An Actionable Threat Intelligence System using a Publish-Subscribe Communications Model

Syam Appala
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
syam1@cisco.com

Nancy Cam-Winget
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
ncamwing@cisco.com

David McGrew
Cisco Systems
13600 Dulles
Technology Drive
Herndon, VA 20171
mcgrew@cisco.com

Jyoti Verma
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
jyoverma@cisco.com

ABSTRACT
We designed a system for securely distributing Threat Intelligence and recommended Courses of Action (CoAs), combining that information with local contextual information, determining which response system(s) can carry it out, and then putting the course of action into effect. Our system uses STIX to express threat information, including CoAs. We identified the problem of matching CoAs with the actions that a response system can carry out as a major design challenge, and found a robust and scalable decentralized solution to that challenge by adopting a publish-subscribe model. We built a solution based on the Extensible Messaging and Presence Protocol (XMPP) architecture and communications protocol as it provided the right security properties as well as the needed extensibility for both data model and transport protocols. We motivate and describe our system, and the use cases of Cyber Threat Prevention, Cyber Threat Detection, and Incident Response.

Categories and Subject Descriptors
D.3.2 [Java]: publish-subscribe communications model; D.3.2 [C]: publish-subscribe bus communications model; I.7.2 [XML]: STIX reports; D.4.6 [Operating Systems]: Security and Protection — Access Control; K.6.5 [Management of Computing and Information Systems]: Security and Protection; K.7.2 [ACM]

General Terms
Security, Experimentation, Design, Measurement, Performance, Verification

Keywords
Threat incidents; indicators of compromise; incident response; publish-subscribe; secure transport; authentication; authorization; controller; synchronous communication; courses of action; remediation; investigation; mitigation

1. INTRODUCTION
Threat Intelligence feeds and systems are the current trend as a result of the increased number of security incidents, breaches and increased awareness of such threats. At the same time, technology continues to evolve on improving the efficacy of threat detection and of threat response.

However, to manage cyber threat response activities effectively, and in a timely manner, there is a need to connect detection and response systems and include these as cohesive components of the security process cycle. This paper describes a Threat Intelligence system that can provide improved automated threat management by correlating information from different Threat Intelligence providers as well as other sources of information, in particular from network elements. More importantly, the system can also provide semi-automation techniques for improving on the mitigation and remediation based on the correlated information.

1.1 Threat Intelligence System Model
A generalized workflow for a Threat Intelligence system is one that can process threat information from potentially one or more sources to generate a comprehensive report and formulate one or more recommendations for an appropriate set of actions. A model for a Threat Intelligence system is shown in Figure 1; Threat Intelligence (TI) providers generate threat intelligence that can be consumed by a Threat Response Management (TRM) ecosystem to help derive defensive actions against known indicators or ongoing threats. The TRM takes action by utilizing one or more Response Systems. Threat Response Systems can be network or endpoint controllers; security elements or data center controllers. Incident responses can be categorized as investigation, mitigation, and remediation as detailed in Section 2.3.1.

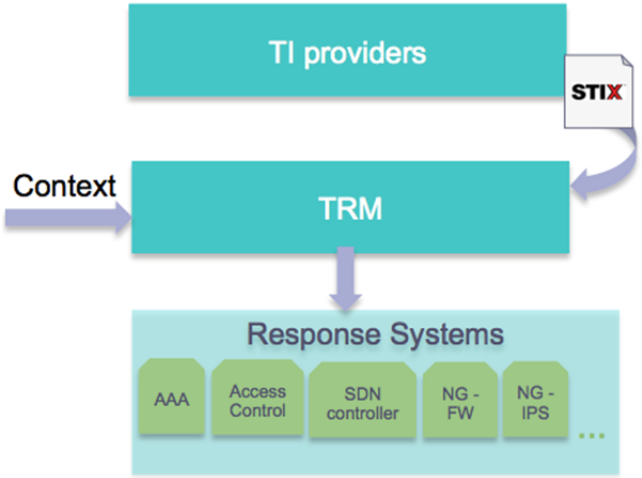

Figure 1 Threat Intelligence System

WISCS'15, October 12, 2015, Denver, CO, USA.

DOI: http://dx.doi.org/10.1145/2808128.2808131

In the proposed Threat Intelligence system, the Structured Threat Information eXpression (STIX) document is used to facilitate how TI providers can feed the threat information to the TRM using a common language. STIX is a formal standard[1][4] language used to represent a wide variety of cyber threat information along with actionable details that can help manage the cyber threat response activities in the network.

To improve on the detection and understanding of threats there may also be other components that can provide further information. An example of this could be information from the network such as session accounting from an Authentication, Authorization and Accounting (AAA) server, or from other network elements that can provide system logs or network statistics (IPFIX)[19]

As the information can now vary from STIX reports to different information from the network, a diverse set of information using different data modes needs to be communicated over a secure communication channel. The information may also be time sensitive and requires timely or on demand delivery.

Beyond the need to provide integrity and confidentiality in the communication channel, the Threat Intelligence components need to ensure authenticity of the communicating peers. That is, a TI provider should ensure that the TRM ecosystem it is communicating with and sending information to, is authorized to receive the data; similarly, the TRM needs to authenticate and authorize the TI providers and providers of specific context information. For example, failure to authenticate and authorize the TI providers and context providers can lead the TRM to suffer different attacks ranging from a simple Denial of Service (DoS) attack to a disruption and untrustworthiness of the data it receives. Failure of the TI to authenticate the TRM may lead to the unintended disclosure of sensitive threat intelligence information to an unauthorized entity. As the number of TI providers and consumers grows, so does the need to control the access to this information. With such security implications, the Threat Intelligence system must consider providing management functions to allow for the secure life cycle management including authentication, authorization, key management and secure communications.

In addition to the security management functions, the Threat Intelligence system must facilitate the ability for the TRM to learn or discover the different types of information available. Similarly, components such as TI providers and context providers should be able to advertise the type of information it can provide; Response systems should also be able to advertise the types of courses of actions it can act upon.

