JIT Mahasarakham University, Thailand
http://jit.it.msu.ac.th

Journal of Applied Informatics and Technology, 3(2), 63-81

Research Article

Intrusion Alert Framework using Semantic Web and Data Mining Approach

Jatuphum Juanchaiyaphum[1,2,*], Preecha Noiumkar[2], Vuttichai Vichaianchai[2]

[1] Information Technology for the Future (ITF)
[2] Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham 44150, Thailand
[*] Corresponding Author: Jatuphum Juanchaiyaphum, jatuphum.j@msu.ac.th

Received:
28 July 2021
Revised:
24 September 2021
Accepted:
10 November 2021

Keywords:

Intrusion alert analysis, alert correlation, ontology, attack scenarios

Abstract: This research proposed an integrated semantic web and data mining approach for examining alert logs and reconstructing attack scenarios, which provide critical evidence for understanding the damaging effects of the attack scenarios. The semantic web is used to filter out irrelevant alerts and then infer candidate attack patterns based on the relationship between alerts defined by the applied Cyber kill-chain concept. After that, an algorithm based on association rules is used to extract frequent attack sequential patterns from candidate attack scenarios. Experiments on the DARPA 2000 LLDOS 1.0 dataset demonstrated that the proposed approach is effective; it reduces false alerts and extracts useful information that can be used to solve direct problems while also shortening the analysis time. The proposed approach outperformed related alert-correlation approaches in terms of completeness and soundness, with the proposed approach achieving 100 percent completeness and soundness, respectively.

1. Introduction

Intrusion detection is a field of computer security which monitors an information system and detects intrusive activities that attempt to compromise the network system. However, using an IDS (intrusion detection system) has its own problems. IDSs generate too many low-level alerts which turn out to be false or irrelevant; they are extremely elementary and not accurate enough to be managed directly by a security administrator. Therefore, they are difficult to analyze, often time-consuming, and labor intensive since the relevant alerts are usually buried under heaps of irrelevant alerts (Njogu *et al*., 2014). Moreover, to examine the logs and understand what damages we may have inflicted, is a challenge of OWASP Top Ten 2017 (OWASP, 2017) (OWASP).

To solve this problem, many researchers have proposed alert correlation approaches which find similarity or causality between the alerts and rebuild the group of attack scenarios. However, these approaches have limitations. For example, the clustering and data mining approaches cannot detect causality between individual attacks. Moreover, the performance of data mining-based models depends on the training dataset. If the training set consists of noises or irrelevant data, the performance will be decreased, which may lead to overlapping alert clusters (García *et al*., 2013). The limitation of knowledge-based approaches can rebuilds both known and unknown attack scenarios as long as the individual attack steps are collected in the knowledge-base. In other words, these approaches use fixed attack patterns for reconstructing attack steps, and therefore, they cannot find and rebuild attack scenarios that do not exist in the pattern database. Moreover, if the IDSs miss a critical alert, the alert correlation module may incorrectly construct attack scenarios.

This research use an intrusion alert framework to analyze low-level alerts and extract information using a combination of semantic web and datamining approach. This method employs a combination of ontology-based and data mining techniques. The analysis module consists of two phases, namely, information acquisition phase and decision phase. In information acquisition phase, knowledge-based ontology is used to extract information by removing irrelevant alerts and creating relationships between alerts. The applied kill-chain concept is used to create relationships between the alerts in each group, as alerts with the same destination host, before generating candidate attack sequences. Ultimately, this method can generate a candidate attack sequence even if some critical alerts are missing. In decision phase, Association rule algorithm is used to discover the real attack patterns from the candidate attack sequences. In other words, this approach extracts qualitative information that will be used in analyzing instead of using low-level data. In essence, the performing effectiveness of this approach is evaluated by

using the DARPA 2000 LLDOS 1.0 dataset (MIT Lincoln Lab, 2002). That results in a series of alerts. Subsequently, they are used to monitor the completeness and soundness metrics in order to compare performance between the proposed approaches with other related approaches (Ning *et al*., 2002).. The former metric assesses how well the method can correlate related alerts together, while the latter evaluate how correctly the alerts are correlated. Regards the evaluation of the proposed approach, outcomes of both the completeness and soundness were as 100% and 100%, respectively.

The remainder of this paper is structured as follows. The second section provides a brief summary of related work, while the third introduces and explains the proposed architecture in detail. The fourth section discusses the alert analysis module used to analyze low-level alert messages in which are applied to extract information and discover attack scenarios from information. The fifth presents an implementation of the framework. The sixth section discusses the performance and results of the proposed method and the final section concludes with a summary and recommendations for future research.

2. Related Work

To solve false alerts of IDS, many researchers have proposed alert correlation methods which group related alerts together by finding similarity or causality between them. Data mining techniques and ontology-based approaches are widely proposed to solve alert-correlation issue. In this section, a brief summary is discussed of related works as follows:

2.1 Clustering and data mining-based alert correlation

de Alvarenga *et al*. (2018) addressed the issue of visualizing huge amount of alert logs. This method used mining process and hierarchical clustering techniques to extract information about the attackers' behavior and discover the attack scenarios that are used in an attempt to compromise the network. The experimental evaluation by using a real IDS alerts dataset from the University of Maryland indicated that this method is capable to represent the attack scenarios used to investigate the alerts manually.

Yu-Xin *et al*. (2008) proposed an improved the Apriori algorithm to find the attack scenarios. The sequence of attack is key feature for mining the relation of each attack. The proposed method was evaluated by The DARPA 1999 dataset. The results demonstrate that the completeness is 76% while the soundness of the approach is 53%.

