MILE
Internet-Draft
Intended status: Informational
Expires: November 18, 2015

C. Inacio
CMU
D. Miyamoto
UTokyo

May 17, 2015

MILE Implementation Report
draft-ietf-mile-implementreport-03

Abstract

This document is a collection of implementation reports from vendors, consortiums, and researchers who have implemented one or more of the standards published from the IETF INCident Handling (INCH) and Management Incident Lightweight Exchange (MILE) working groups.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 18, 2015.

Copyright Notice

Table of Contents

1. Introduction

This document is a collection of implementation reports from vendors and researchers who have implemented one or more of the standards published from the INCH and MILE working groups. The standards include:

- Incident Object Description Exchange Format (IODEF) v1, RFC5070,
- Incident Object Description Exchange Format (IODEF) v2, RFC5070-bis,
- Extensions to the IODEF-Document Class for Reporting Phishing, RFC5901

- Sharing Transaction Fraud Data, RFC5941
- IODEF-extension for Structured Cybersecurity Information, RFCXXXX
- Real-time Inter-network Defense (RID), RFC6545
- Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS, RFC6546.
- Incident Object Description Exchange Format (IODEF) Extension for Structured Cybersecurity Information, RFC7203

The implementation reports included in this document have been provided by the team or product responsible for the implementations of the mentioned RFCs. Additional submissions are welcome and should be sent to the draft editor. A more complete list of implementations, including open source efforts and vendor products, can also be found at the following location:

http://siis.realmv6.org/implementations/

2. Consortiums and Information Sharing and Analysis Centers (ISACs)

2.1. Anti-Phishing Working Group

Description of how IODEF is used will be provided in a future revision.

2.2. Advanced Cyber Defence Centre (ACDC)

Description of how IODEF is used will be provided in a future revision. http://www.botfree.eu/

3. Open Source Implementations

3.1. EMC/RSA RID Agent

The EMC/RSA RID agent is an open source implementation of the Internet Engineering Task Force (IETF) standards for the exchange of incident and indicator data. The code has been released under an MIT license and development will continue with the open source community at the Github site for RSA Intelligence Sharing:

https://github.com/RSAIntelShare/RID-Server.git

The code implements the RFC6545, Real-time Inter-network Defense (RID) and RFC6546, Transport of RID over HTTP/TLS protocol. The code supports the evolving RFC5070-bis Incident Object Description

Exchange Format (IODEF) data model from the work in the IETF working group Managed Incident Lightweight Exchange (MILE).

3.2. NICT IODEF-SCI implementation

Japan’s National Institute of Information and Communications Technology (NICT) Network Security Research Institute implemented open source tools for exchanging, accumulating, and locating IODEF-SCI documents.

Three tools are available in GitHub. They assist the exchange of IODEF-SCI documents between parties. IODEF-SCI is the IETF draft that extends IODEF so that IODEF document can embed structured cybersecurity information (SCI). For instance, it can embed MMDEF, CEE, MAEC in XML and CVE identifiers.

The three tools are generator, exchanger, and parser. The generator generates IODEF-SCI document or appends an XML to existing IODEF document. The exchanger sends the IODEF document to its correspondent node. The parser receives, parses, and stores the IODEF-SCI document. It also equips the interface that enable users to locate IODEF-SCI documents it has ever received. The code has been released under an MIT license and development will continue here.

Note that users can enjoy this software with their own responsibility.

Available Online:

https://github.com/TakeshiTakahashi/IODEF-SCI

3.3. n6

n6 is a platform for processing security-related information, developed by NASK, CERT Polska. Its API provides a common and unified way of representing data across the different sources that participate in knowledge management.

n6 exposes a REST-ful API over HTTPS with mandatory authentication via TLS client certificates, to ensure confidential and trustworthy communications. Moreover, it uses an event-based data model for representation of all types of security information.

Each event is represented as a JSON object with a set of mandatory and optional attributes. It also supports alternative output data formats for keeping compatibility with existing systems - IODEF and

CSV - although they lack some of the attributes that may be present in the native JSON format.

Available Online:

https://github.com/CERT-Polska/n6sdk

4. Vendor Implementations

4.1. Deep Secure

Deep-Secure Guards are built to protect a trusted domain from:

- releasing sensitive data that does not meet the organisational security policy
- applications receiving badly constructed or malicious data which could exploit a vulnerability (known or unknown)

Deep-Secure Guards support HTTPS and XMPP (optimised server to server protocol) transports. The Deep-Secure Guards support transfer of XML based business content by creating a schema to translate the known good content to and from the intermediate format. This means that the Deep-Secure Guards can be used to protect:

- IODEF/RID using the HTTPS transport binding (RFC 6546)
- IODEF/RID using an XMPP binding
- ROLIE using HTTPS transport binding (draft-field-mile-rolie-02)
- STIX/TAXII using the HTTPS transport binding

Deep-Secure Guards also support the SMTP transport and perform deep content inspection of content including XML attachments. The Mail Guard supports S/MIME and Deep Secure are working on support for the upcoming PLASMA standard which enables information centric policy enforcement of data.

