Investigating cyber alerts with graph-based analytics and narrative visualization

Neda AfzaliSeresht
ISILC and Data61
Victoria University and CSIRO
Melbourne, Australia
neda.afzaliseresht@live.vu.edu.au

Yuan Miao
ISILC
Victoria University
Melbourne, Australia
yuan.miao@vu.edu.au

Qing Liu
Data61
CSIRO
Hobart, Australia
Q.Liu@data61.csiro.au

Assefa Teshome
ISILC
Victoria University
Melbourne, Australia
assefa.teshome@vu.edu.au

Wenjie Ye
ISILC
Victoria University
Melbourne, Australia
Wenjie.Ye@vu.edu.au

2020 24th International Conference Information Visualisation (IV) | 978-1-7281-9134-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/IV51561.2020.00090

***Abstract*—In real-world situations, several threat alerts are being investigated by the specialised staff. In order to prompt response to serve incidents or ignore false alarms, alerts are prioritised and analysed. Security professionals rely on information provided in the alert message. Insufficient information in alert messages raises challenges for security analysts that require them to keep track of all internal and external sources to identify the relevant information. In this paper, a Narrative Analytics-Assisted System (NAAS) is proposed, and a knowledge graph is used in the proposed system to present the relationships. The knowledge graph is proposed to capture the complex relationships between the alert and relevant information from the Internal and External knowledge bases to reduce the cognitive effort in information digestion and to understand a wealth of security data. To enable cooperation in the cyber risk management process, it is an inevitable necessity to generate the knowledge graph and interpret it in a human-friendly format. The current machine-friendly formats for reporting incidents from alerts are complex and of an extensive nature. These characteristics hamper the readability and contribution, therefore preventing humans from understanding and being up to date about the incident. NAAS contains four life cycles to assist an analyst to have a better perception of the elements of the environment by involving more staff in the risk management: (1) Analyses the alert, (2) designs the knowledge graph with the natural language sentences, (3) automatically implements the incident report in natural language by applying novel storytelling techniques from the knowledge graph, and (4) maintains it with the contribution of different levels of expertise. The performance of various NAAS's cycles is demonstrated in a case study with an example scenario from the Security Operations Centre (SOC) at an educational institution, highlighting its useability.**

***Keywords*-Cybersecurity, Storytelling, knowledge graph, Human-interactive**

I. Introduction

A. Cyber threat investigation, a problem for security analysts

Every day millions of activities and attempts are recorded in computer systems. Given the huge volume of network data and raised alerts (often uncertain and ambiguous), need to be filtered to identify the most critical ones for analysts [1]. The lack of data leads to an insufficient understanding of security alerts. Situation awareness systems are designed to make sense about what is monitored in the network. Still security data is provided as an individual piece of information, not in a comprehensive model to help analysts to understand the full path of the vulnerability [2]. As compared to the vast volume of logged events, the security team of an organization is few, and the specialised staff is investigating only a fraction of threat alerts. Security analysts are often overwhelmed by a large amount of alerts, especially in a small organization. It is still impractical to process these alerts by the limited cybersecurity professionals; If the alerts can be integrated into a more comprehensible form like summarised stories, the further alert investigation is improved through a better contextualization of cyber situation awareness. A narrative explanation is a state-of-the-art for analysis facilitation; however, rarely efforts have been made in cybersecurity.

B. Knowledge beyond security team is needed for the analysis

Most of the alerts are false alarms. To properly interpret the potential risk of an event, it often requires knowledge beyond security department itself. Here are two examples:

As the first example, consider a scenario where a server of organisation XYZ is used for temporary storage and web testing, which is labelled as a non-critical host. Most of the alerts regarding the server can be omitted unless it is a serious breach. In current financial reporting season, the server was borrows by the financial department for financial reporting and budget planning temporarily. It now keeps critical information and security level shall be lifted to the highest in the organisation to monitor all alerts closely. However, the role transition of the server was not passed on to the security team. Its users at finance department have little expertise in security. A big security hole is left open

2375-0138/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00090

to attackers.

In the second example, consider a scenario where an organisation repeatedly receives a high volume of a security breach from an internal host. This is a typical symptom of an attack, and the security system shall block the host and related ports. A further analysis involving staff from different departments revealed that the host is an experimental server in department A's laboratory; Which is used to test game engines' cloud end under development. The internal host and cloud server require low-level communication and configuration with the corresponding security exceptions.

In both examples, the alerts' analysis requires knowledge from the security team and other departments, which cannot be modeled and integrated with the alert analysis; either false alarms could be triggered, or high-risk alerts could be neglected. Therefore, generally engaging more staff from different departments is needed to solve the issues on the examples. It requires developing a shared understanding of cyber situation awareness (currently restricted to security professionals). Generating a report in a storytelling manner with an analytical graph to present the relationships is human-readable, facilitating comprehension, and effectively allowed human involvement in the process.

C. Association analysis based on up-to-date internal and external information

Security analysts need to gather internal (The system integrated the network infrastructure) and external (vulnerability, cyber threat, and intrusion alert) information from various sources to feed the correlation process and support analysis of security events for reasoning the alerts. [3].

