Journal of Information Security, 2019, 10, 263-277
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Research on University's Cyber Threat Intelligence Sharing Platform Based on New Types of STIX and TAXII Standards

Gang Wang[1], Yuanzhi Huo[2], Zhao Ma[3]

[1]Information Construction and Management Center, Inner Mongolia University of Technology, Hohhot, China
[2]Department of Information Engineering, Inner Mongolia University of Technology, Hohhot, China
[3]Faculty of Engineering, Environment & Computing, Coventry University, Coventry, UK
Email: zmshyz1995@126.com

How to cite this paper: Wang, G., Huo, Y.Z. and Ma, Z. (2019) Research on University's Cyber Threat Intelligence Sharing Platform Based on New Types of STIX and TAXII Standards. *Journal of Information Security*, **10**, 263-277.
https://doi.org/10.4236/jis.2019.104015

Received: July 25, 2019
Accepted: October 26, 2019
Published: October 29, 2019

Abstract

With the systematization of cyber threats, the variety of intrusion tools and intrusion methods has greatly reduced the cost of attackers' threats to network security. Due to a large number of colleges and universities, teachers and students are highly educated and the Internet access rate is nearly 100%. The social status makes the university network become the main target of threat. The traditional defense method cannot cope with the current complex network attacks. In order to solve this problem, the threat intelligence sharing platform based on various threat intelligence sharing standards is established, which STIX and TAXII It is a widely used sharing standard in various sharing platforms. This paper analyzes the existing standards of STIX and TAXII, improves the STIX and TAXII standards based on the analysis results, and proposes a new type of STIX and TAXII based on the improved results. The standard design scheme of threat intelligence sharing platform suitable for college network environment features. The experimental results show that the threat intelligence sharing platform designed in this paper can be effectively applied to the network environment of colleges and universities.

Keywords

STIX, TAXII, Threat Intelligence

1. Introduction

As far as 2019, there were 2879 colleges and universities in China with 37 million current students [1]. Due to the high level of education of personnel and the network usage rate reaching nearly 100%, colleges and universities become ma-

jor targets of cyber security threats. In the same period, threat intelligence, threat information and other related terms became popular in the field of cyber security. With the rapid development of various attack tools and attack technologies, traditional security defenses cannot cope with new types of attacks, and a single point-to-point defense has lagged far behind current attacks. In response to this situation, the concept of threat intelligence began to be raised. As early as 2009, the United States proposed the establishment of a cyber threat intelligence sharing mechanism in the cyberspace policy assessment report—ensuring the security and resilience of information and communication infrastructure. In February 2013, Obama issued Executive Order 13,636, "Enhancing Critical Infrastructure Network Security," clearly stating that cyber threat intelligence sharing must be achieved between government and business. In 2012, MITRE proposed the STIX (Structured Threat Information eXpression) v1.0 framework as a network security threat intelligence expression. In 2013, TAXII (Trusted Automated eXchange of Indicator Information) was proposed as a network security threat intelligence exchange mechanism. After the STIX and TAXII threat intelligence sharing mechanism came out, it was obtained by the US government and business community. The universal application of many units has become one of the standards that are commonly followed in the global threat intelligence field [2] [3]. With the introduction of threat intelligence sharing standards, various threat intelligence sharing platforms have also appeared. The main business of threat intelligence vendors such as isight Parnters, FireEye, and CrowdStrike [4] is to provide users with high-quality threat intelligence information. Although the emergence of these threat intelligence sharing platforms provides a good guarantee for defending against cyber threats, it also has the following shortcomings:

1) No specific platforms for universities

Threat intelligence sharing platforms are generally established by commercial companies, and the types of customers are too complex, such as banks, universities, governments, and other institutions. There is no threat intelligence sharing platform for a certain group of people, such as colleges and universities may be more concerned about whether other universities are subject to a certain type of cyber threat.

2) No detailed solution is provided

The existing threat intelligence sharing platform only provides threat details of threat intelligence, such as threat type, discovery time, intruder IP information, etc. If a site is attacked by a generic vulnerability, then other sites gain access to the details of the attack. A detailed solution to this type of attack can be obtained, such as upgrading the system version, installing patches, etc. to protect against such threats in a timely manner.

3) The authenticity of information

The existing threat intelligence sharing platform has a very important source of threat intelligence for the details of providing cyber threats through "Ethical hacker", such as threat book.cn and other platforms in China. The threat infor-

mation provided by "Ethical hacker" could possibly be fake, and it takes a lot of energy and time to judge the authenticity of the information and the inefficiency.

