Contextual Intrusion Alerts for Scada Networks
An Ontology based Approach for Intrusion Alerts Post Processing

Abdullah Al Balushi, Kieran McLaughlin and Sakir Sezer
Centre for Secure Information Technologies, Queens University Belfast, Belfast BT3 9DT, U.K.

Keywords: Security Ontology, Intrusion Detection System, Modbus TCP, Intrusion Alert Analysis.

Abstract: The complexity of modern SCADA networks and their associated cyber-attacks requires an expressive but flexible manner for representing both domain knowledge and collected intrusion alerts with the ability to integrate them for enhanced analytical capabilities and better understanding of attacks. This paper proposes an ontology-based approach for contextualized intrusion alerts in SCADA networks. In this approach, three security ontologies were developed to represent and store information on intrusion alerts, Modbus communications, and Modbus attack descriptions. This information is correlated into enriched intrusion alerts using simple ontology logic rules written in Semantic Query-Enhanced Web Rules (SQWRL). The contextualized alerts give analysts the means to better understand evolving attacks and to uncover the semantic relationships between sequences of individual attack events. The proposed system is illustrated by two use case scenarios.

1 INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems are used for monitoring and controlling many critical infrastructures including power grids, water distribution, transportation systems, and even nuclear power plants. In the past, most of these systems were physically isolated from outside connections and operated on their own proprietary hardware and communication protocols. However, these systems are moving towards the use of general ICT infrastructures and open standard technologies. This move brought many operational and management benefits, but also exposed the systems to an increased number of cyber security threats. Recently, intrusion detection systems (IDS) have been introduced to SCADA networks. An IDS examines data collected from networks and systems to identify suspicious events and generates appropriate intrusion alerts describing the attack details. However, IDS systems usually provide elementary and low level descriptions of the detected individual attacks, making it a challenging task for security analysts to verify the nature and significance of reported attacks. Further, mapping discrete intrusion alerts to the context of where they occurred, and understanding their impact on the security state of a SCADA system is a challenging task.

In this paper, we propose an ontology-based approach to integrate contextual information and external knowledge into intrusion alerts. We developed three security ontologies (models) for the representation and storage of knowledge on intrusion alerts and Modbus communications context and cyber attacks. Alert information and contextual information about the Modbus network communications are correlated by the use of simple ontology logic rules written in Semantic Query-Enhanced Web Rules (SQWRL). The proposed approach is implemented in a system prototype using open-source tools and is illustrated using two use case scenarios.

The rest of this paper is organized as follows. Section 2 presents related works. Proposed approach is introduced in Section 3. Development of ontologies is explained in Section 4. System implementation and use cases are presented in Section 5. Section 6 concludes the paper.

2 RELATED WORK

Recently, a variety of intrusion detection approaches have been proposed for the surveillance and monitoring of industrial process control infrastructures (Mitchell, 2014). Snort IDS (Roesch, 1999) is an example of general IDS system that became a de facto standard in many environments. Digital Bond Inc. (Peterson, 2009) has developed a set of customized Snort IDS rules for the detection of different attacks on Modbus TCP. Their detection rules consist of 14

Balushi, A., McLaughlin, K. and Sezer, S.
Contextual Intrusion Alerts for Scada Networks - An Ontology based Approach for Intrusion Alerts Post Processing.
DOI: 10.5220/0005745504570464
In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pages 457-464
ISBN: 978-989-758-167-0

signatures and include denial of service (rebooting Modbus servers, listen-only mode commands, and crashing server with a large packets), reconnaissance (e.g., unauthorized attempt to read data, gathering device information), and unauthorized write requests.

Cyber attacks on SCADA systems may require a particular context to proceed. This context includes specific network configurations, communication protocols, system and application configuration, and hardware or system components. Thus, contextual information can undoubtedly play a considerable role in intrusion analysis and understanding the possible impacts of attacks on the systems. Unfortunately, most of the current IDS tend to consider only a little part of the context and generate a huge amount of isolated intrusion alerts (Sadighian et al., 2013). These alerts provide very little attack descriptions and analysts can rarely make decisions about security events without manually analyzing their surrounding context. To address these issues and to enhance the context of intrusion alerts, several approaches have been proposed in (Cuppens et al., 2009; Frye et al., 2012; Sadighian et al., 2013).

