Information Systems Frontiers
https://doi.org/10.1007/s10796-020-10103-7

Toward Automated Cyber Defense with Secure Sharing of Structured Cyber Threat Intelligence

Md. Farhan Haque[1] · **Ram Krishnan**[1]

Accepted: 30 December 2020

Abstract
Cyber Threat sharing helps with defending against cyber attacks in a timely manner. Many frameworks have been proposed for CTI sharing such as Structured Threat Information Expression (STIX) and Trusted Automated Exchange of Intelligence Information (TAXII). However, CTI sharing in a controlled and automated manner is critical. In this paper, we demonstrate Relationship Based Access Control (ReBAC) as an appropriate model for CTI sharing. We also develop an approach for automated threat detection, generation and sharing of structured CTI and taking course of actions to mitigate cyber threats. Finally, we implement an Automated Cyber Defense System in a cloud based environment.

Keywords Cyber Threat Intelligence (CTI) · Structured Threat Information Expression (STIX) · Trusted Automated Exchange of Intelligence Information (TAXII) · Relationship Based Access Control (ReBAC) · OpenStack

1 Introduction and Motivation

Cyber Threat Intelligence (CTI) Intelligence (CTI) is a type of cyber threat information which goes through certain cybersecurity standards through the scrutiny of cybersecurity experts and is collected from reliable sources. CTI provides essential cyber threat information which can be critical to maintain safety and protect integrity of an organization in cyber space. These CTI can also provide valuable insights about cyber attacks and a significant amount of research material to counter against future cyber attacks. In today's data driven world, there is a high demand for CTI sharing in a large quantity. An efficient CTI sharing can boost Cyber Threat Intelligence of an individual organization. Haass et al. (2015) presented the importance of CTI sharing to develop a fast and efficient threat response system. For example: organizations can take faster Course of Actions[1] in response to threat intelligence of malware.

CTI generally contains detailed information related to a cyber attack. For example: a simple Phishing (Jagatic et al. 2007) email attack can have several key features such as attacker information, attack techniques used, target of attack, tools and software used to launch the attack. A well agreed standard is required to express and share several key features of an attack process efficiently. Structured Threat Information Expression (STIX)[2] is a language and serialization format used to exchange CTI maintained by OASIS.[3] STIX enables organizations to share CTI in machine readable manner, allowing other organizations and security communities to get useful insights about an attack and take preventive measures. We focus on sharing CTI in a structured manner and adopt STIX[4] standards into our implementation.

✉ Md. Farhan Haque
md.farhan.haque@utsa.edu

Ram Krishnan
ram.krishnan@utsa.edu

[1] Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX, USA

[1]Course of Action. https://oasis-open.github.io/cti-documentation/stix/intro, accessed: 2019-07-08

[2]Stix : A structured language for cyber threat intelligence. https://oasis-open.github.io/cti-documentation/, accessed: 2019-07-01

[3]Oasis cyber threat intelligence (cti) tc. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti, accessed: 2019-07-09

[4]Stix : A structured language for cyber threat intelligence. https://oasis-open.github.io/cti-documentation/, accessed: 2019-07-01

Published online: 03 February 2021

Organizations may require to share these CTI in a controlled manner. For example: Three organizations A, B and C where A trusts B more than C. Organization A may want to share more with B than C due to information leakage (Chaabane et al. 2012), privacy (Lane et al. 2014) concerns, etc. Organizations can adopt some form of access control (Sandhu and Samarati 1994) over CTI sharing based on the different sharing requirements. There are various forms of access control models such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role Based Access Control (RBAC), Attribute Based Access Control(ABAC), Relationship Based Access Control (ReBAC), etc. It is not well understood the effectiveness of all these access control models for CTI sharing. In our work, we investigate the practical applicability of all these access control models for secure CTI sharing. ReBAC seems to be a natural fit as organizations are able to facilitate different levels of CTI sharing. We adopt Cheng et al.'s (2012) User-to-User Relationship Based Access Control (UURAC) model to control CTI sharing. The advantages of this model are discussed in Sections 4.4.1–2.

Organizations have a need to detect various cyber threats such as malware threats, malicious network traffic threats , etc. in a real time and mitigate those threats. Organizations can also package the encountered cyber threat as CTI and structure into STIX format to share with others. Organizations can learn about a cyber threat when they detect a threat or an another organization shares a threat CTI with them. We provide a practical approach to automate STIX generation from cyber threat detection. Organizations can mitigate or prevent these threats by taking effective course of actions. In this paper, we also discuss about the generation of configurable and automatable course of actions. To summarize, our contributions in this paper are as follows:

1. We develop an automated approach to generate STIX CTI from real cyber threats. (See Sections 4.3.1 & 4.3.3)
2. We demonstrate the applicability of ReBAC for effective sharing of CTI by presenting an example CTI sharing scenario and propose a ReBAC framework for CTI sharing. (See Section 4.4.1). We have published this work in Haque and Krishnan (2019).
3. We develop a practical approach to configure course of actions and demonstrate the automation of course of actions. (See Section 4.5).
4. We develop a prototype implementation of an Automated Cyber Defense System in the context of cloud computing which includes all the above features. (See Section 5).

2 Background

In this section, we discuss a few key concepts involving our work.

2.1 Structured Threat Information Expression

STIX is a standard to express structured CTI and has two key components: STIX Domain Objects (SDO) and STIX Relationship Objects (SRO). SDOs are individual information blocks to express CTI categorically. Each block communicates a high level CTI concept and its built-in properties explain the details about that concept. For example: Threat Actor block represents individuals, groups or organizations which may have malicious intent and more likely to pose cybersecurity threats to other individuals or organizations. Threat Actor has a few properties such as name, goals, motivation and skill level to describe details about the threat actor. The domain object properties are filled with pre established vocabularies and open ended descriptions. There are eighteen domain objects in STIX which involve crucial CTI related to vulnerabilities, attack pattern, malware, course of actions, etc.

STIX relationship objects connect two domain objects and demonstrate inter domain relationships. For example: The Malware domain object represents CTI related to malicious codes or programs. We can link Threat Actor and Malware domain objects by using a "Uses" relationship: "Threat Actor (SDO) Uses (SRO) Malware (SDO)". We can use multiple STIX domain and relationship objects together to represent complicated CTI in a very structured manner. STIX documents are represented in json file format and Fig. 1 shows a few examples.

Course of actions for cyber defense can be any action to prevent or mitigate a cyber threat. We can categorize course of actions into two broad categories: technical and non technical. The technical course of actions would be backing up of resources, monitoring firewall and network traffic, port scanning for malicious traffic, etc. The non technical course of actions would be conducting employee training to deal with phishing emails, creating awareness against social engineering attacks, etc. STIX provides a course of action domain object but does not facilitate automatable course of actions at the moment.