Based on existing threat intelligence sharing platforms, this paper analyzes and improves the existing threat intelligence sharing standards STIX and TAXII. At the same time, it proposes a design scheme of college threat intelligence sharing platform using improved STIX and TAXII.

The paper is structured as follows:

1) Chapter 1 introduces the development background of threat intelligence sharing.

2) Chapter 2 chapter analyzes relevant research, finds out the shortcomings and proposes its own solution.

3) Chapter 3 introduces the knowledge of STIX and TAXII.

4) Chapter 4 introduces the improvement scheme of STIX and TAXII in this paper.

5) Chapter 5 introduces the platform design.

6) Chapter 6 conducts related experiments and results analysis.

7) Chapter 7 summarizes this paper.

2. Related Work

In [5], Elchin Asgarli and Eric Burger analyzed the exchange format of threat intelligence standards and compared them with RDF/OWL exchange formats. [6] designed a new sharing model for TAXII standards. In [7], the field expansion of STIX was carried out, and STIX was used for information interaction between penetration testers. [8] proposed and designed a network threat intelligence management framework (CyTIME) by using the data defined in the TAXII standard. The interface can collect cyber threat intelligence on a regular basis and can be collected periodically without human intervention. [9] analyzes ACTRA's cyber threat intelligence application examples and discusses the real technical problems in the application. In [10], F Fransen *et al*. focused on the rise of threat intelligence sharing platforms, focusing on what types of cybersecurity threat information to share to better defend against attacks.

In the above-mentioned literature [5] [6] [7], although the threat intelligence sharing standard was analyzed or improved [8] [9] [10], the standard was applied to the construction of the threat intelligence platform, but the two did not have corresponding combination, nor A threat intelligence sharing platform specially designed for the characteristics of college networks, this paper studies and proposes corresponding design schemes.

The main contributions of this paper are as follows:

1) Proposed a design scheme of threat intelligence sharing platform suitable for university network characteristics.

2) According to the characteristics of the university network, improve the STIX and TAXII standards, and apply the improved standards to the platform design.

3. Related Knowledge

3.1. STIX

Structured Threat Information Expression (STIX™) is a structured language used to describe cyber threat information, using the same standards for threat intelligence sharing, storage and analysis [11]. It was developed by OASIS (Organization for the Advancement of Structured Information Standards). STIX provides a structured, common framework for expressing cyber threat intelligence, improving intelligence accuracy, interoperability, and automated processing efficiency, effectively supporting the automation of cyber threat management processes and applications, and enabling more advanced cyber security analysis. Provide support for models, frameworks, and specifications such as analyzing cyber threats, characterizing patterns of cyber threats, managing cyber threat response activities, and sharing cyber threat intelligence. The logical relationship of the main components of STIX is shown in **Figure 1**.

STIX defines a total of 12 STIX domain objects, each of which is independent of itself and associated with other objects. As shown in **Figure 3**, the attack activity identifies an attack initiated by the attacker and launches a malicious attack on the vulnerability software or website through the network. All objects are identified in STIX in a uniform format to normalize threat intelligence. Organizations that use STIX threat intelligence expressions can share threat information for early warning and defense.

3.2. TAXII

TAXII's full name is trusted automatic intelligent information exchange. It is a protocol that works at the application layer to deliver cyber threat information in a simple and scalable way [12]. In order to ensure the security of the transmission, TAXII uses HTTPS as a protocol for exchanging network threat intelligence. TAXII can exchange threat intelligence between organizations or companies that use it by defining a specific set of APIs. The logical structure diagram of the TAXII shared transmission mechanism is shown in **Figure 2**.

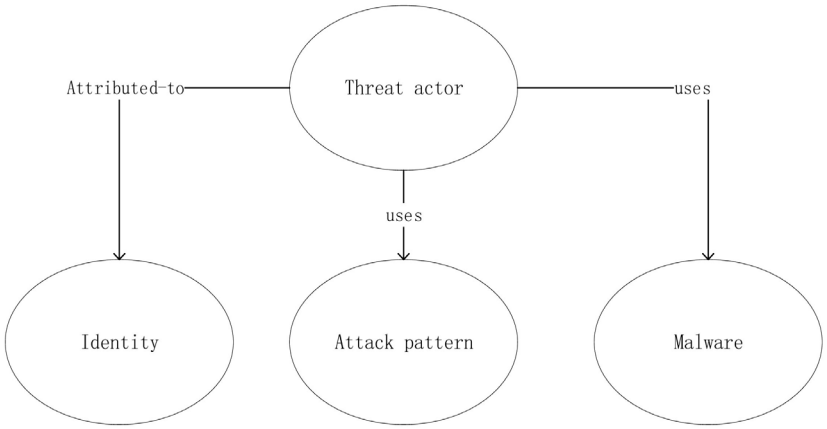

Figure 1. Logical relationship between components.