An ontology is a semantic web technique for knowledge representation and is used for explicit specification of particular domain conceptualities that capture its context. It is a formal way of encoding concepts (classes), properties (relations), axioms, constraints and instances into a machine interpretable language that easily allows sharing semantic information between human and systems.

(Cuppens et al., 2009) proposed an ontology-based approach to map alerts into the attack context. In their approach, context is used to identify network policies that can be used to solve the threat. The use of ontological representation of networks and attacks is presented by (Frye et al., 2012). The authors used ontologies to describe network traffic and generic attacks for the purpose of identifying complex attacks. In (Sadighian et al., 2013), an ontology based approach is proposed for reducing false alerts in multi-sensor environments by the incorporation of contextual information obtained from vulnerability databases and context sensors deployed in the network.

In summary, the concept of ontology techniques have been applied to attack modeling, knowledge representation, and context awareness. However, most of the works focus on general communication networks and do not consider SCADA network context. The complexity of SCADA networks and their associated cyber attacks requires an expressive, but flexible manner, for representing both SCADA domain expert knowledge and collected intrusion evidences. This should be supported by the ability to easily integrate these data for enhanced analytical capabilities and better understanding of attacks. The use of ontology approaches for contextualizing intrusion alerts in SCADA networks can bring many advantages. An ontology enables expressing the knowledge with clear structure and detailed definition in a machine-interpretable format. Moreover, semantics (the meaning behind the data) can be added to data and interpreted using the context definitions and restrictions for new data classifications. Finally, the ontology can help in integrating information from different sources in a flexible way.

3 PROPOSED APPROACH

The proposed ontology-based approach is for correlating and enhancing intrusion alerts with contextual information in SCADA industrial control networks. The essential components of the system are illustrated in Figure 1. These components are labeled with numbers linked to the following main steps in our approach. Detailed information about the technical implementation of each component is provided in Section 5.1.

Step 1. Formal knowledge representation models for traffic, Modbus cyber attacks, and intrusion alerts are developed using ontologies and stored in a knowledge base. These ontologies capture the main properties of cyber-attacks on SCADA systems residing in the communication protocols and systems. They are used by subsequent components of the system to transform input data to ontological format and the integration of SCADA intrusion context.

Step 2. Network traffic containing Modbus packets is captured and fed into the proposed system along with generated intrusion alerts by Snort IDS. This data is parsed and converted to Resource Description Framework (RDF) triples, which are used as ontological representation format. An RDF triple represents data in three elements pairs that are subject, predicate, and object. This format is supported by ontologies and the SPARQL query engine described in the system implementation. These RDF triples are added to the knowledge base as instances.

Step 3. The core analysis engine running on Apache Jena API library and SPARQL queries retrieves correlation rules from the knowledge base and execute them against the ontology instances to integrate contextual information or extract attack relationships in the alerts and packet instances.

Step 4. The output contextual intrusion alert is added to the knowledge base and forwarded to the security analyst for their decision.

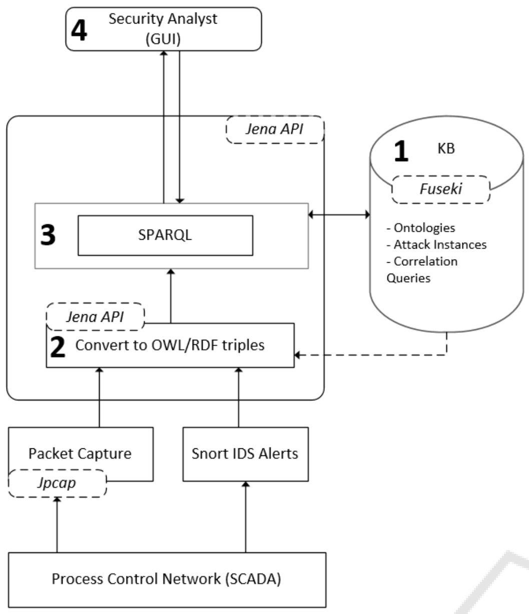


Figure 1: An overview of the System architecture.