In this paper, we present a design for a Threat Intelligence System that is based on STIX and the Extensible Messaging and Presence Protocol (XMPP) [9][10][11], and an architecture and messaging protocols that leverage XMPP's publish-subscribe messaging protocols, federation and security capabilities. XMPP was chosen as the preferred architecture as it also defines a flexible authentication framework that can further be extended to provide the required authorization model for the Threat Intelligence system. XMPP is widely used in collaboration systems and known to scale to support millions of "users" (in our model a user is a Threat Intelligence component). Besides the open standardization of XMPP, it has proven its ability to extend its language to allow for different data models and protocols, which is a hard requirement for a Threat Intelligence system.

1.2 Requirements for Threat information model and system

The role of the TRM is to connect threat intelligence with a response system that can take the appropriate actions based on that intelligence. TI documents consumed by the TRM contain a description of an incident or an indicator of compromise (IoC) and one or more suggested Courses of Action (CoAs). An IoC description is a set of observables and values that are associated with a threat, such as the Internet address of a malware command and control server. A Course of Action is an action that can be taken by the infrastructure to mitigate the threat whose indicators are described by the IoC. An incident description is a set of observables and values associated with a particular incident, such as the address of a (probably compromised) device that contacted a command and control server at a particular time. Incident descriptions are most relevant to the entities responsible for the assets that participated in the incidents, while IoC descriptions contain generic information of more broad interest. STIX contains facilities for detailing both, and provides a rich set of observables. The TRM can consume both incident and IoC descriptions, and then use local contextual information to determine if a response is warranted. To continue our example, it may determine that the device that contacted a malware server contains sensitive information, and thus should be immediately prevented from communicating. We envision the TRM as a way to bring more automation to the security process cycle (Figure 2), in which incidents are detected, appropriate response systems are selected, and responses are taken, followed by ongoing monitoring to ensure that the response was effective. This process is often carried out manually; we see the TRM as a means to automate some or all of the processes. A security administrator can approve a recommended CoA, for instance, or choose a particular CoA from a list of alternatives, then rely on the TRM to carry out the action. Many practitioners see this semi-automated system as an effective way to leverage the expertise of human security analysts while minimizing the mundane tasks they must perform. STIX can convey both human and machine-readable data, making it well suited for this sort of system.

When the TRM determines that an action is needed, it must communicate the description of CoA to a response system that can carry it out. In general, there may be a single system that can take the action, or there could be multiple systems able to do so, or there might be no such system. Examples of such systems include endpoint security software, programmable network functions and new trends in Network Function Virtualization (NFV) [27], DNS sinkholes, and BGP blackhole. Each of these systems has different capabilities, such as perimeter blocking, which cuts off all internal communication with an external address, and internal blocking, which blocks all communication to and from a device with an internal address.

An essential part of the Threat Intelligence System is the ability of the TRM and the response systems to solve the *action/capability matching* problem, of determining which response system(s) are capable of carrying out the CoA. In general, there are multiple response systems, and potentially even multiple components in the TRM (such as the consoles of two different information management systems). Importantly, there may be multiple response systems of a single type; only one of them may be able carry out the CoA, or it may be the case that all need to carry it out. Consider the case of in which there are multiple network controllers (in the Software Defined Networking[25], SDN sense), each of which controls a set of routers that each connect a subnet

to the Internet; each router is capable of installing Access Control List (ACL) entries. The TRM can communicate with a network controller, through its northbound Application Programming Interface (API) in order to have it install ACL entries on routers to block selected traffic and to report when traffic that hits a particular ACL entry for monitoring purposes. Each campus network or data center network may have its own controller, and each controller can cause the configuration of ACLs for a particular set of subnets. When the TRM needs to perform perimeter blocking, it must communicate the CoA to *all* network controllers, so that communication to all subnets is blocked. In contrast, when internal blocking is needed, the TRM must determine which of the network controllers owns the subnet on which the internal address appears. This example highlights several aspects of the action/capability matching problem: the capabilities, configuration and context of each response system determine whether or not it can carry out a CoA, and multiple response systems may need to carry out a particular action.

One approach to solving the action/capability matching problem is to use a capability registration system, in which each response system registers its capabilities with a central entity. Before an action is taken, the registration system is consulted to determine which response system(s) should be actuated. However, this sort of centralization would bring drawbacks: a single system would need to scale to handle registration requests, registration changes, and capability lookups. As there may be many response systems involved in protecting a large information system, and their capabilities may vary (especially when device mobility is involved), scalability is a major concern.

A better alternative is a decentralized and loosely coupled approach in which each response system can independently determine whether or not it can carry out a CoA. We realize this alternative by using a publish-subscribe system, in which each response system subscribes to the CoA capability (or, in XMPP terms, a CoA topic) that it can act upon; conversely the TRM publishes the CoA topic mapping to the desired course of action. In this model, each topic identifies a coarse-grained set of actions (such as setting blocking ACL entries), while each subscriber independently determines whether or not it is capable of carrying out the published action (for instance, by determining if it can block a particular internal address). In addition, the publish-subscribe model contains a many-to-many communication system that naturally accommodates CoAs that must be carried out by multiple response systems, and also accommodates scenarios in which there are multiple components within an TRM that generate CoA messages. In short, the publish-subscribe model facilitates decentralization of both the overall system, and of the TRM itself.

Since each response system independently determines if it can carry out a CoA, redundant responses may occur, in which more than one response system performs the same CoA. This situation can be detected, in a decentralized way, by having the response systems publish a message describing the CoAs that they have taken. Once redundant responses have been detected, the systems involved can determine which one(s) of them should back out their CoA, to eliminate the redundancy. This determination could be done in any of several different ways, but the following simple and effective distributed method is attractive. Each response system publishes "CoA taken" messages, and listens for those messages. Each of those messages includes a random number chosen by the sender, and a response system backs out a CoA if it receives a "CoA taken" message with a response number that is higher than the one that it sent. This process is robust against failure of message delivery, in the sense that no CoA will be backed out by all of the systems. Each response system will need to understand if its own CoA taken is redundant with those taken by other response systems.