Li *et al*. (2007) proposed method of constructing attack scenarios in order to recognize attacker's high level strategies, and predict upcoming attack intentions. Association rule mining is used to mine frequent attack sequential patterns from history high level alert database. This method

can be used to detect novel multistage attack patterns. The performance evaluation using the DARPA 2000 dataset represented attack scenario detection rate of 92.2%.

Al-Mamory & Zhang (2007) proposed a systematic method for constructing attack scenarios. In the proposed approach, the irrelevant alerts are filter out and then similar raw IDS alerts are grouped into meta-alert (MA) messages. An attack scenario is generated using alert clustering and correlation depending on a relation matrix (RM) that defines the similarities between every two MA messages. The result of evaluation by using the DARPA 2000 LLDOS 1.0 dataset indicated that the completeness and the soundness of the proposed approach are 89.7% and 100%, respectively.

Although alert correlation approaches that employ data mining techniques are capable of handling large volumes of IDS alerts and reconstructing novel and unknown attack scenarios, these approaches have limitations, including the technique's inability to reconstruct sophisticated multi-step attack scenarios, as data mining approaches cannot detect causality between individual attack steps (Saad & Traore, 2013). Another significant limitation is that data mining-based models' performance is highly dependent on the training dataset. Performance will be reduced if the training set contains noise or irrelevant data.

2.2 Ontology-based alert correlation

Yuan *et al*. (2020) proposed a method for assessing network vulnerability based on a graph database. The graph database stores network host information, association relationships between hosts, and vulnerability information about the target network; querying and analysis are performed using the graph database query language. The graph database query language is capable of querying and analyzing graph databases. Visualizing the network topology, vulnerability data, and all possible attack paths enables the development of a network security protection strategy. The results of the experiments demonstrate that the method is efficient and facilitates querying and analysis in a large-scale complex network environment.

Barik (2018) proposed a query language for analyzing network vulnerabilities using attack graphs. It is composed of query constructs for performing various attack graph analysis tasks as well as for generating the attack graph itself. The query language's features are based on generic attack graph models, and as such, it can be implemented on top of any type of data store, such as a relational database or a graph database. This query language will aid administrators in the development of network security applications that frequently query the attack graph.

Sadighian *et al*. (2014) suggested ONTIDS, as a highly-flexible, context-aware, ontology-based alert correlation framework. This

approach uses ontologies to represent and store the alerts information, vulnerability information, alerts context, and attack scenarios. To correlate and reduce irrelevant alerts, ONTIDS employs simple ontology logic rules written in Semantic Query-enhance Web Rule Language: SQWRL. The DARPA 2000 and UNB ISCX IDS evaluation datasets were used to illustrate the potential usefulness and flexibility of ONTIDS.

Saad & Traore outlined a new approach for attack scenario reconstruction that analyzes both implicit and explicit relationships between intrusion alerts using semantic analysis and a new intrusion ontology. The proposed approach can rebuild both known and unknown attack scenarios and correlate alerts generated in multi-sensor IDS environments. Moreover, this approach can handle, for the first time, both novel attacks and false negative alerts generated by IDSs. Experimental results using the DARPA 2000 dataset indicated that both the completeness and soundness of the proposed approach as 100% and 99.70%, respectively.

Li & Tian (2010) proposed an alert correlation approach based on XSWRL ontology. This approach consists of agents and sensors, where agents process the information and sensors gather security data in IDMEF (Intrusion Detection Message Exchange Format) (Debar *et al*., 2007). The XSWRL (Extended Semantic Web Rule Language) (Li & Tian, 2008) is used as an automated reasoner to deduce the threat from attack sessions and classify the risk.

However, some previous ontology-based approaches are limited in their ability to find and rebuild attack scenarios, as long as the individual attack steps are collected in their knowledge, because they use fixed attack patterns. Furthermore, all previous approaches performed methods based on perfect alert logs which contain all the critical alerts. As a result, if the IDSs miss a critical alert, the alert correlation module may incorrectly construct an attack scenario.

3. Proposed intrusion alert analysis architecture

The intrusion alert analysis architecture consists of three layers as the resource layer, mediator layer, and application layer (Fig. 1). Details of each layer are as follows:

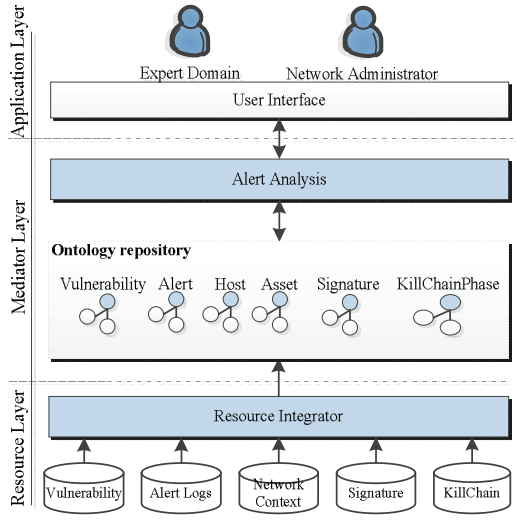

Fig. 1. Intrusion alert analysis architecture

3.1 Resource layer

This layer includes a resource integrator module that integrates heterogeneous resources in structured and unstructured formats, including intrusion alarm logs, intrusion signatures, cyber kill chain, vulnerability database, and network context. The mapping patterns defined by experts are used to extract unstructured format resources from raw files, such as alert logs, intrusion signatures, and network context. Each resource is described using the Resource Description Framework (RDF) and then stored in an ontology repository. The following section contains information about these resources.