4 DEVELOPMENT OF ONTOLOGIES

In this paper, a total of three security ontology models have been developed. The first constructed model captures the main concepts in Modbus TCP communications such as network and application layer features with their semantic interrelationships. This model caters the protocol specifications described in official Modbus documentations (IDA, 2004; Modbus, 2012). It provides the base for transforming network traffic into ontological representation for processing and context extraction. The second model is the cyber attacks ontology, that is used to represent and store Modbus attacks information as well as their interrelationships. The third and last model represents the intrusion alerts. We emphasise that the development of any ontology is an iterative process, and there is no single correct way of mapping knowledge into discrete structure.

4.1 Modbus TCP Ontology

Modbus TCP communication has a simple request-response architecture that is well-defined in the official Modbus documentations(IDA, 2004; Modbus, 2012). Modbus packet can be classified as either a request or a response message, where a client (Master) sends a packet to the Modbus server requesting a function to be executed or a specific data to be returned. The requested function can be reading values, writing values, or performing some server diagnostics. The server then processes this request and returns a response to the client. This response can be either a normal response with the returned data or an exception (error) response to indicate a failure in fulfilling the request. Figure 2 demonstrates the basic flow of Modbus transactions, while detailed packet structure is presented in Figure 3.

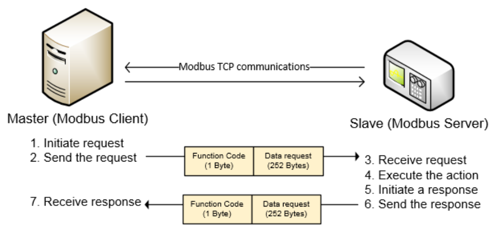


Figure 2: Modbus TCP transactions flow.

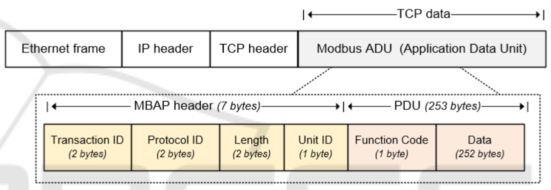


Figure 3: Modbus TCP packet structure.

The main application-level features of Modbus TCP packets are described in Table 1. These features can be exploited by adversaries to perform various attacks such as disrupting process control operations, hiding their attack traces on the system, injecting false measurement data, or exploiting buffer overflow vulnerabilities. For example, an attacker may be able to force a Modbus server into Listen-Mode-Only using a single Modbus packet with function code feature set to 08 and sub-function code of 04. This will result in placing the Modbus server in an inactive state and will subsequently cause a complete desynchronization with other systems.

Figure 4 presents the derived ontology for Modbus TCP communication packets. Any Modbus packet can either be classified as a request or a response message. Thus, we consider these as two main classes in the ontology. Both types of messages share the same header section, so we construct a Header class. In order to allow high level classification of function codes in Modbus TCP packets, we classify them into three different groups; Public, System-defined, and User-defined. The public functions group can be further divided to subclasses, namely Diagnostics codes, Read codes, Write codes. These subclasses are based on the purpose of function codes. An example is that the

