CyTIME: *Cy*ber *T*hreat *I*ntelligence *M*anag*E*ment framework for automatically generating security rules

Eunsoo Kim
Sungkyunkwan University
Suwon, Republic of Korea
eskim86@skku.edu

Kuyju Kim
Sungkyunkwan University
Suwon, Republic of Korea
kuyjukim@skku.edu

Dongsoon Shin
Sungkyunkwan University
Suwon, Republic of Korea
ds.shin@skku.edu

Beomjin Jin
Sungkyunkwan University
Suwon, Republic of Korea
jinbumjin@skku.edu

Hyoungshick Kim
Sungkyunkwan University
Suwon, Republic of Korea
hyoung@skku.edu

ABSTRACT

It is becoming increasingly necessary for organizations to build a Cyber Threat Intelligence (CTI) platform to fight against sophisticated attacks. To reduce the risk of cyber attacks, security administrators and/or analysts can use a CTI platform to aggregate relevant threat information about adversaries, targets and vulnerabilities, analyze it and share key observations from the analysis with collaborators. In this paper, we introduce CyTIME (*Cy*ber *T*hreat *I*ntelligence *M*anag*E*ment framework) which is a framework for managing CTI data. CyTIME can periodically collect CTI data from external CTI data repositories via standard interfaces such as Trusted Automated Exchange of Indicator Information (TAXII). In addition, CyTIME is designed to automatically generate security rules without human intervention to mitigate discovered new cybersecurity threats in real time. To show the feasibility of CyTIME, we performed experiments to measure the time to complete the task of generating the security rule corresponding to a given CTI data. We used 1,000 different CTI files related to network attacks. Our experiment results demonstrate that CyTIME automatically generates security rules and store them into the internal database within 12.941 seconds on average (max = 13.952, standard deviation = 0.580).

CCS CONCEPTS

• **Security and privacy → Network security**; *Intrusion/anomaly detection and malware mitigation*; *Intrusion detection systems*;

KEYWORDS

cyber threat intelligence; intrusion detection systems; security rules

CFI 2018, June 20 22, Seoul, Korea

ACM ISBN 978-1-4503-6466-9/18/06…$15.00
https://doi.org/10.1145/3226052.3226056

1 INTRODUCTION

As increasing number of cybersecurity threats and attacks continuously appear and the environment evolves over time, we need to develop more flexible and efficient security mechanisms that can respond to threats and update security rules to mitigate them in a timely manner. To develop such security mechanisms, it is generally required to gather Cyber Threat Intelligence (CTI) data, and share them with other related entities (e.g., organizations and network/host resources) [5]. In practice, however, CTI data are generally managed in an ad-hoc manner and often manually configured by a few experienced security analysts. Our challenging issues are summarized as follows:

(1) CTI data formats are not standardized and varied among data repositories. We need to incorporate heterogeneous CIT data sources from open IOC (Indicator of Compromise), STIX (Structured Threat Information Expression), ISAO (Information Sharing and Analysis Organization) and Comodo Threat Research Labs (CTRL).
(2) Manually generating security rules is a tedious and time-consuming process. We need to automate this process according to the type of security solution adopted (e.g., Suricata, Snort, and YARA).

In this paper, we present CyTIME (*Cy*ber *T*hreat *I*ntelligence *M*anag*E*ment framework) which is a framework for managing CTI data from heterogeneous and massive streams of data continuously produced in the context of an information security framework. CyTIME can periodically aggregate CTI data from external CTI sources (e.g., open CTI servers and cloud-based malware scanners such as VirusTotal [1]) via the Trusted Automated Exchange of Indicator Information (TAXII) protocol. Moreover, CyTIME is designed to generate security rules without human intervention in an automatic manner and prevent the related attacks as early as possible. We implement a database system using Structured Threat Information eXpression (STIX) because STIX has now been widely adopted as a de facto standard for automated cyber threat information exchange.

To evaluate the performance of CyTIME, we performed experiments to measure the time to complete the task of generating the security rule corresponding to a given CTI data. We used 1,000 different CTI files related to network attacks. The experiment results show that CyTIME automatically generates security rules for each

[1] VirusTotal: https://www.virustotal.com/

security solution (e.g., IDS/IPS and malware scanners) and store them into the internal database within 12.941 seconds on average (max = 13.952, standard deviation = 0.580).