3.1.1 Intrusion alert logs

This resource is the report generated by the IDS. Each alert message commonly consists of eight important attributes, namely, detection time, attack signature, attack classification, source IP address, protocol, source port, destination IP address, and destination port.

3.1.2 Cyber kill-chain

The proposed kill-chain was inspired by the Lockheed Martin kill-chain (Martin, 2014), which is currently used by the National Institute of Standards and Technology (NIST) as a component of the Cyber Security Framework, and cited and applied in various security frameworks (Bryant & Saiedian, 2017; Hahn *et al*., 2015).

This research apply the cyber kill-chain concept from seven to four steps, namely, reconnaissance, delivery, exploitation and system compromised as shown in Fig. 2. The Weaponization phase of the traditional cyber kill-chain was not used because this phase cannot be detected by IDSs, whereas system compromised covers three phases of traditional cyber kill-chain, namely, installation, command and control, and actions on objectives which are activities after successfully exploiting the vulnerability.

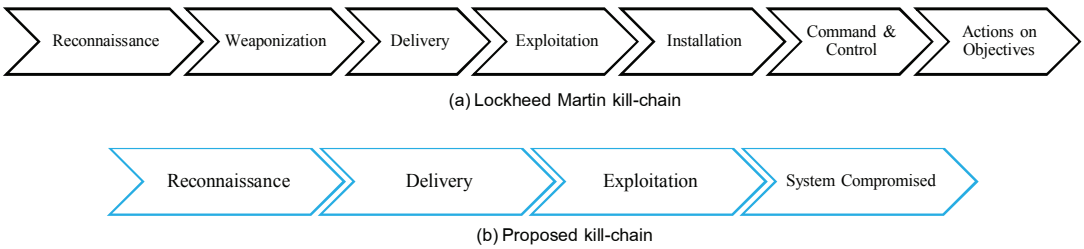

Fig. 2. Two kill-chain models: (a) Lockheed Martin kill-chain, and (b) Proposed kill-chain

3.1.3 Intrusion signature

An intrusion signature (or so-called 'detection rule') is a pattern of known attack that the IDS looks for in network traffic. When these are found, the detection system generates a report (or so-called 'alert log') to the security administrator. To implement the correlation logic, each intrusion signature is mapped to the kill-chain phase, based on its objective activities as shown in Table 1. For example, the Snort's signature number 1911 is mapped to the exploitation phase because its summary detail (Table 2) shows that the objective of this event is to exploit the vulnerability of sadmind.

3.1.4 Vulnerability database

Vulnerability database is a set of security flaws which arise from computer system design, implementation, maintenance, and operation. In this research, the vulnerability database is populated with all the existing vulnerability knowledge in the National Vulnerability Database (NVD) (NIST Computer Security Division, 2005) which is the U.S. Government repository of standards based vulnerability management data to support manual and automated analysis. Each NVD entry associates with a unique identifier following the Common Vulnerability Enumeration (CVE) (The MITRE Corporation, 2006) standard, with classification according to the Common Weakness Enumeration (CWE) (The MITRE Corporation, 2010) catalog. The affected software is identified in the Common Platform Enumeration (CPE) (NIST Computer Security Division, 2007) namespace, and a relevance rank is computed according to the Common Vulnerability Scoring System (CVSS) (NIST Computer Security Division, 2016).

Table 1. Snort signature in each kill-chain phase

Kill-chain phase	Snort signature
Reconnaissance	sid:384, sid:408, sid:585, sid:1957, ...
Delivery	sid:648, sid:1390, sid:2075, sid:2077, ...
Exploitation	sid:718, sid:1251, sid:1911, sid:1912,...
System Compromised	sid:104, sid:105, sid:108, sid:610, ...

Table 2. Summary detail of Snort's signature number 1911

Sid:1911
This event is generated when an attempt is made to exploit a buffer overflow associated with the Remote Procedure Call (RPC) sadmind.

3.1.5 Network context

Network context consists of information relating to the organization's network including IP address, operation system, application, type of hardware, the application vendor, and the hardware vendor.

3.2 Mediator layer

This layer consists of the ontology repository and alert analysis module. As shown in Fig. 3, the ontology repository consists of six main ontologies as alert ontology, signature ontology, kill-chain phase, vulnerability ontology, host ontology, and asset ontology. Instances of proposed ontology are depicted in Fig. 4.

3.3 Application layer

There are two user groups in this system, namely, the expert domain user and the security administrator user. The expert domain user performs the knowledge required for analyzing alert messages. The security administrator uses the information obtained from the analysis system to manage the organization's network.

4. Alert analysis module

Alert analysis module consists of two phases, namely, (1) information acquisition phase and (2) decision phase as shown in