In generic terms, a comprehensive and integrated up-to-date information to security experts includes any information that can be used to characterise the situation of an IT entity that is considered as linking to internally and externally available information [3]. Internal and external information is continuously updating. Implementing approaches to integrate the information into the data model to make full use of cybersecurity-related details from various resources, and associating all these security-related knowledge is difficult and usually has expensive modification cost [4]. One of the major challenge is the rapid varying of the network environments which has a potential impact on security posture, i.e. machines added and removed, various patches applied, applications installed/uninstalled, or confidential data is uploaded or deleted [5].

> "The problem is not lack of information, but rather the ability to assemble disparate pieces of information into an overall analytic picture for situational awareness" [6].

At present, such approaches have been used in the application of knowledge graphs that consolidates data into a comprehensive picture [5]. Narrative reports accompany the analytical graph to compensate for the lack of data, leads to better understanding.

D. Graph-based analytical and Storytelling representation

NAAS is proposed to enables security experts to analyze alerts and represent the incidents' intelligent analysis in enriched textual narrative reports. This approach combines an innovative way of presenting the alert and corresponding data in a graph with an interactive human-friendly component to analyze and edit the threat information, as well as the interpretation of the graph's knowledge in a human-readable narrative report.

We choose our analytical strategy as a graph-based model to bring together isolated data and the varying update intervals of the data sources from the internal and external of an organization. The knowledge graph is built to carry out an alert correlation analysis. The knowledge graph is gracefully supported human-friendly sentences that might be needed for revising/expanding the information due to digging more security knowledge. It is suited for the integration of different types of information that security analysts might want to correlate with. It also would engage staff with different levels of expertise from many departments. The graph-based model is used for expanding the predictions of network attacks by retrieving the vulnerabilities of the target host and then retrieving the vulnerability related to the alert.

NAAS converts the alerts and association knowledge into human-comprehensible stories since the success and effectiveness of any security measure rely heavily on the contribution of security experts [7]. The summarised stories generate a holistic view for better tracing security alerts, even aware people with no security specialised knowledge about the incident. Detecting and tracing malicious events with a combination of both powers of machines and human beings is a more reliable method in the processing of a large number of alerts [8]. With the summarized stories, our method will help to establish an understandable common ground among human beings and machines.

Storytelling is a method to assist or engage people to explore and interpret complex real-world problems. In other words, telling stories in the problem formulation stage merges synthesis and analysis, and makes abstract concepts more concrete [9]. Storytelling can be used as a knowledge representation method to highlight the semantic and implied information from events into a human-readable format [10].

Given the huge volume of events and corresponding alerts, the stories need to be generated automatically. In this paper, interpreting security alerts from different aspects, from a holistic view to technical solutions, in a natural language is proposed. Therefore, expert and non-expert analysts could analyse data beyond the security rules and policies.

The rest of the paper is organised as follows. First, we provide an overview of related works. NAAS and its development cycles are introduced in section III. Then, the

aspects of the designing NAAS will be explained. Section IV illustrates the usefulness of the NAAS by means of an example use case. The paper is ending with a conclusion and an outlook on the current limitations.

II. Background

The first place analysts begin their threat hunting investigation is an alert message. Alert correlation is carried out with a mix of machine algorithms and human investigation [11]. Given the complexity of modern systems and cyber attacks, algorithms have not successfully applied sufficient context to the message or have enough intelligence to understand why certain alerts were important. Furthermore, human beings have to be involved in the analysis process, human-as-a-security-sensors into security analytic [12]. A major limitation exists in the current alert analysis process, which relates to the messages' verbosity without annotation to cope with the enormous volume of events [13].

The abundance of available cybersecurity knowledge raises challenges for security analysts and professionals who have to keep track of all the available sources and identify relevant information within them [14]. Even the overload of cybersecurity knowledge available in various formats to human analysts and their incapability to determine the most relevant information which is not easily readable for humans can be seen as a data quality problem [15].

The data collected from humans may be generated automatically or manually into machine data to update the knowledge bases. To deriving relationships to machine data, rule-based correlation and aggregation are the famous approaches. In order to facilitate the definition of rules, it can be helpful to visualize the generated data and separate the rules from the detection model [12]. Many state-of-the-art studies have been carried out, such as [7], [2], [16] and [17]. They proposed visualization of the knowledge graph to aid security analysts in their investigation.

In this paper, to enable cooperation in exchanging knowledge between humans and avoid peering into its internal structure in machine-readable formats, the rules are defined in a human-understandable format. Besides, generating narrative reports with the interpretation of the knowledge graph about the threat alerts is facilitate comprehension because of human-readability, and effectively provides better ground for faster response.

III. Narrative Analytics-Assisted System (NAAS)

This research's initially overarching aim was to present the vulnerabilities and threat factors and internal elements associated with the threat alert, which is effectively understandable by humans for alert validation. NAAS was aimed to reduce the cognitive load imposed on cybersecurity analysts while processing for alerts based on the proposed idea by Afzaliseresht et al. in [18]. A primarily revised analytical strategy of knowledge graph mode is presented in this paper, which is able to generate the automatic story by exploring the subgraphs. A shared understanding of cyber situation awareness is developed to engaged more people in updating the knowledge.