Courses of action recommended by the TI provider, and CoAs determined by the TRM, should be as generic as possible, to maximize the likelihood that there is a response system that can carry out that CoA. For instance, internal blocking can be accomplished by endpoint security software, or by installing an ACL entry on a wireless system or network switch close to the endpoint, or by using an AAA (RADIUS [20]) change of authorization to remove the network access privileges of a device. When a CoA requests internal blocking, what is essential is that each response systems that can carry out that action, can match that request against their capabilities, and thus understand that they can and should take the action, regardless of what mechanism they would use. This can be achieved simply by using an information model in which each abstract action (such as internal blocking and perimeter blocking) corresponds to a set of concrete actions (endpoint ACL, switch ACL, etc.). Suggested and recommended CoAs contain abstract actions, while response systems are aware of the concrete actions that they can carry out and the abstract actions to which they correspond. Once a CoA is taken, the concrete action is then documented. STIX provides the flexibility to convey a concrete CoA taken, as well as an abstract action. In a decentralized system, the CoA taken message must be generated by the response system; a publish-subscribe topic can be used to propagate these messages to the IR components.

In summary, building an effective and scalable Threat Intelligence System requires a decentralized solution to the action/capability matching problem, and we identify a solution of that type in the use of the publish-subscribe messaging paradigm along with an information model for CoAs that enables abstract actions to be requested and concrete actions to be matched against those requests.

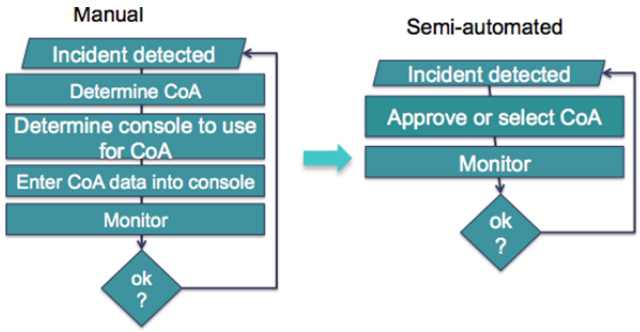

Figure 2. The security monitoring and response cycle

Our contributions in this paper are:

1. Proposing a reference architecture for a threat intelligence system with TI providers, Context providers and TRM
2. Introducing a secure publish subscribe communications model leveraging XMPP to help realize the actionable threat intelligence system
3. Defining common response type categories and proposing a valid implementation of the CoAs for these categories

1.3 Secure Communications Requirement

Given the system's focus, the Threat Intelligence system must provide security mechanisms to ensure that the communications between the components, e.g. TI provider, context providers. TRM and Response systems must be secure. Additionally, the Threat Intelligence system must provide mechanisms to ensure the components interacting with each other are identifiable (e.g. authenticated) and authorized to perform their functions.

Considering the heterogeneous deployments of the Threat Intelligence system, the following architectural considerations and requirements exist:

- **Federated communications:** as TI providers may be applications that are outsourced or providers from third party sources, the communications channel must allow for (network) transports outside the enterprise's domain.
- **Secure communications:** at minimum, the information exchanged from TI provider to TRM and from TRM to the Response systems must provide authenticity and integrity. Some deployments may also require confidentiality.
- **Extensible interfaces:** as the Threat Intelligence system must adapt to new threats and analytic algorithms, the interfaces will also need to adapt and evolve.
- **Data model agnostic:** while TI providers can standardize and share threat information using STIX; context providers' information can vary widely based on their context, e.g. syslog, ipfix or netflow, (network) session information or statistics. To allow for the diverse types of information flowing through the Threat Intelligence system, the communications model must allow for different data models and data operations.
- **Flexible content retrieval mechanisms:** as breaches, security events and recommendation actions can be time sensitive, the interfaces must allow for publish-subscribe, peer-to-peer direct queries and out-of-band communications.
- **Distinct control/management and data planes**: to allow for the diversity of data models, operations and transports while ensuring security. A distinct control plane must be defined to allow for the security negotiations, content retrieval negotiation and discovery of the Threat Intelligence components (and their capabilities).

In addition to the overall system requirements, the transport protocol and associated services requirements for achieving an effective incident response include:

- An effective solution to the action/capability matching problem described above, that ensures that an action will be taken if there are response system(s) that can carry it out, and that detects and eliminates redundant responses.
- Dynamic ability to determine and dispatch course of action to one or multiple incident response actor systems based on the network topology, location affinity, capability of the consumers.
- Traceability of the CoA across its life cycle (suggested, requested, and taken). To improve on the visibility and management of the overall Threat Intelligent system, the CoA history and status should be provided
- Provide policy-based authorization control to ensure the right system(s) are processing and acting upon the course of action.

The rest of this paper is organized as follows: Section 2 gives an overview of STIX and describes it as a model for threat events and response; Section 3 describes the secure publish-subscribe communications model based on XMPP; Section 4 goes over the Threat Intelligence use cases and how they can be realized using STIX and the XMPP based controller; Section 5 concludes the work and calls out references to future work in this area.

2. STIX: a Model for Threat Events and Response Information

STIX is an evolving standard [2] for representing a wide variety of cyber threat information in a structured manner that is expressive, flexible, extensible, automatable and readable.

STIX provides a common mechanism for addressing structured cyber threat information across and among the full range of the following cyber security needs:

- Analyzing cyber threats
- Specifying indicator patterns for cyber threat
- Managing cyber threat response activities
- Sharing cyber threat information

STIX reports can include specific recommended actions for incidents, and abstracted recommendations for indicators that can be translated to specific actions. These actions can fall under the three generic categories - investigation, mitigation and remediation. STIX directly leverages the Cyber Observable eXpression (CybOX) [3] schema.

STIX provides loose-coupling mechanisms and default implementations for leveraging the following constituent schemas as appropriate:

- Common Attack Pattern Enumeration and Classification (CAPEC) [21]
- Malware Attribute Enumeration and Characterization (MAEC) [22]
- Common Vulnerability Reporting Framework (CVRF) [23]
- OASIS Customer Information Quality (CIQ) xPRL [24]

2.1 STIX Components

While STIX is a very rich language, it provides an underlying set of 8 well-defined components that are given shared context in STIX reports.

1. Observables – An observable as defined by CybOX is a set of properties or characteristics that describe an entity within a cyber operational domain. Observables are standalone objects, which can provide context to Incidents and Indicators by answering the question "What activity has been or might be seen?"
2. Indicators – contain observable patterns mapped to a TTP context and other metadata like confidence assertion, likely impact, known sightings etc. to answer the question "What threats should I look for on my networks and systems and why?"
3. Incidents - discrete instances of indicators describing an ongoing threat describing a specific adversary action with affected assets, nature of the compromise, parties involved etc. to answer the question "Where has this threat been seen?"
4. Tactics, Techniques, and Procedures(TTP) – represent and characterize adversary behavior including attack patterns, malware, exploits, kill chains, tools,

infrastructure, victim targeting etc. to answer the question "What does the threat do?"