4.2. IncMan Suite, DFLabs

The Incident Object Description Exchange Format, documented in the RFC 5070, defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents. IncMan Suite implements the IODEF standard for exchanging details about incidents, either for exporting and importing activities. This has been introduced to enhance the capabilities of the various CSIRT, to

facilitate collaboration and sharing of useful experiences, conveying awareness on specific cases.

The IODEF implementation is specified as an XML schema, therefore all data are stored in an xml file: in this file all data of an incident are organized in a hierarchical structure to describe the various objects and their relationships.

IncMan Suite relies on IODEF as a transport format, composed by various classes for describing the entities which are part of the incident description: for instance the various relevant timestamps (detect time , start time, end time, report time), the techniques used by the intruders to perpetrate the incident, the impact of the incident, either technical and non-technical (time and monetary) and obviously all systems involved in the incident.

4.2.1. Exporting Incidents

Each incident defined in IncMan Suite can be exported via a User Interface feature and it will populate an xml document. Due to the nature of the data processed, the IODEF extraction might be considered privacy sensitive by the parties exchanging the information or by those described by it. For this reason, specific care needs to be taken in ensuring the distribution to an appropriate audience or third party, either during the document exchange and subsequent processing.

The xml document generated will include description and details of the incident along with all the systems involved and the related information. At this stage it can be distributed for import into a remote system.

4.2.2. Importing Incidents

IncMan Suite provides a functionality to import incidents stored in files and transported via IODEF-compliant xml documents. The importing process comprises of two steps: firstly, the file is inspected to validate if well formed, then all data are uploaded inside the system.

If an incident is already existing in the system with the same incident id, the new one being imported will be created under a new id. This approach prevents from accidentally overwriting existing info or merging inconsistent data.

IncMan Suite includes also a feature to upload incidents from emails.

The incident, described in xml format, can be stored directly into the body of the email message or transported as an attachment of the email. At regular intervals, customizable by the user, IncMan Suite monitors for incoming emails, filtered by a configurable white-list and black-list mechanism on the sender's email account, then a parser processes the received email and a new incident is created automatically, after having validated the email body or the attachment to ensure it is a well formed format.

4.3. Surevine Proof of Concept

XMPP is enhanced and extended through the XMPP Extension Protocols (or XEPs). XEP-0268 (http://xmpp.org/extensions/xep-0268.html) describes incident management (using IODEF) of the XMPP network itself, effectively supporting self-healing the XMPP network. In order to more generically cover incident management of a network and over a network, XEP-0268 requires some updates. We are working on these changes together with a new XEP that supports "social networking" over XMPP, enhancing the publish-and-subscribe XEP (XEP-0060). This now allows nodes to publish any type of content and subscribe to and therefore receive the content. XEP-0268 will be used to describe IODEF content. We now have an alpha version of the server-side software and client-side software required to demonstrate the "social networking" capability and are currently enhancing this to support Cyber Incident management in real-time.

4.4. MANTIS Cyber-Intelligence Management Framework

MANTIS provides an example implementation of a framework for managing cyber threat intelligence expressed in standards such as STIX, CybOX, IODEF, etc. The aims of providing such an example implementation are:

- To aide discussions about emerging standards such as STIX, CybOX et al. with respect to questions regarding tooling: how would a certain aspect be implemented, how do changes affect an implementation? Such discussions become much easier and have a better basis if they can be lead in the context of example tooling that is known to the community.

- To lower the entrance barrier for organizations and teams (esp. CERT teams) in using emerging standards for cyber-threat intelligence management and exchange.

- To provide a platform on the basis of which research and community-driven development in the area of cyber-threat intelligence management can occur.

5. Vendors with Planned Support

5.1. Threat Central, HP

HP has developed HP Threat Central, a security intelligence platform that enables automated, real-time collaboration between organizations to combat today's increasingly sophisticated cyber attacks. One way automated sharing of threat indicators is achieved is through close integration with the HP ArcSight SIEM for automated upload and consumption of information from the Threat Central Server. In addition HP Threat Central supports open standards for sharing threat information so that participants who do not use HP Security Products can participate in the sharing ecosystem. General availability of Threat Central will be in 2014. It is planned that future versions also support IODEF for the automated upload and download of threat information.