The NAAS comprises four cycles to assist an expert in cyber situation awareness. As a result, full awareness about the alert situation from various heterogeneous sources such as different departments and owners can be achieved. The main development life cycles are illustrated in Figure 1.

- **Analysis Cycle**: although monitoring systems help filter through millions of logged events and generate security alerts, final human assessment is still part of the process. The analyst analyses the alert by using the extracted information from the Internal and External knowledge bases. A draft threat report is written for sharing the knowledge with others. The knowledge bases are regularly updated based on the available updated information (from other staff or public resources).
- **Design Cycle**: the analyst puts the analytical results in a cyber threat intelligence report within a sequence of sentences, that is converted into the simple structure, then into Cypher queries. Given the machine-friendly rather than human-friendly format of such alerts, the interpretation of raised alerts and cyber threat intelligence information is still required. The cyber threat intelligence shared informally as text by cybersecurity analyst. Writing much relevant information in a human-friendly form deducts cognitive overload on currently limited cybersecurity resources. The sentences automatically convert into the well-defined structured format, then into the Cypher queries that are visualized in a knowledge graph.
- **Implementation Cycle**: a script was used to query the knowledge graph. The query results are specifications for matching subgraph patterns to complete the incident report template of interest. Storytelling is used as a knowledge representation method to highlight the graph's implicit and explicit knowledge and convert it into a human-readable format. The capability to automatically provide the story at different levels of details enabled to cater to various information needs and intended aims (i.e., low-level for the user at the financial department, high-level for top management).
- **Maintenance Cycle**: the graph, which is the narrative story accompaniment, is presented as a report, a ticket, or a post to the particular audiences (I.e., device administrator, risk owner, the manager, analytical experts, and others.). And the human-friendly of the report contributed towards broader audience engagement into cyber situation awareness (currently restricted to security analyst). As an example scenario, the infected device user can receive the storytelling report with the graph and obtains insight into the cyber situation in-

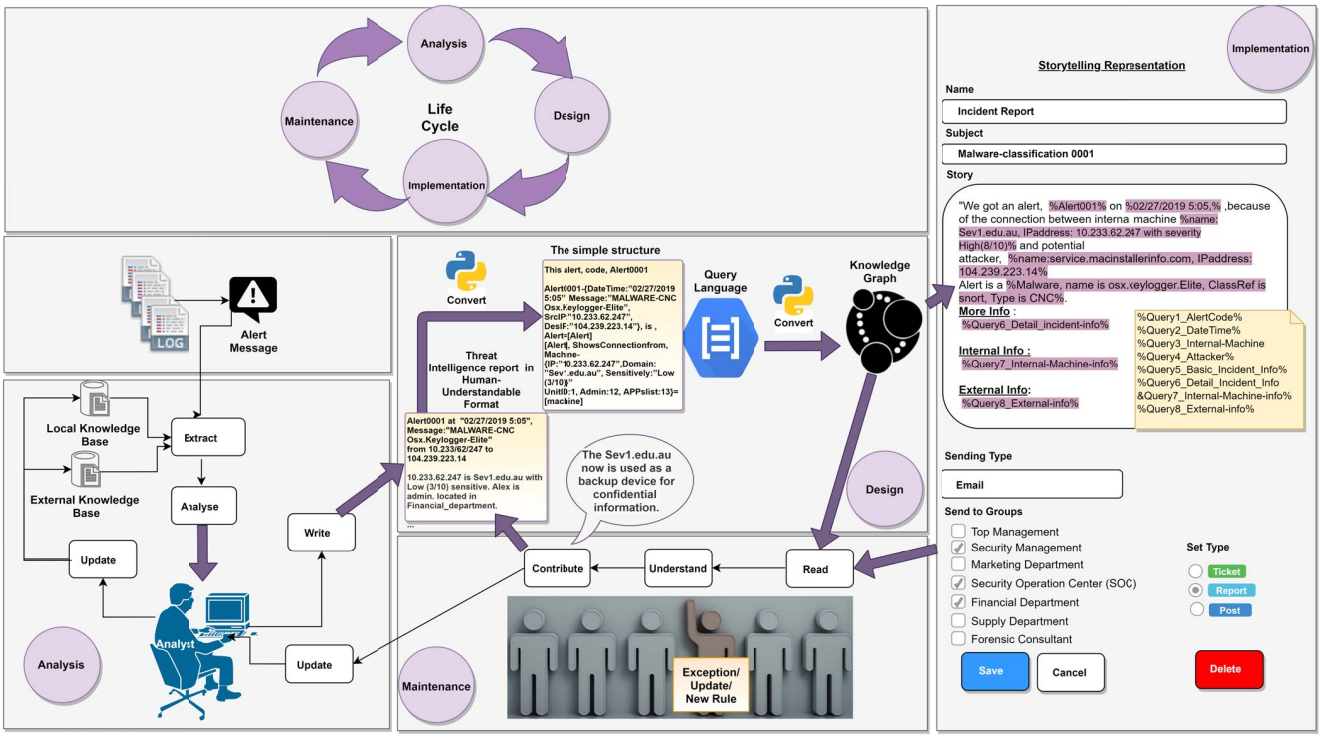

Figure 1. Overview of the NAAS's development life cycles made of four cycles (The story representation is the output of the implementation cycle which capable of revised based on the updated knowledge from the maintenance stage).