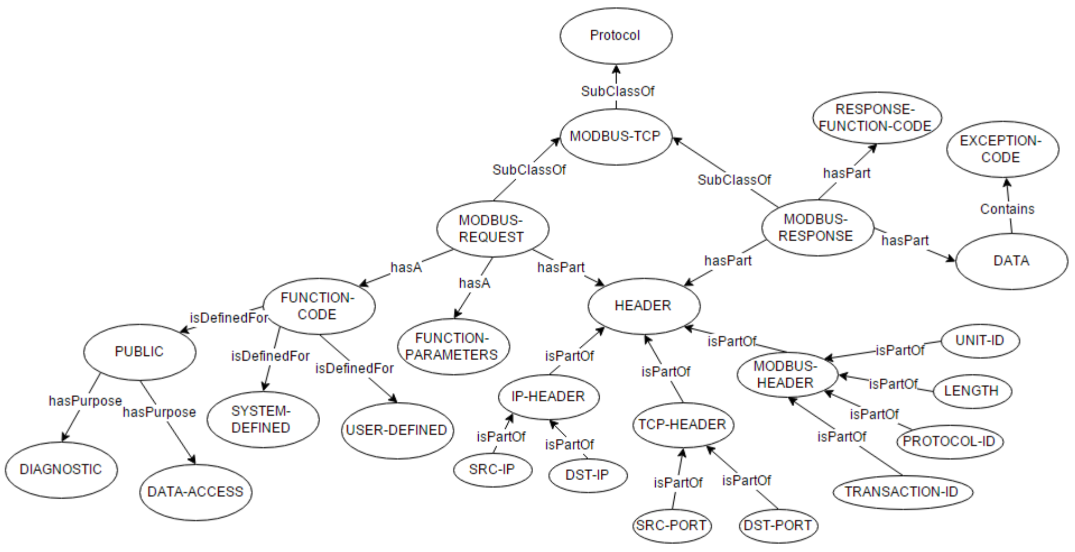


Figure 4: Modbus TCP communication ontology.

Table 1: Main application-level features in Modbus TCP.

Section	Feature	Description
Modbus Application (MBAP) Header	Trans ID	A numerical identifier for the current transaction for the purpose of pairing request and response.
	Protocol ID	Describes the protocol in use. Should be always set to zero for Modbus TCP.
	Length	Specifies the length of the remaining part of the packet.
	Unit ID	Unique device ID when Modbus gateway is used. If not, then it is set to 0xFF.
Protocol Data Unit (PDU)	Function Code	Specifies the action to be performed by the slave device.
	Data	Contains data of variable size with the data corresponding to the request or response.

public diagnostic codes can be used for system audit related operations such as Get Com event counter, Get Com Event Log, Return Diagnostic Register, Clear Counters and Diagnostic Registers.

In ontologies, classes and their subclasses are organized in a hierarchy. The main classes in the Modbus TCP communication ontology are described in Table 2. Each of the presented classes may have a relationship with other classes. These relations are important to define whether two classes should be disjoint from each other or one is parent class. For example, IP-Header is disjoint from TCP-Header. An ontology allows two types of properties; object and data. Object properties are used to describe relationships between classes, while data properties are used to store data value. A partial list of these properties is presented in Table 3.

Table 2: Main Classes in Modbus TCP ontology.

Class	Description
HEADER	This class represents the main header features that exist in any Modbus packets over TCP. Subclasses: IP header, TCP header, Modbus header
MODBUS REQUEST	This class represent a Modbus request message sent from client to Modbus server. Subclasses: Function-Code, Data
MODBUS RESPONSE	This class represent a Modbus response message sent from Modbus server to client. Subclasses: Function-Code, Data, Exception-Code

Table 3: Object and data properties in Modbus TCP.

Object Properties	SubclassOf, hasA, isDefinedFor, hasPurpose, isPartOf, Contains
Data Properties	hasUnitID, hasTransactionID, hasProtocolID, hasLength, hasSrcIP, hasDstIP, hasFunctionCode, hasExceptionCode, hasSrcPort, hasDstPort

4.2 Modbus Attacks Ontology

This model is used to capture and maintain information about cyber attacks on SCADA systems that are utilizing Modbus TCP protocol as an attack vector. Constructing an ontological representation of these attacks enables viewing them in high-level abstraction and extracting their shared attributes such as common impacts on SCADA system. To construct this ontology we started by analyzing cyber-attacks categories against Modbus TCP presented in (Huitsing et al., 2008; Zhu et al., 2011; Hadžiosmanović et al., 2014; Drias et al., 2015). The main concepts in the proposed attacks model are Attack and Attacker. Figure 5 illustrates the relationship exist between these main classes and their next level subclasses.

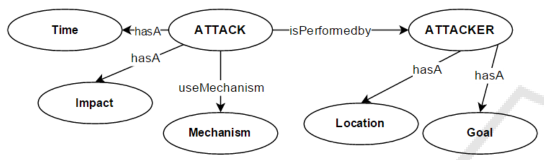


Figure 5: Ontological relationship of main classes in attacks ontology.