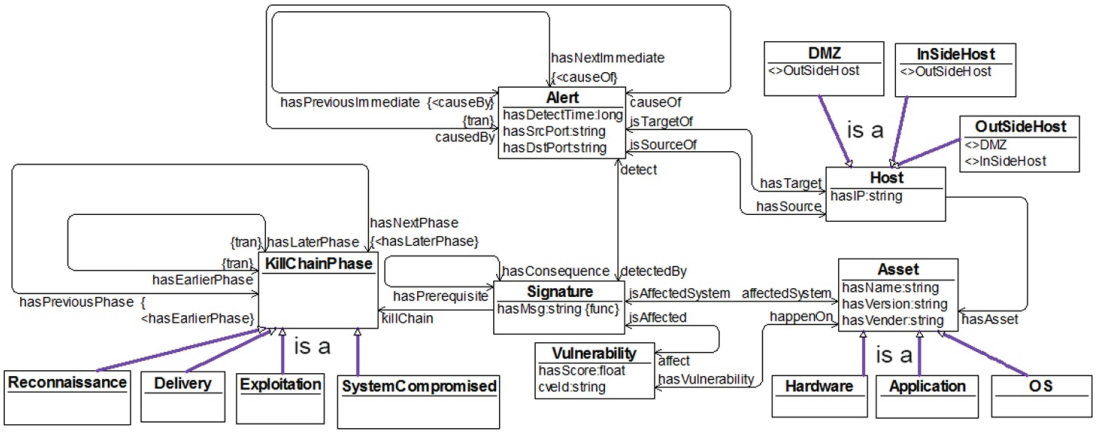

Fig. 3. Conceptual ontology of intrusion alert analysis approach

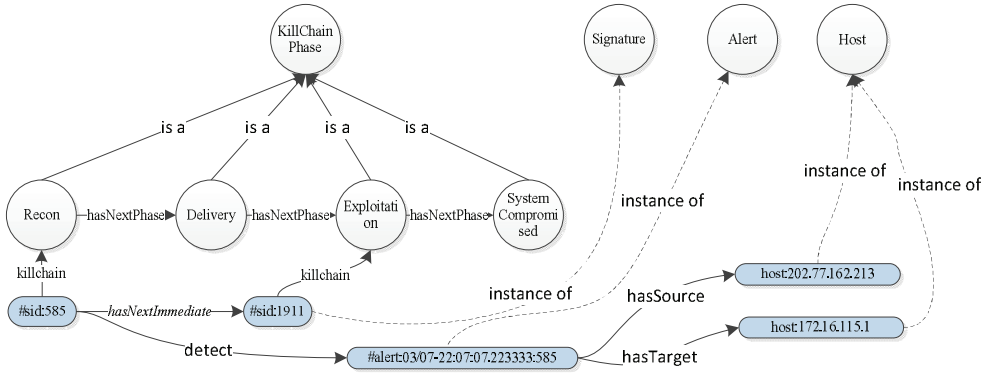

Fig. 4. Example of instances in the proposed ontology

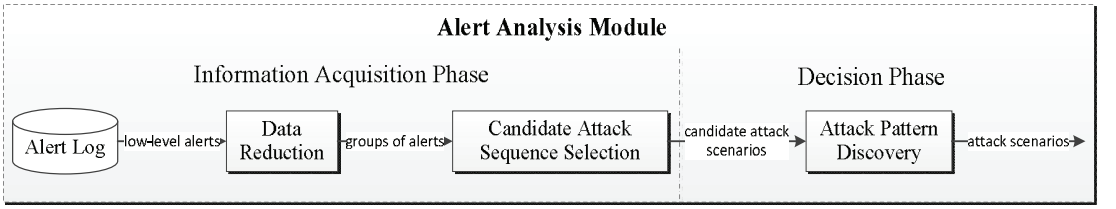

Fig. 5. Processes of Alert analysis module

Fig. 5. Firstly, information acquisition phase analyzes low-level alert messages and extracts information (i.e., the candidate attack scenarios) using the expert knowledge represented in the ontology. Secondly, decision phase discovers the attack scenarios from information, which is the result of previous phase. Each phase is described in detail as follows.

4.1 Information acquisition phase

The objective of this phase is to extract the candidate attack patterns from low-level alerts. There are two processes, namely, data reduction process and candidate attack sequence selection process.

4.1.1 Data reduction process

Regards compromising victim host, an attacker may technically use single or multiple sources to break and enter a target host until the goal is reached or the destination host is changed.To cope this circumstance, the goal of data reduction process is to reduce data complexity by removing irrelevant alerts that are divided into groups according to the destination host and as this research analyzes vulnerability in the local network, some alerts with a destination host outside the local network in each group are removed by SPARQL in Fig. 6.

```
PREFIX :<http://it.msu.ac.th/iams.owl#>
DELETE {?alert ?predicate ?object}
WHERE{
   ?alert :hasTarget ?host.
   ?alert ?predicate ?object.
   ?host a :OutSideHost.
}
```

Fig. 6. SPARQL for removing outside target *al*erts

```
PREFIX :<http://it.msu.ac.th/iams.owl#>
DELETE {?alert ?predicate ?object}
WHERE {
   ?alert :hasTarget ?host.
   ?alert :detectedBy ?sid.
   ?alert :hasDetectTime ?time.
   ?alert ?predicate ?object.
   {
      select ?sid ?host (min(?time) as ?min)
      WHERE
      {
         ?alert :hasDetectTime ?time.
         ?alert :detectedBy ?sid.
         ?alert :hasTarget ?host.
      }
      group by ?sid ?host
   }
   filter(?time > ?min)
}
```

Fig. 7. SPARQL for removing duplicate alerts on each host

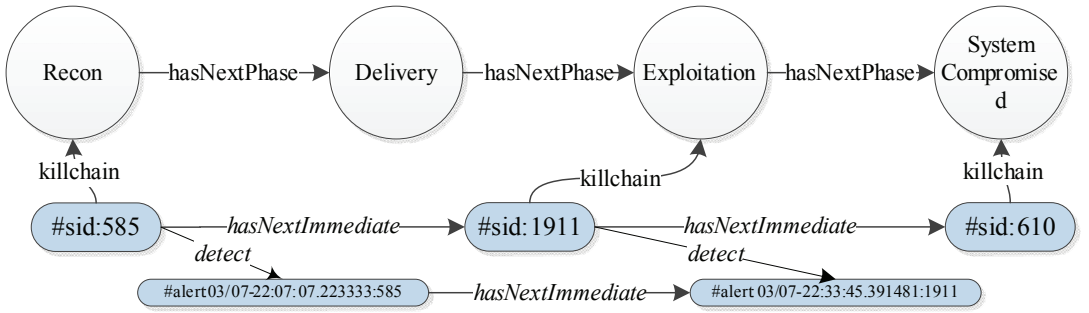

Fig. 8. The example of Candidate Attack Sequence Selection Process

Duplicate alerts generated by the same IDS signature on each target host are then removed by SPARQL in Fig. 7