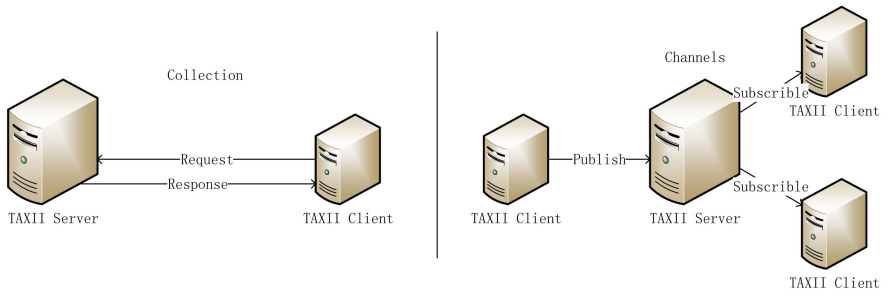

Figure 2. TAXII sharing mechanism.

Centered around the TAXII server, each TAXII's client acts as both a producer and a consumer. The organization normalizes the threat intelligence through STIX and transmits it to the TAXII server using the TAXII transport mechanism. All client subscribed to the TAXII server can obtain the latest threat information from the TAXII server. These clients, who act as consumers, can also become producers, releasing their own threat intelligence to the TAXII server for threat intelligence sharing. In the solution designed in this paper, TAXII is run on the cloud center.

In addition, TAXII and STIX are two completely independent mechanisms. TAXII can also transmit threat information in other formats, greatly increasing the flexibility of threat intelligence sharing. Organization and organization that use other specifications for threat intelligence sharing can use TAXII. Share threat intelligence in other formats.

4. STIX/TAXII Standard Analysis and Improvement

4.1. STIX Standard Analysis and Improvement

The complete network threat activity cycle is defined in the STIX standard. In a scenario of a target site under attack, the target site is compromised and a malicious URL link to the backdoor is inserted. The corresponding STIX format is as follows:

1) Indicator

The indicator uniquely identifies the overall information of the threat event, such as the ID of the threat information, creation time, modification time, label, description, etc., and summarizes the threat information in a general way. The indicator fields are as follows:

a) Id: Indicator number

b) Created: The creation time of the identification indicator

c) Modified: The final modification time of the identification indicator. This field is the same as the creation time when it was first created.

d) Name: The indicator points to the name of the threat event

e) Description: Describe the details of the threat event

f) Labels: Classification of threat events

g) valid_from: Identifies the expiration time of the indicator

2) Killing chain stage

The kill chain is a description model of the life cycle of a network attack. Malware in network security incidents will have different classifications and be at different stages of the kill chain depending on the function. The backdoor Trojans that occur in web intrusions are at the stage of establishing a foothold.

a) Kill_chain_name: Identifies the kill chain name

b) Phase_name: Identifies the stage name of the specific stage of the malware in the model

3) Malware

a) Id: Malware number

b) Created: About the creation time of malware threat information

c) Modified: Final modification time of threat information

d) Name: Malware name

e) Labels: Classification tags for malware

f) Description: Functional information of malware

g) Kill_chain_phases: Identifies the established foothold

The malware associated with the URL in this scenario is a backdoor that can be modeled using the STIX malware SDO. Like indicator objects, malware objects can be further categorized using the labels from the malware tag and can also be used to capture kill chain information about malware instances. The malware tries to build a backdoor and download the remote file. This is indicated by the malware object using kill_chain_phases, which contains the name of the kill chain used and the stage in the kill chain. For this scenario, the Mandiant attack lifecycle model is used as a termination chain and uses the "stage name" to establish a foothold. Other kill chain models can also be used for representation, such as Lockheed Martin's kill chain.

Relationship SROs are used for link indicators and malware objects, and URL indicators indicate backdoor malware objects. In this relationship, the indicator ID is source_ref and the malware ID is target_ref. Figure 3 shows the relationship between metrics and malware SDO and SRO.

In order to address the hazards posed by cyber attacks, the STIX standard has developed a course of action (COA) to describe the actions taken against cyber attacks, including automated responses (applying patches, reconfiguring firewalls), or higher Levels of measures, such as training on employee safety knowledge and changes to overall security policies. The attributes contained in the COA are shown in Table 1.