We describe the main classes and their associated properties as follows:

Attacker Class. This class represents the attacker object and its attributes such as the source IP address of an attacker, his previous intrusive activities obtained from historical intrusion alerts and his goals. An attacker may aim for one of the following goals, as captured by the ontology: to change data, disrupt operations, gain control, steal data, or prepare for another objective. An ontological representation of these concepts are illustrated in Figure 6.

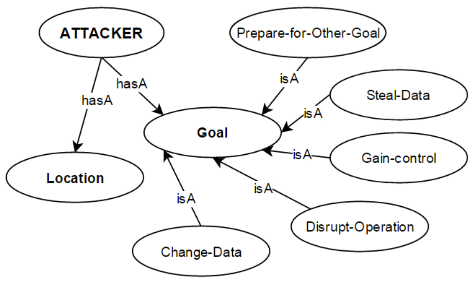


Figure 6: Ontological view of attacker class.

Attack Class. This class is used to represent the associated attributes of an attack such as the time of an attack instance, possible impact outcome and description of the mechanism used to conduct the attack. For instance, the impact of an attack can be on Data-Integrity, Denial-of-Service, Physical-Damage, Process-Integrity, System-Integrity, Reconnaissance, and Process-Reconnaissance. For example, an administrative command (Function code 8, Sub Code 0A) that clear counters and diagnostic audit records on the server is associated with System Integrity impact. Other commands such as enforcing listen only mode or restarting the TCP communication can result in Denial of service impact. In addition, an attack can be performed using different mechanisms where four of the related mechanisms are added to the attack ontology model as shown in Figure 7.

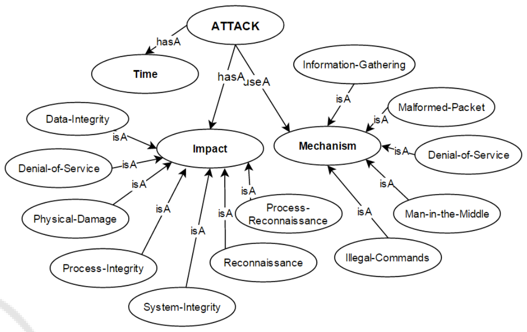


Figure 7: Ontological view for the Attack Class.

4.3 Intrusion Alerts Ontology

In order to contextualize intrusion alerts, an ontological representation (RDF triples) of these alerts is required. The conversion of alerts and semantic relationships extraction are defined in the intrusion alert model. We restrict this ontology to features that exist in Snort IDS alerts which are Time, Source IP, Source Port, Destination IP, Destination Port, Signature ID, Protocol, Alert Classification and Alert Message. We also add some additional features derived from the semantic analysis, which are necessary, but cannot be provided by Snort. These features include the context protocol in application layer (e.g., Modbus), the impact on SCADA system, affected systems, attack description, and recommended action. For example, the protocol feature in raw Snort alerts is described as TCP. However, multiple protocols can be encapsulated in TCP frames including HTTP and MODBUS. Thus,we describe the protocol more precisely by setting relevant feature to Modbus-TCP.

5 IMPLEMENTATION AND USE CASES

In order to functionally test the proposed approach, we constructed a system prototype using the three ontologies developed in section 4 and open-source tools that are described in the following sections.

5.1 Reference Implementation

The high-level system architecture was presented in Figure 1. This system was developed using Java on top of open-source Apache Jena API library. The proposed system consists of three ontologies for Modbus TCP communications, Modbus TCP attacks and Intrusion alerts. In order to derive and implement these ontologies, we used the Ontology Web Language Description Logic (OWL-DL) language through an open-source Protege ontology editor and knowledge acquisition framework. The resulted ontologies are then stored in the knowledge base which is based on Apache Fueski v2. The main inputs are network packet captures (PCAP) and Snort IDS intrusion alerts. The input data is mapped to the ontological representation models defined in Modbus TCP communication and intrusion alerts ontologies. We used the Resource Description Framework (RDF) triples format to store the converted input data in the knowledge base as instances. This mapping process is performed by the system using Java and Apache Jena API. After having all input data stored as ontology instances in the knowledge base, the core analysis engine can perform queries using Semantic Query-Enhanced Web Rule Language (SQWRL) through Apache Jena API library. The result of these queries are then presented to the security analyst and can be added as new ontology instances to the knowledge base.