6. Other Implementations

6.1. Collaborative Incident Management System

Collaborative Incident Management System (CIMS) is a proof-of-concept system for collaborative incident handling and for the sharing of cyber defence situational awareness information between the participants, developed for the Cyber Coalition 2013 (CC13) exercise organized by NATO. CIMS was implemented based on Request Tracker (RT), an open source software widely used for handling incident response by many CERTs and CSIRTs.

One of the functionality implemented in CIMS was the ability to import and export IODEF messages in the body of emails. The intent was to verify the suitability of IODEF to achieve the objective of collaborative incident handling. The customized version of RT could be configured to send an email message containing an IODEF message whenever an incident ticket was created, modified or deleted. These IODEF messages would then be imported into other incident handling systems in order to allow participating CSIRTs to use their usual means for incident handling, while still interacting with those using the proof-of-concept CIMS. Having an IODEF message generated for every change made to the incident information in RT (and for the system to allow incoming IODEF email messages to be associated to an existing incident) would in some way allow all participating CSIRTs to actually work on a "common incident ticket", at least at the conceptual level. Of particular importance was the ability for users to exchange information between each other concerning actions taken in the handling of a particular incident, thus creating a sort of common action log, as well as requesting/tasking others to provide information or perform specified action and correlating received

responses to the original request or tasking. As well, a specific "profile" was developed to identify a subset of the IODEF classes that would be used during the exercise, in an attempt to channel all users into a common usage pattern of the otherwise flexible IODEF standard.

6.2. Automated Incident Reporting - AirCERT

AirCERT was implemented by CERT/CC of Carnegie Mellon’s Software Engineering Institute CERT divison. AirCERT was designed to be an Internet-scalable distributed system for sharing security event data. The AirCERT system was designed to be an automated collector of flow and IDS alerts. AirCERT would collect that information into a relational database and be able to share reporting using IODEF and IDMEF. AirCERT additionally used SNML to exchange information about the network. AirCERT was implemented in a combination of C and perl modules and included periodic graphing capabilities leveraging RRDTool.

AirCERT was intended for large scale distributed deployment and eventually the ability to sanitize data to be shared across administrative domains. The architecture was desgined to allow collection of data at a per site basis and to allow each site to create data sharing based on its own particular trust relationships.

6.3. US Department of Energy CyberFed

The CyberFed system was implemented and deployed by Argonne National Laboratory to automate the detection and response of attack activity against Department of Energy (DoE) computer networks. CyberFed automates the collection of network alerting activity from various perimeter network defenses and logs those events into its database. CyberFed then automatically converts that information into blocking information transmitted to all participants. The original implementation used IODef messages wrapped in an XML extension to manage a large array of indicators. The CyberFed system was not designed to describe a particular incident as much as to describe a set of current network blocking indicators that can be generated and deployed machine-to-machine.

CyberFed is primarily implemented in Perl. Included as part of the CyberFed system are scripts which interact with a large number of firewalls, IDS/IPS devices, DNS systems, and proxies which operate to implement both the automated collection of events as well as the automated deployment of blacking.

Currently CyberFed supports multiple exchange formats including IODef and STIX. OpenIOC is also a potential exchange format that DoE is considering.

6.4. TrendMicro Sharing System

More information to come.

7. Implementation Guide

The section aims at sharing the tips for development of IODEF-capable systems.

7.1. Code Generators

For implementing IODEF-capable systems, it is feasible to employ code generators for XML Schema Document (XSD). The generators are used to save development costs since they automatically create useful libraries for accessing XML attributes, composing messages, and/or validating XML objects. The IODEF XSD was defined in section 8 of RFC 5070, and is availabe at http://www.iana.org/assignments/xml-registry/schema/iodef-1.0.xsd.

However, there still remains some problem. Due to the complexity of IODEF XSD, some code generators could not generate from the XSD file. The tested code generators were as follows.

- XML::Pastor [XSD:Perl] (Perl)
- RXSD [XSD:Ruby] (Ruby)
- PyXB [XSD:Python] (Python)
- JAXB [XSD:Java] (Java)
- CodeSynthesis XSD [XSD:Cxx] (C++)
- Xsd.exe [XSD:CS] (C#)

For instance, we have used XML::Pastor, but it could not properly understand its schema due to the complexity of IODEF XSD. The same applies to RXSD and JAXB. Only PyXB, CodeSynthesis XSD and Xsd.exe were able to understand the schema.

There is no recommended workaround, however, a double conversion of XSD file is one option to go through the situation; it means XSD is serialized to XML, and it is again converted to XSD. The resultant XSD was process-able by the all tools above.

