**Leveraging a Cloud-Native Architecture to enable Semantic Interconnectedness of Data for Cyber Threat Intelligence**

**Abstract**

Cloud technologies have several merits including the elimination of cost incurred when traditional technologies are adopted. Despite the benefits, the cloud is still facing security challenges thereby calling for cyber threat intelligence capable of identifying threats and providing possible solutions. However, dependence on traditional security mechanisms and approaches for security solutions within cloud environments presents challenges. This calls for cloud-native solutions which leverages cloud features for design and development of solutions for data and applications hosted and running within the cloud. Past studies have suggested the adoption of semantic technologies for cloud-based security mechanisms. However, the semantic processing of data faces challenges of data interconnectedness due to aggregation of data from diverse heterogenous sources. Hence, this study proposes a cloud-native architecture capable of connecting security-related data from different sources in the cloud to enhance cyber threat intelligence. It presents a proof-of-concept implementation of the proposed solution on Amazon AWS cloud, within an auto-scaling group for scalability and across multiple availability zones for high availability.

**Keywords:** Cyber Threat Intelligence, Cyber Incident Response, Cloud Computing, Semantic Knowledge Graph, Data Interconnectedness

1. **Introduction**

Cyber-attacks experienced in cloud environments include attacks based on XML, SQL injection attacks, distributed denial of service attacks, attacks based on virtualization, and cloud-based data storage attacks. Attacks based on XML involve the manipulation of XML format of SOAP messages with which applications communicate in the cloud (Shaikh, 2016). SQL injection attack involves an unauthorized user’s entry of arbitrary queries from the web application interface, followed by execution in the server which exposes confidential data or manipulates existing data within the database (Alsaleem et al., 2019). A DDoS (Distributed Denial of Service) attack involves an attacker disrupting cloud services thereby disallowing users’ access to these services and use of these services when required. Attacks can send a massive amount of traffic towards overwhelming victims so that services will be unavailable to those who have the right access (Somani et al., 2017; Rajendran et al., 2018). Attacks based on virtualization focus on the weaknesses of virtualization in the cloud to take over guest machines and host machines to corrupt CPU, memory, and resources, launch arbitrary code, manipulate data, modify configurations, inject malicious code, and more, thereby influencing the data’s availability and integrity (Yucel and Romuald, 2015; Sri et al., 2017; Kumar et al., 2017). Cloud-based data storage attacks are attacks on data stored in the cloud towards illegal analysis of data, compromising a database’s integrity, taking control of one or more storage resources, and polluting it to make data unavailable.

The identification of threats and organising responsive action towards mitigation is important. When it comes to cyber incident response, detection and recovery are the major features due to the necessity of monitoring the implications of the solutions to an attack and observing new behaviour coming from the incident. When an attack is detected, mitigated and the system is recovered, information about the events and recovery processes may be useful for similar attacks, thereby forming response actions. Such response actions may be useful in mitigating several cyber incident examples. There are several frameworks meant to handle incidents related to computer security and some researchers have come up with models to handle security incidents in the cloud environments. Main global standards and guidelines for incident handling models are defined and published by organisations such as Computer Emergency Response Team Coordination Centre (CERT/CC), NIST Special Publication (NIST SP 800-61), International Organization for Standardization (ISO), European Network and Information Security Agency (ENISA), SANS Institute, and Information Technology Infrastructure Library (ITIL). CERT/CC published four guidelines to manage information security incidents. They describe four phases of the incident handling process model: receiving an incident report, triage, incident response, and analysis, consisting of 14 subphases. The design of the Handbook for Computer Security Incident Response Teams (CSIRT) is to guide organisations specifically to form and operate a CSIRT (WestBrown et al., 2003). The State of the Practice for CSIRTs (Killcrece, 2003) is aimed at assisting new and existing teams to understand best practices and recommendations to handle incidents and services related to CSIRT. The focus of Organisational Models for CSIRT is to select the appropriate model for incident response capabilities in an organisation (Killcrece et al., 2003). Defining incident management processes for CSIRTs: A work in progress is about incident management, including processes and functions overview, and how to support the individuals, technology, and procedures involved (Alberts et al., 2004).

In addition, NIST Special Publication (NIST SP 800-61) explains computer security handling guidelines (Cichonski et al., 2012). There are four incident handling phases in the NIST guideline, including preparation, detection and analysis, containment, eradication and recovery, and post-incident activity. The five phases of ISO/IEC 27035:2011 include planning and preparation, detection and reporting, assessment and decision-making, responses, and lessons learned. It offers reporting form templates for information security vulnerabilities, incidents, and events (ISO, 2011). ENISA’s Incident Management Guide offers practical information and guidelines to manage incident handling phases and consists of six major sequence components, including incident report, report registration, triage, incident resolution, incident closure, and post-analysis (ENISA, 2010). The Incident Handler's Handbook from the SANS Institute, which offers information to produce organisational incident response policies, standards, and teams, consists of six phases: preparation, identification, containment, eradication, recovery, and lessons learned (Kral, 2011). ITIL’s Foundations of IT Service Management Based on ITIL V3 features incident management, consisting of five key phases: incident detection and recording, classification and initial input, investigation and diagnosis, resolution and recovery, and incident disclosure (British Standards Institution, 2007).