2 BACKGROUND

In this section, we briefly summarize the description of some components used in CyTIME.

STIX is a language for having a standardized communication for the representation of CTI [1]. It is developed for the specification, capture, characterization and communication of standardized threat information. STIX also provides a common mechanism for addressing structured CTI across and among the full range of analyzing cyber threats, specifying indicator patterns, managing cyber threat responses and many more to improve consistency, efficiency, and overall situational awareness.

TAXII is an application layer protocol for the communication of CTI in a simple and scalable manner [2]. TAXII is a type of protocol used to exchange CTI over HTTPS so that it can enable organizations to share CTI by defining APIs that align with common sharing models. It also defines data formats for securely exchanging CTI for detection, prevention, and mitigation of cyber threats in real time. The advantages of using TAXII is situational awareness about emerging threats, and it enables organizations to easily share the information while leveraging existing relationships and systems.

Suricata is one of the good examples of open-source IDS/IPS available on all platforms [3]. It identifies an attack by inspecting network data against predefined standard signature rule-set available from emerging threats. Suricata provides name, severity, and type of the attack. To keep the system up-to-date, a logging agent contacts the administration server to check the availability of new signatures in the internal database. If a new signature is found, logging agent automatically updates Suricata rule-set.

YARA is a pure indicator layer technology that describes regular expression patterns and behavior. YARA is an engine and language for scanning files and memory blocks. When a rule matches a pattern, YARA presumes to classify the subject according to the rule's behavior. YARA can match various string formats like ASCII, UTF, and other encodings; YARA can also parse on PERL regular expressions and has Python and Ruby bindings.

Cuckoo sandbox is an open source automated malware detection system. It is capable of analyzing different malicious files as well as malicious websites under various platforms, such as Windows, Linux, Mac OS and Android virtualized environment. It is also able to trace API calls and general behavior of the file and distill this into high level information and signatures comprehensible by anyone. Cuckoo sandbox can easily be integrated into existing framework and backed in the way a user (systems administrator) wants.

MISP (Malware Information Sharing Platform) [6] is a trusted platform that allows the collection and sharing of important indicators of compromise (IOC) of targeted attacks, but also threat information like vulnerabilities or financial indicators used in fruad cases. The aim of MISP is to help in setting up preventive actions and counter-measures used against targeted attacks, and to enable detection via collaborative knowledge sharing about existing malware and other attacks.

3 PROPOSED FRAMEWORK

CyTIME is designed to integrate heterogeneous CTI data sources under a global JSON format and automatically generate network security rules from the incorporated CTI data in a seamless manner. CyTIME operates in three major stages: (1) Data acquisition, (2) Data conversion, and (3) Security rule generation. Figure 1 shows the high level architecture of CyTIME.

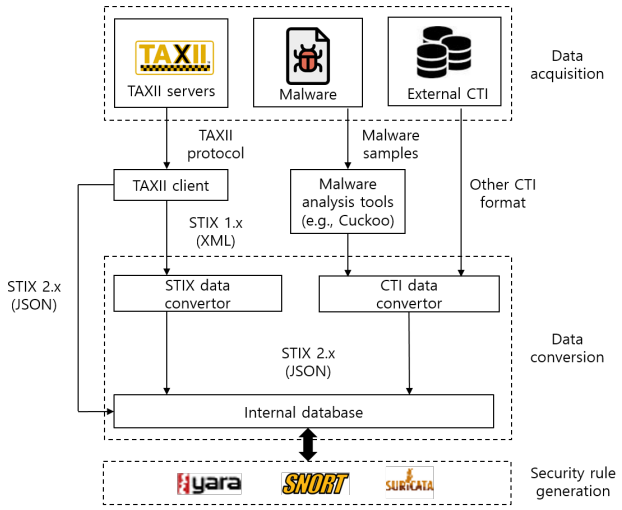

Figure 1: Overview of CyTIME.

In the data acquisition stage, we obtain CTI data from various sources. A possible way is to use TAXII protocol. Open CTI data on TAXII servers can be delivered through the use of a TAXII client. We use malware analysis tools to collect the information about malware samples. We can also support many other CTI data formats such as Malware Attribute Enumeration and Characterization (MAEC) and Malware Information Sharing Platform (MISP) data files.

