STIX 2 IDS

Arnold Sykosch[1,2] and Matthias Wübbeling[1,2]

[1]University of Bonn - Working Group IT Security

[2]Fraunhofer FKIE - Cyber Security Department

Abstract The possibility to run threat intelligence against the IT infrastructure to protect, creates opportunities for protection enhancement as well as threat intelligence sharing group management. We introduce our approach to link STIX formatted data to an infrastructure's assets.

I. Introduction

Knowing a threat before being confronted with it, is an invaluable advantage for the defender. This provides one with the opportunity to prepare for emerging threats. Therefore, communities are established, to share threat intelligence with each other. Thus, participants learn from each other and may increase their own protection. Threat intelligence is often structured in threat reports. These reports are usually expressed in a format called STIX (*Structured Threat Information eXpression*). We will use STIX compliant vocabulary from here on. [2]

The fact that one obtains threat intelligence to protect oneself, does not mean that it was obtained before falling victim to the threat actor. To connect the own infrastructure to given threat intelligence is a challenging task. Protection based on STIX formatted data is not fully technically supported yet. This contribution takes first steps towards that direction. It enables security analysts to run threat intelligence through their own infrastructure. This allows the search for indicators of compromise of a present threat as well as the evaluation of the information that describes it.

II. SITX

STIX data is meant to describe a threat's meta information like the *threat actor*, the *campaign* he is running and the *tools, tactics and procedures* (TTP) he is employing to achieve his goals. Along with this TTP, a threat report might also describe *indicators*. These indicators ”convey specific Observable patterns combined with contextual information intended to represent artifacts and/or behaviors of interest within a cyber security context.”[9]

These Observables are defined in a format called *Cyber Observable eXpression* (CybOX). CybOX objects contain a comprehensive description of the object itself by characterizing its attributes. These attributes describe an object, which can be identified with them accordingly. A typical example is the description of a file by its name, location, checksum, type, etc. These objects are the link between a threat description and the infrastructure in which it may be observed. [8]

III. Intrusion Detection Systems

The description of an object, may be interpreted as its signature. Ideally the object is identified before it reaches a host. Most networks employ some kind of network-based intrusion detection system (NIDS). NIDSs are developed specifically for this purpose, the identification of objects or actions in a network. We therefore created a mapping between CybOX formatted objects and NIDS signatures.

Shortcomings exist, when it comes to network based detection. Not every object is visible to the detection mechanisms. Traffic might be encrypted or the object might be created on the host and never transmitted over the network. There might also be situations where the reconstruction of an object from its representation during transmission is computational to expensive, e.g. when a file has to be identified by its MD5 checksum. Therefore, a host based approach usually complements the detection. Since the only link between the infrastructures' objects and STIX data is the CybOX object, an alert should reflect the id of a CybOX object.

A. Network Based Intrusion Detection Systems

Monitoring network packets to detect suspicious traffic is the purpose of network-based intrusion detections systems (NIDS). Usually, they are located at a network's border or between specific parts of a network. The detection algorithms of common NIDSs are either signature-based or anomaly-based. Each and every packet that bypasses the network interface is examined accordingly. Having a set of signatures, derived from CybOX objects, each signature formally represents a pattern of a known attack. The first signature a packet matches leads to a corresponding action. Depending on the NIDS setup, possible actions range from simple event logging up to dropping the packet or adjusting external firewalls in order to prevent further attacks. [3][10]

Several types of CybOX represent network traffic related objects. They contain attributes of network connections and their metadata as well as their content. Such information is valuable in a general manner and may be manually mapped to nearly all IDSs. STIX data might include specific NIDS signatures as instance of a test mechanism for the described threat. Nevertheless, they are only valuable to those STIX consumers running the corresponding NIDS. Thus, specific signatures are rarely distrubuted via STIX.

Snort and Suricata are signature-based NIDSs sharing the same signature language, which has been established by the Snort project. Both are well known tools in the security community due to their open-source licenses and the maturity of Snort [6].