5.2 Use Case Scenarios

This section presents two use case scenarios. The first scenario focuses on analysing individual raw intrusion alerts received from Snort IDS and integrating them with contextual SCADA information. The aim of this scenario is to show the advantages of enhancing contextual information in intrusion alerts for security analysts. The second scenario extends the use of contextual information for the identification of complex attack scenarios utilizing the relationship of the enhanced individual alerts. For these scenarios, we used Snort IDS with customised Modbus TCP detection rules that were developed by Digital Bond Quick-Draw (Peterson, 2009).

5.2.1 Semantic Integration of Contextual Information

In the first scenario, we demonstrate the process of semantically integrating diverse information within the ontologies knowledge base into a rich context intrusion alert. The benefit of this is to provide security analyst with descriptive attack information considering the context of Modbus communications. Let us consider an attack scenario where the attacker performs a series of simple attack steps to cause data desynchronization between the Modbus master and clients. Assuming the attacker has limited knowledge about the Modbus server implementation under attack, he may attempt sending spoofed Modbus packets to read the device identification or perform a function scan to determine which functions are allowed. After determining the server implementation, the attacker may abuse some of the allowed function codes to cause a denial of service such as Force-Listen-Only-Mode (Function code: 08, Sub code 04). Each of these attack activities may be detected by IDS systems. However, the intrusion alert messages presented to security analysts usually provide only low-level information. For example, one attack step of forcing the Modbus server into Listen only mode is reported by Snort IDS as follow:

```
09/24-11:02:51.531149 [**] [1:1111001:1] SCADA_IDS:
Modbus TCP-Force Listen Only Mode [**] [Classification:
attempted-dos][Priority:1]{TCP} 192.168.3.100:49880->
192.168.3.20:502
```

This alert provides little description about the attack step and reports the protocol feature as TCP with general attack impact of attempted DoS. However, security analysts may require more descriptive attack information such as what is the Force listen only attack and how can it be mitigated. In addition, the impact of an attack on a Modbus network may be described more precisely. For instance, the reconnaissance attacks on Modbus TCP can be associated with two types of impacts that are Process-Reconnaissance or System-Reconnaissance. The Process-Reconnaissance is where the attacker analyses the structure of industrial process such as specific functions implementations and process flow. While in the system reconnaissance the attacker analyses the Modbus server functionality and product version. Determining these specific attack impacts can reveal possible attacker objectives.

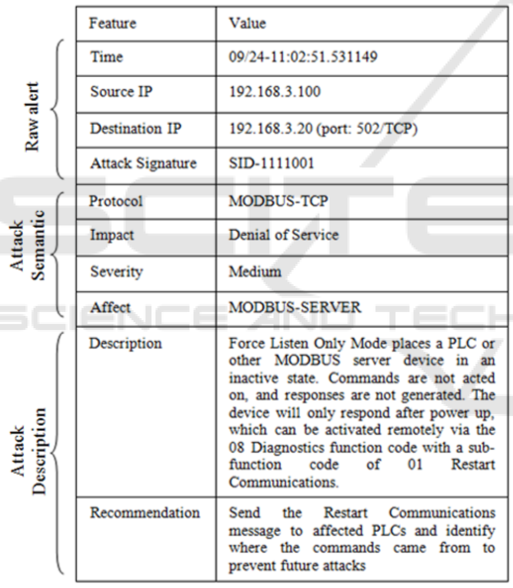
To address these concerns, the following SPARQL query is performed to extract attack description from the Modbus TCP attacks ontology and integrate it with the raw alert features presented in the above alert example.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX intrusions: <http://server1/ontologies/IntrusionAl
 erts.owl#>
PREFIX attacks:<http://server1/ontologies/ModbusAttacks.
 owl#>

  SELECT ?alerttime ?alertsrcip ?alertdstip ?protocol
  ?impact ?severity ?affect ?description
```