To take this further, the focus of the current study is towards designing a novel semantic-based methodology for cloud-native cyber incident response. This implies considersing the distinct features of cloud computing technology to come up with a relevant method specific to the cloud domain towards incident response. This study wants to leverage semantic technologies like ontologies and knowledge graphs to facilitate the retrieval and analysis of data related to security challenges in the cloud environment. This paper explores a semantic-based approach to cyber incidents in the cloud. A comprehensive study is carried out to evaluate and review the existing threats cloud users or consumers face regularly and the practices that should be implemented by all involved parties to ensure security and privacy as well as management of the risks that may arise. A wide range of security standards for cloud computing and IT management are reviewed in conjunction with the common security practices utilized and implemented by cloud providers all over the world by taking into consideration and analysing the security-themed whitepapers provided on these cloud services companies web pages. This paper explores cyber incidents in the cloud through a review of existing literature and opinions regarding the semantic approach for cyber incidents in the cloud and identifying the main challenges faced by cloud-based services and analysing these challenges and threats to the security of the cloud platform. We have also investigated existing research efforts towards addressing the challenges faced with cyber incident response in the cloud. The remaining sections are as follows: section 2 investigates research efforts regarding cloud-based cyber incident response or handling solutions, frameworks, and models; section 3 analyses the research gaps and highlights the requirement for data interconnectedness across diverse heterogeneous sources with different data structures based on the gaps identified and recommendation of past studies. In section 4, the proposed solution is designed, with detailed description of its methodology while section 5 describes a proof-of-concept implementation and some evaluation metrics, based on available sample dataset. Section 6 provides a conclusion to the paper.

1. **Literature Review**

The framework by the European Union Agency for Cybersecurity (ENISA, 2010) on how to report major cloud security incidents focuses on guiding government authorities about handling of security incidents but they are likewise applicable to organisations. Periodical risk assessments should be carried out towards identifying crucial parts of the IT infrastructure as well as the core services which depend on the cloud (Dekker et al., 2013). Mogull et al. (2017) came up with tabletop exercises and threat modelling towards determining the most effective response to different types of attack and Hengst (2020) thinks they should cater for responses required for IaaS (infrastructure-as-a-service), PaaS (platform-as-a-service), and SaaS (software-as-a-service). Ab Rahman and Choo (2015) introduced a conceptual cloud incident handling model consisting of incident handling, digital forensic and the Capability Maturity Model Services towards further effective organizational handling of incidents in the cloud. Their model integrates forensic activities into every phase of the incident handling model. A distinctive feature of their study is response selection which emphasized the critical function of response selection techniques as they can ideally facilitate rapid deployment of incident response processes. The three response selection techniques identified are static mapping, dynamic mapping, and cost-sensitive mapping. Static mapping, which can be carried out using ontology or probabilistic cognitive maps, helps to map a predefined incident to predefined responses. However, the downside of static mapping despite its facilitation of rapid deployment is that it helps attackers to also expect response actions. This downside is eliminated by dynamic mapping because it involves the deployment of various advanced approaches. Potential mapping techniques include game theory, machine learning, as well as risk assessment methods. The mapping strategies provide a means of selecting the most appropriate mapping type based on an understanding of each one. While these can be used within cloud-based systems, they are not limited to use within the cloud and can be adopted or adapted for different scenarios of incident handling solutions. Dynamic response selection does not consider the cost of response and damage. Cost-sensitive considers the cost of response and damage (Ab Rahman and Choo, 2015). These are summarised in Table 1.

Table 1 – Response Selection Techniques for Incident Response Processes

|  |  |  |
| --- | --- | --- |
| **Response Selection Techniques** | **Strengths** | **Weaknesses** |
| Static Mapping | Facilitates rapid deployment by mapping predefined incidents to predefined responses  | Implies that attackers also expect response actions |
| Dynamic Mapping | Addresses the challenges of static mapping with techniques such as game theory and machine learning | Ignores cost of response and any potential damage |
| Cost-Sensitive Mapping | Addresses the challenge of dynamic mapping | Can be more time-consuming |

Furthermore, Ab Rahman et al. (2016a) investigated attacks on cyber-physical cloud systems (CPCS) and proposed a conceptual CPCS forensic-by-design model. In another study, Ab Rahman et al. (2016b) presented an integrated cloud incident handling and forensic-by-design model and validated the model by deploying to diverse cloud repositories. Monfared and Jaatun (2012) suggested an approach capable of handling compromised components in an IaaS cloud installation which demonstrated the adaptability of NIST incident handling guidelines for deployment in the cloud computing space and introduced cloud specific strategies for the five phases: specific cloud incident handling approaches, responsible stakeholder(s) for the approaches, service impacted, enforcement challenges, and specific platform and library dependencies. Zhang et al (2010) suggested an information risk management framework which covers several cloud services and deployment models. Within the framework are three phases; architect and establish, implement, and operate as well as monitor and review comprising of different processes.