2.2 Trusted Automated Exchange of Intelligence Information

Trusted Automated Exchange of Intelligence Information or TAXII (Connolly et al. 2014) is a suggested application protocol to exchange CTI over the network. CTI in STIX format can also be transported with other communication

```
{
   "type": "malware",
   "id": "malware--d8fac658-32fa-4d51-931c-b8e61b02520a",
   "created": "2020-03-29T17:40:06.052Z",
   "modified": "2020-03-29T17:40:06.052Z",
   "name": "malware-1",
   "malware_types": [ "Ransomware" ]
}
```

```
{
   "type": "threat-actor",
   "id": "threat-actor--3e6f3fcf-ec41-4a83-8143-33e88e13fc23",
   "created": "2020-03-29T17:40:06.055Z",
   "modified": "2020-03-29T17:40:06.055Z",
   "name": "attacker-1",
   "labels": [ "hacker", "criminal" ]
}
```


Fig. 1 STIX representation in json

protocols. TAXII supports two sharing models: Collection and Channel.

1. **Collection:** Collection operates on a request response model where data can be hosted on a TAXII server and consumer can get data by request. We adopt this model of CTI sharing into our work.
2. **Channel:** Channel sharing operates on publish subscribe model. CTI producers publish data on TAXII server and consumers subscribe to get CTI.

2.3 Relationship Based Access Control

Access control is a known mechanism to control access to resources in computer based systems. There are several forms of access control models such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), Attribute Based Access Control (ABAC) and Role Based Access Control (RBAC) (Sandhu and Samarati 1994), etc. There is a more recent form of access control model named as Relationship Based Access Control (ReBAC) proposed by Gates (2007). ReBAC grants access to resources based on the relationship between the accessor and the owner. ReBAC is popular in online social networks (Garton et al. 1997) scenario because of its intuitive relationship based structure. We use ReBAC in our implementation because organizations may be unrelated or loosely related with each other and only come together to share different levels of CTI. These types of sharing requirements can easily be facilitated by establishing sharing relationships.

3 Related Work

Johnson et al. (2016) defined cyber threat information is as any information that can help an organization identify, assess, monitor, and respond to cyber threats. The authors put emphasize on the importance of CTI sharing and provided a few use cases for cyber threat information sharing such as nation state attacks against a specific industry sector, distributed denial of service attack against another industry sector, financial conference phishing attack, etc. Haass et al. (2015) demonstrated a case study for information sharing challenges within a public/private not for profit partnership organization called ACTRA: Arizona Cyber Threat Response Alliance, Inc.

STIX and TAXII are an approach to represent and share CTI in an automated and machine readable manner. STIX and TAXII are maintained by OASIS[5] and a well accepted standard for CTI. For example: a malware STIX communicates important malware related information such as malware name, malware type etc in a structured way. There are plenty of opportunities to perform analysis on structured threat intelligence to extract meaningful information and apply them to better organizational cyber defense. Iannacone et al. (2015) provided an ontology to develop for cybersecurity knowledge graph similar to Google's knowledge graph which incorporates information from both structured and unstructured information sources. Syed et al. (2016) proposed Unified Cybersecurity Ontology (UCO) which integrates and incorporates data from various cybersecurity standards and also mapped with archived STIX 1.0 (Barnum 2012). We plan to develop a robust cyber defense system with the capability of STIX analyses.

Gates (2007) introduced Relationship Based Access Control (ReBAC) where access to a resource depends on the relationship between owner and accessor. Over the years, several numbers of ReBAC models have been proposed in the literature. Fong (2011) proposed a modal logic based relationship based access control policy in a social network context. Crampton and Sellwood (2014) provided a relationship based access control policy based on path conditions which are similar to regular expressions. Cheng et al. (2012) provided a regular expression based relationship based access control model for online social networks. Cheng et al.'s model makes an authorization decision based on multiple policies which is beneficial for our CTI sharing requirements.

Cyber threat detection such as intrusion detection, intrusion prevention, malware threat detection, etc. has a rich literature. The machine learning based detection approaches of these threats are the current state of the

[5]Oasis cyber threat intelligence (cti) tc. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti, accessed: 2019-07-09

arts research. Debar et al. (1992) developed a neural networks based approach for intrusion detection. Li and Liu (2010) developed a SNORT based intrusion prevention using support vector machine (SVM). Abdelsalam et al. (2018) developed a convolutional neural networks approach to detect malware in the system. We have used K-means (Hartigan and Wong 1979) algorithm for threat detection. Course of actions is also one of the 18 domain objects defined in STIX. Collaborative Automated Course of Action Operations (CACAO) is a group working on the development of automated course of actions to mitigate cyber threats (see[6]). Automated course of actions are an integral part of an effective cyber defense.

4 Automated Cyber Defense System

4.1 CTI Sharing Requirements

Let us consider different CTI sharing requirements based on geographical locations (Intracity) and collaboration with law enforcement agencies (Lawenforcement).

1. **Sharing Requirement 1 - Intracity:** Imagine that there is a surge of Ransomware (Mansfield-Devine 2016) attacks directed towards critical organizations in San Antonio, Texas such as banks, airports, hospitals, etc. These attacks are circulated through Email spoofing (Pandove et al. 2010) and Social engineering (Thornburgh 2004) tactics. Health institutions in San Antonio understand these cyber threats against the city and can agree to share malware CTI.
2. **Sharing Requirement 2 - Lawenforcement:** Cyber criminals (Burden and Palmer 2003) can launch attacks which may have serious consequences in real world and pose security risks to infrastructures and employees of an organization. These cyber crimes may need to be reported to law enforcement agencies. Organizations can agree to share threat actor (attacker information) CTI with law enforcement agencies.

4.2 Current CTI Sharing Framework: STIX and TAXII

STIX provides a standard to structure threat intelligence and TAXII provides a mechanism to store and share those information. Organizations can share CTI using STIX and TAXII. Figure 2 shows the current infrastructure for CTI sharing between two organizations named Ace Health SA and Sacred Lake SA. TAXII Client communicates with TAXII Server to exchange (push and pull) STIX documents. For example: Ace Health SA's TAXII Client pushes STIX documents into the STIX database and Sacred Lake SA's TAXII Client can send a request to pull STIX documents from Ace Health SA's TAXII Server.

However these standards are well accepted for CTI sharing, we investigate the following research questions:

1. How can we automate threat detection and generate STIX documents?
2. How can we automate controlled sharing of STIX through TAXII?
3. How can we automate actionable course of actions by analyzing STIX ?

Our contribution is the Automated Cyber Defense System which has the following components to answer the above questions (see Fig. 3).

1. Automated Threat Detection and STIX Generation (marked in green in Fig. 3) answers question 1.
2. Automated CTI Sharing(marked in black in Fig. 3) answers question 2.
3. Automated Course of Actions(marked in red in Fig. 3) answers question 3.

4.3 Automated Threat Detection and STIX Generation

We have developed Threat Detection System, Data Handler REST Project and STIX Generation System showed in Fig. 3 to perform automated threat detection and STIX generation.