It should be noted that IODEF uses ’-’ (hyphen) symbols in its classes or attributes, listed as follows.

- IODEF-Document Class; it is the top level class in the IODEF data model described in section 3.1 of [RFC5070].

- The vlan-name and vlan-num Attribute; according to section 3.16.2 of [RFC5070], they are the name and number of Virtual LAN and are the attributes for Address class.

- Extending the Enumerated Values of Attribute; according to section 5.1 of [RFC5070], it is a extension techniques to add new enumerated values to an attribute, and has a prefix of "ext-", e.g., ext-value, ext-category, ext-type, and so on.

According to the language specification, many programing language prohibit to contain ’-’ symbols in the name of class. The code generators must replace or remove ’-’ when building the librarlies. They should have the name space to restore ’-’ when outputting the XML along with IODEF XSD.

7.2. iodeflib

iodeflib is an open source implementation written in Python. This provides a simple but powerful APIs to create, parse and edit IODEF documents. It was designed in order to keep its interface as simple as possible, whereas generated libraries tend to inherit the complexity of IODEF XSD. As well as the interface, iodeflib involves functions of hiding some unnecessarily nested structures of the IODEF schema, and adding more convenient shortcuts.

This tool is available through the following link:

http://www.decalage.info/python/iodeflib

7.3. iodefpm

IODEF.pm is an open source implementation written in Perl. This also provides a simple interface for creating and parsing IODEF documents, in order to facilitate the translation of the a key-value based format to the IODEF representation. The module contains a generic XML DTD parser and includes a simplified node based representation of the IODEF DTD. It can hence easily be upgraded or extended to support new XML nodes or other DTDs.

This tool is available through the following link:

http://search.cpan.org/~saxjazman/

7.4. Usability

Here notes some tips to avoid problems.

- IODEF has category attribute for NodeRole class. Though various categories are described, they are not enough. For example, in the case of web mail servers, you should choose either "www" or "mail". One suggestion is selecting "mail" as the category attribute and adding "www" for another attirbute.

- The numbering of Incident ID needs to be considered. Otherwise, information, such as the number of incidents within certain period could be observed by document receivers. For instance, we could randomize the assignment of the numbers.

8. Acknowledgements

The MILE Implementation report has been compiled through the submissions of implementers of INCH and MILE working group standards. A special note of thanks to the following contributors:

John Atherton, Surevine

Humphrey Browning, Deep-Secure

Dario Forte, DFLabs

Tomas Sander, HP

Ulrich Seldeslachts, ACDC

Takeshi Takahashi, National Institute of Information and Communications Technology Network Security Research Institute

Kathleen Moriarty, EMC

Bernd Grobauer, Siemens

Dandurand Luc, NATO

Pawel Pawlinski, NASK

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

This draft provides a summary of implementation reports from researchers and vendors who have implemented RFCs and drafts from the MILE and INCH working groups. There are no security considerations added in this draft because of the nature of the document.

11. Informative References

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.

[RFC5901] Cain, P. and D. Jevans, "Extensions to the IODEF-Document Class for Reporting Phishing", RFC 5901, July 2010.

[RFC5941] M'Raihi, D., Boeyen, S., Grandcolas, M., and S. Bajaj, "Sharing Transaction Fraud Data", RFC 5941, August 2010.

[RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)", RFC 6545, April 2012.

[RFC6546] Trammell, B., "Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS", RFC 6546, April 2012.

[XSD:CS] Microsoft, "XML Schema Definition Tool (Xsd.exe)", <http://www.microsoft.com/>.

[XSD:Cxx] CodeSynthesis, "XSD - XML Data Binding for C++", <http://www.codesynthesis.com/>.

[XSD:Java] Project Kenai, "JAXB Reference Implementation", <https://jaxb.java.net/>.

[XSD:Perl] Ulsoy, A., "XML::Pastor", <http://search.cpan.org/~aulusoy/XML-Pastor-1.0.4/>.

[XSD:Python] Bigot, P., "PyXB: Python XML Schema Bindings", <https://pypi.python.org/pypi/PyXB>.

[XSD:Ruby] Morsi, M., "RXSD - XSD / Ruby Translator", <https://github.com/movitto/RXSD>.

Authors' Addresses

Chris Inacio
Carnegie Mellon University
4500 5th Ave., SEI 4108
Pittsburgh, PA 15213
US

Email: inacio@andrew.cmu.edu

Daisuke Miyamoto
The Univerisity of Tokyo
2-11-16 Yayoi, Bunkyo
Tokyo 113-8658
JP

Email: daisu-mi@nc.u-tokyo.ac.jp