In the data conversion stage, we convert the obtained CTI data into a single JSON format. The CTI files obtained using TAXII server are represented in either STIX 1.x (XML) or STIX 2.x (JSON). Because CyTIME stores all CTI files under the STIX 2.0 format, STIX 1.x files are first converted into STIX 2.0 and then stored in the CyTIME database. Malware samples are first analyzed by malware analysis tools (e.g., VirusTotal or Cuckoo sandbox) and the analysis reports can be converted into the STIX 2.0 format. Other CTI data formats can be translated into the STIX 2.0 format because they also have a structured format under an XML schema.

The rule generation stage is responsible for generating security rules from the internal CTI database records. For CTI records in the CyTIME database, security rules can be automatically generated for a target security application (e.g., Suricata, Snort, and YARA). In general, some feature strings for malicious URLs or blacklisted IP addresses can be extracted to generate security rules. The extracted feature strings can properly be mapped into the fields in the security rule for the target security application.

4 IMPLEMENTATION

In this section, we describe a prototype implementation of CyTIME in more detail. We used Ubuntu 16.04 LTS x64 to run CyTIME. In our implementation, we deployed a TAXII client to periodically receive fresh CTI data from open source TAXII servers that are available to public TAXII clients. We also installed the Cuckoo host component, and then configured the Cuckoo host component to analyze given malware samples in an automatic way. Here, the Cuckoo sandbox was used to generate static and dynamic analysis reports by executing the program with a malicious sample. The analysis results were finally incorporated into the CyTIME database in order to generate a proper security rule for the target security application if it is needed.

To support various CTI data formats, we implemented a module for translating source XML files into STIX 2.0 format. We modified an existing open source parser named stix2elevator. Since each STIX XML file has its own header and namespace that differs from a conventional XML, we first need to parse it and map the parsed fields to database tables and columns under the STIX 2.0 format.

We also developed an automatic policy generation module which extracts the necessary fields to generate a security rule for a target security application such as Snort and Suricata using the CTI data stored in the CyTIME internal database. In our prototype implementation, we only focused on transforming the CTI data into network security rules so that the rule generation is relatively simple and straightforward compared with other security rules and their effectiveness can also be verified by security analysts.

As described in Section 3, we considered three different CTI data sources: open TAXII servers providing STIX files, malware samples, and other external CTI databases. To obtain CTI data from malware samples, we incorporated malware analysis tools using VirusTotal and Cuckoo sandbox to scan malware samples and store the analysis results into the CyTIME internal database. We implemented a conversion module to translate various CTI data formats in STIX XML, IOC XML, MAEC and MISP into a single united format under STIX 2.0. Finally, the translated JSON data fields were stored in the internal CyTIME database which was implemented by the MySQL 14.14 database. The security rule generator automatically identify the important feature strings from the database columns and generate a proper security for a target security application. Our prototype implementation can support Snort and Suricata only. The detailed description of how security rules are generated for those applications will be presented in the next section.

5 SECURITY RULE GENERATION

To generate security rules, we aim to properly map key strings in each CTI file into the field values in a security rule for IDS/IPS and malware scanners.

We first extract the feature strings such as malicious IP addresses, domain names and malware signature from CTI files gathered from open TAXII servers, external CTI databases, and malware analysis tools. Next, we generate a network security rule according to a pre-defined rule format. In our prototype implementation, we used Suricata. Figure 2 shows how the extracted feature strings are mapped. In this example, IP addresses in a CTI file can be

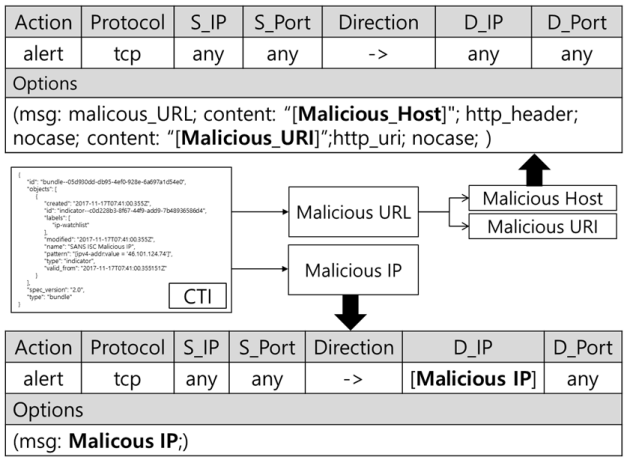

Figure 2: Rule generation example.