4.3.1 Threat Detection System

Threat Detection System is a conceptual unit which has the capabilities to detect cyber threats and able to generate necessary data to produce valid STIX documents. The system is an abstract idea to detect varieties of cyber threats. This can be an Intrusion Detection System (IDS) (Liao et al. 2013) which monitors network traffics such as SNORT. Threat Detection System can also be human driven. An admin can manually monitor cyber threats using different tools and can generate threat data. Some human usable COTs tools are Splunk, MITRE ATT&CK MATRIX, Wireshark, etc. Threat Detection System can also be automated such as various automated threat detection tools, antivirus software, open source projects, etc.

[6]CACAO: a future for collaborative cybersecurity course of action. https://www.lookingglasscyber.com/blog/cacao-a-future-for-collaborative-cybersecurity-course-of-action/, accessed: 2020-03-10, OASIS Collaborative Automated Course of Action Operations (CACAO) for Cyber Security TC. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cacao, accessed: 2020-03-10

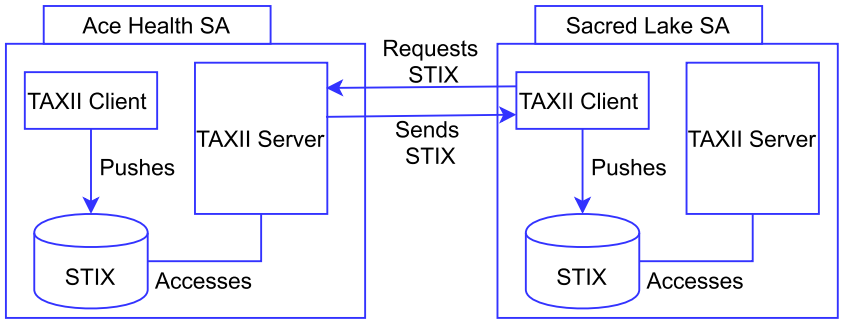

Fig. 2 Current state of STIX and TAXII

4.3.2 Data Handler REST Project

STIX Generation System should be integrable with any threat detection system. Any COTs threat detection tool should be able to plugged into STIX Generation System. Data Handler REST Project works as an intermediary between Threat Detection System and STIX Generation System. Data Handler REST Project allows Threat Detection System to send threat data and then stores the data in an internal database. Threat Detection System communicates with Data Handler REST Project through REST API. An example instance of threat data sent to the api endpoint "⟨restproject − serverIP⟩/malware" URL is {"malware − name" :"ddos − 1","malware − type" : "DDOS"}. STIX Generation system can get the the malware data by sending get request to endpoint ⟨restproject − serverIP⟩/malware" of Data Handler REST Project. The internal application logic prevents any duplicate data from storing in the database.

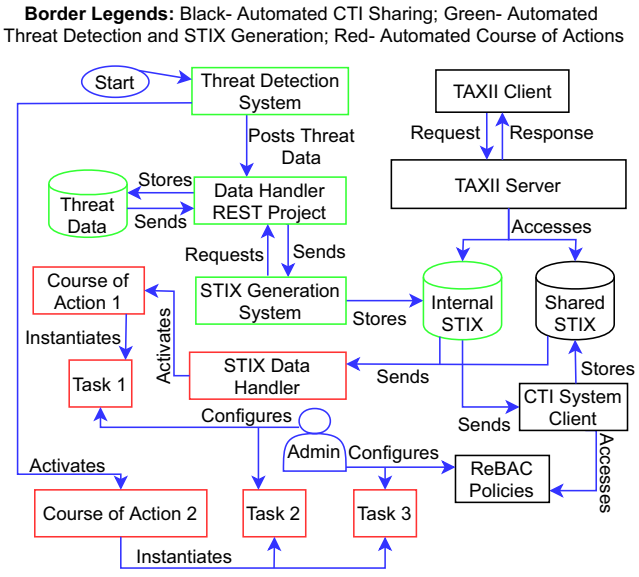

Fig. 3 Automated cyber defense system

4.3.3 STIX Generation System

STIX provides a standard to structure and categorize CTI aligned with variable sharing requirements. STIX provides some predefined property requirements to be valid. STIX also allows to set user defined property requirements. For example: some predefined STIX Report categories are Malware, Threat Actor, Attack Pattern , etc. We can set user defined property requirements such as a Malware Report must include at least one Malware domain object. Malware domain object also has STIX defined required properties such as malware name and malware labels or types. STIX Generation System requests threat data from Data Handler REST Project and creates STIX documents that meet both valid STIX schema requirements and user defined properties requirements and stores them in an internal database.

4.4 Automated CTI Sharing

We have designed a centralized authentication and authorization system named as CTI System (outside of Automated Cyber Defense System) to facilitate controlled CTI sharing between different organizations. CTI System implements ReBAC as access control. The CTI System and client, TAXII server and client in Fig. 3 perform automated CTI sharing with other organizations.

4.4.1 ReBAC Policies

We have explored the applicability of major forms of access controls such as MAC, DAC, RBAC, ABAC and ReBAC for CTI sharing requirements in Fig. 4.1.

1. **Motivation to Adopt ReBAC for CTI Sharing:** Access Control List (ACL) (Sandhu and Samarati 1994) is one of the DAC approaches where we have to maintain a list of subject's access rights for each object. In our scenario, each individual STIX type would be objects and organizations would be subjects. Then we have to maintain ACLs for every STIX types

for each organization. For example: we keep a list of organizations allowed to read Threat Actor type STIX of Ace Health SA. We can also do the same for other STIX types. This kind of approach is cumbersome work for an individual organization and consumes huge amount of system and human resources. RBAC (Sandhu and Samarati 1994) is a popular form of access control in enterprise scenario where access to a resource is granted based on the roles adopted by the user. But RBAC may be ineffective in organizational CTI sharing scenario for two reasons. First, organizations may only want to share CTI when there is an active sharing need. That will cause frequent assignment and removal of roles as an organization. Second, the sharing requirements between two organizations from one sector can vary greatly from two other organizations from the same sector. That can cause RBAC model to face with Role Explosion (Elliott and Knight 2010) problem. Fong (2011) demonstrated a few advantages of ReBAC with respect to RBAC model.

Another popular access control model is Attribute Based Access Control (ABAC) (Hu et al. 2015). An ABAC model makes authorization decision based on user, resource and environmental attributes. Let assume, we have chosen user or organization attribute as security clearance and resource attribute as data sensitivity. Independent organizations do not follow any such hierarchical structures. That would limit flexible and variable resource sharing between organizations as an organization may want to share resources with another organization now but later, they may not want share resources anymore. In that case, changing data sensitivity of resources may interrupt with other existing sharing arrangements. Similarly, we can also consider relationship as an attribute for ABAC. However, it is not well understood how the relationship attribute will handle multi-level or indirect relationships between organizations. Other environmental attributes such as data location, access time do not make sense for CTI sharing.

