MILE Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2017

P. Kampanakis
Cisco Systems
M. Suzuki
NICT

July 8, 2016

IODEF Usage Guidance
draft-ietf-mile-iodef-guidance-06

Abstract

The Incident Object Description Exchange Format v2 [I-D.ietf-mile-rfc5070-bis] defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents. Since the IODEF model includes a wealth of available options that can be used to describe a security incident or issue, it can be challenging for security practicioners to develop tools that can leverage IODEF for incident sharing. This document provides guidelines for IODEF practicioners. It also addresses how common security indicators can be represented in IODEF and use-cases of how IODEF is being used so far. The goal of this document is to make IODEF's adoption by vendors easier and encourage faster and wider adoption of the model by Computer Security Incident Response Teams (CSIRTs) around the world.

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 January 9, 2017.

Table of Contents

1. Introduction

The Incident Object Description Exchange Format v2 in [I-D.ietf-mile-rfc5070-bis] defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents. The IODEF data model consists of multiple classes and data types that are defined in the IODEF XML schema.

The IODEF schema was designed to be able to describe all the possible fields that would be needed in a security incident exchange. Thus, IODEF contains plenty data constructs that could potentially make it harder for IODEF implementers to decide which are the most important ones to use. Additionally, in the IODEF schema, there exist multiple fields and classes which do not necessarily need to be used in every possible data exchange. Moreover, there are fields that are useful only in data exchanges of non-traditional security events. This document tries to address these issues. It also addresses how common security indicators can be represented in IODEF. It points out the most important IODEF classes for an implementer and describe other ones that are not as important. Also, it presents some common challenges for IODEF implementers and how to address them. The end goal of this document is to make IODEF's adoption by vendors easier and encourage faster and wider adoption of the model by Computer Security Incident Response Teams (CSIRTs) around the world.

Section 3 discusses the recommended classes and how an IODEF implementer should chose the classes to implement. Section 4 presents common considerations a practicioner will come across and how to address them. Section 5 goes over some common uses of IODEF.

2. Terminology

The terminology used in this document follows the one defined in [RFC5070] and [RFC7203].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

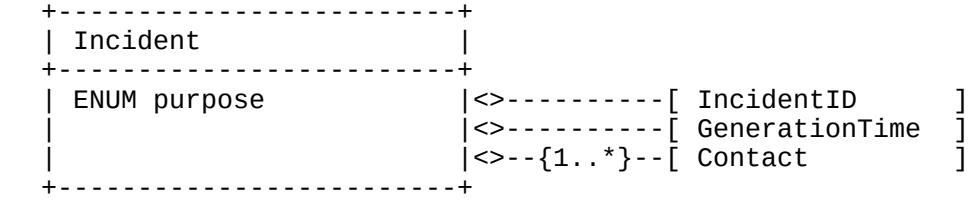
3. Implementation Strategy

It is important for IODEF practicioners to be able to distinguish how the IODEF classes will be used in incident information exchanges. It is critical to follow a strategy according to which of the various IODEF classes will be implemented. It is also important to know the most common classes that will be used to describe common security incidents or indicators. Thus, this section will describe the most

important classes and factors an IODEF implementer should take into consideration before designing the implementation or tool.

3.1. Minimal IODEF document

An IODEF document MUST include at least an Incident class and a version attribute. An Incident MUST contain three minimal mandatory-to-implement classes. An Incident class needs to have a Generation time and at least one Contact and IncidentID class. The structure of the minimal-style Incident class follows below.



Minimal-style Incident class

This minimal Incident class needs to include a purpose attribute and the IncidentID, GenerationTime, and Contact elements.

The Contact class requires the type and role attributes, but no elements are required by the IODEF v2 specification. Nevertheless, at least one of the elements in the Contact class, such as Email class, need to be implemented so that the IODEF document can be practical.

Implementers can refer to Appendix A and Section 7 of [I-D.ietf-mile-rfc5070-bis] for example IODEF and IODEF v2 documents respectively.