```
WHERE {
  #alert information from intrusion alert ontology
  ?alert rdf:type intrusions:IntrusionAlert .
  ?alert intrusions:hasTime ?alerttime .
  ?alert intrusions:hasSrcIP ?alertsrcip .
  ?alert intrusions:hasDstIP ?alertdstip .
  ?alert intrusions:hasDstPort ?alertdstport .
  ?alert intrusions:hasAttackSid ?alertAttacksid .
  #attack information from Modbus attacks ontology
  ?attack attacks:hasAttackSid ?attackSID .
  ?attack attacks:useProtocol ?protocol .
  ?attack attacks:hasImpact ?impact .
  ?attack attacks:hasSeverity ?severity .
  ?attack attacks:hasAffecton ?affect .
  ?attack attacks:hasDescription ?description .
     FILTER (sameTerm(?attackSID,?alertAttacksid))
 }
```

The result of executing the above SPARQL query provides new contextual information as shown in Figure 8 which describes the attack in more details and classify intrusion alert in more specific way in term of description the exact protocol and possible impact. This new contextual alert is then added to the ontology knowledge base as ModbusTCPAlert.



	Feature	Value
Raw alert	Time	09/24-11:02:51.531149
	Source IP	192.168.3.100
	Destination IP	192.168.3.20 (port: 502/TCP)
	Attack Signature	SID-1111001
Attack Semantic	Protocol	MODBUS-TCP
	Impact	Denial of Service
	Severity	Medium
	Affect	MODBUS-SERVER
Attack Description	Description	Force Listen Only Mode places a PLC or other MODBUS server device in an inactive state. Commands are not acted on, and responses are not generated. The device will only respond after power up, which can be activated remotely via the 08 Diagnostics function code with a sub-function code of 01 Restart Communications.
	Recommendation	Send the Restart Communications message to affected PLCs and identify where the commands came from to prevent future attacks

Figure 8: The contextual intrusion alert.

As Figure 8 shows, new attack semantic features have been added to the alert context. The security analysts can understand the intrusion context in a more descriptive way. In addition, the extended features such as determined application layer protocol, specified attack impact, and the affected system allows for new search capabilities among intrusion alerts. For instance, it is possible to retrieve all intrusion alert instances of ModbusTCPAlert that share the same attack impact or affect a specific target system while in raw Snort IDS alerts, the search may be limited to IP addresses, ports, attack signatures, generic protocols (e.g., TCP, UDP).

5.2.2 Semantic Correlation for Complex Attack Detection

As an example of complex attacks, we describe a distributed denial of service in SCADA networks. In this attack scenario, we assume that the attacker successfully introduced a malware infection in the network resulting in compromising multiple engineer host machines. The attacker now is able to instruct these infected machines to perform individual attacks on a Modbus server. The attacker now instructs all infected machines to send malicious (but legitimate looking) Modbus TCP packets to the same Modbus server target with Modbus TCP communication restart command (function code 08, sub function 01). This command will force the Modbus server to restart and power up self-tests. During this power up process, the Modbus server will be unavailable for short time. However, repeatedly sending these requests can force the Modbus server into a restart loop making it unavailable for a long time. These attack activities are detected by Snort IDS, but reported in isolation without suspicion that they are part of a complex distributed denial of service as they are originating from different machine sources. Since these individual alerts are received by our system and added to the ontologies knowledge base as ModbusTCPAlert instances with shared impact of denial of service, the system is able to retrieve all individual instances against the same Modbus server target. To do this, the system execute the following SPARQL query:

```
SELECT ?Time ?AttackerSourceIP ?ModbusServerIP
?impact ?affect

WHERE {
 #Modbus-TCP alert instances information
 ?ModbusTCPalert rdf:type intrusions:ModbusTCPAlert .
 ?ModusTCPalert intrusions:hasTime ?Time .
 ?ModbusTCPalert intrusions:hasSrcIP ?AttackerSourceIP .
 ?ModbusTCPalert intrusions:hasDstIP ?ModbusServerIP .
 ?ModbusTCPalert intrusions:hasImpact ?impact .
 ?ModbusTCPalert intrusions:hasAffectOn ?affect .
    FILTER regex(str(?impact), "Denial of Service")
 }
```

The above query will search the knowledge base for existence of Modbus TCP alert instances that report denial of service attack and are targeting same Modbus server which indicates a possible distributed denial of service attack if they originate from different machine IP sources. If this matches, a summary of the attacker source IP addresses, the Modbus server IP address, and the attack impact is retrieved. The system processes the query result in the background and provide the security analyst with the following predefined attack summary:

```
    Possible DDoS attack detected!