stantly, thus preventing further problem escalation. For instance, the sev1.edu.au was a primary server without confidential information. Although the server's security level was low because of non-critical information stored in it, the owner used it as a backup device for the confidential data without passing the role transition of the server to the security team. By reading the report and the analytical graph, the owner understood the current situation of the server is not updated, and the security professionals were not aware of its status. The user can contribute to complete the intelligent threat report and update the server's sensitivity.

NAAS includes the following four aspects.

A. *Knowledge Graph Construction*

The synthesized knowledge is visualized using a graph-like structure for supporting the analytical reasoning [19]. Usually, a historical collection of domain-specific knowledge is designed and developed in prior to constructing a knowledge graph [20]. But here, a predefined set of vocabularies and relationships are not required. The graph is intended to be a flexible middle-ware analytical tool to avoid a firm-solid set of rules. In other words, the initial alert messages and corresponding information from the Internal and External knowledge bases are combined in the graph to visualize the ongoing threats.

Since we want to have a flexible graph to be updated quickly and affordably, a human-friendly interactor is designed to convert simple vocabularies to nodes, properties, and relationships. The cyber ontologies and resources are not usually publicly available, and those were available did not provide sufficiently detailed to represent the repositories we target. On the other hand, the internal network environments are changed frequently and need to be updated manually. Therefore, a Python script taking human-readable sentences as inputs helps gather and facilitate mapping knowledge from different units of an organization, from/to other existing conceptualisations or ontologies.

We chose the Neo4j to explain graph data modeling [21], although it seems the simplified exchange syntax for ontology language is Resource Description Framework (RDF). Compared to Neo4j, RDF triples are machine-friendly syntax that are not proper for humans to understand. RDF is very strongly index-based, which should be defined in the triple-oriented-language, while Neo4j is navigational (implements index free adjacency) and stores the connections between connected entities without scanning indexes [21].

In addition, Neo4j is open source, significantly improved the processing efficiency of massive RDF data replication. And its Cypher query language is a very expressive query language, which is built ground-up for humans to perform graph queries. [22].

The following concepts are used to model a Neo4j graph [22]:

- **Nodes**: Concepts or entities in the domain.
- **Lables**: A tag for adding more meaning to nodes besides adding constraints and indices.
- **Relationships**: A directed, semantically connection between nodes to depict the relations between them.
- **Properties**: A key-value pairs that depict more information about nodes and relationships.

B. Knowledge bases

A knowledge graph is used to integrate multi-source heterogeneous data [4]. Two main domain knowledge bases (Internal and External) required to obtain contextual insight about an alert with the aggregation of isolated data.

The Internal knowledge base includes available information with domain knowledge of experts and the raw data collected from the security devices. The Internal knowledge base allows the exchange of explicit knowledge about the situation of an incident relevant to the company/institution.

The External knowledge base contains supplementary information that is collected by external companies and researchers. Existing instance data in the External knowledge base is comprised of the following information. (1) Whois[1] and additional information about IP address and domain registrants, (2) online scan engines such as Virus Total[2] or Threat Miner[3] that generate the "malicious" labels for a given URL or file (3) the Snort community rule set[4] (4) online open threat exchange repositories such as AlienVault[5] , Windows Defender Security Intelligence (WDSI)[6], and Symantec[7].

C. Natural language sentences into Cypher query

A graph consists of nodes, attributes and relationships. The knowledge entities as nodes and properties of entities as attributes must be recognized in the domain. Besides, how these entities and attributes are related to one another, and what entities are introduced at particular times should be captured [23]. In cybersecurity area, basic standards in information exchange formats were proposed to represent various aspects of cybersecurity. The most relevant standard is a Structured Threat Information eXpression (STIX) [24] by MITRE. STIX captures concepts at a high level for information sharing on cyber threat intelligence [23]. The machine-readable format makes it extremely challenging to understand the components and the relations between them [18].

To reason over statements expressed in these vocabularies, a human-friendly format improves the cyber threat intelligence accessibility for security experts. A Python script converts the natural language sentences into Cypher queries, which quickly transforms into a knowledge graph. The script assists the integration of different schemas and formats for the presentation of the graph. The NLTK library, which offers a sentence tokenizer function to split the words based on the grammar, is used to extract the entities, attributes, and relationships between entities from each human-readable report. Then, the extracted terms are categorized into nouns, verbs, and adjectives to describe the entities, relationships, and properties. The groups are automatically converted into a simple structure that can be transformed into Cypher queries. A few simple rules need to follow when simple structure sentences are created, for example:

1) {} is used to describe the property of the entity that could be used for searching
2) Relationship has to be one word, for example, in the expression: "Malware, is a, OSX-keylogger", "is a" is not correct, it shall be "is_a".
3) Shortcode [] is used to represent a whole entity, either the head or tail. For example, the investigation Alert=[Alert0]; later, then we can use [Alert0] to refer to this node.