2. **ReBAC in CTI Sharing Scenario:** We now present a relationship based organizational CTI sharing scenario in Fig. 4.

Organizations have established different types of sharing relationships to facilitate various levels of CTI sharing. We consider two sharing relationships: Intracity and Lawenforcement relevant to our sharing requirements. The directional lines represent one directional relationships while the non directional lines represent bidirectional relationships. We can now state a few sample ReBAC policies consistent with sharing requirements:

(a) **Intracity:** Will give access to Malware Report STIX.
(b) **Lawenforcement:** Will give access to Threat Actor Report STIX.

There are two major ReBAC models in research literature. They are: Fong's ReBAC and Cheng et. al's UURAC ReBAC. Fong's ReBAC model is a single policy per resource ReBAC model and can be used in our implementation. On the other hand, Cheng et. al's ReBAC has three types of policies: requester, resource and system policies which are more realistic for organizations to have their own separate policies. These multiple access control policies together make authorization decisions and provide more finer grained control over the sharing of resources. We adopt this ReBAC model into our implementation due to this feature. The model defines both user and resource as potential targets for an authorization decision in online social networks (Garton et al. 1997). An example of user as target is when an user performs an action on another user such as poking in Facebook.

In our sharing scenario, users are organizations and we do not consider them as actionable targets. We rather focus on CTI resources owned by an organization as targets. We consider three policies from Cheng et al.'s access control policy taxonomy. They are system specified policy (SP) for a resource, Accessing User Policy (AUP) for outgoing actions of organizations as users and Target Resource Policy (TRP) for incoming actions to a resource in an organization.

(a) **System Specified Policy:** CTI System admin sets up a System specified policy (SP) which determines the access or denial of a system wide Access Request (Cheng et al. 2012) from an authorized user of a requesting/accessing organization to access another organization's CTI. An instantiation of SP for our CTI sharing scenario:

⟨read, ThreatActor, (RequestingOrganization, (Lawenforcement∗, 5))⟩,
⟨read, Malware, (RequestingOrganization, (Intracity, 5))⟩

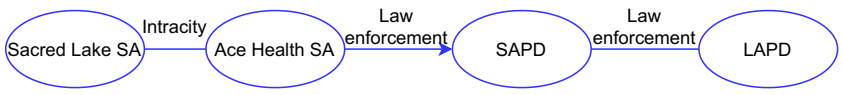

Fig. 4 Organizational CTI sharing scenario

The above policy is defined for read operation of two different STIX types: Threat Actor and Malware. The word "read" is the policy action, "ThreatActor" is the resource name, "RequestingOrganization" is the policy evaluation start point, "Lawenforcement*" is the regex based relationship type and "5" is the maximum allowable hop count distance between the requesting and the resource owner organization.

(b) **Accessing User Policy:** Each organization's system admin sets up an Accessing User Policy (AUP) to control all the outgoing requests from the employees and prevents any unsolicited outgoing request from that organization. For example: Sacred Lake admin may not want any of it's employees to send access requests to FBI Tyler. An instantiation of AUP for our ReBAC based CTI sharing scenario:
$\langle$read, (LAPD, (Intracity $*$ $-$Lawenforcment$*$, 3))$\rangle$,
$\langle$read, (SacredLakeSA, (Intracity $*$ $-$Lawenforcment$*$, 2))$\rangle$

(c) **Target Resource Policy:** System admin of an organization also sets up Target Resource Policy (TRP) for each STIX type of that organization to control the access of their own CTI. This policy provides organizations an extral level of control over their own CTI as organizations do not have any control over joining or leaving organizations in the CTI sharing ecosystem. In Fig. 4, Ace Health SA trusts SAPD with Lawenforcement relationship and wants to share Threat Actor CTI. CTI System maintains System specified Policies (SP) that may allow the sharing of Threat Actor CTI with any two organizations having direct or indirect Lawenforcement relationship between them. Later when SAPD establishes another Lawenforcement relationship with LAPD, LAPD will then gain the access to Ace Health SA's Threat Actor CTI according to SP. But if Ace Health SA is unwilling to share it with LAPD, they can control LAPD's access to their Threat Actor CTI through the enforcement of their own Target Resource Policy (TRP) for Threat Actor. An instantiation of TRP for our CTI sharing scenario:
$\langle$read^{-1}, ThreatActor, (AceHealthSA, (Lawenforcment$*$, 1))$\rangle$,
$\langle$read$-$1,Malware, (AceHealthSA, (Intracity$*$, 1))$\rangle$

3. **ReBAC Authorization Decision Process:** The CTI System makes authorization decisions by considering three policies. They are: AUP of accessing/requesting organization, TRP of resource owner organization for the requested resource and SP of CTI System for the same resource. These three policies are verified against the CTI sharing ecosystem. Each policy evaluation result is represented by a boolean result of true or false. If the requesting organization and resource owner organization are matched with relationship type and are within the maximum hop count limit specified in the policy, the policy evaluation yields in a true result.

The policy evaluation results of these three types of policies may individually yield in different boolean results and can cause a decision conflict. In case of a decision conflict, Cheng et al. proposed disjunctive, conjunctive and prioritized approaches to resolve the conflict. We have adopted the conjunctive approach into our implementation which means access is granted if all the three policy evaluation results are true. In Fig. 4, Sacred Lake SA and Ace Health SA have Intracity sharing relationship. CTI System would allow Sacred Lake SA to read Ace Health SA's Malware Reports according to all the three types of policies (AUP, TRP and SP).

4.4.2 CTI System and Client

Each organization has a CTI System Client which Communicates with CTI System and other organization's CTI System Client to share and receive CTI. The CTI System Client accesses organizational ReBAC polices setup by an admin and securely shares them with CTI System. The CTI System Client also stores CTI received from other organizations inside Shared STIX database.

4.4.3 TAXII Server and Client

A TAXII Server responds to a request from a TAXII Client. When Sacred Lake SA's CTI System Client sends a request to access a CTI from Ace Health SA in Fig. 4, CTI System checks the authorization of Sacred Lake SA for the requested resource and securely shares the authorization decision with Ace Health SA's CTI System Client. Based on the decision, Ace Health SA's CTI System Client instructs the TAXII Client to pull the resource from their TAXII Server and securely send towards Sacred Lake SA's CTI System Client.

4.5 Automated Course of Actions

A course of action is any action that can be used to prevent a cyber attack or mitigate an already encountered attack. In our Automated Cyber Defense System, a course of action is automatically activated in response to a cyber threat detection. A cyber threat can be detected in two ways inside our system:

1. **Internal Threat Detection:** The Threat Detection System inside an organization can detect a cyber threat (see Fig. 3).
2. **External Threat Detection:** An organization can receive a CTI about cyber threats from an another organization.

The STIX Data Handler and Course of Actions 1 & 2 in Fig. 3 perform automated course of actions. In order to automate course of actions, we have designed course of actions to be modular. A course of action can be formed by combining several smaller actions which we have named as tasks.