In addition, real-time analysis of attacked systems and log files requires improved techniques and should be prioritized (Grobauer and Schreck, 2010). Performing real-time forensics on running systems in cloud computing can provide valuable information especially for PaaS and SaaS where the log files of CSP contain critical evidence sources. Hence, this calls for an approach capable of generating and analysing such information (Grobauer and Schreck, 2010), even from data sources such as the devices of CSU and off-site CSP data centres (Ab Rahman and Choo, 2015). Real-time approaches which help to automate processes can capture any metadata at the time of the alert and pause the virtual machine towards retaining the memory state. They can be used to analyse network flows to determine whether network isolation was successful, examine configuration data towards identifying similarly affected instances, and review data access logs towards determining possible effects of the attack on the cloud platform itself (Mogull et al., 2017). Report sharing is a critical aspect of cyber incident response as it facilitates effective mitigation processes across organizations. Security teams can discuss best practices and enhance the way security incidents are handled when they share incidents or summaries of security incidents (Dekker et al., 2013). These define what the best practices are, as recommendations for designing and developing solutions with their actual implementation and subsequently, their effect subject to several other factors, including design and architectural principles of their adoption. Frøystad et al. (2016) proposed a simplified method to share incidents and it makes sure that everyone involved understands the information pertaining to them. Semantic-based approaches can also enhance the ability to retrieve information about similar attacks. According to Martini and Choo (2012), it is necessary to have metadata retention and recommend possible evidence sources like checking of file integrity, extensive logging, and centralised auditing. Virtualization, resource pooling and rapid elasticity are cloud features relevant in developing tools for forensic evidence. The dynamic scaling and elasticity nature of cloud, with multi-tenancy capabilities implies that forensic data between multiple tenants in a variety of cloud structures and locating forensic data with timestamps can be separated (Purnaye et al., 2022). These are achievable by adopting semantic-based methods as they can explore data and linking relevant information. Due to the complexity of cloud environments because of their distributed nature when compared to traditional on-premises systems, traditional cyber incident response approaches are less effective in the cloud (Guo et al., 2021). Likewise, collecting, preserving, and analysing data related to security incidents in the cloud can be really tasking due to multiple jurisdictions and multiple systems. Most studies on cloud incident response provided cloud-native recommendations complementing traditional incident response approach based on the incident response lifecycle provided by the National Institute of Standards and Technology (NIST).

**2.1 Semantic-Based Incident Response**

Regarding semantic-based incident response solutions, Baskerville et al. (2014) discovered that conventional security approaches focus majorly on detecting and preventing attacks relative to incident response. There is a need for a powerful security model capable of implementing an incident response process that makes the most of web semantics, facilitates appropriate knowledge representation, and permits certain inference level (Moreira et al., 2018). Blackwell (2010) proposed a three-layer architectural security model including social layer (people and organisations), physical layer, and logical layer (computers and networks), and security incident ontology which considers organisations and systems. Both the architecture and ontology facilitate holistic analysis of incidents. However, the ontology seems to be more like taxonomy than ontology as it provides definition of certain concepts of incident, cyber defense and attack, without formalizing them as ontology. The researchers recommended the adoption of OWL (Web Ontology Language) in further studies (Moreira et al., 2018). Mundie et al. (2014) introduced an ontology based on a past study which defined a meta-model meant to handle processes of incident management. Documenting, comparing, and analysing Computer Security Incident Response Teams (CSIRTs) are the objectives of the proposed ontology. A high level is involved in the modelled processes with more focus on teams’ relationships and roles. However, it did not go into the incident’s technical details. ISO/IEC 27035:2011 formed the basis of an ontology presented by Silva and Fagundes (2014) to manage incidents in the information security domain. The purpose of the ontology is to train incident response teams. By comparing past studies, the study presented eight super classes as well as the incident class properties. However, it lacks the definition of the concept of “event” despite being a basic part of ISO 27035 (Moreira et al., 2018).

An ontology representing cybernetic assets to model scenarios for simulating computer networks’ defence and increase their resilience was introduced by O’Sullivan and Turnbull (2015). Cyber Simulation Terrain (CST) and Cyber Effects Simulation Ontology (CESO) are the two divisions in the ontology. The former provides definitions of concepts like computers, network connectivity, users, software, vulnerabilities, and exploits. CESO helps in modelling the effects of a cyber-attack on systems within a complex network. However, according to Moreira et al. (2018), it lacks consideration for security events and temporal aspects. The Computer Security Incident Handling Ontology (CSIHO) in OWL format is an ontology meant to handle security incidents and is characterized by easy extension, integration with other models, facilitation of logical inferences and simplification of the knowledge transfer within a collaborative cyber defense context. CSIHO ontology helps in handling security incidents, defining and implementing the basic concepts of security events, and recording an incident’s temporal aspects (Moreira et al., 2018).

1. **Research Gap Analysis**

A notable challenge facing cyber incident response in the cloud is the dependence on traditional incident handling approaches. However, despite the possibility of adapting components of traditional handling mechanisms, it is pertinent to evaluate distinctive cloud features. SAN’s survey revealed that the absence of standards, training and tools are the biggest challenges in investigating cloud incidents (Henry et al., 2013). Depending on traditional incident handling frameworks to handle incidents in the cloud will be challenging since such frameworks have their demerits as shown in Table 2. The analysis carried out by Grobauer and Schreck (2010) regarding appropriate approaches and challenges for incident handling and response in the cloud considered five common steps: detection, analysis, containment, eradication and recovery, and preparation/continuous improvement. The study highlighted the challenge of security personnel’s limited knowledge regarding Cloud Service Provider’s (CSP) architecture. Ozer et al. (2020) added another step; cloud configuration to the traditional incident response plan towards increasing security personnel’s familiarity with the underlying technology of cloud computing and sharing current best practices.