Table 1: Feature strings for Snort and Suricata rules.

CTI (JSON)	Snort/Suricata	Required
Name, Description	Message content	Mandatory
Pattern: URL value	Source IP or URL	Mandatory
First created	Signature ID	Mandatory
Source/Destination ref	Reference	Optional
Labels	Classtype	Optional

Table 2: Example of a generated security rule.

Action	Protocol	S_IP	S_Port	Direction	D_IP	D_Port
drop	ip	213.202.230.14	any	->	$HOME_NET	any
Options						
(msg: "SANS ISC Malicious IP;" classtype: ip-watchlist; sid: 20180120-1; rev:1;)						

extracted for Snort/Suricata rules. In Table 1, we describe some feature strings used in this example. The `Labels` field represents a `Classtype` value to express which type of class the indicator belongs to. The `Labels` field can be used to specify whether it is a blacklisted IP address or malicious application. The `Name` and `Description` fields can be used for optional strings to describe the name of the malicious activity and how it affects the system. The `First created` field represents the moment of the CTI data creation. This date value can be used with a sequential number as the signature ID – a sequential number is required to differentiate those that have the same `First created` value. The `Source` and `Destination` fields are optional, and can be included to represent the location information such as malicious `URL` or `IP` address. In Snort and Suricata, `Classtype` and `Reference` are parts of an optional message field which can be added to describe for which malicious activity should the security rule be applied, and where does the information about such malware come from. These attributes are not marked as mandatory, and therefore can be left out during the rule generation stage. The generated security rule is shown in Table 2.

Algorithm 1: Security rule generation algorithm.

Input : CTI files *Files*
Output: IDS rules R

$R = \{\emptyset\}$
for *each* f_i *in Files* **do**
  if f_i *contains blacklisted IP addresses* **then**
    Extract IP addresses, creation time, indicator, reference from f_i
    Generate the corresponding security rule r_i with IP addresses, creation time, indicator and reference
    Add r_i to R
  else if f_i *contains blacklisted domain names* **then**
    Extract domain names, creation time, indicator, reference from f_i
    Generate the corresponding security rule r_i with domain names, creation time, indicator and reference
    Add r_i to R
  else if f_i *contains malware* **then**
    Extract malware name, malware type, malware signature, creation time, indicator, reference from f_i
    Generate the corresponding security rule r_i with malware name, malware type, malware signature, creation time, indicator and reference
    Add r_i to R
  ...
end
Return R

Algorithm 1 describes how the CTI files are converted into the corresponding security rules. For each file in the collected CTI files, we first identify the file type (e.g., by checking `type` in STIX) and then extract fields which are necessary for the file type to generate the corresponding security rule.

```
{
    "type": "malware",
    "id": "malware--0c7b5b88-8ff7",
    "created": "2016-05-12T08:17:27.000Z",
    "modified": "2016-05-12T08:17:27.000Z",
    "name": "Cryptolocker",
    "description": "...",
    "labels": ["ransomware"]
}
```

Figure 3: Example of malware object.

For example, if "`type`" in a STIX malware object is "`malware`" (see Figure 3), we can extract its "`type`", "`id`", "`created`", "`labels`" and "`name`" field values and generate a security rule with those values.

6 EVALUATION

To show the feasibility of our rule generation method, we generated security rules from 1,000 CTI files about network security attacks. For performance evaluation, we measured the time taken from the CTI data acquisition step until the rule generation step. The time for the CTI data to be converted, validate and stored in the internal database is also measured. Figure 4 shows how the processing time for storing CTI data and rule generation changed with the number of rules. For improved visualization, the processing time is represented in log scale on the y-axis. As shown in this figure, the total processing time was overwhelmingly dominated by data import operations although there was a slight increase in the processing time for automatic rule generation with the number of rules.