4.5.1 Tasks

Tasks can be defined as various sub actions of a course of action. For example: an employee training program to teach about Phishing attacks is a course of action. Each employee must get a handout of the training program before the training starts. Thus the distribution of handouts is a task to be performed to conduct employee training course of action. We now discuss the process of configuring and using the tasks.

1. **Task Configuration:** An admin in the organization identifies the required information to perform a task. The training program admin needs to know the attending employee numbers to distribute handouts. So, the distribution of handouts task needs one parameter: attending employee numbers. This task can be reused by instantiating attending employee numbers with different values. The distribution of handouts task's parameter definition is as follows: HandoutsDistribution :⟨Attendingemployeenumbers⟩.
2. **Task Instantiation:** An admin inside an organization instantiates a task by setting values to the task parameters. An example of two instances of distribution of handouts tasks for both HR and Legal departments are $\text{HandoutsDistribution}_{\text{HR}}$:⟨AttendingEmployee Numbers : 50⟩ and $\text{HandoutsDistribution}_{\text{Legal}}$:⟨AttendingEmployeeNumbers : 70⟩. Each organization can configure and instantiate the tasks for course of actions according to their own requirements.

4.5.2 Course of Actions

An admin inside an organization creates a course of action by instantiating at least one or more of these tasks. The admin can create the employee training program course of action as: EmployeeTraining :⟨$\text{HandoutsDistribution}_{\text{HR}}$ &$\text{HandoutsDistribution}_{\text{Legal}}$⟩. Multiple task instantiations can be included in a course of action with ANDs and ORs.

5 Implementation

The major operations of Automated Cyber Defense System and the components that perform those operations are:

- **Automated Threat Detection and STIX Generation:** Threat Detection System, STIX Data Handler and STIX Generation System.
- **Automated CTI Sharing:** CTI System & Client, TAXII server & Client.
- **Automated Course of Actions:** STIX Data Handler and Course of Actions System.

The three types of admins and users such as CTI System admins, organizational admins and users perform initial system setup. Table 1 shows a few example of admin and user operations.

5.1 Experimental Setup of Automated Cyber Defense System

Figure 5 shows the experimental implementation setup. We now discuss different components of Fig. 5.

5.1.1 OpenStack Nova and Web Server

OpenStack is an open source IaaS provider which offers different cloud services. OpenStack Nova is the compute service which allows to create, delete and manage servers. We have setup an apache web server using compute service which returns a static web page upon request. We have not established any protection mechanisms such as firewalls to block certain traffics, Intrusion Detection Systems , etc. on the server to conduct the experiment seamlessly. The server has a linux based ubuntu operating system.

Legitimate clients are used to generate realistic web traffic to the web server. We have established 5 different clients in different machines. In order to simulate realistic Pareto ON/OFF (Bohnert and Monteiro 2005) network

Table 1 System admin and user operations

Admin/User	Operations
CTI System Admin	Manages organizations and relationships
CTI System Admin	Manages organization's ReBAC policies
CTI System Admin	Enforces ReBAC
CTI System Admin	Manages & monitors sharing organizations
CTI System Admin	Manages & monitors sharing relationships
Organizational Admin	Creates and updates ReBAC policies
Organizational Admin	Initiates relationship request protocol
Organization Admin	Configures course of actions
Organizational User	Initiates resource request protocol

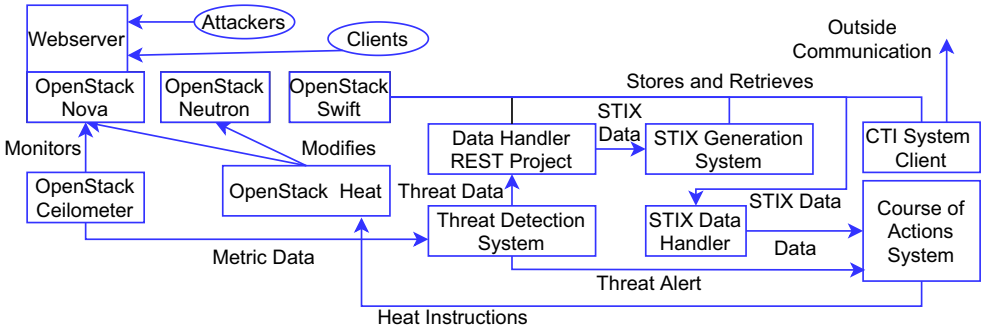

Fig. 5 Experimental implementation setup

traffic, we have developed scripts for clients to burst requests and stay idle at random intervals. We have also setup 5 different clients to work as attackers. We have used Metasploit, a popular penetration testing tool to launch DDOS attacks. In this experiment, we have launched a particular form of DDOS attack named Slowloris which tries to exhaust the web server by keeping as many connections open for as long as possible. Each attacker client is capable of opening 150 connections making a total of 750 connections from 5 clients at a single time.

5.2 OpenStack Neutron and OpenStack Swift

OpenStack Neutron provides networking services. We have made the web server accessible to both clients and attackers from outside networks using Neutron. OpenStack Swift is a storage service. We have stored internal and shared STIX documents and raw threat data from the Threat Detection System in different databases using Swift service.

5.3 Threat Detection System

We have developed a threat detection system as a web service which detects threats through the analysis of metric data of the web server. We have used K-means clustering machine learning algorithm (Hartigan and Wong 1979) to detect threat anomalies by analyzing metric data. Here are a few details about our work flow process.

- K-means Algorithm: K-means algorithm clusters data points based on the similar behaviors. We have chosen K= 2 clusters approach where one cluster is for normal web server behavior and the another one is for web server behavior during an attack.
- Data Collection: OpenStack Ceilometer[7] is metering and data collection service. We have collected web server metrics data shown in Table 2 using this service. We have collected one hour of metrics data during legitimate traffic and one hour of metrics data during DDOS (Slowloris) attack with 10 seconds of granularity.
- Data Normalization: Web server metric data are normalized using Min-Max normalization. Normalization is needed because K-means algorithm can be sensitive to higher number values. Min-Max Normalization for a metric Y can be defined as:

$$Y' = \frac{Y - Ymin}{Ymax - ymin}$$

- Training Process: We have trained the K-means algorithm with one hour of web server metric data during legitimate traffic and one hour of metric data with DDOS (Slowloris) attack and legitimate traffic.
- Testing Process: We have collected web server metric data on different intervals during both legitimate and DDOS (Slowloris) traffic and predicted the appropriate cluster. If there is any metric sample that belongs to anomalous (DDOS attack) cluster, a threat alarm is raised and the threat data is sent to Data Handler REST Project for further processing.