4.1.2 Candidate attack sequence selection process

This process aims to create a correlation between the alerts that are linked to informing alerts as the real attack pattern. The completed linking is based on the kill-chain phase in order to link hasNextImmedaiate (as object property) between alerts. For example, alerta that belongs to the Recon phase will initially be linked by hasNextImmedaiate and then it will be sent to alertb that belongs to the next phase called Delivery. However, if no alerts are reported in the nearest phase, process still links alters to the nearest adjacent phase as shown in Fig. 8. Alert1(#sid:585) in Recon phase is linked by hasNextImmedaiate to alert2(#sid:1911) in the Exploitation phase. This occasion happens due to no alerts in the Delivery phase reported and this performance is carried out by SPARQL as shown in Fig. 9. It is a calculation to find the nearest phase of individual alerts that later will be linked to another nearest adjacent phase. We can create the attack pattern although lack of informing alerts from any phase. Overall, the attack pattern is derived using SPARQL, and it appears to be the real attack pattern that will be considered in the next process to eventually find the real attack pattern that affects the target host.

```
PREFIX :<http://it.msu.ac.th/iams.owl#>
DELETE {?subject :hasNextImmediate ?object .}
INSERT {?subject :hasNextImmediate ?object .}
WHERE {
 ?subject :hasTarget ?host.?object :hasTarget ?host.
 ?host :hasIP ?ip.?sids :detect ?subject.
 ?sido :detect ?object.?sids :hasMsg ?msgs.
 ?sido :hasMsg ?msgo.?sids :killChain ?killchains.
 ?sido :killChain ?killchaino.
 ?killchains :hasLaterPhase ?killchaino.
 { select ?killchains ?killchainco (count(?killchainp) as ?path){
    ?killchains :hasNextPhase+ ?killchainp.
    ?killchainp :hasNextPhase* ?killchaino.
  } group by ?killchains ?killchaino
 } .
 { select ?s (min(?path2) as ?min){
    ?s :detectedBy ?sids.?s :hasTarget ?host.
    ?sids :killChain ?killchains. ?sido :killChain ?killchaino.
   ?killchains :hasLaterPhase ?killchaino.
    ?object :detectedBy ?sido.?object :hasTarget ?host.
    { select ?killchains ?killchaino (count(?killchainp) as ?path2)
      { ?killchains :hasNextPhase+ ?killchainp.
        ?killchainp :hasNextPhase* ?killchaino.
      } group by ?killchains ?killchaino
    } .
  } group by ?subject
 }. filter (?path = ?min).
}
```

Fig. 9. SPARQL for creating relationship between alerts

4.2 Decision Phase

To launch a DDoS attack, the attacker must control a large number of hosts in order to attack a target host. To begin, the attacker searches for and compromises vulnerable hosts in order to seize control of them. As a result, the target host's attack is successful. Typically, the attacker use the same tools and techniques to compromise a variety of hosts on the same network. Thus, this research employs an association rule algorithm to discover attack situations through an analysis of their frequency.

Table 3. The example of Candidate Attack Sequence

Candidate Attack Sequence	Target Host
sid:1957,sid:1911,sid:610	172.16.112.10
sid:384,sid:1911,sid:610	172.16.112.10
sid:585,sid:1911,sid:610	172.16.112.10
sid:15934,sid:1679,sid:610	172.16.112.50
sid:1957,sid:1679,sid:610	172.16.112.50
sid:384,sid:1679,sid:610	172.16.112.50
sid:585,sid:1679,sid:610	172.16.112.50
sid:716,sid:1679,sid:610	172.16.112.50
sid:15934,sid:1911,sid:610	172.16.112.50
sid:1957,sid:1911,sid:610	172.16.112.50
sid:384,sid:1911,sid:610	172.16.112.50
sid:585,sid:1911,sid:610	172.16.112.50
sid:716,sid:1911,sid:610	172.16.112.50
sid:1957,sid:1911,sid:610	172.16.115.20
sid:384,sid:1911,sid:610	172.16.115.20
sid:402,sid:1911,sid:610	172.16.115.20
sid:408,sid:1911,sid:610	172.16.115.20
sid:409,sid:1911,sid:610	172.16.115.20
sid:585,sid:1911,sid:610	172.16.115.20

4.2.1 Attack pattern discovery process

While the association rule can represent the implicit correlation between alerts in a collection of multi-step attacks, it cannot represent the attack sequence. Thus, rather than a set of low-level alerts, this research uses the association rule concept to discover real attack patterns from a set of candidate attack patterns, as shown in Table 3.

An association rule has the form of $X \rightarrow Y$ where is the itemset of attack steps before the final step of the candidate attack sequence, Y is the final step of the candidate attack sequence and $X \cap Y = \phi$. The significance of an association rule is determined by two measurements, support and confidence (Jungja, Ceong, & Yonggwan, 2009). The support measurement is the percentage of the candidate attack sequence ($X \cup Y$) that exists in the dataset. The confidence measurement is an indication of how often the attack sequence has been found to be true. The computations of support and confidence are shown in equations (1) and (2), respectively, where P(S) is the probability of the attack sequence which contains the itemset.

$$Support(X \rightarrow Y) = P(X \cup Y) \quad (1)$$

$$Confidence(X \rightarrow Y) = \frac{P(X \cup Y)}{P(X)} \quad (2)$$

The minimum support and confidence thresholds in this study are 10% and 60%, respectively.