Furthermore, cloud environments are characterised by distributive data across multiple systems or jurisdictions and usually multiple tenants. Data collection can be difficult when security teams lack full access to CSPs’ sources (Grobauer and Schreck, 2010; Ab Rahman et al., 2016b), possibly due to lack of direct points of contact with the CSP or are limited to standard support (Mogull et al., 2017). Privacy issues may also prevent disclosure of certain data (Grobauer and Schreck, 2010). Hengst (2020) in addition, noted that data collection is among the major challenges in cloud incident handling. That is, data collection to conduct proper incident handling is tasking due to the need for identification of possible data sources. Another challenge is the complexity involved in the consolidation of data from disparate structured, unstructured, and semi-structured sources. Since security-related data comes from different sources, how to streamline data becomes a remarkable challenge because of the need to integrate heterogeneous data streams. There is a need to standardize data to analyse such data effectively and accurately. Hence, the solution proposed in this study focuses on how to connect security-related data coming from across diverse heterogeneous sources with different data structures in the cloud through the cloud architecture and semantic technologies. Based on these, Table 2 summarises features from traditional security incident handling frameworks, including identification of their coverage for feature recommendations across the frameworks.

Table 2 – Features of traditional security incident handling frameworks

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Features** | **TAL** | **CVE** | **NVD** | **CPE** | **CWE** | **CAPEC** | **ATT&CK** |
| **Enumeration** | Enables focus on attacker’s mindset | Yes | No | No | No | No | Yes | Yes |
| Defines attacker’s tactics | Yes | No | No | No | No | Yes | Yes |
| Defines attacker’s attributes | Yes | No | No | No | No | Yes | Yes |
| Describes attacker’s technique | No | No | No | No | No | No | Yes |
| Ensures information sharing | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Identifies threat or vulnerability | No | Yes | Yes | Yes | Yes | Yes | No |
| Defines threat or vulnerability | No | Yes | Yes | Yes | Yes | Yes | No |
| Provides fix information | No | No | Yes | No | No | No | Yes |
| Provides severity of threat or vulnerability | No | No | Yes | Yes | Yes | Yes | No |
| Provides impact of threat or vulnerability | No | No | Yes | Yes | Yes | Yes | No |
| Provides mechanism of attack | No | No | No | No | No | Yes | No |
| Provides attack domain | No | No | No | No | No | Yes | Yes |
|  | **Features** | **TAL** | **CVE** | **NVD** | **CPE** | **CWE** | **CAPEC** | **ATT&CK** |
| **Scoring Systems** | Enables focus on attacker’s mindset | Yes | No | No | No | No | Yes | Yes |
| Defines attacker’s tactics | Yes | No | No | No | No | Yes | Yes |
| Defines attacker’s attributes | Yes | No | No | No | No | Yes | Yes |
| Describes attacker’s technique | No | No | No | No | No | No | Yes |
| Ensures information sharing | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Identifies threat or vulnerability | No | Yes | Yes | Yes | Yes | Yes | No |
| Defines threat or vulnerability | No | Yes | Yes | Yes | Yes | Yes | No |
| Provides fix information | No | No | Yes | No | No | No | Yes |
| Provides severity of threat or vulnerability | No | No | Yes | Yes | Yes | Yes | No |
| Provides impact of threat or vulnerability | No | No | Yes | Yes | Yes | Yes | No |
| Provides mechanism of attack | No | No | No | No | No | Yes | No |
| Provides attack domain | No | No | No | No | No | Yes | Yes |
|  | **Features** | **TAL** | **CVE** | **NVD** | **CPE** | **CWE** | **CAPEC** | **ATT&CK** |
| **Sharing Standards** | Enables focus on attacker’s mindset | Yes | No | No | No | No | Yes | Yes |
| Defines attacker’s tactics | Yes | No | No | No | No | Yes | Yes |
| Defines attacker’s attributes | Yes | No | No | No | No | Yes | Yes |
| Describes attacker’s technique | No | No | No | No | No | No | Yes |
| Ensures information sharing | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Identifies threat or vulnerability | No | Yes | Yes | Yes | Yes | Yes | No |
| Defines threat or vulnerability | No | Yes | Yes | Yes | Yes | Yes | No |
| Provides fix information | No | No | Yes | No | No | No | Yes |
| Provides severity of threat or vulnerability | No | No | Yes | Yes | Yes | Yes | No |
| Provides impact of threat or vulnerability | No | No | Yes | Yes | Yes | Yes | No |
| Provides mechanism of attack | No | No | No | No | No | Yes | No |
| Provides attack domain | No | No | No | No | No | Yes | Yes |
| **Table header acronyms’ meaning and references**TAL (Casey, 2017): Technology Alert ListCVE (MITRE Corporation, 2021a): Common Vulnerabilities and ExposuresNVD (US NVD, 2021): National Vulnerability DatabaseCPE (MITRE Corporation, 2021b): Common Platform EnumerationCWE (MITRE Corporation, 2021c): Common Weakness EnumerationCAPEC (MITRE Corporation, 2021d): Common Attack Pattern Enumeration and ClassificationATT & CK (MITRE Corporation, 2021e): Adversarial Tactics Techniques and Common Knowledge |

Hence, a cloud-native incident response or handling mechanism that also leverages semantic technologies can address these challenges as semantic data models facilitate automation for generating, consuming, and interpreting knowledge. The creation of standardized taxonomies which consider the characteristics of cloud computing systems and the security concerns in the cloud environments will facilitate integrations. Since ontologies can be reused and extended, the development of cloud-native semantic-based incident response ontology will offer flexibility required to add new concepts without regards to the multiple tenancy feature in the cloud. Cloud Service Provider’s (CSP) architecture can integrate such framework such that different tenants are involved. Semantic technologies are characterized by scalability, so they can handle the data collection issues, the need for identification of possible data sources and introduction of new data sources in the cloud. Ontologies are semantic data models for defining the types of entities within a domain and the properties useful in describing them.