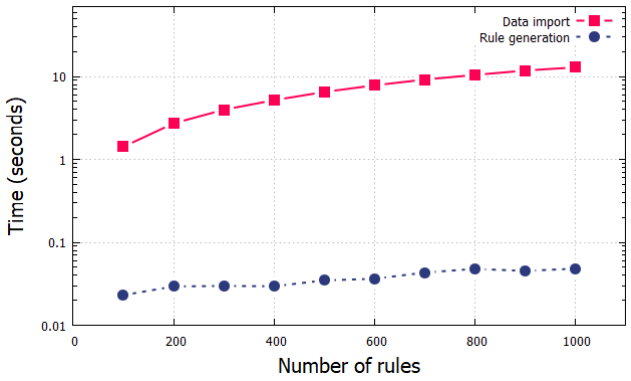

Figure 4: Processing time for storing CTI data and rule generation.

In summary, when the number of security rules is 1,000, CyTIME automatically generated security rules and stored them into the internal database within 12.941 seconds on average (max = 13.952, standard deviation = 0.580). Moreover, the time taken to generate a security rule is only within about 0.0481 seconds on average (max = 0.0556, standard deviation = 0.004).

7 RELATED WORK

Because cyber threat intelligence is becoming increasingly important in industry, several CTI management software products were recently introduced.

Industrial Control System Information Sharing and Analysis Center (ICS-ISAC) particularly developed Soltra Edge (https://www.soltra.com/). Soltra Edge can be used to share CTI information between multiple organizations in real time. Soltra Edge uses TAXII protocol to periodically gather CTI information and also provides a web interface to manually manage CTI information for administrators. Similarly, Threatconnect (https://www.threatconnect.com/) has been demonstrated for government agencies and large enterprises to aggregate all available threat data, analyze them, automate orchestration and suggest proper tactical, operational and strategies against cyber threats. More recently, Eclectic IQ (https://www.eclecticiq.com/) was introduced as a CTI sharing and analysis platform. Eclectic IQ uses STIX format and enables sharing of threat intelligence data through the TAXII protocol. Unlike our

proposal, however, their products only supports a few data formats (e.g., STIX 2.0) and do not provide the functionality to generate security rules.

Several researchers have also proposed CTI management platforms. Qamar et al. [4] presented a threat analytic framework based on Web Ontology Language (OWL) for formal specification, semantic reasoning, and contextual analysis, allowing the derivation of network associated threats from a large volume of threat sources. Their framework provides an automated mechanism to investigate cyber threats related to network systems, categorize those threats, analyze the likelihood of those threats and predict their impacts on assets. Wanger at al. [6] proposed a CTI sharing platform called MISP that gathers the information about malware samples and shares it with other organizations. MISP not only supports popular threat information formats such as STIX and MAEC but also creates its own data format. MISP also includes a flexible import tool to manage STIX, OpenIOC, Snort and etc. In this paper, we extend MISP by developing CyTIME to support other popularly used CTI data formats (e.g., STIX 1.1 and VirusTotal results) and automatically converts them into STIX 2.0 data format that can efficiently be stored in a database server.

8 CONCLUSION AND FUTURE WORK

In this paper, we introduce a CTI management framework (named CyTIME) to manage CTI data and automatically generate security rules for IDS/IPS and malware scanners. CyTIME is capable of importing various CTI formats including STIX. We demonstrate the feasibility of CyTIME with 1,000 real CTI files.

As part of our future work, we plan to extend CyTIME to develop a more general security rule mapping algorithm. It would also be interesting to evaluate the performance of CyTIME on real malicious network activities.

ACKNOWLEDGMENTS

This research was supported in part by the MIST (2015-0-00914) and the ITRC program (IITP-2017-2015-0-00403).

REFERENCES

[1] Barnum, S. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *The MITRE Corporation* (2012), 1–22.

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

[3] Park, W., and Ahn, S. Performance comparison and detection analysis in snort and suricata environment. *Wireless Personal Communications 94* (2017), 241–252.

[4] Qamar, S., Anwar, Z., Rahman, M. A., Al-Shaer, E., and Chu, B.-T. Data-driven analytics for cyber-threat intelligence and information sharing. *Computers & Security 67* (2017), 35–58.

[5] Skopik, F., Settanni, G., and Fiedler, R. A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing. *Computers & Security 60* (2016), 154–176.

[6] Wagner, C., Dulaunoy, A., Wagener, G., and Iklody, A. Misp: The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 1st ACM Workshop on Information Sharing and Collaborative Security* (2016).