5. Exploit Targets - describe vulnerabilities, weaknesses, or configurations that are targeted for exploitation by TTPs of threat actors and can include handling guidance. They help answer the question "What weaknesses does the threat exploit?"
6. Courses of Action – preventive, prescriptive or predictive measures that can be deployed in the network or on endpoints to apply defenses against known threat indicators or to address ongoing threats. They help answer the question "What can I do about the threat?"
7. Campaigns – instances of Threat actors pursuing a shared intent, potentially across organizations as described by sets of incidents and/or TTPs to answer the question "Why does the threat do this and how have others dealt with it"
8. Threat Actors – identification and/or characterization of the adversary representing a cyber attack including characterization of identity, suspected motivation, suspected intended effect, historically observed behavior etc. to answer the question "Who is responsible for this behavior?"

STIX is both flexible and extensible to allow for representation of cyber threat information pertaining to a variety of use cases. Even though the STIX schema is highly verbose, almost everything is optional so that use cases could leverage only the portions of the specification that are relevant for them without being overwhelmed by the rest. Existing standardized languages may be leveraged as optional extensions where appropriate and numerous flexibility mechanisms are designed into the language. For example, STIX reports can be represented in XML and JSON formats today. Specific subsets of STIX capabilities can be defined and agreed to beforehand in the form of profiles for use within sharing communities, by tools, etc.

2.2 Using STIX for Incidents and Indicators

Indicators of Compromise (IOCs) or indicators – Indicators convey a method of an infiltration attempt. As an example, IOCs are virus signatures, file hashes, URLs or IP addresses that tie a compromise together. STIX indicators can represent IOCs by tying together stateful properties/events (Observables) potentially mapped to a related TTP context and other relevant metadata like confidence, restrictions, time window, impact, sightings, related campaigns, related indicators, source and suggested course of actions.

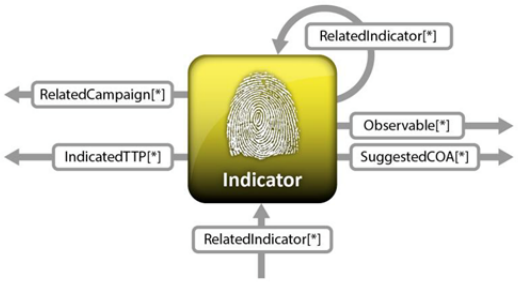

Figure 3: STIX indicator with relationships

Incidents – Incidents are discrete instances of indicators affecting an organization. They consist of data such as time, parties involved, assets affected, impact assessment, related Indicators, related Observables, leveraged tactics, techniques, and procedures (TTP), attributed Threat Actors, response Course of Action requested, response Course of Action taken, confidence in characterization etc.

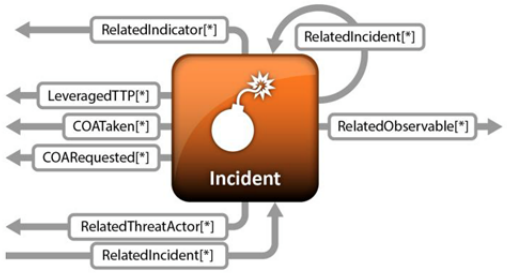

Figure 4: STIX incident with relationships

2.3 Using STIX for Threat Response

Actionable details that go along with the threat information could be used to harden the network in a preventive manner so as to defend against known threat indicators; in a prescriptive manner to mitigate/remediate ongoing threats and to aid in forensic investigation of past threats. These actions could be specified in STIX reports using the following constructs:

- Suggested Course of Action (SuggestedCOA) - The action(s) indicated by the SuggestedCOA field of the STIX indicator can be used to program defensive COAs for Indicators of Compromise (IOC) learned from a variety of different sources and/or generated from security tools. For Example, for blocking access to a known botnet Command and Control (C2C) server, the SuggestedCOA tied to the STIX indicator could specify perimeter blocking actions such as BGP blackhole to block access to the external IP or DNS sinkhole to blacklist and sinkhole the malicious domain.
- Requested Course of Action (RequestedCOA) - For ongoing threats, the RequestedCOA field of the STIX incident can be used for mitigation and/or remediation in the incident response workflow. For example, if an internal network host was suspected to be communicating with a C2C server, there could be a series of actions specified by the RequestedCOA to copying its traffic to an inspection device (to determine the scope of the attack), blocking the host from doing lateral reconnaissance of its neighbors to eliminating the malware by containing the host to a remediation service.
- Course of Action taken (COATaken) – the actions taken to resolve an incident can be tabulated in the STIX Incident's COATaken field. This enhanced incident information can aid in better handling of future incidents of similar type.

2.3.1 Response actions

The response actions as described in this paper can be divided into three main categories and can be network based or endpoint based. The TRM can invoke any or a combination of these actions from the response systems capable of delivering the functionality.

- Investigate - Obtain more information about a threat
- Mitigate - Block, but not eliminate, a threat
- Remediate - Fix or eliminate a threat

Type of Action	Network based	Endpoint based
Investigate	• Inspect with IPS / NGIPS • Netflow/IPFIX monitoring • Packet capture	Scanning an endpoint
Mitigate	• Perimeter blocking using BGP blackhole, DNS sinkhole, ACL • Interior blocking using 802.1X Change of Authorization, ACL • Containment using VLAN tagging, SGT tagging	• Kill process • Delete file
Remediate	Containment to remediation server or service	• Reimage host • Remove software • Reinstall software

Table 1 Response actions

3. A Secure Management and Transport model

A Secure Threat Intelligence System that meets the requirements described in Section 1.3 has been implemented. Our implementation uses STIX as the Threat Intelligence language by which TI providers and the TRM can represent, understand and process the threat information consistently. For the data functions and its communication channels, the Threat Intelligence system leverages XMPP's communication protocols to facilitate the different required data acquisition mechanisms through the XMPP query or publish-subscribe protocols.