D. Story Representation

After a graph is constructed, subgraphs can be retrieved over Neo4j, by using a Python API. Py2neo is a Python Neo4j API with imperative and declarative features [25]. It can be used to run a more necessary and performant method of querying. The queries by traversing the graph, explore terms that can be used to fill the text templates. Both templates and queries are modifiable and can be customised based on an organization's preference and its internal policy. We installed Neo4j APOC library extension to enable the analyst with more power and flexibility for crafting queries, that can be successively constrained while maintaining a simple and readable syntax [26]. These extracted results from the queries are applied to generate an automatic story.

IV. NAAS Test Use Case

In this section, we illustrate the applicability of the NAAS. For testing purposes, a threat alert example was selected, which was raised from an external vendor's tool - the Secureworks[8]. Secureworks is the commercial cybersecurity analytical tool that the educational institute SOC team responses the potential cyber threats that are detected among the monitored log [18].

The example of the alert message produced by Secureworks is as follows:

[1]WhoIs. http://www.whois.com/s
[2]https://www.virustotal.com/
[3]https://www.threatminer.org/
[4]https://www.snort.org/downloads
[5]https://otx.alienvault.com/pulse/
[6]https://www.microsoft.com/en-us/wdsi/threats
[7]https://www.symantec.com/security-center/a-z/
[8]https://www.secureworks.com/

Table I
PART OF INTERNAL AND EXTERNAL KNOWLEDGE ASSOCIATED WITH THE THREAT ALERT.

Internal	External
Alert_DateTime(02/27/2019 5:05)	Snort_Header(MALWARE-CNC Osx.Keylogger.Elite variant outbound connection)
Alert_Message(MALWARE-CNC Osx.Keylogger-Elite)	ThreatExchange_Definition (OSX.Keylogger is a spyware program for Mac OSX that records keystrokes may take screenshots and may also send the information to a predetermined email address)
Alert_SrcIP(10.233.62.247)	ThreatExchange_Name(Syemantec site)
Host_SrcDomain(Sev1.edu.au)	WhoIs_DesDomain(service. macinstallerinfo.com)
Host_Seneitivity(Low(3/10)	ScanEngine_Name(ThreadMiner)
Host_Location(Fincanctial_Unit, Block3 Level2)	ScanEngine_URL (http://service.macinstallerinfo. com/ Mac/getInstallerSpecs/ ?&channel=3Db5002& info=3D238749466&encinfo= 3D1&)
Unit_Confidentiaial(High(8/10)	
Host_Admin(Alex,alexvu.edu.au)	
Host_Apps (Schart, CoNsoleK-itMicrosoftVisual, C++)	
Alert_DesIP(104.239.223.14)	

```
''MALWARE-CNC Osx.Keylogger-Elite - 10.233.62.247
→ 104.239.223.14 02/27/2019 5:05 PM''
```

The threat alert is correlated to the Internal and External knowledge bases to represent a possible threat scenario. The extracted knowledge from knowledge bases is shown in a human-readable format in Table I to be comparable with the threat intelligence report. As Table I shows, to represent the knowledge the NameOfSource_Feild (Value) is used as a structure, where: (1) *NameOfSource* is the name of the source in which the data is retrieved from, (2) *Feild* is the property, which is searching from the source, and (3) *Value* presents the extracted value. I.e., Alert_SrcIP(10.233.62.247): Alert is an internal source that source IP (SrcIP) is searched there, and 10.233.62.247 is the extracted value for source IP.

By using the knowledge, the threat intelligence report is automatically converted into the simple structure format and then into Cypher queries. Where the alert_id is the key as an essential identification for each alert, and it used in corresponding nodes and relationships. A snapshot of the simple structure format transformed from the threat intelligence report is as follows.

This alert, code, Alert0001

create(alert_2_shortform_0_:sentence_entity_shortform {content:'Alert',content_lower:'alert',short_form:'Alert',alert_id: 'Alert0001' })
create(alert_2_shortform_1_:sentence_entity_shortform {content: 'Machine',content_lower: 'machine',IP:"10.233.62.247", Domain:"Sev1.edu.au",Sensitively:"low(3/10)", UnitID:1, Admin:12, APPslist:13, short_form: 'machine', alert_id:'Alert0001'})
create(sentence_entity_alert_2head_2:sentenc_entity {content:'Alert0001', content_lower:'alert0001',alert_id:'Alert0001', DateTime:"02/27/2019 5:05", Message:"MALWARE-CNC Osx.Keylogger-Elite",SrcIP:"10.233.62.247", DesIP:"104.239.223.14"})
create (sentence_entity_alert_2head_2)-[:is] → (alert_2_shortform_0_)
create (alert_2_shortform_0_)-[:shows_connection_from] → (alert_2_shortform_1_)
create(sentence_entity_alert_2tail_6:sentence_entity { Content:'Department',content_lower:'department',alert_id:'Alert0001', Unit:1,Name:"Financial_Unit", Address:"Block3 Level2", Security_rank:"High(8/10)"})
create (alert_2_shortform_1_)-[:located_in] → (sentence_entity_alert_2tail_6)