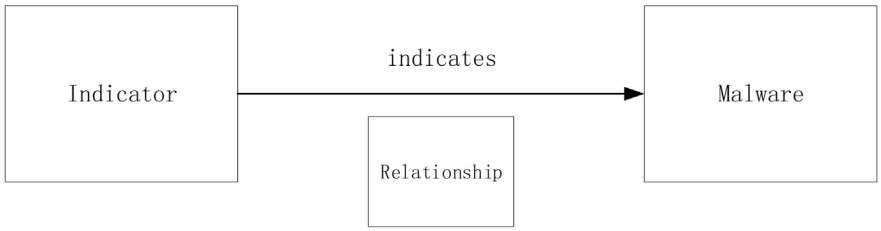

Figure 3. SDO and SRO Relationship.

As can be seen from Table 1, although the description scheme of the threat measures is specified in STIX, all the descriptions are written in the same field, and the level is not clear enough. It is not applicable to the threat intelligence sharing platform for college networks.

It is extended based on the original COA identification field. The extended field format is shown in Table 2.

As shown in Table 2, the relevant fields of the solution are added to the fields defined by the original COA, and the meanings are as follows.

1) Solution_url: The reference URL associated with the solution, such as the download address of the patch.

2) Solution_layer: The solution acts on the first layer of the TCP/IP five-layer reference protocol, such as the application layer.

3) Solution_start Time: The time when the solution starts running, such as the vulnerability patch starts at 17:30:00.

4) Solution_end Time: The time when the solution ends, such as the vulnerability patch installed at 17:30:10.

5) Solution_cost: The cost of the solution (time, money, etc.).

6) Solution_supplier: The solution provider, such as the solution provided by ABC University. If a university encounters the same type of cyber threat, the

Table 1. COA attribute table.

Field	Meaning	Sample Content
Type	Action process	"type": "course-of-action"
Name	The name of the action process	"name": "close port 445"
Description	Describe the details of the action process	"description": "This is a detailed description of how to close the port..."
Action	Reserved fields—capture structured and automated action plans	None

Table 2. COA extension field table.

Field	Meaning	Sample Content	Data type
Solution_url	Reference URL link related to solution content	"solution_url": "https://abc.go.com/"	string
Solution_layer	Solution applied to the TCP/IP five-layer reference protocol location	"solution_layer": "Application layer"	string
Solution_startTime	The start time of the solution	"solution_start Time": "2019-7-18 17:30:00"	string
Solution_endTime	The end time of the solution	"solution_end Time": "2019-7-18 17:30:10"	string
Solution_cost	The cost of the solution (time, money, etc.)	"solution_cost": "time-10s"	string
Solution_supplier	Solution provider	"solution_supplier": "ABC University"	string

solution cannot handle the threat according to the solution, and can provide solutions based on the information provided in this field. Universities get more detailed information.

By extending the content of COA, STIX has an information field that provides a more detailed solution for threat intelligence, which allows consumers of threat intelligence to defend against threats more quickly.

4.2. TAXII Standard Analysis and Improvement

The communication protocol used in TAXII is HTTPS. HTTPS is a "secure version of HTTP" with the SSL layer added to the HTTP protocol. The HTTP life cycle is defined by Request, which is a Request and a Response. In HTTP1.0, this HTTP request ends.

Improved in HTTP 1.1, the HTTP request header contains a keep-alive, which means in a HTTP connection, multiple Requests can be sent to receive multiple Responses. But the number of requests is equal to response, as specified in the HTTP protocol which means a request can only have one response. And this response is also passive and cannot be initiated. Due to the limitation of HTTP's own mechanism, the client needs to re-establish a connection with the server every time it wants to send data to the server, which causes the consumption of server resources. When the number of requests is too large, it may even cause the server to crash. There are a large number of colleges and universities, and each university also owns thousands of servers. In order to ensure the stability and real-time communication of such a large number of colleges and universities communicating with the threat intelligence sharing platform, this paper proposes to use a secure version of websocket (WSS) replaces the HTTPS protocol used in the TAXII standard.

Websocket is a protocol for full-duplex communication over a single TCP connection, a persistent protocol compared to HTTP [13]. Websocket makes it easier to exchange data between the active defense terminal and the server, allowing the server to actively send data to the active defense terminal. In the Websocket API, the client and server only need to complete a handshake, and a persistent connection can be created between the two, and bidirectional data transmission. The advantages of Websocket are as follows:

1) Less control overhead

When data is exchanged between the server and the active defense terminal after the connection is created, the header used for protocol control is relatively small. In the case of no extensions, the header size is only 2 to 10 bytes (depending on the packet length) for server-to-client content. For client-to-server content, the header needs to be extra 4 words. This overhead is significantly reduced by carrying a complete header each time relative to a HTTP request.