The messaging framework also includes a control plane to facilitate the management operations including:

- Dynamic discovery: enabling Threat Intelligence components to discover other components and their capabilities especially the abstract actions that they can carry out. For example, TRM can discover available TI providers, context providers and Incident Response systems. When a system registers for a topic associated with an abstract action. Note that the discovery is bidirectional; e.g. TI providers can also discover available TRM, as Incident Response systems can discover available TRM and so on.

- Authentication and Authorization of Threat Intelligence components: enabling the components to authenticate and be authorized to be part of the Threat Intelligence system. As part of the Threat Intelligence system, the management function ensures that components are authorized to function as that component; For example, the TRM needs to be authorized to consume and process STIX information from TI providers.

- Secure communications: the management function can aid in establishing secure data communications either synchronously (via a direct query) or asynchronously (via a publish-subscribe).

The management and data transport system in our implementation is the Platform eXchange Grid (pxGrid) [11][10]. PxGrid is a messaging framework that enables secure, bi-directional sharing of information between participating network elements and applications. PxGrid is built on top of Extensible Messaging and Presence Protocol (XMPP), where XMPP servers deployed in cluster mode with message routing constitute the data plane communication. The pxGrid Controller, an external XMPP component, provides the centralized policy-based control plane for pxGrid.

Based on XMPP, pxGrid leverages and extends XMPP's architecture and protocol extensions to provide the required security and communication properties. XMPP's authentication is based on SASL [26]; it is used to enforce strong mutual authentication and secure communication channels based on TLS. Extensions to the XMPP management architecture were made in pxGrid to enforce a centralized, policy-based authorization while retaining a decentralized data management plane. XMPP was chosen as it is widely used in a federated environment such as instant messaging systems and for its ability to evolve and extend to suit the sharing of different data constraints, from datagrams to sound to video that require different data models, compression requirements as well as time latencies; requirements that are also needed in a Threat Intelligence system.

With the goal of fostering an open secure ecosystem for sharing information, the XMPP transport and control plane aspects of pxGrid have been submitted [13] for standardization to the Internet Engineering Task Force (IETF) as the proposed control plane and transport for the Security Automation and Continuous Monitoring (SACM) working group. Note that while this paper focuses the use of pxGrid to Threat Intelligence, the pxGrid has wider applicability in facilitating the secure sharing of information as is documented in the IETF draft, e.g. showing its applicability for sharing posture information.

3.1 Use of XMPP based publish-subscribe

A Threat Intelligence system can consume a wide variety of security information ranging from indicators of compromise, incident descriptions, intrusion detection events, as well as contextual information provided by different network elements. The information can vary in datagram size and be made available with different timing constraints. For instance, a AAA server may provide updates as users come onto the network but also can provide a full view of all the current users that is currently on the network. In the case of the updates, the AAA server should make this information available through a publication or notification to the TRM in real-time and can achieve this in a reasonably sized datagram where as a full user view can be a very large data exchange as in large deployments potentially thousands to hundreds of thousands of users could be on the network. In the latter case, it is expected that these full view updates would be requested directly from the TRM to context providers, as it needs the information to either initialize or synchronize its own state.

The need for affecting different forms of communications (e.g. direct query versus a notification) implies the need for a messaging system that can adapt to datagram sizes, time latencies and network protocol agility. A simple analogy is to envision a Threat Intelligence system as a social media system; whereby there is a need to facilitate the socialization or sharing of communication or data in a secure way; where the data may vary from short bursts of data updates (e.g. notifications of users

coming or leaving a network) to very large data requests to synchronize a system (e.g. requesting full network packet captures or requesting a full view of the active network sessions) XMPP is a proven architecture and open source standard that evolved from a social collaboration tool, Jabber. There are now hundreds of XMPP extensions in use to facilitate different types of information such as video, audio and from a security context, IODEF [13] and addressing each of their requirements; such as compression and timing synchronization for video and audio. Having such a proven record for meeting the collaborative requirements securely and in an open source environment, XMPP is a good choice to adopt for a Threat Intelligence system.

3.2 Use of a secure controller to broker information sharing

The Threat Intelligence system creates capabilities through XMPP topics such that authorized components can act as subscribers or publishers to the topics on the XMPP based Threat Intelligence management service. Note that while the management service can be co-hosted in any of the Threat Intelligence system component, in our implementation we decoupled the management service to an effective controller that is based on XMPP (e.g. an XMPP based server). Each component of the Threat Intelligence system must authenticate into the system, e.g. the XMPP based controller. As shown in Figure 5, a Threat Intelligence component must first authenticate and obtain authorization to be a part of the Threat Intelligence system. Note that the authorization is per operation.

For example, a context provider will be authorized to be a publisher of context information but not threat intelligence; it may also not be authorized to consume any other information. Once authorized, it can then register to obtain information through a publish-subscribe registration or can obtain information directly through a query operation. A Threat Intelligence component may be a publisher as it shares information (e.g. a TI provider or context provider) or it may be a consumer (e.g. a TRM). Note that these are basic functions in the communications channel as an actual Threat Intelligence component may be both a publisher and a consumer; e.g. a response system may subscribe to obtain the CoAs but publishes the "CoA taken" information.

Once authenticated and authorized through the control plane, the producers and consumers of information can exchange data in real-time through publish-subscribe notifications, or on-demand peer-to-peer or out-of-band queries.

3.2.1 Establishing Authentication and Authorization

Client systems such as a TI provider or TRM connecting to pxGrid go through an authentication, account approval and authorization before they can successfully publish or subscribe to XMPP topics and transact information. The XMPP server, acting as the pxGrid secure controller supports Simple Authentication and Security Layer (SASL) [26], to facilitate the authentication of publishers and consumers. SASL provides flexible authentication options such as TLS, or plain challenge-response schemes to the connecting clients. Certificate based mutual authentication is the de facto authentication mechanism pxGrid uses for enhanced trust and security. The XMPP server restricts the authenticated clients' access within the pxGrid controller, in order for the controller to approve client accounts and apply granular authorizations on clients' subscription and query requests. On successful authorization with the pxGrid controller, the authorized clients have access to data plane communication with other authorized participants.