Table 2 Web sever metrics

Metric name	Metric type	Unit
Cpu	Instance	Nanoseconds
Disk device allocation	Instance disk	Bytes
Disk device usage	Instance disk	Bytes
Disk device write bytes	Instance disk	Bytes
Disk device write latency	Instance disk	Nanoseconds
Disk device write requests	Instance disk	Request count
Memory resident	Instance	Megabytes
Memory usage	Instance	Megabytes
Network incoming bytes	Instance network interface	Bytes
Network incoming packets	Instance network interface	Packet count
Network outgoing bytes	Instance network interface	Bytes
Network outgoing packets	Instance network interface	Pakcet count

[7] Ceilometer. https://www.openstack.org/software/releases/ussuri/components/ceilometer, accessed: 2019-07-01

5.4 Data Handler REST Project

The Data Handler REST project is a web service hosted on OpenStack server. The service is capable of handling http/s requests. We have opened api endpoints with post requests to send threat data and get requests to retrieve the threat data. In our experiment, the threat information is sent to "baseapi/DDOS" endpoint. The web application logic handles any duplication of data and stores threat information into a database using Swift.

The get request process works a bit differently. When a client sends a get request to "baseapi/DDOS", the endpoint would return all the DDOS information available. These apis are meant to be consumed by internal services and should not be open for public consumption. A DAC authorization mechanism is placed for services to send and retrieve threat data from these endpoints.

5.5 STIX Generation System

STIX Generation system is also a web service hosted on OpenStack server. The service does not interact with the Threat Detection System directly as it may interrupt the detection system and gather raw app data. Instead STIX generation system requests threat data from Data Handler REST Project to generate STIX. This system works as a core for internal STIX generation and then dumps the generated STIX documents into an internal database. We have used STIX 2.0 Python library to generate valid STIX schemas.

5.6 CTI System and Client

An organization's CTI System Client communicate with CTI System and other organization's CTI System Clients through secure communication protocols and CTI System processes sharing relationship requests and resource requests from organizations through the enforcement of Cheng et. al's ReBAC.

5.6.1 Secure Communication Protocols

We have developed two protocols for secure processing of communication requests between organizations. The first protocol is sharing relationship addition request protocol which demonstrates the communications among two organizations and CTI System to securely establish a sharing relationship between organizations. The second protocol is resource request protocol which shows the secure processing of a resource request from an organization to the resource owner organization.

These protocols are server to server three way communication protocols between two organizations and CTI System and are built on top of known communication protocols such as Meadows (1996). CTI System makes decisions to allow or deny resource requests based on access control policies and identities of organizations. Both the protocols have two implementation prerequisites in order to establish a secure communication. First prerequisite is to implement Meadows (1996) public key protocol to mutually authenticate two participating organizations and CTI System. We have chosen public key version of Needham Schroeder to avoid the "Key exchange problem". Second prerequisite is to share a session key between those two organizations in a secure manner after Needham Schroeder for further communications and resource transfers.

5.6.2 Prerequisite 1: Needham Schroeder Public Key Protocol

The Needham–Schroeder protocol is a popular authentication protocol and has two variations: symmetric key and public key. We have adopted the public key protocol with the assumption that each organization and CTI System have their respective RSA public private key pairs. We have implemented the modified version of the protocol free from man in the middle attack. An instantiation of Needham Schroeder public key exchange among Sacred Lake SA, CTI System and Ace Health SA is shown in Fig. 6.

5.6.3 Prerequisite 2: Session Key Share

Since Needham Schroeder pubic key protocol does not establish a shared session key; Sacred Lake SA and Ace Health SA then need to share a session key for secure communication and data exchange. Ace Health SA generates a session key through symmetric key generation algorithm and securely sends to Sacred Lake SA. Figure 6 also shows secure sharing of session key between Sacred Lake SA and Ace Health SA after the Needham Schroeder protocol. These two prerequisites are required before both protocol 1 and 2.

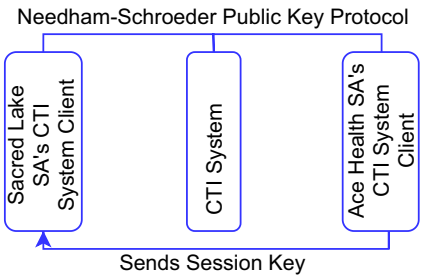

Fig. 6 Needham Schroeder public key protocol & session key share (protocol details in (Skm detailed protocols. https://github.com/farhan071024/SKM-Detailed-Protocols-, accessed: 2020-09-13))

5.6.4 Protocol 1: Relationship Request Protocol

Needham schroeder implementation ensures the identities of both Sacred Lake SA and Ace Health SA. CTI System is the central body which processes any relationship requests from any of the organizations. Figure 7 shows an example of the protocol.

Let us consider, Sacred Lake SA sends an encrypted and Integrity[8] protected request to Ace Health SA to establish an Intracity relationship. Ace Health SA verifies the integrity of the request and forwards the request to CTI System along with their own signed approval request. After successful verification of signed requests from both Sacred Lake SA and Ace Health SA, CTI System establishes the Intracity relationship between them. CTI System keeps records of organizations and relationships in Neo4j graph database where nodes are organizations and edges are relationships.

5.6.5 Protocol 2: Resource Request Protocol

Let us consider, Sacred Lake SA wants to read DDOS course of action CTI owned by Ace Health SA. Sacred Lake SA sends an encrypted and signed request to Ace Health SA after both prerequisite 1 and 2 have been completed. Ace Health SA verifies the request

and forwards the request to CTI System along with their own signed request. After successful verification of signed requests from both Sacred Lake SA and Ace Health SA, CTI System makes an authorization decision by verifying the Accessing User Policy (AUP) of Sacred Lake SA, Target Resource Policy (TRP) for DDOS course of action CTI of Ace Health SA and System specified Policy (SP) for DDOS course of action of CTI System. CTI System then securely sends the authorization decision to Ace Health SA. Based on the authorization decision, Ace Health SA's TAXII Client pulls DDOS course of action STIX from their TAXII Server, encrypts the STIX with the shared session key and securely sends to Sacred Lake SA. Figure 8 shows an example of the protocol. In our experiment, the CTI System Client requests and receives DDOS course of action STIX from an outside organization using these protocols.

5.7 STIX Data Handler

STIX Data Handler is a web service hosted on OpenStack server. In our experient, STIX Data Handler communicates with STIX databases through REST api and sends a get request to the url: "⟨basedatabaseapi⟩/DDOS". The api is an internal api and STIX Data Handler needs to

[8]Confidentiality, Integrity, Availability: The three components of the CIA Triad. https://security.blogoverflow.com/2012/08/confidentiality-bintegrity-availability-the-three-components-of-the-cia-triad/, accessed: 2019-08-13

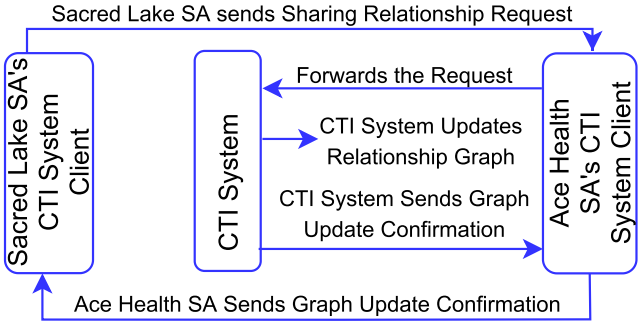

Fig. 7 Relationship request protocol (protocol details in (Skm detailed protocols. https://github.com/farhan071024/SKM-Detailed-Protocols-, accessed: 2020-09-13))

be authenticated to make the successful requests to the database. The database then return all the DDOS related STIX and STIX Data Handler parses relevant information according to the request of a Course of Actions system. In our experiment, Course of Actions system requests Slowloris course of actions from STIX Data Handler.

