
[41] C. Sonnenberg and J. Vom Brocke, "Evaluations in the science of the artificial - Reconsidering the build-evaluate pattern in design science research," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7286 LNCS, pp. 381–397.
[42] P. Brereton, B. Kitchenham, D. Budgen, and Z. Li, "Using a protocol template for case study planning," in *12th International Conference on Evaluation and Assessment in Software Engineering, EASE 2008*, 2008.
[43] J. Wagner *et al.*, "Carving database storage to detect and trace security breaches," *Digit. Investig.*, vol. 22, pp. S127–S136, 2017.
[44] L. K. Branting, "Data-centric and logic-based models for automated legal problem solving," *Artif. Intell. Law*, vol. 25, no. 1, pp. 5–27, 2017.
[45] E. Yunizal, K. Surendro, and J. Santoso, "A Method of Simplifying the Asset Dependency Cycle in Security Risk Analysis," 2021.
[46] G. Stergiopoulos, D. Gritzalis, and V. Kouktzoglou, "Using formal distributions for threat likelihood estimation in cloud-enabled IT risk assessment," *Comput. Networks*, vol. 134, pp. 23–45, 2018.
[47] H. Zhou and J. Li, "A dynamic instrumentation tool for obtaining software logs," *J. Phys. Conf. Ser.*, vol. 1684, no. 1, 2020.
[48] M. Abdulrazzaq and Y. Wei, "Industrial Control System (ICS) Network Asset Identification and Risk Management," 2018.
[49] M. Lyu, H. Habibi Gharakheili, C. Russell, and V. Sivaraman, "Mapping an Enterprise Network by Analyzing DNS Traffic," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11419 LNCS, pp. 129–144.
[50] D. B. Speights, D. M. Downs, and A. Raz, *Essentials of modeling and analytics: Retail risk management and asset protection*. 2017.