The pxGrid controller manages the clients' authorizations per XMPP topic; e.g. authorization to publish or subscribe is granted per XMPP topic. This alleviates the need for the controller to authorize each publish/subscribe message as only authorized clients participate in the topic. For directed peer-to-peer and out-of-band bulk queries, the controller evaluates the authorization policies on-demand before allowing the subscriber to request the capability provider.

3.2.2 Securing Communication Transport

Beyond the requirement to authenticate the clients and authorize them for an XMPP topic for a pxGrid operation such as publish/subscribe, directed or bulk queries etc., the pxGrid controller also enforces the protection of data in motion. That is, based on the configured policies and authorizations, the data may be further protected at the (network) transport layer to provide confidentiality and integrity.

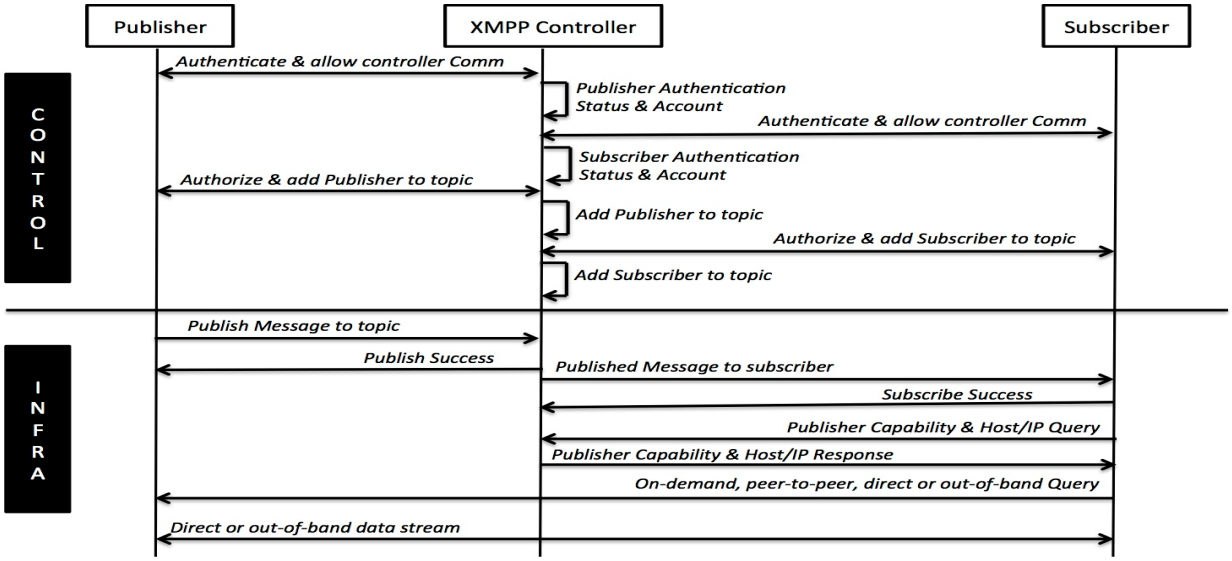

Figure 5 Establishing Secure Communications

These considerations are critical, especially in Threat Intelligence where mitigations can occur. Those remediation or mitigation actions must be protected from attack.

3.3 Comparison of other models

While many publish-subscribe communication transports such as MQTT[15], RabbitMQ[17] and Kafka1[18] exist, they lack sufficient centralized management along with a decentralized data plane The most relevant communication model to the STIX based Threat Intelligence system proposed here, is the Trusted Automated eXchange of Indicator Information (TAXII). TAXII is the current de-facto method defined by MITRE [1] for exchanging STIX information. It defines services, protocols and messages for cyber threat information producers and consumers to exchange cyber threat information related to detection, mitigation or prevention of cyber threats. As a protocol, TAXII supports hub-and-spoke, peer-to-peer and source-subscriber threat sharing models. While TAXII is not tied to a specific network transport or format, it is mainly transported over HTTPS and enforces a centralized data communication model whereby all consumers of information must establish a connection with all producers. From a general security perspective TAXII leaves access control or content access rights to the producers and consumers of information. In contrast, the pxGrid controller performs centralized authorization on all clients' publish/subscribe or query requests. TAXII core components mandate support for polling, inbox/push services and specific APIs provided are producer prerogative, and message filters are left for consumers to deal with. In pxGrid producers of information define specific model, real-time, on-demand and/or bulk query APIs, sub-topics with filtering criteria they support, and specify allowed operations/actions in order for controller to enforce authorization policies. PxGrid also supports on-demand topic creation with authorization control performed by pxGrid controller. TAXII does not clearly specify out-of-band data plane negotiation, API, data model extensibility, message filtering criteria and sub topics or dynamic topics creation.

Table 2 below provides a comparison between TAXII and pxGrid

Feature	TAXII	pxGrid
Security	●	●
Transport bi-directionality	●	●
API & Data Model Extensibility	◐	●
Information filtering	◐	●
Integration with other transport	! ! ●	●
Dynamic data collection service	●	●
Open Source	●	◐

● Supported ● Not Supported ◐ Partially Supported

Table 2 TAXII vs. pxGrid

As we observe that TAXII is network transport agnostic, it is possible to use TAXII's protocol encapsulation within pxGrid to leverage XMPPs publish-subscribe model as well as its directed query mechanism. The consideration to enable this is to leverage pxGrid's ability to decentralize and further allow for improved scalability.

Other considerations that can be supported within the pxGrid architecture is the ability to provide anonymity of the originating sources or the information being shared. With a centralized control plane enforcing authorization, the authorization policies can be defined to anonymize the publishers (or a subset of publishers) as well as filtering or obscuring the information (or subset of information) being published.

4. Threat Intelligence Use cases

A Threat Intelligence system can have many applications and use cases. In our implementation, we focused on three main use cases: threat detection, threat prevention and incident response. This section will further describe these three use cases.

4.1 Threat Prevention

Security administrators evaluate potential preventative courses of action for identified relevant threats in the industry and select appropriate actions in order to harden and prepare the network for such attacks. As new IOCs are learned, indicators with the suggested courses of action could be implemented in the network. For example, in the case of a confirmed phishing attack, a STIX indicator for the phishing attack can be generated calling out a suggestive course of action (SuggestedCOA) to implement a blocking rule at the email gateway. In another situation if an administrator learns about a new botnet attack, a STIX indicator calling out the details of the attack might be generated with a suggestive course of action to implement a blocking rule at the web security gateway.