5.8 Course of Actions System and OpenStack Heat

The Course of Actions system is a web service hosted on OpenStack server. The Course of Actions system is notified when the Threat Detection System detects a threat. In our experiment, Course of Actions is notified about Slowloris DDOS attack from Threat Detection System. Course of Actions then requests STIX Data Handler for course of actions related to Slowloris. STIX Data Handler returns the following: { "boot":"5", "network": "alternate", "image": "secure-backup"}. It basically means to scale 5 different web servers in a different network to ease DDOS attack and each web server must boot from a known secure state.

Course of Actions system identifies two different tasks to be performed to execute Slowloris course of actions. These two tasks are: boot and network segmentation. OpenStack Heat is an OpenStack orchestration service to launch

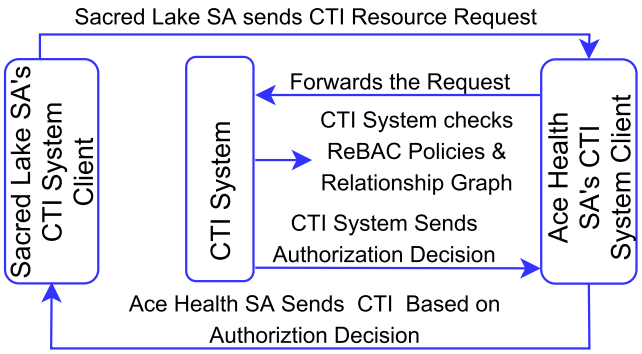

Fig. 8 Resource (CTI) request protocol (protocol details in (Skm detailed protocols. https://github.com/farhan071024/SKM-Detailed-Protocols-, accessed: 2020-09-13))

multiple composite cloud applications based on templates in the form of text files that can be treated like code. Boot web server and network segmentation tasks can also be performed by instantiating a predefined Heat templates with required parameters. Course of Actions system instantiates the heat template to boot 5 web servers in a different network and instructs Heat service to execute the template. Heat then performs modifications in the existing OpenStack infrastructure by communicating with OpenStack Nova and Neutron services.

6 Evaluation Results

The type of evaluation for the proposed Automated Cyber Defense System can be categorized into: Threat Detection System's performance evaluation, security evaluation of CTI sharing and effectiveness of course of actions.

6.1 Threat Detection System's Performance Evaluation

We have used three evaluation metrics to measure Threat Detection System's performance.

$$Accuracy = \frac{Correct\,Predictions}{All\,Predictions}$$

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives}$$

$$Recall = \frac{True\,Positives}{True\,Positives + False\,Negatives}$$

We have evaluated two types of input metrics data. One is Mixed data(contains one hour of metrics data during legitimate traffic and one hour of metrics data during Slowloris attack) and another one is Malicious data (contains only one hour of metrics data during Slowloris attack). Figure 9 shows analysis results. Accuracy and Recall values drop significantly for Malicious data as compared with Mixed data. Threat Detection system's performance drops with only one kind (malicious) of data which is rare in real web servers during an attack. We can overcome this challenge in our Threat Detection System by taking samples at different intervals and then perform the detection.

6.2 Security Evaluation of CTI Sharing

During automated CTI Sharing, data is transferred from organization to another. The two major security concerns are data security (Confidentiality and Integrity) in the transport and proper authorization to access the data. We have introduced secure communication protocols and ReBAC to address those issues. However, proper security evaluation can be done by performing security audits (protocol security and authorization log audits) in real systems and is out of scope of this paper.

6.3 Effectiveness of Course of Actions

In our experiment, we have demonstrated scaling as a possible course of action against Slowloris (DDOS) attack. In real systems, there will be other layered security controls such as Intrusion Detection Systems, firewalls, antivirus, etc. which will work together with scaling to optimize costs and provide effective availability. Our system can be tested with other security controls activated to measure the availability which is out of scope of this paper.

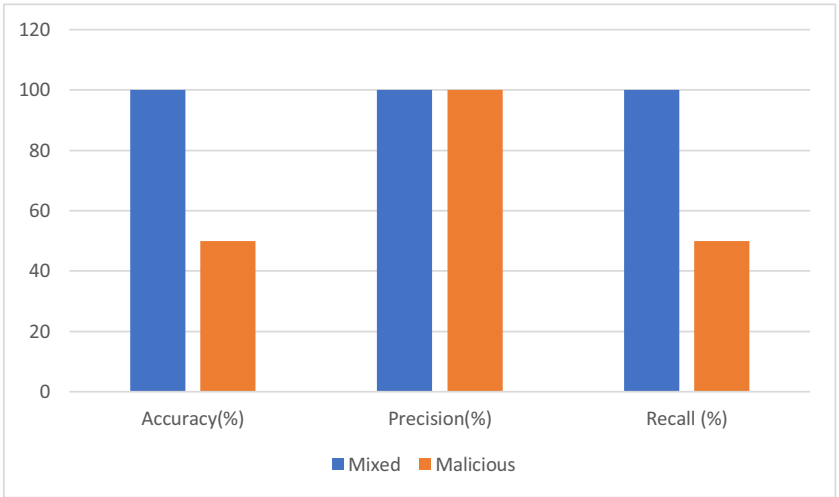

Fig. 9 Threat Detection System's performance evaluation

7 Future Work

We have presented an Automated Cyber Defense System which involves threat detection, STIX generation, threat intelligence sharing and taking course of actions. We now propose some future works:

1. **Cyber Threat Intelligence Knowledge Graph:** The knowledge graph is a knowledge base used by Google and it's services to enhance the search engine's results with information gathered from a variety of sources. A machine learning based approach could be applied to develop a similar type of knowledge graph for structured CTI. The graph could provide useful relevant information such as attacks of similar nature, previous course of actions taken for similar attacks, etc.
2. **OpenStack Native Services:** We have developed different components of our Automated Cyber Defense system such as Threat Detection system, STIX Generation system, Course of Actions system, etc. as general web services. There is potential to develop these services as OpenStack native services like OpenStack Nova or Neutron. That would provide a cloud tenant with automated cyber threat detecion, STIX generation, threat intelligence sharing and course of actions capabilities.