Figure 2. A snapshot of the generated Cypher queries from the threat intelligence report(human-understandable format)

Alert0001-{DateTime:"02/27/2019 5:05", Message: "MALWARE-CNC Osx.Keylogger-Elite", SrcIP:"10.233.62.247", DesIP:"104.239.223.14"}, is , Alert=[*Alert*] [*Alert*], Shows_Connection_from, Machine-{IP:"10.233.62.247", Domain:" Sev1.edu.au",Sensitively:"low(3/10)", UnitID:1,Admin:12, APPslist:13}=[*machine*] [*machine*], located_in, Department-{Unit:1,Name:"Financial_Unit", Address:"Block3 Level2", Security_rank:"High(8/10)"}

By defining the sentence "This alert, code, Alert0001", a new alert_id as a new key is generated. Figure 2 shows a snapshot of the output generated from the Python script where human-readable sentences converted into Cypher queries after transforming to the simple structure format. The output queries can be easily copy-pasted into Neo4j to generate the corresponding graph. Nodes, relationships, and properties are created based on the knowledge that was provided in human-readable sentences and corresponding queries. As Figure 2 shows, at first, nodes and their properties are created, then the head and tail of a connection are defined then relationships are linked. Since establishing nodes and links with Cypher is complicated for humans, the Python script translates the human-readable sentence to Cypher queries. Thus, provides a bridge for people to pass this stage with a human-readable format.

The graph is generated directly from the Cypher queries. Figure 3 shows the generated graph for the threat alert with node labels and relationships type in the Neo4j. The corresponding knowledge related to the alert is shown by

nodes and relationships in the graph; The internal knowledge is presented through the Machine node and its dependencies, and the external knowledge is shown through the potential Attacker and Malware nodes and their associations. The nodes are shown in circles and they are classified into two groups: **sentence_entity** defines the pink circles in the graph that brings a fact or isolated piece of knowledge. And the red circles are defined as **sentence_entity_shortform**, that deeper insights about them are provided in the graph (The nodes were converted by adding [] to represent a whole entity). Cypher is very similar to SQL, consists of clauses, keywords, and expressions like predicates and functions [21]. Each node represents an entity table, and its properties are such as the columns of the table. For instance, Machine as the highlighted node in Figure 3 contains properties like IP, Domain, Sensitively, and others. It has relations with other nodes (Application, staff, and department). For a Python graph database, Neo4j is installed on a system and then accessed via it's binary and HTTP APIs, though the Neo4j Python driver, i.e.,Database.driver [21]. Then, the graph for representing the knowledge of interest gives the analyst the power and flexibility for crafting queries. The queries statements are easily and affordably defined and manipulated by users. A procedure on Cypher (APOC), an add-on library for Neo4j, is used for querying flexibly and traversing the knowledge graph. The APOC library consists of many procedures to expand the subgraph nodes reachable from the start node following relationships to max-level adhering to the label filters. For example, in the below query, the collection of nodes in the subgraph, and the collection of relationships between all subgraph nodes are returned. The given condition constrained the analytic results to focus on those that were linked from the node called Machine.

```
MATCH (p:sentence_entity_shortform {content:'Machine'+
",alert_id:"+str(AlertCode)+ "" })
CALL apoc.path.subgraphAll(p, {
relationshipFilter: '>',
minLevel: 0,
maxLevel: 5
})
YIELD nodes, relationships
RETURN nodes, relationships;"
```

A node in the knowledge graph is labeled by its name, as a noun, and its properties are as adjectives, and its acting by a verb shows a relationship. Retrieved subgraphs as results of the queries, brings the opportunities to generate sentences with nouns, adjectives, and verbs automatically. For example, the paragraph *"Machine contains APPS, App1 is Schart, App2 is CoNsoleKit Microsoft Visual, App3 is C++. Machine administrated_by Staff, Name is Alex, Mail is alex@vu.edu.au. Machine located_in Department, Name is Financial_Unit, Security_rank is High(8/10), Unit is 1, Address is Block3 Level2."* has generated automatically after traversing the graph and returning the nodes, relationships, and properties associated with the specific node,"Machine". The flexibility of creating a graph, and extracting information from it provides advantages to make an automatic report based on the preferences. A generated story from the threat alert is shown in Figure 4.

V. Conclusion

We have demonstrated that NAAS with knowledge graph an narrative report can assist security professionals to have a better perception of the elements of the environment by involving more staff in the risk management that originated from the threat alerts. The transformation from the human-readable sentences to query language, and interpretation of the graph's knowledge in a human-readable narrative report, facilitated comprehension and effectively allowed human involvement in the process of risk management.