**4.0 Design Methodology for Semantic Data Interconnectedness**

This section expatiates the principle of the proposed cloud architecture solution for cyber incident response. It describes how data is collected from different sources and the processes involved in achieving semantic data interconnectedness. Semantic interconnectedness for data in this context refers to an automated means of linking large amounts of data entities and data points based on a knowledge representation model such as an ontology. By leveraging other semantic technologies alongside ontologies, the automatically linked data exposes unknown relationships between diverse data entities. It also facilitates inferencing and reasoning from data beyond what is applicable with disparate data entities. Ontologies create relationships between entities and are used to represent and communicate facts and relationships (Aboubacar et al., 2020). Logs in each cloud service can be easily collected and made available universally. The adoption of knowledge graphs (KGs) further facilitates data collection, processing, and analysis within the cloud. Hence, in the case of a cloud-native incident response, a KG is generated from data by instantiating security concepts in the cloud as retrieved from a cloud-based ontology devoted to security issues and incident response. Construction of KGs is iterative as more data (structured data, semi-structured data as well as unstructured data) are updated. This can eliminate the problem of limited knowledge of information suffered by some security personnel, as they will no longer need to wait for the CSP to provide information. Figure 1 shows the architecture of the proposed solution, and it consists of heterogeneous data sources and Amazon services involved.



Figure 1 – Cloud Architecture with Semantic Data Interconnectedness for Cyber Incident Response

The major step towards cloud-native incident response involves factoring IR requirements into the development of any cloud environment to ensure the automation of incident response in an orchestrated manner. When deploying cloud services, the most important sections to be considered include governance, which is responsible for regulation and compliance, visibility across various systems, and the cloud allocating roles to various stakeholders within an organization to be responsible for incident response plans. Infusing incident response requirements in the planning of a cloud environment requires a lot of security planning and execution; seeing as the successful implementation of this plan will save cloud service providers a lot of money and help in preserving reputation. Controlled access, scanning of uploaded data, multiple verifications, and verification of IP addresses are all important to the cloud-native approach in this cloud security proposal. The creation of a cloud environment will see the implementation of all these entities in the development process; concerning controlled access, employees, customers, and organization executives will have specified access to data and at no point will a party in the organization or outside of the organization have access to data uploaded by the customer. However, the nature of the data being uploaded by the customer to the cloud platform will be scanned for malicious entities before permission is granted to upload such data on the cloud platform given the possibility of cyber actors exploring means to attack the cloud service provider.

The cloud environment will also take note of the original IP address used by the customer to open a cloud account on the platform to efficiently identify the change in IP address of data upload which could signify that the customer's cloud account has been hijacked; in cases like these, more verification will be required to be certain of no malicious practices. Multiple verification is possible in case of account hijacking because customers will be required to provide more than one means of identification and the answer to a secret question which will be asked when the cloud network notices any form of unusual activity as they may be an indication of cyberattacks being attempted. Compared to the traditional response to cloud incidents, the native approach prevents cyber threats before they get to infiltrate and cause damage to the cloud service; hence, security protocols and mechanisms are considered when creating the cloud environment. These protocols will be triggered and actions that ensure security come alive automatically; what distinguishes native-cloud response to incidents from the traditional response is ‘time-conscious automation’. Based on the sample dataset utilised, the graph model, represented in Figure 2 comprises of the following:

* Event: This node is taken from Event object within the sample dataset utilised. The properties for this node are id, date, info.
* EventType: There are nodes from this node, they are Report and Malware, this two are extracted from the XML file names.
* Item: Event objects in the XML file has attribute names as Item, these nodes are generated from the attribute of each event object, Item nodes have properties as comment, value, id
* ItemCategory: These nodes are generated from the category of each Item in the XML file
* ItemType: These nodes are generated from type of each Item in the XML file
* Relations, which include:
	+ isA: from node event to node EventType
	+ hasItem: from node event to node Item
	+ hasType: from node item to node itemType
	+ hasCategory: from node item to node itemCategory



