SDN と STIX を組み合わせた情報共有とインシデント対応の自動化

満永 拓邦[†1]　松田 亘[†1]　藤本 万里子[†1]

概要：サイバー攻撃による被害の深刻化が進んでおり、組織にとってサイバー攻撃のリスクは看過できないものになっている。セキュリティ対策の一環として組織を超えた情報共有が行われているが、共有に伴う手続きや処理は手動で実施されることが多く、効率的とは言い難い。またインシデントの発生時においても対応処理の大部分が手動で行われている。本稿は、動的なネットワーク構成変更を可能にする Software Defined Networking(SDN) とサイバー攻撃活動を記述するための技術仕様である Structured Threat Information eXpression(STIX) を組み合わせて、情報共有やインシデント対応を自動化することにより、セキュリティ被害の早期発見や未然防止を図ることを目的とする。

キーワード：情報共有，インシデント対応，自動化，SDN，STIX

Automated System for Information Sharing and Incident Response with SDN and STIX

Takuho Mitsunaga[†1]　Wataru Matsuda[†1]　Mariko Fujimoto[†1]

Abstract: Damages of cyber attacks become serious along with the sophisticating the attack techniques. To mitigate the damages, information sharing form victim organization is conducted among Japanese organizations. However, the method is not efficient since the processes for information sharing are conducted manually. In this paper, we propose an automated system for information sharing and incident response with Software Defined Networking and Structured Threat Information eXpression.

Keywords: Information Sharing, Incident Response, Automated System, SDN, STIX

1. Background

A large number of cyber attack cases in Japan are seen, along with the global trend. Attack techniques are becoming more sophisticated each year and security countermeasures need to be taken based on the assumption that intrusion can occur anytime, to anyone. One of the activities that is taking place is the sharing of threat information which is collected from victim organizations. In Japan, cyber security organizations such as NISC, JPCERT/CC, IPA and the National Police Agency provide such threat information sharing frameworks[1][2][3]. Many organizations especially critical infrastructure organization, take part in such activities. When this information is shared, recipients can look into their logs in their proxy servers and security devices in order to check if there is any access from their network to the URL in question, or take preventive action out of the information.

Depending on the information that is shared, the action need to be taken is different. The recipients need to understand the information correctly and take measures accordingly. In a case of malware C&C server information. recipient organizations need to check their outbound proxy logs and firewall logs against the shared list. In another case, if DDoS attack information is shared, they need to check inbound communication towards their web server or Web application Firewall. And actions such as network shutdown or isolation of infected computer depend on the threat level of cyber attacks when malware infection is found. Summarizing the steps for shared information, these actions are conducted manually because of two reasons.

(1) the recipients need to understand the information correctly, and take proper actions as mentioned above
(2) the recipients need to judge the threat level of shared Information to take proper action

In this paper, we propose, an automated system for information sharing and incident response with the technique combining Software Defined Network(SDN) and Structured Threat Information eXpression(STIX).

2. Preliminary

2.1 STIX

As for problems regarding to the understanding of recipients for information sharing, one approach is to follow the standard format and procedures to describe and share the information. As

†1 東京大学情報学環
Interfaculty Initiative in Information Studies

a breakthrough to the problems, a standard format called STIX, The Structed Threat Information expression, was introduced by athe MITRE, supported by the US government.[4]. STIX is one of the major standard formats for describing indicators. It uses XML or JSON format to indicate C&C server URL and attack period. This means all recipients have the same understanding of the data. As a result, STIX can minimize gaps between the sender and recipients, and enables smooth, automatic sharing of information.

Judging the threat level, further effort needs to be made. Each type of cyber attacks has different threat levels. Dealing with the all in the same priority is not an effective approach. For example, when just looking at malware, there are too many kinks-adware, ransomware, banking trojan..etc. Depending on the type of malware, the priority for response is different. However, for someone who are not familiar with malware type, it would be difficult to judge the theat level by the names detected on antivirus software. It may panic them even if it was low-risk malware infection. As an unfortunate example, a Japanese organization detected malware that has a high treat level. And when they asked anti-virus vendors, they were informed that the malware is not a kind that leaks information outside. As a result, the organization did not take the risk seriously, which ended up in the massive personal information leakage. Later on, the malware turned out to be “Emdivi” which is used in the APT operation against Japanese Organizations. As you can see, sometimes threat levels of cyber attacks may not be judged correctly. In term of judging the threat level and countermeasure, we decided to automate decision making process.

2.2 Software Defined Networking (SDN)

The proposed system makes a database out of received indicators, and analyses its threat level. Also, by linking it with SDN, route switching can be made dynamically, and we can change communication from client computers. Based on indicators according to the threat level of the information. For example, if the indicator matches with logs in the organization network, analysis engine finds it and kick the API to the SDN controller. Then, OpenFlow SW changes the communication from client which is likely-infected. And if the threat level of indicator is high, the communication is blocked immediately.