8 Conclusion

In this paper, we have provided an Automated Cyber Defense System integrable with STIX and TAXII standards. We have demonstrated a machine learning based approach for threat detection. We have also developed an automated STIX generation approach independent of cyber threat detection platforms. We have presented the necessities to share CTI in an organizational scenario and provided a ReBAC framework to share threat intelligence in a controlled and secure manner. Our adoption of Cheng et al.'s ReBAC model demonstrates the applicability of this type of access control outside social networking contexts. We have used TAXII protocol to share CTI in a request response model. We have developed an approach to configure and automate course of actions to mitigate or prevent cyber threats. We have also showed an instantiation course of actions in terms of OpenStack auto scaling. We plan to extend the proposed Automated Cyber Defense System as an OpenStack native service available for OpenStack tenants.

Acknowledgements This work is partially supported by DoD ARO Grant W911NF-15-1-0518, NSF CREST Grant HRD-1736209 and NSF CAREER Grant CNS-1553696.

References

Abdelsalam, M., Krishnan, R., Huang, Y., Sandhu, R. (2018). Malware detection in cloud infrastructures using convolutional neural networks. In *2018 IEEE 11th International conference on cloud computing (CLOUD)* (pp. 162–169): IEEE.

Barnum, S. (2012). Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation*, *11*, 1–22.

Bohnert, T., & Monteiro, E. (2005). A comment on simulating lrd traffic with Pareto on/off sources. In *Proceedings of the 2005 ACM conference on Emerging network experiment and technology* (pp. 228–229).

Burden, K., & Palmer, C. (2003). Internet crime: Cyber crime—a new breed of criminal? *Computer Law & Security Review*, *19*(3), 222–227.

Chaabane, A., Acs, G., Kaafar, M.A., et al. (2012). You are what you like! information leakage through users' interests. In *Proceedings of the 19th annual network & distributed system security symposium (NDSS). Citeseer*.

Cheng, Y., Park, J., Sandhu, R. (2012). A user-to-user relationship-based access control model for online social networks. In *IFIP Annual conference on data and applications security and privacy* (pp. 8–24): Springer.

Connolly, J., Davidson, M., Schmidt, C. (2014). The trusted automated exchange of indicator information (taxii). The MITRE Corporation, 1–20.

Crampton, J., & Sellwood, J. (2014). Path conditions and principal matching: a new approach to access control. In *Proceedings of the 19th ACM symposium on access control models and technologies* (pp. 187–198): ACM.

Debar, H., Becker, M., Siboni, D. (1992). A neural network component for an intrusion detection system. In *null* (p. 240): IEEE.

Elliott, A., & Knight, S. (2010). Role explosion: acknowledging the problem. In *Software Engineering research and practice* (pp. 349–355).

Fong, P.W. (2011). Relationship-based access control: protection model and policy language. In *Proceedings of the first ACM conference on Data and application security and privacy* (pp. 191–202): ACM.

Garton, L., Haythornthwaite, C., Wellman, B. (1997). Studying online social networks. *Journal of Computer-Mediated Communication*, *3*(1), JCMC313.

Gates, C. (2007). Access control requirements for web 2.0 security and privacy. *IEEE Web, 2*(0).

Haass, J.C., Ahn, G.J., Grimmelmann, F. (2015). Actra: a case study for threat information sharing. In *Proceedings of the 2nd ACM workshop on information sharing and collaborative security* (pp. 23–26): ACM.

Haque, M.F., & Krishnan, R. (2019). Toward relationship based access control for secure sharing of structured cyber threat intelligence. In *International conference on secure knowledge management in artificial intelligence era* (pp. 21–37): Springer.

Hartigan, J.A., & Wong, M.A. (1979). Algorithm as 136: a k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108.

Hu, V.C., Kuhn, D.R., Ferraiolo, D.F., Voas, J. (2015). Attribute-based access control. *Computer*, *48*(2), 85–88.

Iannacone, M.D., Bohn, S., Nakamura, G., Gerth, J., Huffer, K.M., Bridges, R.A., Ferragut, E.M., Goodall, J.R. (2015). Developing an ontology for cyber security knowledge graphs. *CISR*, *15*, 12.

Jagatic, T.N., Johnson, N.A., Jakobsson, M., Menczer, F. (2007). Social phishing. *Communications of the ACM*, *50*(10), 94–100.

Johnson, C., Badger, M., Waltermire, D., Snyder, J., Skorupka, C. (2016). Guide to cyber threat information sharing. Tech. rep., National Institute of Standards and Technology.

Lane, J., Stodden, V., Bender, S., Nissenbaum, H. (2014). *Privacy, big data, and the public good: frameworks for engagement*. Cambridge University Press.

Li, H., & Liu, D. (2010). Research on intelligent intrusion prevention system based on snort. In *2010 International conference on computer, mechatronics, control and electronic engineering*, (Vol. 1 pp. 251–253): IEEE.

Liao, H.J., Lin, C.H.R., Lin, Y.C., Tung, K.Y. (2013). Intrusion detection system: a comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16–24.

Mansfield-Devine, S. (2016). Ransomware: taking businesses hostage. *Network Security*, *2016*(10), 8–17.

Meadows, C.A. (1996). Analyzing the Needham-Schroeder public key protocol: a comparison of two approaches. In *European symposium on research in computer security* (pp. 351–364): Springer.

Pandove, K., Jindal, A., Kumar, R. (2010). Email spoofing. *International Journal of Computer Applications*, *5*(1), 27–30.

Sandhu, R.S., & Samarati, P. (1994). Access control: principle and practice. *IEEE Communications Magazine*, *32*(9), 40–48.

Syed, Z., Padia, A., Finin, T., Mathews, L., Joshi, A. (2016). Uco: a unified cybersecurity ontology. In *Workshops at the thirtieth AAAI conference on artificial intelligence*.

Thornburgh, T. (2004). Social engineering: the dark art. In *Proceedings of the 1st annual conference on information security curriculum development* (pp. 133–135): ACM.

Md. Farhan Haque received M.S. degree in Computer Engineering from The University of Texas at San Antonio (UTSA) in 2015. He is currently working on doctoral degree at the Electrical and Computer Engineering Department at UTSA. His research interests include automated and secure threat intelligence sharing and sharing infrastructure development.

Ram Krishnan is an Associate Professor of Electrical and Computer Engineering at the University of Texas at San Antonio, where he holds Microsoft President's Endowed Professorship. His research focuses on (a) applying machine learning to strengthen cybersecurity of complex systems and (b) developing novel techniques to address security/privacy concerns in machine learning. He actively works on topics such as using deep learning techniques for runtime malware detection in cloud systems and automating identity and access control administration, security and privacy enhanced machine learning and defending against adversarial attacks in deep neural networks. He is a recipient of NSF CAREER award (2016) and the University of Texas System Regents' Outstanding Teaching Award (2015). He received his PhD from George Mason University in 2010.