Figure 2 – Semantic graph model generated from the CTI dataset

**4.1 Proof-of-Concept Implementation**

For the proof-of-concept implementation, we design and develop a cloud architecture that provides dynamic scalability and high availability, among several other features of AWS well-architected framework to facilitate semantic data interconnectedness for cyber threat intelligence. The architecture also leverages Neo4j semantic graph database to store the graph model and processed data in graph format. To generate the proposed graph database, we need to take each XML file from the sample dataset and process it, extract nodes and relations and finally generate csv files by the way they will be used to generate graph database. For these purposes we have a python script. The XML files are stored into a folder for example, CTI dataset. These are transferred to the Neo4j import folder on running compute instances. A connection to a compute instance is established, followed by stopping neo4j service and moving the folders transactions and databases into a backup folder. Upon completion of data importing and processing, the neo4j service is started. For the design and implementation workflow, data from different sources gets streamed into an s3 bucket. Uploading data such as documents, videos, and pictures to S3 requires the creation of an S3 bucket in one of the AWS Regions. These data are uploaded to the cloud via the command line interface. Other options available are through a graphical user interface or data pipelines, with total upload time being dependent on the data size. This gives room to upload any number of objects to the bucket. To implement this, s3 offers APIs for the management of buckets and objects. Uploading of data into buckets is carried out when S3 APIs to send requests to S3. Immediately new data enters the s3 bucket, a pre-configured CloudWatch Event triggers a Lambda function. The CloudWatch Event provides an almost real-time stream of system events describing changes in AWS resources. It can match events and routing them to one or more target functions or streams. CloudWatch Event responds to changes in operations by sending messages that activate functions, make changes, and capture state information. For this deployment, the Lambda function is the target of the CloudWatch Event. When a Lambda function is triggered, it processes the data within an event (a JSON-formatted document), eventually invoking an ETL script. The configuration of the Lambda function could include connections to subnets and security groups in a virtual private cloud. Private resources are accessed by connecting Lambda functions to private subnets and there are inbound and outbound rules for security groups. When a function is connected to a public subnet, no internet access or a public IP address is provided. Network address translation (NAT) is used if the function needs internet access. Connecting a function to a virtual private cloud prompts lambda’s creation of an elastic network interface for each subnet within the virtual cloud environment. Figure 3 illustrates the data workflow from multiple raw datasets to a cloud-native semantic graph database.



Figure 3 – Data workflow from raw datasets to a cloud-native semantic graph database

For the data transformation, an ETL (Extract, Transform and Load) script to extract data from the sources, transform, and load them into targets is executed. In this instance, the ETL script reads data into Neo4j graph database on a compute instance within the auto-scaling group. Neo4j graph database is responsible for efficient implementation of the property graph model down to the storage level. The database stores data as it is on the whiteboard and utilizes pointers for navigating and traversing the graph. The auto-scaling group consists of compute instances existing as a logical grouping to achieve automatic scaling and management. The auto scaling features include scaling policies and check replacements, ensuring the maintenance of pre-defined number of compute instances and enabling dynamic launching of additional instances when necessary. Scaling ensures that the desired number of instances is running even if an instance fails and enables automatic increase or decrease in the number based on changes in demand. Hence, the graph database is elastic based on real-time traffic and computing resources utilisation. Over time, redundant data is deleted from the graph dataset using “delete” cypher query statements. In addition, the compute instance(s) provides an endpoint (web interface) for the Neo4J graph database such that users can use the Cypher query language to query the database for different purposes. Attaching a load balancer target group to an auto-scaling group implies any instance attached is registered with the load balancer. In cases where a load balancer target group is attached to an auto-scaling group, both the load balancer and the instance need to be within same virtual private cloud. The load balancing fosters automatic distribution of incoming application traffic across all running compute instances. It achieves the management of incoming requests through optimal routing of traffic to prevent any instance from being overwhelmed. Furthermore, it monitors registered targets’ health and ensures that only healthy targets receive traffic. The load balancer functions by receiving requests, evaluating listener rules in priority order towards determining applicable rules, and selecting a target from the target group to respond to the request. The application traffic’s content can serve as basis for configuring listener rules to route requests to different target groups. One of the merits of using an application load balancer rather than a classic one is the possibility of registering lambda functions as targets. Hence, an application load balancer is applicable to the current solution. The availability zones contain registered targets which the load balancer routes requests to when enabled. Overall, the main goals for adopting a cloud platform are based on other features such as high level of availability, scalability, rapid provisioning, agility, and high processing power available on demand.

**4.2 Implementation Evaluation**

To evaluate the implementation, sample cypher queries are run against the Neo4j graph database, based on the graph model developed and illustrated in Figure 2. To see graph model prototype of the database, the following cypher query is executed:

match (a:EVENTTYPE),(b:EVENT),(c:ITEM),(d:ITEMTYPE),(e:ITEMCATEGORY)

match (a)<-[r1]-(b)-[r2]->(c)

match (c)-[r3]->(d)

match (c)-[r4]->(e)

where id (c)=50 return a,b,c,d,e limit 1

Additionally, to retrieve the number of items by its type, we run the following cypher query, with the result set presented in Figure 4: match (a:ITEM)-[:hasType]->(b:ITEMTYPE) return distinct b.itemType,count(\*)



Figure 4 – Result set for sample cypher query to return number of items by category

Also, to retrieve the number of events that occurred in a specific year, for instance 2012 by item type, we run the following cypher command, as illustrated in Figure 5: match (c:EVENT)-[:hasItem]->(a:ITEM)-[:hasType]->(b:ITEMTYPE) where tostring(c.eventDate)>"2012-01-01T00:00:00" and tostring(c.eventDate)<"2013-01-01T00:00:00" return distinct b.itemType,count(\*)



Figure 5 – Result set for sample cypher query to return number of logged events per type

In addition, result sets can be retrieved for specific event types. The number of malware events by category as it occurred in a specific year (for example, 2012) is retrieved by running the following command, as shown in Figure 6: match (d:EVENTTYPE {etName:'Malware'})<-[:IS\_A]-(c:EVENT)-[:hasItem]->(a:ITEM)-[:hasCategory]->(b:ITEMCATEGORY) where tostring(c.eventDate)>"2012-01-01T00:00:00" and tostring(c.eventDate)<"2013-01-01T00:00:00" return distinct b.itemCat,count(\*)



Figure 6 – Result set for sample cypher query to return number of diverse malware events within a year.