3. Related Work

Sato and Sekiya introduced a monitoring and communication control mechanism at the end of the network to detect and respond to infection at a location close to the infection source and designed a system that minimizes infection spread and information leakage damage[5][6]. Shiota et al. developed administration policy description method to reduce the burden. By presenting description forms to the administrator based on the use frequency of scenario patterns and receiving the selection of resources, the association between the resources can be easily described.[7]

4. Proposed Method

The proposed system makes a database out of received indicators, and analyses its threat level. Also, by linking it with SDN, route switching can be made dynamically, and we can change communication from client computers. Based on indicators according to the threat level of the information. For example, if the indicator matches with logs in the organization network, analysis engine finds it and kick the API to the SDN controller. Then, OpenFlow SW changes the communication from client which is likely-infected. And if the threat level of indicator is high, the communication is blocked immediately. Or if the threat level of indicator is middle, the communication goes to the internet through packets capture. As a first step, we made a database out of indicators and blacklists of malicious hosts, access logs. The analysis engine is designed to extract malicious hosts which are possibly used for actual attack activities by examining the indicators and other confirmed malicious hosts. This system can import STIX format data to database. Let me show you a demonstration to visualize the indicator and its analysis results.

First method is pattern matching. Characteristics in the letters, for example in domain names, are examined to see if it matches any existing data. And Black lists have attributes such as this domain is used by APT attacker or Ransom attacker. So, if there are any data which matches existing data with some attribution. The threat level can be found. It also examines relations to other attackers through Open Source Intelligence. For example, if the domain registrant is identical to other malicious hosts, it is considered that there is a correlation. If the analysis engine judges that the host potentially has a high risk, communication to the host needs to be blocked immediately. Here I would like to show you an example of route controlling through SDN. The SDN is linked with a controller, which centrally controls the network flow. It can dynamically change the network flow based on the destination IP address and other information. This function is used to control or allow communication to certain hosts. We decided to classify the analysis results into the following three levels. First is the highest severity, where the host is judged as malicious through the analysis. In this case, communication to the host is blocked. The second is a suspicious case where the indicator has something in common with other existing, malicious indicators. For such hosts, communication is still allowed, but the packet capture will be taken so that the

communication can be analysed later. The third is the harmless hosts. Communication is allowed as normal. What we are still working on is the timing for implementing the network flow. One option is to set up a rule at the time when threat information is received. With this option, every time we received STIX file, the rules of SDN is changed. Another option is to set up a rule when communication to such hosts is observed, by analyzing logs. In this case, the rule is changed when malware-infected computer is found.

For the first option, since rules are set in advance, process which occurs at the time of communication is relatively small. This also saves the load on the controller. However, it requires a lot of rules to be set beforehand. Also, IP addresses are resolved from URLs at the time of receiving information and there may be changes in due course. It is also possible that all the IP addresses cannot be mapped. For the other option, opposite problems are expected. The advantage is the small number of rules to define. However, for early detection, it needs to almost real time of communication between proxy logs and analysis engine. The other challenge is the rules for blocking. Communication to hosts that match the blacklist are blocked. Also, through log analysis, if any devices that have communicated with the blacklisted hosts are found, these devices are likely to be infected with malware. In order to prevent further damage caused by the attack activities, we are considering that this system can be improved by blocking any communications from likely-infected devices, not just to blacklisted hosts. Since SDN can only define rules up to Layer 4, domain names in the indicators need to be resolved to IP addresses separately and registered to the system. In order to keep the list up to date, it would be better to check name resolution regularly.

5. Conclusion

We developed an automated system contributing in reducing the time for initial response to shared information with the technique of STIX and SDN. Normally, once organizations receive indicators, they start detection, analysis and containment processes. With this system, this part can be done automatically. This enables speedy incident handling, which does not depend on skills of the security personnel

References

[1] “標的型攻撃に関する情報共有体制", https://www.nisc.go.jp/active/infra/pdf/cc_c4tap.pdf (参照 2017-08-28).

[2] “早期警戒情報とは", https://www.jpcert.or.jp/wwinfo/wwdata.html (参照 2017-08-28).

[3] “サイバー情報共有イニシアティブ", https://www.ipa.go.jp/security/J-CSIP/" (参照 2017-08-28).

[4] “Structured Threat Information eXpression (STIX™) 1.x Archive Website", https://stixproject.github.io/ (参照 2017-08-28).

[5] 佐藤 康次. 関谷 勇司. SDN を用いた Network 監視によるデータ漏えい防止機構の検討. 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報.. 115(484):2016.3.3・4. IN2015-139. 183-188 ISSN 0913-5685

[6] 佐藤 康次. 関谷 勇司. 出口対策に向けた耐感染性を有したネットワーク監視並びに防御システムの検討. 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報.. 116(361):2016.12.15・16. IN2016-82. 91-96 ISSN 0913-5685

[7] 塩田 実里. 山口 由紀子. 嶋田 創. 他. 自動ネットワーク記述システムにおける管理ポリシー記述手法の実装. 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報.. 114(71):2014.6.5・6. ICSS2014-10. 49-54 ISSN 0913-5685

[8] Itoh, S. and Goto, N.. An Adaptive Noiseless Coding for Sources with Big Alphabet Size. IEICE Transactions. 1991, vol. E74-A, no. 9, p. 2495-2503.

[9] Foley, J. D. et al.. Computer Graphics: Principles and Practice in C. 2nd ed., Addison-Wesley Professional, 1990, 1200p.

[10] 千葉則茂, 村岡一信. レイトレーシング CG 入門. サイエンス社, 1990, 282p.

[11] Chang, C. L. and Lee, R. C. T.. Symbolic Logic and Mechanical Theorem Proving. Academic Press, 1973, 331p.