The knowledge graph is used in NAAS, visualized the relationships between the alert and relevant information from the Internal and External knowledge bases through a shared understanding of cyber situation awareness contextualization. Alerts integrated into a more comprehensible form like summarised stories to represent security events based on the knowledge graph. Thus alerts investigation is improved through the better contextualization of cyber situation awareness and helps to establish an understandable common ground among human beings.

In terms of the current limitations, we only focused on malware taxonomy for approach demonstration in this paper. Still, the model can be easily adapted to other types of incidents by providing complementary sources in the Internal and the External knowledge bases. Also, since the enriched report for a security alert in a story design is not available, we were not able to perform the direct comparison with the proposed storytelling model. Besides, the impact of the understanding of the narrative format and engaging people is not directly measurable. Thus usability and suitability of NAAS were presented in a real case study.

References

[1] S. Jajodia, S. Noel, P. Kalapa, M. Albanese, and J. Williams, "Cauldron mission-centric cyber situational awareness with defense in depth," in *2011 - MILCOM 2011 Military Communications Conference*, 2011, pp. 1339–1344.

[2] S. Noel and S. Jajodia, "A suite of metrics for network attack graph analytics," in *Network Security Metrics*. Springer, 2017, pp. 141–176.

[3] A. Sadighian, J. M. Fernandez, A. Lemay, and S. T. Zargar, "Ontids: A highly flexible context-aware and ontology-based alert correlation framework," in *Foundations and Practice of Security*, J. L. Danger, M. Debbabi, J.-Y. Marion, J. Garcia-Alfaro, and N. Zincir Heywood, Eds. Cham: Springer International Publishing, 2014, pp. 161–177.

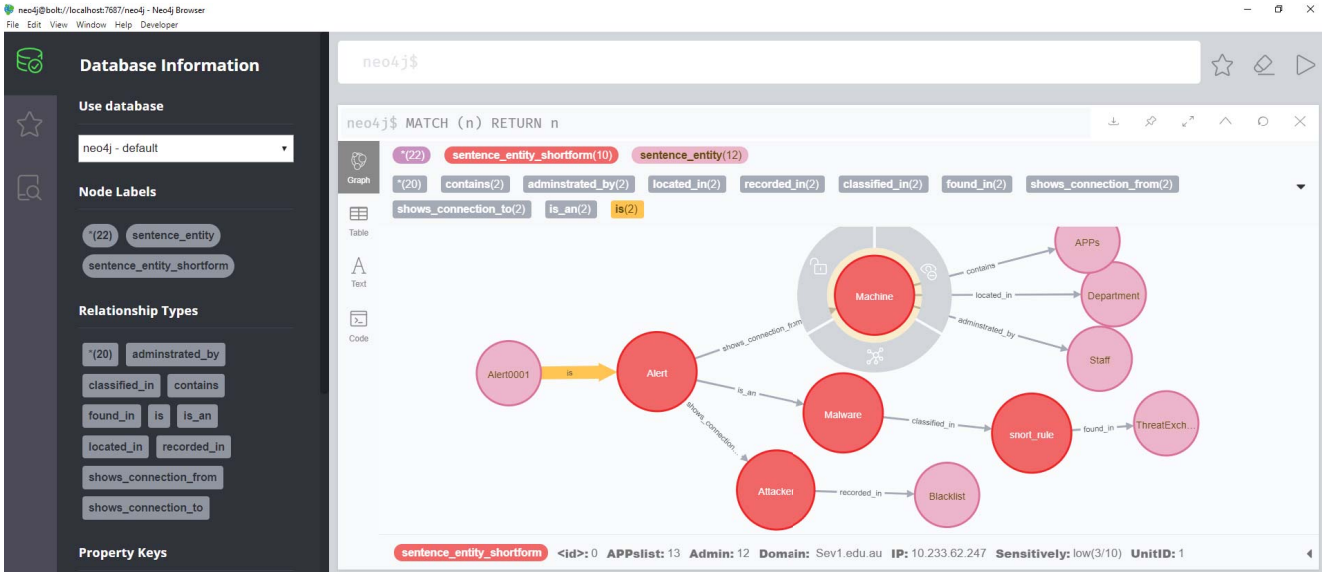

Figure 3. The generated graph in Neo4j from the Cypher queries(nodes are illustrated as circles, and the relationships are shown as directed arrows).

"We got an alert, Alert0001, Alert0001, on 02/27/2019 5:05, because of the connection between Internal Machine,name: Sev1.edu.au, IPaddress: 10.233.62.247 with severity low(3/10) and Potential attacker, name:service.macinstallerinfo.com, IPaddress: 104.239.223.14

Alert is a Malware, Name is Osx.Keylogger.Elite, ClassRef is snort, Type is CNC.

More Info:

Malware classified_in snort_rule, Classification is malware-CNC, Title is MALWARE-CNC Osx.Keylogger.Elite variant outbound connection. It found_in ThreatExchange, Defintion is OSX.Keylogger is a spyware program for Mac OSX that records keystrokes may take screenshots and may also send the information to a predetermined email address, Reference is Symantec.