In the system called out here, the TI providers receive indicators of compromise (IOC) from various sources and convert them into the STIX format. The STIX reports are then published to the TRM where suggestive courses of action are added and the respective rules are applied to the security tools and network devices. The suggestive courses of action are published on the XMPP based controller and received by the subscribers of that functionality. For example, an SDN controller can implement a network block action.

Figure 6 below shows a flow wherein the TI provider receives an indicator for a new malicious command and control server and a block action is programmed for the C2C.

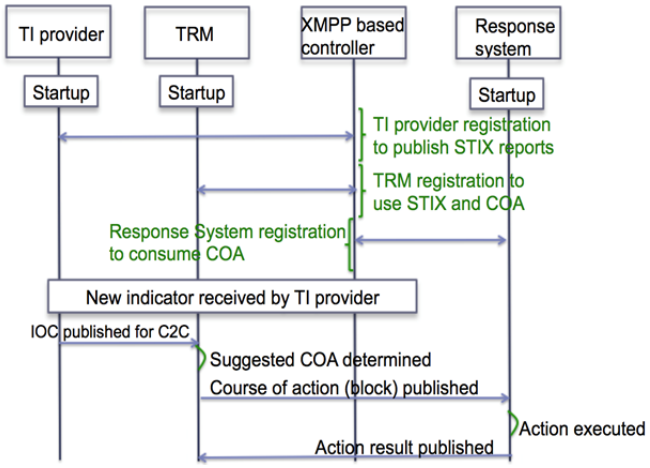

Figure 6: Flow diagram for Threat Prevention use case

4.2 Threat Detection

Security administrators apply manual, automated or semi-automated means to monitor and assess network and endpoint activity in order to detect the occurrence of specific threats whether in the past through historical evidence (from stored logs, packet captures etc.) or currently ongoing through dynamic situational awareness. Threat detection tools look at different facets of the operational data environment and employ different techniques to determine threats. Some of these techniques involve checking reputation of entities involved and drawing parallels between the activities with known indicator patterns.

For example, in the case of a confirmed phishing attack with defined indicators, cyber operations personnel may harvest any specified observable patterns from defined indicators of the attack and apply them appropriately within the operational environment to detect any evidence of the phishing attack occurring and the internal IP addresses involved in the attack. In order to achieve this, the cyber security analysts will run queries on top of Security Information and Event Management tools (SIEM) to draw a correlation between the observables in indicators and their operational environment.

In the system called out here, the TI providers correlate observable information from the published indicators with the operational environment and generate incident reports in STIX if an activity is determined. These reports are then published over the XMPP based controller to the SIEM for responders to review.

Figure 7 below shows the flow for a new incident generated by the TI provider.

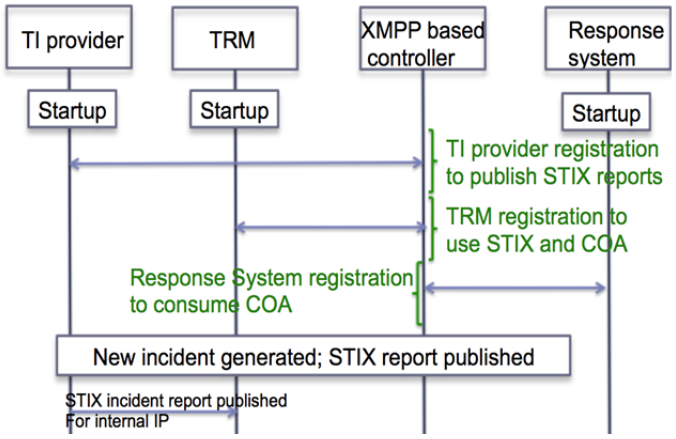

Figure 7: Flow diagram for Threat Detection use case

4.3 Incident Response

Cyber operations personnel respond to detections of potential cyber threats through ***investigation*** of what has occurred or is occurring, scope the nature of the actual threat, perform ***mitigation*** to block the threat from infecting the neighbors and ***remediation*** to fix the attack.

For example, in the case of a confirmed phishing attack, cyber operations personnel may conduct investigative activities to determine whether the phishing attack was successful in carrying out an exploit (e.g., was malware installed or run) and if so, attempt to characterize in detail those effects (e.g., which systems were affected by malware, what data was ex-filtrated, etc.). Once the effects are understood, cyber operations personnel would implement appropriate mitigating or corrective courses of action (e.g. wipe and restore systems, block exfiltration channels, etc.).

In the system described here, the TRM will receive incidents from the various TI providers and identity context from the context providers over the XMPP based controller. It will extract the observables from the incidents and correlate them with any available context. Appropriate actionable details can then be added to this contextual information by the incident responder manually or be derived automatically by the SIEM. The response will then be published over the XMPP based controller to be addressed by the response systems. For example, if an incident reported that an internal host was infected with malware, the incident responder may wish to observe the behavior of the malware by running it in a sandboxed environment. For this, he may contain the host within a network segment and will publish the response to the XMPP based controller. The action can be picked up by response systems can sandbox the host.

Figure 8 below shows the flow for an incident response use

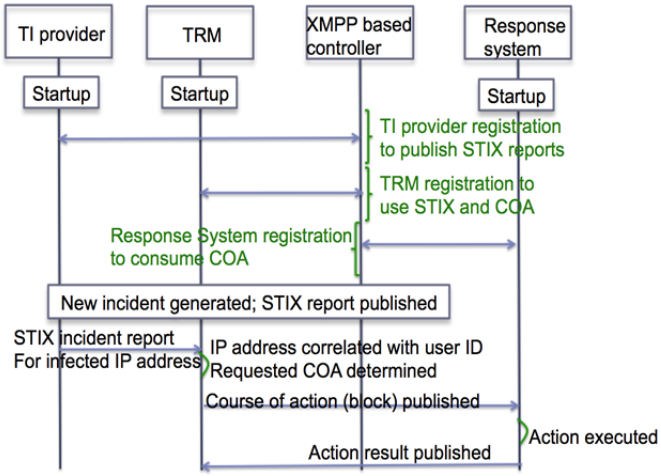

Figure 8: Flow diagram for Incident Response use case

5. Overall Scalability

While the use cases described above detailed the actions required for the particular scenarios, this section will describe the scalability considerations. As our implementation uses pxGrid to facilitate the information flowing into and out of the TRM, this section further discusses how the use of the control plane is applied to help broker and scale the overall solution.