2) Stronger real-time performance

Because the protocol uses a full-duplex mechanism, the server can actively send data to the client at any time. Compared to HTTP requests, it is necessary to wait for the client to initiate a request to the server to respond, and the delay

is significantly less; even if compared with long polls like Comet, it can transfer data more times in a short time.

3) Stay connected

Unlike HTTP, Websocket needs to create a connection first, which makes it a stateful protocol, after which some state information can be omitted during communication. HTTP requests may require status information (such as identity authentication, etc.) to be carried on every request.

Since Websocket has the above advantages over http, combined with the huge number of universities, websocket is used instead of the HTTPS protocol used by TAXII.

5. Platform Model Design

5.1. Shared Model Design

The threat intelligence sharing platform of colleges and universities designed in this paper is a platform for threat intelligence sharing among all universities. The improved STIX and TAXII are used as threat intelligence standards and communication standards, and the radial shape model specified by TAXII is used as a sharing model. The radial shape model is shown in **Figure 4**.

In the radial sharing model, an organization serves as a central node for all participating nodes. Any participating node first sends the information to the central node, and then shares it with other participating nodes. The central node may analyze and filter before sharing information. In this model, information can flow from the participating nodes to the central node or from the central node to the participating nodes. Each university is a participating node in the model, and the TAXII server acts as a central node. When a university wants to share threat information, it sends threat information to the central node according to the new STIX standard, and other universities can obtain the latest threat information and corresponding solutions through the central node.

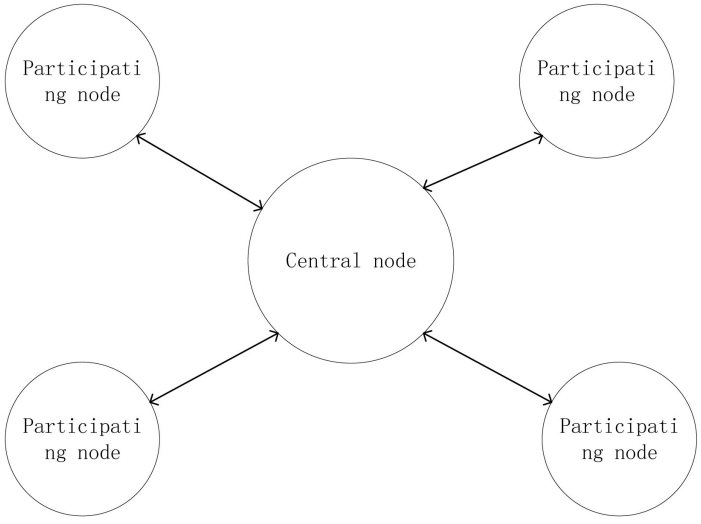

Figure 4. Radial shape model.

5.2. Frame Design and Technology Selection

5.2.1. Overall Frame Design

Based on security considerations, the TAXII server is deployed on a cloud server and enables defense devices such as firewalls. Since the working hours of colleges and universities are generally from 8:00 am to 5:00 pm, in order to ensure the real-time and high efficiency of threat intelligence sharing, the participating nodes cannot be deployed in the office area. In the design of this paper, the participating nodes are deployed in the equipment room and use Raspberry pi as a hardware carrier. The Raspberry Pi is only a credit card-sized microcomputer, and its system is based on Linux. The Raspberry Pi is deployed in the server room of a university, saving space and cost. The platform model is shown in Figure 5.

Each participating node represents the Raspberry Pi deployed in each university computer room. When the participating nodes communicate with the central node for the first time, the two parties establish a persistent connection mechanism through WSS. When the participating nodes send threat information to the central node, if the central node succeeds upon receipt of threat intelligence which means all participating nodes can obtain the latest threat intelligence and corresponding solutions by sending a request to the central node and the heads of various universities can implement this process by remotely logging in to the Raspberry Pi.

5.2.2. Related Technology

Based on the selection of the platform design model in the previous section, the technical choice of platform design is divided into two parts: the participating node and the central node, as shown in Table 3.

TAXII server functions are divided into two major functional modules: authorization authentication and threat intelligence sharing.