**6.0 Conclusion**

The graph model prototype of the database was displayed when the query was run. Other queries also ran successfully, and it shows that it is possible to retrieve information such as number of items by its type, number of events which occurred in a particular year by item type, as well as number of malware events that occurred in a specific year by category. The proof-of-concept implementation alongside evaluation of its usability demonstrates how semantic technologies running on a cloud-native architecture for cyber incident response can facilitate data interconnectedness and foster timely and appropriate response for cyber incidents. With more rich-content data, the capabilities become even more evident, as it provides more basis for a very robust graph model and relationships between diverse data elements. Hence, in this paper, we were able to establish that adoption of traditional security approaches within a cloud environment is challenging as they do not consider the features of cloud technologies and services. Therefore, it was necessary to design and develop cloud-native solutions. Likewise, we have established that adoption of semantic technologies for cloud-native security solutions in cyber threat intelligence is faced by the challenge of semantic data interconnectedness. These two challenges have been addressed in this study by developing a solution on AWS cloud, with built-in features that natively leverages cloud capabilities such as dynamic scaling of resources, across multiple availability or geographic zones. Evaluation of the proof-of-concept implementation also shows the possibility of connecting multiple data sources to retrieve a variety of information and infer more knowledge from data in the cloud towards enhanced cyber threat intelligence.

**References**

Aboubacar, M. S., Castelltort, A. & Laurent, A. (2020). Knowledge graph on cybersecurity: A survey, Doctoral Congress 2020.

Ab Rahman, N. H., Glisson, W. B., Yang, Y., & Choo, K. R. (2016a). Forensic-by-design framework for cyber-physical cloud systems. In IEEE Cloud Computing, vol. 3, no. 1, pp. 50-59, Jan.-Feb. 2016, doi: 10.1109/MCC.2016.5.

Ab Rahman, N. H., Cahyani, N. W. & Choo, K. K. R (2016b). Cloud incident handling and forensic-by-design: cloud storage as a case study. In Concurrency Computation 29.14 (July 2017), ISSN 15320634.

Ab Rahman, N. H. & Choo, K. R. (2015). A survey of information security incident handling in the cloud. Computers & Security, Volume 49, 2015, Pages 45-69, ISSN 0167-4048. https://doi.org/10.1016/j.cose.2014.11.006.

Alberts, C, Dorofee, A, Killcrece, G, Ruefle, R & Zajicek, M. (2004). Defining incident management processes for CSIRTs: A work in progress.

Alsaleem, L. S., Alqahtani, S. A., Alharbi, S. F., & Agrouba, R. (2019). Cloud computing-based attacks and countermeasures: A survey. Journal of Theoretical and Applied Information Technology, Vol.97. No 19.

Amara, N., Zhiqui, H. and Ali, A. (2017). Cloud computing security threats and attacks with their mitigation techniques. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2017, pp. 244-251, doi: 10.1109/CyberC.2017.37.

Baskerville, R., Spagnoletti, P., & Kim, J. (2014). Incident-centered information security: Managing a strategic balance between prevention and response. Information & management, 51(1):138–151.

Blackwell, C. (2010). A security ontology for incident analysis. In Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, page 46. ACM.

British Standards Institution (2007). BIP 0107:2008 Foundations of IT Service Management Based on ITIL V3, UK.

Casey, T., 2017. Threat Agent Library helps identify information security risks. Intel White Paper, 2.

Cichonski, P., Millar, T, Grance, T., & Scarfone, K. (2012). Computer security incident handling guide. International Journal of Computer Research, vol. 20, no. 4, pp. 459– 530.

Dekker, M., Liveri, D., & Lakka, M. (2013). Cloud security incident reporting - Framework for reporting about major cloud security incidents. December. 2013, p. 38. ISBN: 9789279000775. doi: 10.2788/14231.

ENISA (2010). Good practice guide for incident management. ENISA, Athens.

Frøystad, C., Gjære, E. A., Tøndel, I. A., and Jaatun, M. J. (2016). Security incident information exchange for cloud services. In: Scitepress, May 2016, pp. 391–398. doi: 10.5220/0005953803910398.

Grobauer, B. & Schreck, T. (2010). Towards incident handling in the cloud, in Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop (CCSW 10), pp. 77–85.

Gupta, P., Seetharaman, A., & Raj, J. R. (2013). The usage and adoption of cloud computing by small and medium businesses. International Journal of Information Management, vol. 33, no. 5, pp. 861–874, 2013.

Hengst, K. (2020). Best practices in cloud incident handling. Master’s Thesis, University of Twente.

Henry, P., Williams, J., & Wright, B. (2013). The SANS survey of digital forensics and incident response. Tech. rep. SANS Institute.

ISO (2011). ISO/IEC 27035:2011 Information Technology - security techniques - information security incident management, Geneva.

Kaufman, L. M. (2009). Data security in the world of cloud computing. IEEE Security & Privacy, vol. 7, no. 4, pp. 61–64, 2009.

Killcrece, G. (2003). State of the practice of computer security incident response teams (CSIRTs), CMU/SEI, Pittsburgh.

Killcrece, G., Kossakowski, K. P., Ruefle, R. & Zajicek, M. (2003). Organizational models for computer security incident response teams (CSIRTs), CMU/SEI, Pittsburgh.

Kral, P. (2011). Incident Handler's Handbook. SANS Institute.