3.2. Decide what IODEF will be used for

There is no need for an practicioner to implement IODEF classes and fields other than the minimal ones (Section 3.1) and the ones that are necessary for his use-cases. The implementer SHOULD carefuly look into the schema and decide classes to implement (or not).

For example, if we have has DDoS as a potential use-case, then the Flow class and its included information are the most important classes to use. The Flow class describes information related to the attacker hosts and victim hosts, which information may help automated filtering or sink-hole operations.

Another potential use-case is malware command and control. After modern malware infects a device, it usually proceeds to connect to one or more command and control (c2) servers to receive instructions from its master and potentially exfiltrate information. To protect against such activity, it is important to interrupt the c2 communication by filtering the activity. IODEF can describe such activities using the Flow and the ServiceName classes.

For use-cases where indicators need to be described more than events themselves, the IndicatorData class and the necessary included in it classes will be implemented instead of the EventData class and its classes.

In summary, an implementer SHOULD identify the use-cases and find the classes that are necessary to support in IODEF v2. Implementing and parsing all IODEF classes can be cumbersome in some occasions and is not always necessary. Other external schemata can also be used in IODEF to describe incidents or indicators which should be treated accordingly only if the implementer’s IODEF use-cases require external schema support.

3.3. Indicators vs Events

[I-D.ietf-mile-rfc5070-bis] contains classes that can describe attack Methods, Events, Indicents, how they were discovered and the Assessment of the reprecussions of the incident to the victim. It is important for implementers to know the distinction between these classes in order to decide which ones fullfulls their use-cases.

An IndicatorData class depicts a threat indicator or observable that could be used to describe a threat that does not necessarily mean that an exploit happened. For example, we could see an attack happening but it might have been prevented and not have resulted in an incident or security event. On the other hand an EventData class usually describes a security event and can be considered as a incident report of something that took place.

Classes like Discovery, Assessment, Method, RecoveryTime are used in conjuction with EventData as they related to the incident report described in the EventData. The RelatedActivity class can reference an incident, an indicator or other related threat activity.

While deciding what classes are important for the needed use-cases, IODEF users SHOULD carefuly evaluate the necessary classes and how these are used in order to avoid unecessary work. For example, if we want to only describe indicators in IODEF, the implementation of Method or Assessment might not be important.

4. IODEF considerations and how to address them

4.1. External References

The IODEF format includes the Reference class that refers to externaly defined information such as a vulnerability, Intrusion Detection System (IDS) alert, malware sample, advisory, or attack technique. To facilitate the exchange of information, the Reference class was extended to the Enumeration Reference Format [RFC7495]. The Enumeration Reference Format specifies a format to include enumeration values from external data representations into IODEF like CVE, and manages references to external representations using IANA registry. Practicioners SHOULD only support external enumerations that are expected to be used in IODEF documents for their use-cases.

4.2. Extensions

The IODEF data model ([RFC5070]) is extensible. Many class attributes and their values can be extended using using the "ext-*" prefix. Additional classed can also be defined by using the AdditionalData and RecordItem classes. An extension to the AdditionalData class for reporting Phishing emails is defined in [RFC5901].

Additionally, IODEF can import existing schemata by using an extension framework defined in [RFC7203]. The framework enables IODEF users to embed XML data inside an IODEF document using external schemata or structures defined by external specifications. Examples include CVE, CVRF and OVAL. Thus, [RFC7203] enhances the IODEF capabilities without further extending the data model.

IODEF practicioners can consider using their own IODEF extensions only for data that cannot be described using existing standards or importing them in and IODEF document using [RFC7203] is not a suitable option.

Information about extending IODEF classes attributes and enumarated values can be found in Section 5 of [I-D.ietf-mile-rfc5070-bis].