5. Implementation

This research implemented an intrusion alert analysis framework using the concepts discussed in the previous section. The program was written in JAVA, with Jena ontology API (Apache, 2017), to perform the ontology. To store and infer the ontology, GraphDB 7.1 (Ontotext, 2017) was used as the ontology repository. Alert message logs were generated by Snort 2.9.7.6 ("Snort," 2017) with 1,737 rules. The experiments were conducted using a machine with an Intel Core i5 2.50 GHz and 8 GB of RAM with Windows 10.

6. Experiment and Evaluation

6.1 Dataset

To evaluate the performance of the proposed approach, this research used the DARPA 2000 LLDOS 1.0 dataset (MIT Lincoln Lab, 2002) in the experiments. The LLDOS 1.0 dataset is a DDoS attack scenario which compromises a variety of hosts and launches a DDoS attack at an off-site server from the compromised hosts. The data set includes the network audit data collected in both the DMZ (Demilitarized Zone) and inside the evaluation network. There are two datasets in each evaluation network. First, the content dataset consists of both normal and attack network traffics. Second, the labeling dataset, which is the answer of the dataset, contains only attack network traffics.

In the experiment, Alert logs were generated by replaying the content and

labeling datasets in an isolated network monitored by a Snort IDS. The performance of the proposed approach is represented by the comparison of the analysis results of the content dataset with the analysis results of the labeling dataset.

6.2 Experimental results

To evaluate the effectiveness of the proposed method in reducing irrelevant alerts, this research compared the experiment results of the content data with the labeling data. The results indicated that the proposed method can reduce irrelevant alerts from the content data similar to the labeling data The proposed method can complete the analysis of inside data for 34,298 alerts within about 70 seconds, and DMZ data for 1,008 alerts within about 10 seconds.

As shown in Fig. 10, the number of alerts in DMZ data is decreased from 1,008 to 15, a decrease represented by 98.5% of the number of alerts before performing analysis process. Similarly, the number of alerts in Inside data as shown in Fig. 11 is decreased from 34,298 to 15, a decrease represented by 99.9% of the number of alerts due to irrelevant alerts in both datasets are removed before performing analysis process.

To demonstrate the effectiveness of this method in reducing false alerts, this research compared the attack graph without the reducing method with the attack graph with the reducing method. Fig. 12 (a) shows the attack graph without the reducing method for host 172.16.112.50, whereas Fig. 12 (b) shows the attack graph with the reducing method when false alerts have been removed.

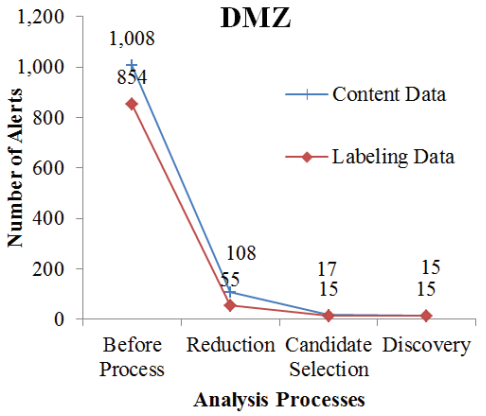

Fig. 10. Comparison of the number of remaining alerts in DMZ data

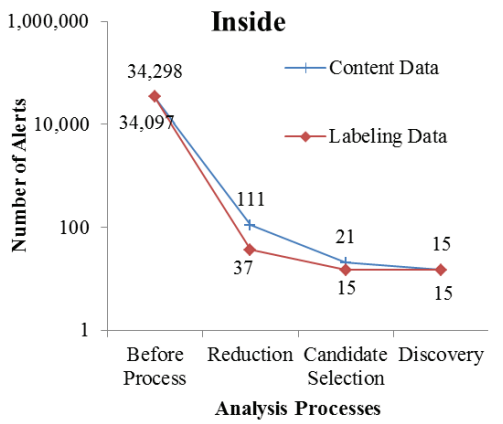

Fig. 11. Comparison of the number of remaining alerts in inside data

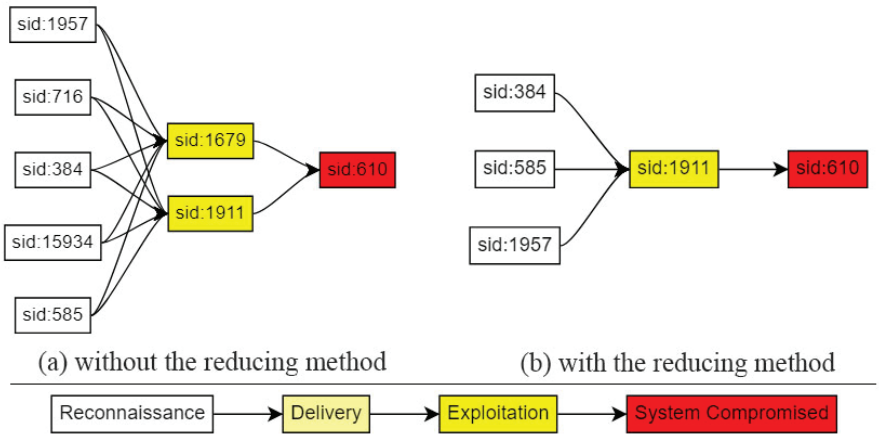

Fig. 12. Comparison of attack graph on host 172.16.112.50

The alert messages represented in the attack graph are described in Table 4.