Kumar, J., Rajendran, B., Bindhumadhava, B. S. & Babu, N. S. C. (2017). XML Wrapping attack mitigation using positional token, In International Conference On Public Key Infrastructure and its applications (Pkia), Bangalore, India. Digital Investigation 9.2 (2012), pp. 71–80. issn: 17422876. doi:10.1016/j.diin.2012.07.001.

Martini, B. and Choo, K. K. R. (2012). An integrated conceptual digital forensic framework for cloud computing. Digital Investigation 9.2 (2012), pp. 71–80, ISSN: 17422876.

MITRE Corporation. (2021a) MITRE: Common Vulnerabilities and Exposures (CVE). [Online]. Available from: <https://cve.mitre.org/>

The MITRE Corporation. (2021b) MITRE: Common Platform Enumeration (CPE). [Online]. Available from: <https://cpe.mitre.org/>

MITRE Corporation. (2021c) MITRE: Common Weakness Enumeration (CWE). [Online]. Available from: <https://cwe.mitre.org/>

MITRE Corporation. (2021b) MITRE: Common Attack Pattern Enumeration and Classification. [Online]. Available from: <https://capec.mitre.org/>

MITRE Corporation. (2021e) MITRE: Adversarial Tactics, Techniques and Common Knowledge (ATT & CK). [Online]. Available from: <https://attack.mitre.org/>

Monfared, A. and Jaatun, M. G. (2012). Handling compromised components in an IaaS cloud installation. Journal of Cloud Computing: Advances, Systems and Applications, vol. 1, no. 1, pp. 1–21.

Mogull, R., Arlen, J., Lane, A., Peterson, G., Rothman, M., and Mortman, D. (2017). Security guidance for critical areas of focus in cloud computing v4.0. Tech. rep. Cloud Security Alliance, 2017.

Moreira, G. B., Calegario, V. M., Duarte, J. C., & Santos, A. F. (2018). CSIHO: An Ontology for Computer Security Incident Handling. https://sol.sbc.org.br/index.php/sbseg/article/view/4239/4170

Mundie, D., Ruefle, R., Dorofee, A., McCloud, J., Perl, S., & Collins, M. (2014). An incident management ontology. CEUR Workshop Proceedings. 1304. 62-71.

O’Sullivan, K. and Turnbull, B. (2015). The cyber simulation terrain: Towards an open source cyber effects simulation ontology. In Australian Information Warfare Conference, pages 14–23. Security Research Institute, Edith Cowan University, 05 Nov. de 2017.

Ozer, M., S. Varlioglu, B. Gonen, V. Adewopo, N. Elsayed and S. Zengin (2020). Cloud incident response: Challenges and opportunities. International Conference on Computational Science and Computational Intelligence (CSCI), 2020, pp. 49-54, doi: 10.1109/CSCI51800.2020.00015.

Rajendran, R., Kumar, S., Palanichamy, Y., & Arputharaj, K. (2018). Detection of dos attacks in cloud networks using intelligent rule based classification system. Cluster Computing, 2018.

Purnaye, P. and Kulkarni, V., 2022. A comprehensive study of cloud forensics. Archives of Computational Methods in Engineering, 29(1), pp.33-46.

Šendelj, R. and Ognjanović, I. (2014). Semantically enhanced cyber security over clouds: Methodological approach. International Journal of Advances in Computer Networks and Its Security – IJCNS, 4(3).

Shaikh, A. A. (2016). Attacks on cloud computing and its countermeasures. In 2016 International Conference On Signal Processing, Communication, Power And Embedded System (Scopes), Paralakhemundi, India, 2016.

Silva, P. C. D. and Fagundes, L. L. (2014). Simo: Security incident management ontology. In SimposioBrasileiroemSeguranc¸a da Informac¸ ´ ao e de SistemasComputacionais, pages 302–305, Bras´ılia. SociedadeBrasileira de Computac¸ao. 05 nov. de 2017.

Soman, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in Cloud Computing: Issues, Taxonomy, and Future directions. Computer Communications, Vol. 107.

Sri, N. O, Tapas, K., & Vedula, V. (2017). A survey on security aspects of server virtualization in cloud computing. International Journal of Electrical and Computer Engineering (Ijece), Vol. 7, No. 3, Pp. 1326-1336.

Srinivasan, J. (2015). Semantic cloud architecture an integration of cloud and semantic web. IMS Manthan. The Journal of Innovations, 8(2).

U.S. National Vulnerability Database. NVD (2021): Common Vulnerability Scoring System (CVSS) v2 [Online]. Available from: https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator

West-Brown, M. J, Stikvoort, D., Kossakowski, K. P., Killcrere, G., Ruefle, R. & Zajicek, M. (2003). Handbook for computer security incident response teams (CSIRTs), 2nd edn, Carnegie Mellon/SEI, Pittsburgh.

Yucel, T. and Romuald, K. K. (2015). Cloud computing virtualization and cyber attacks: Evidence centralization. CivilComp Press, Stirlingshire, Scotland.

Zhang, S., Zhang, S., Chen, X., and Huo, X. (2010). Cloud computing research and development trend. Second International Conference on Future Networks, 2010, pp. 93-97, doi: 10.1109/ICFN.2010.58.

Zhang, X., Wuwong, N., Li, H. and Zhang, X. (2010). Information security risk management framework for the cloud computing environments. In: Proceedings - 10th IEEE International Conference on Computer and Information Technology, CIT- 2010, 7th IEEE International Conference on Embedded Software and Systems, ICESS-2010, ScalCom-2010. 2010, pp. 1328–1334.