4.3. Indicator predicate logic

An IODEF [I-D.ietf-mile-rfc5070-bis] document can describe incident reports and indicators. The Indicator class can include references to other indicators, observables and more classes the contain details about the indicator. When describing security indicators, it is often common to need to group them together in order to form a group of indicator that constitute a security threat. For example, a botnet might have multiple command and control servers. For that

reason, IODEF v2 introduced the IndicatorExpression class that is used to add the indicator predicate logic when grouping more than one indicators or observables.

It is important for implementers to be able to parse and apply the boolean logic offered by an IndicatorExpression in order to evaluate the existance of an indicator. As explained in Section 3.29.5 of [I-D.ietf-mile-rfc5070-bis] the IndicatorExpression element operator defines the operator applied to all the child element of the IndicatorExpression. If no operator is defined "and" SHOULD be assumed. IndicatorExpressions can also be nested together. Child IndicatorExpressions should be treated as child elements of their parent and they SHOULD be evaluated first before evaluated with the operator of their parent.

In the following example the EventData class evaluates as a Flow of one System with source address being (10.10.10.104 OR 10.10.10.106) AND target address 10.1.1.1

```
<!-- ...XML code omitted... -->
 <IndicatorData>
   <Indicator>
     <IndicatorID name="csirt.example.com" version="1">
     G90823490
     </IndicatorID>
     <Description>C2 domains</Description>
     <IndicatorExpression operator="and">
       <IndicatorExpression operator="or">
         <Observable>
           <System category="source" spoofed="no">
             <Node>
               <Address category="ipv4-addr">
                 10.10.10.104
               </Address>
             </Node>
           </System>
         </Observable>
         <Observable>
           <System category="source" spoofed="no">
             <Node>
               <Address category="ipv4-addr">
                 10.10.10.106
               </Address>
             </Node>
           </System>
         </Observable>
       </IndicatorExpression>
       <Observable>
         <System category="target" spoofed="no">
           <Node>
             <Address category="ipv4-addr">
               10.1.1.1
             </Address>
           </Node>
         </System>
       </Observable>
     </IndicatorExpression>
   </Indicator>
 </IndicatorData>
<!-- ...XML code omitted... -->
```

Similarly, the FileData Class can be an observable in an IndicatorExpression. The hash values of two files can be used to match against an indicator using boolean "or" logic. In the following example the indicator consists of either of the two files with two different hashes.

```
<!-- ...XML code omitted... -->
 <IndicatorData>
   <Indicator>
     <IndicatorID name="csirt.example.com" version="1">
     A4399IWQ
     </IndicatorID>
     <Description>File hash watchlist</Description>
     <IndicatorExpression operator="or">
         <Observable>
           <FileData>
             <File>
               <FileName>dummy.txt</FileName>
               <HashData>
                 <Hash>
                  <ds:DigestMethod Algorithm=
                  "http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>
                   141accec23e7e5157de60853cb1e01bc38042d
                   08f9086040815300b7fe75c184
                  </ds:DigestValue>
                 </Hash>
               </HashData>
             </File>
           </FileData>
         </Observable>
         <Observable>
           <FileData>
             <File>
               <FileName>dummy2.txt</FileName>
               <HashData>
                 <Hash>
                  <ds:DigestMethod Algorithm=
                  "http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>
                   141accec23e7e5157de60853cb1e01bc38042d
                   08f9086040815300b7fe75c184
                  </ds:DigestValue>
                 </Hash>
               </HashData>
             </File>
           </FileData>
         </Observable>
     </IndicatorExpression>
   </Indicator>
 </IndicatorData>
<!-- ...XML code omitted... -->
```

4.4. Disclosure level of IODEF

The information conveyed in IODEF documents SHOULD be treated carefully since the content may be confidential. IODEF provides a disclosure level indicator, but its enforcement depends on operations at the practicioner's side.