One target of this approach is to automatically utilize threat intelligence data to improve the NIDS's set of signatures. A first step is to map CybOX attributes from network related objects to elements of Snort/Suricata signatures. A naive mapping approach very likely results into a hugh number of generated signatures. This leads to a second step, the optimization and aggregation of them. Based on the topological position of CybOX objects, signatures are combined to reduce the NIDS's matching effort.

By the time of writing, a prototype for the first step already exists. Preparations have been made to evaluate performance effects on NIDS and possible improvements of generated signatures.

B. Host Based Intrusion Detection Systems

When it comes to host based intrusion detection (HIDS) the typical approach is log based. All log entries reflect events and therefore changes in state that are recorded by the system. If an event is marked as *security related* an alert is generated. The majority of CybOX objects describe the state of a system's object. To match an CybOX object with an HIDS it is necessary to reconstruct the state of the system's object. This exposes two specific issues:

1) The log has to reflect changes to all attributes of an object.
2) The initial state has to be known.

An approach is needed that enables the retrieval of the current state of a system. This motivates the usage of a live forensics framework. With *GRR Rapid Response* the possibility to check an object's state is given implicitly. It facilitates a client server model. The client is reflected by an installed agent, which delivers results based on server side defined queries. These results typically consist of the described object itself. [5]

GRR supports two alternative implementations. One possibility is to use the selection method of the client to retrieve an object and check for the rest of the attributes on the server side. Another one would be to implement the rest of the checking methods into the client and only report *alerts*, with the specific CybOX object id. If a match is found the object might be retrieved by GRR standard methods on demand for further investigation.

IV. Related Work

There are two similar host based approaches worth mentioning focusing the linkage between threat intelligence and the infrastructure. The first one is an app called SPLICE for the SPLUNK SIEM. It indexes STIX data and makes it addressable for custom alert rule creation [4]. However, since SPLUNK ist log driven the downsides laid out in Section III-B apply.

A tool to check threat intelligence directly is Redline® by MANDIANT [7]. It provides the possibility to check for IOCs, that are shaped in OpenIOC [1]. This format does not support meta information like STIX does and Redline® only supports Windows. A solution that maps CybOX objects into NIDS signatures as described in Section III-A is not known to the authors by the time of writing.

V. Conclusion

The ability to check threat intelligence against the whole infrastructure enables the use of threat intelligence to estimate impact. It further enables evaluation of the given information's quality in terms of false positives generation. This opens up the opportunity to evaluate information sources by a reputation system, providing positive and negative feedback.

References

[1] OpenIOC - An Open Framework for Sharing Threat Intelligence. http://www.openioc.org/, Retrieved: April 1st, 2015.

[2] S. Barnum. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX™). *MITRE Corporation*, 2012.

[3] Joachim Biskup. *Security in Computing Systems: Challenges, Approaches and Solutions*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[4] Cedrix Le Roux. SA-SPLICE. https://splunkbase.splunk.com/app/2637/, Retrieved: April 1st, 2015.

[5] Google (GRR). GRR Rapid Response Documentation. https://github.com/google/grr-doc, Retrieved: April 1st, 2015.

[6] Algis Kibirkstis. Intrusion Detection FAQ: What Are The Top Selling IDS/IPS and What Differentiates Them from Each Other?, November 2009. http://www.sans.org/security-resources/idfaq/top-selling-ids-ips.php, Retrieved: April 1st, 2015.

[7] MANDIANT. Redline®. https://www.mandiant.com/resources/download/redline, Retrieved: April 1st, 2015.

[8] MITRE Corporation. Cyber Observable eXpression Documentation. https://cyboxproject.github.io, Retrieved: April 1st, 2015.

[9] MITRE Corporation. Structured Threat Information eXpression Documentation. https://stixproject.github.io, Retrieved: April 1st, 2015.

[10] Pramod Pandya. *Computer and Information Security Handbook*. Springer Publishing Company, Incorporated, 2nd edition, 2013.