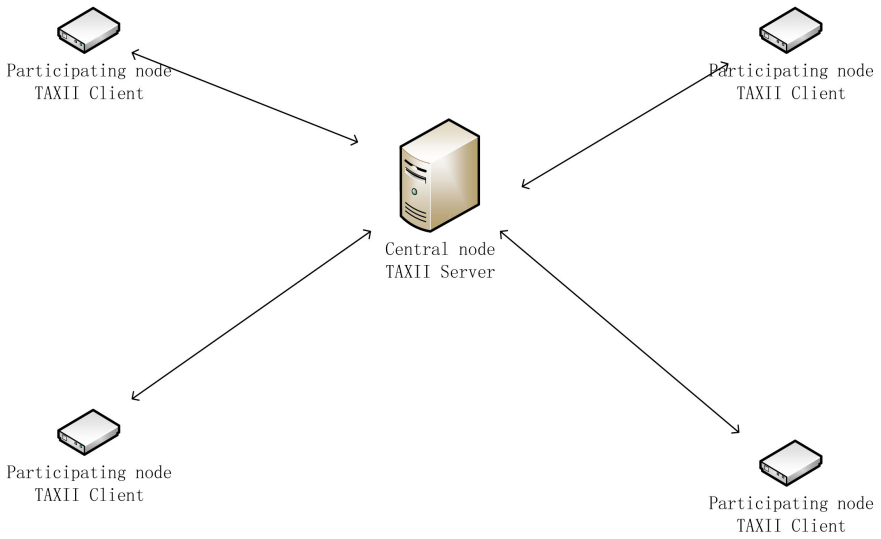

Figure 5. Platform model.

Table 3. Technical parameters table.

Node Type	Platform hardware and software environment	Achieve function
Central node	Windows 10 Enterprise Edition 64-bit 17,134 Python 3.6.4 64-bit Django 2.0.1 MySQL 8.0 64 bit	As the implementation framework of TAXII server, Django is designed to meet the format standard specified by TAXII. MySQL stores client authentication data and threat intelligence data.
Participating node	Raspbian system Python 3.6.4 64-bit	The participating nodes use the Websocket module to transmit data with the central node.

1) Authorization authentication

Used to verify the identity of the TAXII client. The client authenticated by the client can interact with the server.

2) Threats to report shared data

The authenticated client sends the request according to the standard format specified by TAXII, and the TAXII server returns the corresponding Collections, API Root and other information from the database.

The code for the authorization authentication function of the TAXII server is as follows.

```
def login(request):
...
Username = uf.cleaned_data ['username']
Password = uf.cleaned_data ['password']
User = User.objects.filter (username__exact = username, password__exact = password)
...
```

The model definition part of the STIX format threat intelligence in the database is as follows.

```
...
Name = models. Char Field (max_length = 50)
Description = models. Text Field (blank = True)
Validator = models. Foreign Key ('Validator', Blank = True, null = True)
Date_created = models. Date Time Field (auto_now_add = True)
Date_updated = models. Date Time Field (auto_now = True)
...
```

6. Experimental Results and Analysis

The experiment tests the stability, real-time and security of the proposed platform design model. The experimental software and hardware environment is the same as the platform design environment. As shown in Table 3.

6.1. Real-Time

As a threat intelligence sharing platform, the core metric is whether the response is rapid and the timeliness of the communication between the participating nodes and the central node. As shown in Figure 6.

By sending 1000 times of threat intelligence transmission tests to 10 participating nodes and calculating the average response time per 100 times, it can be seen from the figure that the transmission delay is up to 72 milliseconds and the transmission delay is at least 48 milliseconds when the network is unblocked. Threat intelligence sharing within this delay range guarantees real-time performance.

6.2. Platform Stability

The experiment selected 10 participating nodes to communicate with the central node for communication, using the common HTTPS protocol and WEBSOCKETS protocol. It is more appropriate to determine which communication protocol is more appropriate by counting the impact of changes in the requests of multiple participating nodes over time on the load of the central node. The load capacity of the central node as a function of time is shown in Figure 7.

Figure 7 shows the trend of the number of TCP links in the TCP/IP stack of the system as a function of time when 10 participating nodes send requests to the central node at the same time. The number of TCP links is counted every 10 seconds. The straight line showing the horizontal trend is used. WSS is used as the communication protocol. Because the characteristics of WEBSOCKETS are long links, when the participating nodes establish a good communication channel with the central node, real-time full-duplex communication can be performed, and 10 participating nodes establish a stable 20 states of ESTABLISHED. This greatly improves the communication stability of the system. The curve shows the communication using HTTPS protocol. The central node will