IODEF has a common attribute, called "restriction", which indicates the disclosure guideline to which the sender expects the recipient to adhere to for the information represented in the class and its children. That way, the sender can express the level of disclosure for each component of an IODEF document. Appropriate external measures could be implemented based on the restriction level. One example is when RID is used to transfer the IODEF documents, it can provide policy guidelines for handling IODEF documents by using the RIDPolicy class.

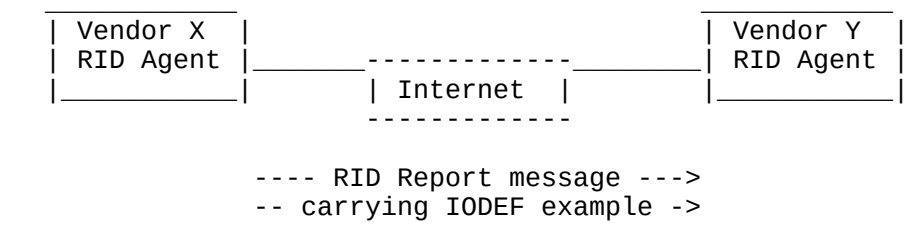
The enforcement of the disclosure guidelines goes beyond IODEF. The recipient of the IODEF document needs to follow the guidelines, but these guidelines themselves do not provide any enforcement measures. For that purpose, practicioners SHOULD consider appropriate measures, technical or operational.

5. Current uses of IODEF

IODEF is currently used by various organizations in order to represent security incidents and share incident and threat information between security operations organizations.

5.1. Inter-vendor and Service Provider Exercise

Various vendors organized and executed an exercise where multiple threat indicators were exchanged using IODEF. The transport protocol used was RID. The threat information shared included incidents like DDoS attacks. Malware and Spear-Phishing. As this was a proof-of-concept (PoC) exercise only example information (no real threats) were shared as part of the exchanges.



PoC peering topology

The figure above shows how RID interactions took place during the PoC. Participating organizations were running RID Agent software on-premises. The RID Agents formed peering relationships with other participating organizations. When Entity X had a new incident to exchange it would package it in IODEF and send it to Entity Y over TLS in a RID Report message. In case there was an issue with the message, Entity Y would send an RID Acknowledgement message back to Entity X which included an application level message to describe the issue. Interoperability between RID agents and the standards, [RFC6545] and [RFC6546], was also proven in this exercise. Appendix A includes some of the incident IODEF example information that was exchanged by the organizations' RID Agents as part of this proof-of-concept.

The first use-case included sharing of Malware Data Related to an Incident between CSIRTs. After Entity X detected an incident, she would put data about malware found during the incident in a backend system. Entity X then decided to share the incident information with Entity Y about the malware discovered. This could be a human decision or part of an automated process.

Below are the steps followed for the malware information exchange that was taking place:

(1) Entity X has a sharing agreement with Entity Y, and has already been configured with the IP address of Entity Y's RID Agent

(2) Entity X's RID Agent connects to Entity Y's RID Agent, and mutual authentication occurs using PKI certificates.

(3) Entity X pushes out a RID Report message which contains information about N pieces of discovered malware. IODEF is used in RID to discribe the

(a) Hash of malware files

(b) Registry settings changed by the malware

(c) C&C Information for the malware

(4) Entity Y receives RID Report message, sends RID Acknowledgement message

(5) Entity Y stores the data in a format that makes it possible for the back end to know which source the data came from.

Another use-case was sharing Distributed Denial of Service (DDoS) as presented below information: Entity X, a Critical Infrastructure and Key Resource (CIKR) company detects that their internet connection is saturated with an abnormal amount of traffic. Further investigation determines that this is an actual DDoS attack. Entity X's computer incident response team (CIRT) contacts their ISP and shares information with them about the attack traffic characteristics. In addition, Entity X has an information sharing relationship with Entity Y. It shares information with Entity Y on characteristics of the attack to watch for. Entitty X's ISP is being overwhelmed by the amount of traffic, so it shares attack signatures and IP addresses of the most prolific hosts with its adjacent ISPs.