5.1 Scaling the number of participants

The pxGrid framework allows multiple simultaneous XMPP topics to be created, where each topic could have multiple publishers, publishing the same data model, and registered subscribers. The subscription list can scale by deploying a number of XMPP servers organized in message routing cluster mode and there is virtually no restriction on the number of XMPP servers. Similarly, the control plane scales directly proportional to the number of pxGrid Controllers (where a pxGrid controller could have a one-to-one association with each XMPP server) organized in cluster mode, sharing control plane meta-data among themselves in real-time.

5.2 Scalability of the information flow

As discussed in Section 5.1, XMPP offers scalability in a clustered deployment with a number of servers routing messages between them. Clients can connect to any of the XMPP servers and the XMPP message router and publish-subscribe functionality figures how to route the published messages to the subscribers.

We benchmark performance for a user session topic to be published by an Access Control Server where attributes such as user identity, user group, device type, device posture, location and other attributes are published (the published message size being approximately 1 KB). The message is published as users are authenticated and authorized by the Access Control server. On a single quad-core, 16 GB memory server, with 100 subscribers, a no drop publish rate of 200 events per second, 1500 directed queries per second and 3000 out-of-band bulk queries per second was benchmarked. These numbers could be scaled to more subscribers in a pxGrid cluster mode deployment and with publisher side optimizations to achieve further scalability in publish and query rates.

6. Conclusions and Future Work

Our goal was to build a threat response management system that can consume threat intelligence from multiple sources, combine that information with local contextual information, make or accept a course of action, determine which response system(s) can carry it out, then put the course of action into effect. We also sought to provide strong access control on threat intelligence, so that sensitive data is not inappropriately shared. We found STIX to be well suited to our needs; it can express a wide range of threat-relevant data, including a course of action, and it is both machine and human readable, facilitating semi-automated systems. We identified the action/capability matching problem as a major design challenge, which is outside the scope of STIX, but which is important in realizing a system that can put threat intelligence into action. We found a good decentralized solution to that challenge in the adoption of a publish-subscribe model. We built a solution on top of pxGrid, which brings with it a strong access control system that meets the needs for the protection of threat intelligence data.

Our system has been built and tested at a small scale, using some extensions to STIX that express a machine-readable course of action. It would be valuable to flesh out the information model for courses of action, so that it encompasses more types of response systems; we hope to see this work in a future standard. We have not built or tested a large-scale system, which would be interesting future work. It would also be interesting to explore the use of many different types of response systems, and investigate the scaling of our system in cases with many such systems.

7. ACKNOWLEDGMENTS

The authors would like to express our thanks to Bret Hartman for supporting this project and to Scott Pope for his continued passion to promote an open secure API ecosystem to improve security solutions. We would also like to acknowledge Panos Kampanakis for reviewing both our work and this paper.

8. REFERENCES

[1] MITRE Corporation, http://www.mitre.org/

[2] Structured Threat Information eXpression, https://stix.mitre.org

[3] Cyber Observable eXpression (CybOX™) http://cybox.mitre.org/

[4] OASIS Cyber Threat Intelligence (CTI) Technical Committee, https://stixproject.github.io/oasis-cti-info.html

[5] Sean Barnum, Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX™) https://stix.mitre.org/about/documents/STIX_Whitepaper_v1.0.pdf

[6] Trusted Automated eXchange of Indicator Information https://taxii.mitre.org/

[7] David McGrew, Jyoti Verma, Making Threat Intelligence Actionable – Recommending Actions using STIX https://www.rsaconference.com/events/us15/agenda/sessions/1775/making-threat-intelligence-actionable-recommending

[8] Cisco Identity Services Engine (ISE) data sheet, http://www.cisco.com/c/en/us/products/collateral/security/identity-services-engine/data_sheet_c78-656174.html?cachemode=refresh

[9] P. Saint-Andre, Extensible Messaging and Presence Protocol (XMPP): Core, RFC 6120, https://tools.ietf.org/html/rfc6120

[10] P. Saint-Andre, Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, RFC 6121, https://tools.ietf.org/html/rfc6121

[11] P. Saint-Andre, Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM), RFC3922, https://tools.ietf.org/html/rfc3922

[12] P. Saint-Andre, End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP), RFC3923, https://tools.ietf.org/html/rfc3923

[13] Syam Appala, Nancy Cam-Winget et al, XMPP Protocol Extensions for Use in SACM Information Transport, Internet Draft, https://tools.ietf.org/html/draft-salowey-sacm-xmpp-grid-00

[14] Cisco Platform Exchange Grid, pxGrid, http://www.cisco.com/c/dam/en/us/products/collateral/security/identity-services-engine/at_a_glance_c45-728420.pdf

[15] MQ Telemetry Transport, MQTT, http://mqtt.org/

[16] A. Hefczyc, et al, XEP-0268: Incident Handling, http://xmpp.org/extensions/xep-0268.html

[17] Rabbit MQ, https://www.rabbitmq.com/

[18] Apache Kafka, http://kafka.apache.org/

[19] B. Claise, Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, RFC5101, https://tools.ietf.org/html/rfc5101

[20] M. Chiba, et al, Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS), RFC 5176, https://tools.ietf.org/html/rfc5176

[21] Common Attack Pattern Enumeration and Classification (CAPEC™) http://capec.mitre.org/

[22] Malware Attribute Enumeration and Characterization (MAEC™) https://maec.mitre.org/

[23] Common Vulnerability Reporting Framework (CVRF) http://www.icasi.org/cvrf

[24] OASIS Customer Information Quality (CIQ) xPRL https://www.oasis-open.org/committees/ciq/

[25] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks" (PDF). *White paper*, April 13, 2012.

[26] A. Melnikov, Simple Authentication and Security Layer (SASL), IETF RFC 4422, https://tools.ietf.org/html/rfc4422

[27] European Telecommunications Standards Institute (ETSI), Network Functions Virtualization – An Introductory White Paper, October 22, 2012.