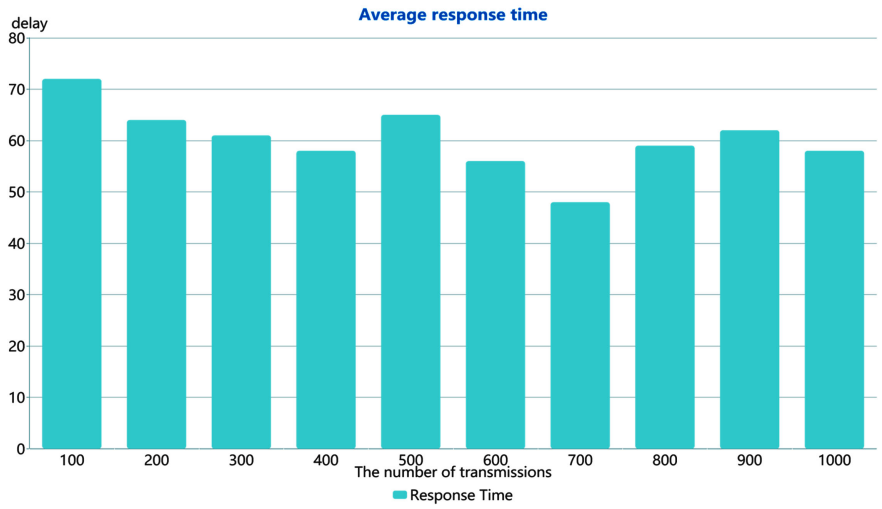

Figure 6. Average response time.

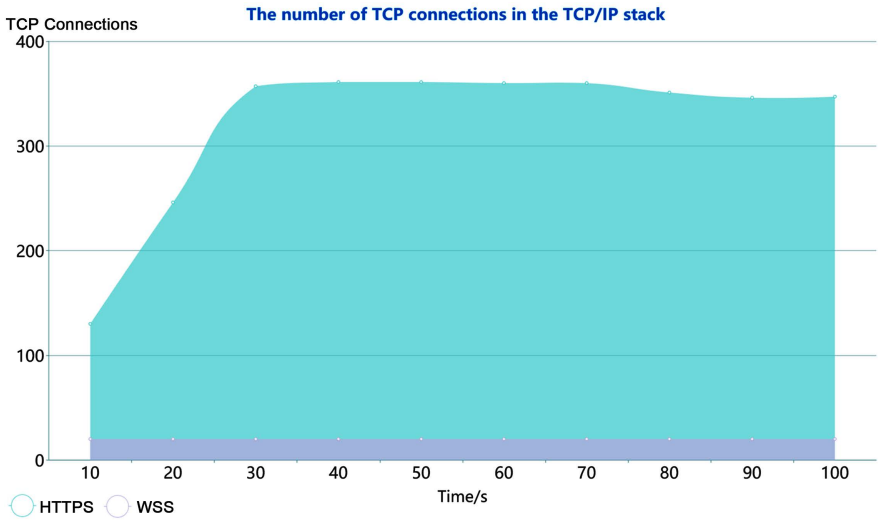

Figure 7. Load capacity curve.

open a new socket each time to communicate with the participating nodes. The data sent by the participating nodes. With the characteristics of small amount of data and many times, when the interaction is completed, the participating nodes will actively disconnect. In the initial 10 seconds, the number of TCP links of the central node increases rapidly, and the state of the number of added TCP links is TIME_WAIT. In 10-30 seconds, the TIME_WIAT state in the TCP/IP stack is exploding. After 30 seconds, the number of links is maintained at around 350.

As the number of participating nodes increases, the TCP/IP stack [14] of the central node is occupied by a large number of TCP links in the TIME_WAIT state, consuming a large amount of server resources. This situation is caused by the party that actively closes the communication. After FIN, TCP sends an ACK packet and enters the TIME-WAIT state to ensure that the remote TCP receives the connection interrupt request confirmation, which largely ensures that both parties can end normally. But there are also problems. To pass the next connection needs to wait for the 2MSL time. In order to ensure the real-time and stability of threat intelligence sharing, the participating nodes need to transmit data with the central node from time to time, and the transmission time is much smaller than the MSL time, so over time, the central node protocol stack The TCP link in the TIME_WAIT state will be more and more, eventually be kept at a certain number and consume consumption of server resources. The experimental results show that using WSS as the communication standard of TAXII has better stability and real-time performance.

6.3. Platform Security

Acunetix [15] [16] was used for penetration testing of TAXII servers to scan for security vulnerabilities, especially for Owasp top 10 vulnerabilities. The results of Owasp top 10 [17] are shown in **Table 4**.

Table 4. Owasp top 10 scan results.

Number	Vulnerability type	Scan result
1	Injection	Safety
2	Invalid authentication and session management	Safety
3	Cross-site XSS	Safety
4	Insecure object direct	Safety
5	Forged Cross-Site Request (CSRF)	Safety
6	Security error configuration	Low risk
7	Restrict URL access failure	Safety
8	Unverified redirects and forwards	Safety
9	Apply known vulnerability components	Safety
10	Sensitive information exposure	Low risk

As shown in Table 4, there are no high-risk vulnerabilities in the central node. These vulnerabilities include injection, XSS, and unsafe object direct references. The security misconfiguration and sensitive information are exposed to a low-risk state because the cloud center outputs some debugging information at runtime, but the information does not contain sensitive information. It can be seen from the scan results that the safety of the center ground is higher.