Internal Info:

Machine contains APPs, App1 is Schart, App2 is CoNsoleKit Microsoft Visual, App3 is C++ Machine administrated_by Staff, Name is Alex, Mail is alex@vu.edu.au Machine located_in Department, Name is Financial_Unit, Security_rank is High(8/10), Unit is 1, Address is Block3 Level2

External Info:

Attacker recorded_in Blacklist, IP is 104.239.223.14, Reference is ThreadMiner, URL1 is http://service.macinstallerinfo.com/Mac/getInstallerSpecs/?&channel=3Db5002&info=3D238749466&encinfo=3D1& "

Figure 4. A generated story from the alert

[4] K. Zhang and J. Liu, "Review on the application of knowledge graph in cyber security assessment," *IOP Conference Series: Materials Science and Engineering*, vol. 768, p. 052103, mar 2020.

[5] S. Noel, E. T. Harley, K. H. Tam, M. Limiero, and M. Share, "Chapter 4 – cygraph: Graph-based analytics and visualization for cybersecurity," *Handbook of Statistics*, vol. 35, pp. 117–167, 2016.

[6] M. C. Study, "Graph technology powers cybersecurity situational awareness that's more scalable, flexible and comprehensive," https://neo4j.com/case-studies/mitre/, 2019.

[7] F. Böhm, F. Menges, and G. Pernul, "Graph-based visual analytics for cyber threat intelligence," *Cybersecurity*, vol. 1, pp. 1–19, 2018.

[8] S. Liu, W. Xiting, L. Mengchen, and Z. Jun, "Towards better analysis of machine learning models: A visual analytics perspective," *Visual Informatics*, pp. 48–56, March 2017.

[9] J. Vink, "Storytelling — design research techniques," http://designresearchtechniques.com/casestudies/storytelling/, 2010.

[10] Q. Wu, Z. Shen, C. Leungy, H. Zhang, Y. Cai, and C. Miao, "Internet of things based data driven storytelling for supporting social connections," pp. 383–390, 2013.

[11] C. I.-L. Testing, "Cbest implementation guide, bank of england," [online]: w.bankofengland.co.uk/-/media/boe/files/financial-stability/financial-sector-continuity/cbest-implementation-guide.pdf, accessed: 2019-10-30.

[12] M. Vielberth, F. Menges, and G. Pernul, "Human-as-a-security-sensor for harvesting threat intelligence," *Cybersecurity*, vol. 2, pp. 1–15, 2019.

[13] F. Amato, G. Cozzolino, A. Mazzeo, and F. Moscato, "Detect and correlate information system events through verbose logging messages analysis," *Computing*, vol. 101, no. 7, pp. 819–830, 2019.

[14] E. Kiesling, A. Ekelhart, K. Kurniawan, and F. Ekaputra, "The sepses knowledge graph: An integrated resource for cybersecurity," in *The Semantic Web – ISWC 2019*, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, and F. Gandon, Eds. Cham: Springer International Publishing, 2019, pp. 198–214.

[15] D. Schlette, F. Böhm, M. Caselli, and G. Pernul, "Measuring and visualizing cyber threat intelligence quality," *International Journal of Information Security*, pp. 1–18, 2020.

[16] W. Wang, R. Jiang, Y. Jia, A. Li, and Y. Chen, "Kgbiac: Knowledge graph based intelligent alert correlation framework," in *Cyberspace Safety and Security*, S. Wen, W. Wu, and A. Castiglione, Eds. Cham: Springer International Publishing, 2017, pp. 523–530.

[17] A. Elitzur, R. Puzis, and P. Zilberman, "Attack hypothesis generation," in *2019 European Intelligence and Security Informatics Conference (EISIC)*. IEEE, 2019, pp. 40–47.

[18] N. Afzaliseresht, Y. Miao, S. Michalska, Q. Liu, and H. Wang, "From logs to stories: Human-centred data mining for cyber threat intelligence," *IEEE Access*, vol. 8, pp. 19 089–19 099, 2020.

[19] Y. B. Shrinivasan and J. J. van Wijk, "Supporting the analytical reasoning process in information visualization," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 1237–1246.

[20] R. Haberlin, P. C. G. da Costa, and K. B. Laskey, "A reference architecture for probabilistic ontology development." *STIDS*, vol. 2013, pp. 10–17, 2013.

[21] "Neo4j graph platform," https://neo4j.com/.

[22] M. Lal, *Neo4j graph data modeling*. Packt Publishing Ltd, 2015.

[23] B. E. Ulicny, J. J. Moskal, M. M. Kokar, K. Abe, and J. K. Smith, "Inference and ontologies," in *Cyber Defense and Situational Awareness*. Springer, 2014, pp. 167–199.

[24] STIXTM, "Structured threat information expression (stixtm)," https://docs.google.com/document/d/1IcA5KhglNdyX3tO17bBluC5nqSf70M5qgK9nuAoYJgw/, 2017.

[25] G. Drakopoulos and A. Kanavos, "Tensor-based document retrieval over neo4j with an application to pubmed mining," in *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, 2016, pp. 1–6.

[26] "Apoc user guide 3.5," https://neo4j-contrib.github.io/neo4j-apoc-procedures/#introduction., 2019.