Table 4. Snort alert messages

Signature ID	Alert Message
sid:384	PROTOCOL-ICMP PING
sid:585	PROTOCOL-RPC portmap sadmind request UDP attempt
sid:716	INFO TELNET access
sid:610	PROTOCOL-SERVICES rsh root
sid:1679	ORACLE describe attempt
sid:1911	PROTOCOL-RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt
sid:1957	PROTOCOL-RPC sadmind UDP PING
sid:15934	PROTOCOL-DNS dns response for rfc1918 172.16/12 address detected

Table 5 shows the results of the discovery process. The candidate attack patterns which have support greater than 10% and confidence greater than 60% are selected. These attack sequences can be reconstructed as attack graphs as shown in Fig. 12 (b).

Table 5. Results of the discovery process on Inside Data

Attack Pattern	Support (%)	Confidence (%)
{sid:1957,sid:1911}sid:610	16%	100%
{sid:384,sid:1911}sid:610	16%	100%
{sid:585,sid:1911}sid:610	16%	100%

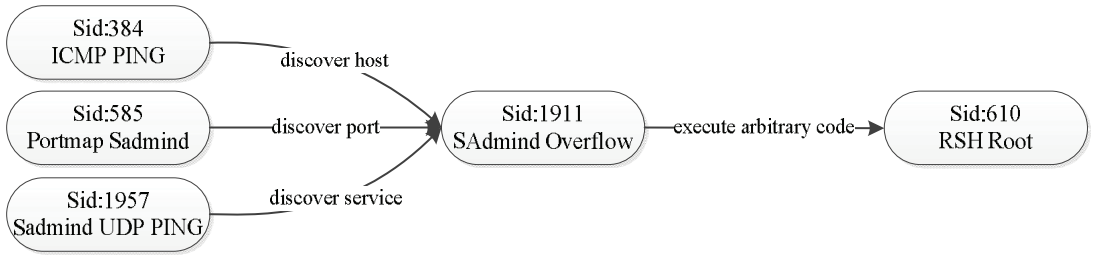

Fig. 13. Attack scenario graph

The final result of the experiments shows the attack graphs on three hosts: 172.16.112.10, 172.16.112.50, and 172.16.115.20. This result indicates the multi-stage attack scenarios and the compromised hosts. According to the promising cyber kill-chain, the alerts in the system compromised phase indicate that these hosts are compromised by attacker. The vulnerability of each host can be determined from the alerts in the exploitation phase. For example, Fig. 13 shows that sid:1911 (PROTOCOL-RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_D-OMAIN overflow attempt), which is generated when the intruder tries to attack a vulnerable sadmind (CVE-1999-0977) to obtain root access to the remote host. This information can help security administrators to solve the direct problem and save analysis time.

The proposed approach was compared to other alert-correlation-related techniques. The evaluation was performed on the DARPA 2000 LLDOS 1.0 dataset. To demonstrate the proposed approach's performance, the most widely used metrics, namely completeness and soundness, are used. Completeness quantifies the capacity to correlate relevant alerts as the ratio of accurately correlated alerts to the total number of related alerts in the same attack scenario. The soundness metric measures how accurately the alerts are correlated as a ratio of the number of correctly correlated alerts to the total number of correlated alerts.

As shown in Table 6, the experimental evaluation of the proposed approach is better than the related alert-correlation approaches in terms of completeness and soundness. The key factor of the proposed approach is to achieve the high effectiveness by a combination of knowledge-based ontology and data mining techniques. In essence, not only the knowledge-based ontology is used to remove irrelevant alerts and create the candidate attack scenarios from low-level alert log but also the association rule is used to discover the real attack scenarios from the candidate attack scenarios. In other words, the proposed approach performs based on high-quality data without irrelevant data.

7. Conclusions and Discussion

We proposed an intrusion alert analysis framework using a combination of semantic web and data mining approach. We

Table 6. Comparison of alert correlation approaches using DARPA 2000 LLDOS 1.0 dataset

Approach	Completeness	Soundness
Ning, Cui, and Reeves (2002)	93.96%	93.96%
Safaa O Al-Mamory and Hong Li Zhang (2007)	88.7%	100%
W. Li *et al*. (2007)	92.2%	Not provided
Saad and Traore (2013)	100%	99.70%
Proposed Method	100%	100%

implemented and evaluated the framework through experiments using the LLDOS 1.0 dataset. The results indicated that the proposed approach can reduce false alerts and extract information from the alert logs including the compromised hosts, the attack patterns, and the vulnerability of the compromised hosts. The comparison of the proposed approach with the related alert-correlation approaches shows that the proposed approach is better than the related alert-correlation approaches in terms of completeness and soundness.

However, there are two limitations of the proposed approach. First, this method maps IDS signatures to the kill-chain manually. The concept of this research is to extract information from many low-level alerts using expert knowledge, and a mapping process is performed by expert domain to avoid mistakes. Another limitation is that the detection rate depends on the capability of the IDS and its signature rules. This approach can reduce the false alarm rate but cannot improve the IDS detection rate. In future work, I plan to investigate a method to improve the IDS detection rate by selecting appropriate signature rules on each local network.

8. Acknowledgements

The researcher would like to express the gratitude to Faculty of Informatics, Mahasarakham University for the support of budget for this research.