87 Modbus requests with possible impact of denial of
```

```
service have been detected over the last 3 mintues.

Alerts count:    87
Impact:          Denial of Service
Affected item:   PLCs and other field devices
                 that contain MODBUS servers
Attack vectors:  Modbus TCP restart(84 times),
                 Force-Listen-Only-Mode (3 times)
Target server:   192.168.3.20
Attack sources:  192.168.3.100, 192.168.3.102,
                 192.168.3.107
Start time:      09/27-09:03:23.435311
End time:        09/27-09:06:57.229114
```

This message provides the security analysts with very descriptive attack information instead of flooding them with every single alert (total of 87 alerts). This enables a quicker response to perform further investigation or take a corrective actions. The attack summary provides useful information about the detected attack scenario. This includes the number of generated alerts, the impact, the affected asset, summary of the attacks vectors, the Modbus server under attack, and more importantly the list of machine IP addresses that are performing the suspicious activities.

6 CONCLUSION

This paper has proposed an ontology-based approach for knowledge representation and integration for contextualizing intrusion alerts in SCADA networks. An ontology enables expressing the domain knowledge and collected intrusion alerts with clear structure and detailed definition in a machine-interpretable format. Moreover, the semantic meaning of Modbus control commands are extracted and new classifications are possible to derived. Furthermore, as all knowledge data are represented in the same format using the developed models, it is possible to integrate them in a flexible way. The contextualized intrusion alerts produced by our approach provides better attack descriptions and new classifications based on the semantic meaning of the extracted control commands. This provides the analysts the means to better understand evolving attacks and to uncover the semantic relationships between sequences of individual attacks. In future work it is intended that the proposed approach will be evaluated against a more comprehensive Modbus dataset.

REFERENCES

Cuppens, N., Cuppens, F., Autrel, F., and Debar, H. (2009). An ontology-based approach to react to network attacks. *IJICS*, 3(3-4):280–305.

Drias, Z., Serhrouchni, A., and Vogel, O. (2015). Taxonomy of attacks on industrial control protocols. In *ICPE and NTDS Conference, 2015*, pages 1–6. IEEE.

Frye, L., Cheng, L., and Heflin, J. (2012). An ontology-based system to identify complex network attacks. In *IEEE (ICC)*, pages 6683–6688. IEEE.

Hadžiosmanović, D., Sommer, R., Zambon, E., and Hartel, P. H. (2014). Through the eye of the plc: semantic security monitoring for industrial processes. In *30th ACSAC*, pages 126–135. ACM.

Huitsing, P., Chandia, R., Papa, M., and Shenoi, S. (2008). Attack taxonomies for the modbus protocols. *IJCIP*, 1:37–44.

IDA, M. (2004). Modbus messaging on tcp/ip implementation guide v1. 0a.

Mitchell, Robert, C. I.-R. (2014). A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)*, 46(4):55.

Modbus (2012). Modbus application protocol specification v1. 1b3. *Modbus Organization, Inc., April*, 26.

Peterson, D. (2009). Quickdraw: Generating security log events for legacy scada and control system devices. In *CATCH'09*, pages 227–229. IEEE.

Roesch, M. (1999). Snort nids.

Sadighian, A., Zargar, S. T., Fernandez, J. M., and Lemay, A. (2013). Semantic-based context-aware alert fusion for distributed intrusion detection systems. In *(CRiSIS)*, pages 1–6. IEEE.

Zhu, B., Joseph, A., and Sastry, S. (2011). A taxonomy of cyber attacks on scada systems. In *4th CPSCom*, pages 380–388. IEEE.