7. Conclusion

Based on the environmental characteristics of colleges and universities, this paper improves the existing STIX and TAXII standards, and analyzes the shortcomings of the existing network threat intelligence sharing platform. It proposes a college cyber threat intelligence sharing platform using the improved STIX and TAXII standards. The experiment proves that the platform designed in this paper can effectively apply the sharing of network security threat intelligence in colleges and universities and protect the network security environment of colleges and universities.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Ministry of Education of the People's Republic of China (2019) 2019 National Higher Education List. http://www.moe.gov.cn/jyb_xxgk/s5743/s5744/201906/t20190617_386200.html

[2] Yang, Z.-M., Li, Q., Liu, J.-R., *et al*. (2015) Research on Threat Intelligence Sharing and Utilization for Attack Source Tracing. *Information Security Research*, **1**, 31-36.

[3] Thomas, R.K., *et al*. (2019) System and Method for Modeling and Analyzing the Impact of Cyber-Security Events on Cyber-Physical Systems. U.S. Patent Applica-

tion No. 15/264,028.

[4] Xu, L.-P. and Hao, W.-J. (2016) The Status Quo of Threat Intelligence in US Government and Enterprise Networks and Its Enlightenment to China. *Information Network Security*, No. 9, 278-284.

[5] Elchin, A. and Burger, E. (2016) Semantic Ontologies for Cyber Threat Sharing Standards. 2016 *IEEE Symposium on Technologies for Homeland Security*, Waltham, MA, 10-11 May 2016, 1-6. https://doi.org/10.1109/THS.2016.7568896

[6] Gong, Y. (2017) Research on Threat Intelligence Usage and Sharing Method. Chinese Computer Society. *Proceedings of the* 32*nd National Computer Security Academic Exchange Conference*, 4.

[7] Liu, Y., Zhang, H.-F., Zhang, L., *et al*. (2018) Study on a Penetration Testing Collaboration Scheme Based on STIX Information Interaction. *Information Technology and Network Security*, **37**, 1-5.

[8] Kim, E., Kim, K., Shin, D., Jin, B. and Kim, H. (2018) CyTIME: Cyber Threat Intelligence ManagEment Framework for Automatically Generating Security Rules. *Proceedings of the* 13*th International Conference on Future Internet Technologies*, Seoul, 20-22 June 2018, Article No. 7.

[9] Haass, J.C., Ahn, G.-J. and Grimmelmann, F. (2015) ACTRA: A Case Study for Threat Information Sharing. *Proceedings of the* 2*nd ACM Workshop on Information Sharing and Collaborative Security*, Denver, CO, 12 October 2015, 23-26. https://doi.org/10.1145/2808128.2808135

[10] Frank, F., Smulders, A. and Kerkdijk, R. (2015) Cyber Security Information Exchange to Gain Insight into the Effects of Cyber Threats and Incidents. *e & i Elektrotechnik und Informationstechnik*, **132**, 106-112. https://doi.org/10.1007/s00502-015-0289-2

[11] Bedini, I., Matheus, C., Boran, A., Patel-Schneider, P.F. and Nguyen, B. (2011) Transforming XML Schema to OWL Using Patterns. 2011 *IEEE* 5*th International Conference on Semantic Computing*, Palo Alto, CA, 18-21 September 2011, 102-109. https://doi.org/10.1109/ICSC.2011.77

[12] Connolly, J., Davidson, M. and Schmidt, C. (2014) The Trusted Automated Exchange of Indicator Information (Taxii). The MITRE Corporation, Bedford, MA, 1-20.

[13] Fette, I. and Melnikov, A. (2011) The Websocket Protocol. https://doi.org/10.17487/rfc6455

[14] Dunkels, A. (2001) Design and Implementation of the lwIP TCP/IP Stack. *Swedish Institute of Computer Science*, **2**, 77.

[15] Attack, Cross Site Scripting (2014) Audit Your Website Security with Acunetix Web Vulnerability Scanner. https://www.acunetix.com/websitesecurity/cross-site-scripting/

[16] Daud, N.I., Abu Bakar, K.A. and Md Hasan, M.S. (2014) A Case Study on Web Application Vulnerability Scanning Tools. 2014 *Science and Information Conference*, London, 27-29 August 2014, 595-600. https://doi.org/10.1109/SAI.2014.6918247

[17] Wichers, D. (2013) Owasp Top-10 2013. OWASP Foundation.