9. References

Al-Mamory, S.O., & Zhang, H.L. (2007). Scenario discovery using abstracted correlation graph. In *the International Conference on Computational Intelligence and Security (CIS)* (pp. 702-706). IEEE. https://doi.org/10.1109/CIS.2007.21

Apache. (2017). *Apache Jena*. Retrieved 20 March 2017, Retrieved from http://jena.apache.org/

Barik, M.S. (2018). AGQL: A query language for attack graph based network vulnerability analysis. In *the Fifth International Conference on Emerging Applications of Information Technology (EAIT)* (pp. 1-4). https://doi.org/10.1109/EAIT.2018.8470430

Bryant, B.D., & Saiedian, H. (2017). A novel kill-chain framework for remote security log analysis with SIEM software. *Computers & Security*, 67, 198-210. https://doi.org/10.1016/j.cose.2017.03.003

de Alvarenga, S.C., Barbon, S., Miani, R.S., Cukier, M., & Zarpelão, B.B. (2018). Process mining and hierarchical clustering to help intrusion alert visualization. *Computers & Security*, 73, 474-491. https://doi.org/10.1016/j.cose.2017.11.021

Debar, H., Curry, D.A., & Feinstein, B.S. (2007). *The intrusion detection message exchange format (IDMEF)*. Request for Comments (Experimental).

García, V., Mollineda, R.A., & Sánchez, J.S. (2008). On the k-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications*, 11, 269-280. https://doi.org/10.1007/s10044-007-0087-5

Hahn, A., Thomas, R.K., Lozano, I., & Cardenas, A. (2015). A multi-layered and kill-chain based security analysis framework for cyber-physical systems. *International Journal of Critical Infrastructure Protection*, 11, 39-50. https://doi.org/10.1016/j.ijcip.2015.08.003

Jungja, K., Ceong, H., & Yonggwan, W. (2009). Weighted association rule mining for item groups with different properties and risk assessment for networked systems. *IEICE TRANSACTIONS on Information and Systems*, 92(1), 10-15.

Li, W., & Tian, S. (2008). XSWRL, an extended semantic web rule language. In *the Intelligent Information Technology Application (IITA)* (pp. 437-441). IEEE. https://doi.org/10.1109/IITA.2008.411

Li, W., & Tian, S. (2010). An ontology-based intrusion alerts correlation system. *Expert Systems with Applications*, 37(10), 7138-7146. https://doi.org/10.1016/j.eswa.2010.03.068

Li, W., Zhi-tang, L., Dong, L., & Jie, L. (2007). Attack scenario construction with a new sequential mining technique. In *the Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD)* (pp. 872-877), IEEE. https://doi.org/10.1109/SNPD.2007.395

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250, 113-141. https://doi.org/10.1016/j.ins.2013.07.007

Martin, L. (2014). *Cyber Kill Chain®*. Retrieved from http://cyber.lockheedmartin.com/hubfs/Gaining_the_Advantage_Cyber_Kill_Chain.pdf

MIT Lincoln Lab. (2002). *DARPA intrusion detection scenario specific datasets*. Retrieved from https://www.ll.mit.edu/ideval/data/2000/LLS_DDOS_1.0.html

Ning, P., Cui, Y., & Reeves, D.S. (2002). Constructing attack scenarios through correlation of intrusion alerts. In *the 9th ACM Conference on Computer and Communications Security (CCS)* (pp. 245-254). ACM. https://doi.org/10.1145/586110.586144

NIST Computer Security Division. (2005). *NVD-national vulnerability database*. Retrieved from https://nvd.nist.gov

NIST Computer Security Division. (2007). *CPE-common platform enumeration*. Retrieved from https://nvd.nist.gov/cpe.cfm

NIST Computer Security Division. (2016). *CVSS-common vulnerability scoring system*. Retrieved from https://nvd.nist.gov/cvss.cfm

Njogu, H.W., Jiawei, L., Kiere, J.N., & Hanyurwimfura, D. (2013). A comprehensive vulnerability based alert management approach for large networks. *Future Generation Computer Systems*, 29(1), 27-45. https://doi.org/10.1016/j.future.2012.04.001

Ontotext. (2017). *Graph DB™. 7.1*. Retrieved from: http://ontotext.com/products/graphdb/

OWASP. (2017). *Application security risks-2017, open web application security project (OWASP)*.

Saad, S., & Traore, I. (2013). Semantic aware attack scenarios reconstruction. *Journal of Information Security and Applications*, 18(1), 53-67. https://doi.org/10.1016/j.jisa.2013.08.002

Sadighian, A., Fernandez, J.M., Lemay, A., & Zargar, S.T. (2014). ONTIDS: A highly flexible context-aware and ontology-based alert correlation framework. In Danger, L.J., Debbabi, M., Marion, J.-Y., Garcia-Alfaro, J., & Heywood, Z.N. (Eds.), *Foundations and Practice of Security: 6th International Symposium (FPS)* (pp. 161-177). Springer: Cham.

Snort. (2017). Retrieved from: https://www.snort.org/

The MITRE Corporation. (2006). *CVE-common vulnerabilities and exposures*. Retrieved from: https://cve.mitre.org/

The MITRE Corporation. (2010). *CWE-common weakness enumeration*. Retrieved from https://cwe.mitre.org

Yu-Xin, D., Hai-Sen, W., & Qing-Wei, L. (2008). Intrusion scenarios detection based on data mining. In *the International Conference on Machine Learning and Cybernetics (ICMLC)* (pp. 1293-1297). IEEE. https://doi.org/10.1109/ICMLC.2008.4620604

Yuan, B., Pan, Z., Shi, F., & Li, Z. (2020). An attack path generation methods based on graph database. In *the 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (pp. 1905-1910). IEEE. https://doi.org/10.1109/ITNEC48623.2020.9085039