Below are the steps followed for a DDoS information exchange:

(1) Entity X has a sharing agreement with Entity Y, and has already been configured with the IP address of Entity Y's RID Agent

(2) Entity X's RID Agent connects to Entity Y's RID Agent, and mutual authentication occurs using PKI certificates.

(3) Entity X pushes out a RID Report message which contains information about the DDoS attack. IODEF is used in RID to discribe the

(a) Start and Detect dates and times

(b) IP Addresses of nodes sending DDoS Traffic

(c) Sharing and Use Restrictions

(d) Traffic characteristics (protocols and ports)

(e) HTTP User-Agents used

(f) IP Addresses of C&C for a botnet

(4) Entity Y receives RID Report message, sends RID Acknowledgement message

(5) Entity Y stores the data in a format that makes it possible for the back end to know which source the data came from.

One more use-case was sharing spear-phishing email information as explained in the following scenario: The board members of several defense contractors receive an email inviting them to attend a conference in San Francisco. The board members are asked to provide their personally identifiable information such as their home address, phone number, corporate email, etc in an attached document which came with the email. The board members were also asked to click on a URL which would allow them to reach the sign up page for the conference. One of the recipients believes the email to be a phishing attempt and forwards the email to their corporate CSIRT for analysis. The CSIRT identifies the email as an attempted spear phishing incident and distributes the indicators to their sharing partners.

Below are the steps followed for a spear-phishing information exchange between CSIRTs that was part of this PoC.

(1) Entity X has a sharing agreement with Entity Y, and has already been configured with the IP address of Entity Y’s RID Agent

(2) Entity X’s RID Agent connects to Entity Y’s RID Agent, and mutual authentication occurs using PKI certificates.

(3) Entity X pushes out a RID Report message which contains information about the spear-phishing email. IODEF is used in RID to discribe the

   (a) Attachment details (file Name, hash, size, malware family

   (b) Target description (IP, domain, NSLookup)

   (c) Email information (From, Subject, header information, date/ time, digital signature)

   (d) Confidence Score

(4) Entity Y receives RID Report message, sends RID Acknowledgement message

(5) Entity Y stores the data in a format that makes it possible for the back end to know which source the data came from.

5.2. Implementations

In order to use IODEF, some tools that cope with IODEF documents, such as the IODEF parser, are needed. Though arbitrary implementations can be done, some guidelines are provided in [I-D.ietf-mile-implementreport]. IODEF , but [I-D.ietf-mile-implementreport] provides guidelines for implementers. The document does not specify any specific MTI but provides a list of implementations the authors have surveyed at the time of its publication as well as some tips on the implementations. Implementers are encourage to read the draft.

5.3. Other

IODEF is also used in various projects and products to consume and share security information. Various vendor incident reporting products have the ability to consume and export in IODEF format [implementations]. Perl and Python modules (XML::IODEF, Iodef::Pb, iodeflib) exist in order to parse IODEF documents and their extensions. Additionally, some worldwide CERT organizations are already able to use receive incident information in IODEF.

Future use-cases of IODEF could be:

(1) ISP notifying a national CERT or organization when it identifies and acts upon an incident and CERTs notifying ISPs when they are aware of incidents.

(2) Suspected phishing emails could be shared amongst organizations and national agencies. Automation could validate web content that the suspicious emails are pointing to. Identified malicious content linked in a phishing email could then be shared using IODEF. Phishing campaigns could thus be subverted much faster by automating information sharing using IODEF.

(3) When finding a certificate that should be revoked, a thrid-party would forward an automated IODEF message to the CA with the full context of the certificate and the CA could act accordingly after checking its validity. Alternatively, in the event of a compromise of the private key of a certificate, a third-party could alert the certificate owner about the compromise using IODEF.

6. Updates

version -06 updates:

(1) Updated wording in various sections to make content clearer.

(2) Updated Predicate Logic section to reflect the latest IndicatorExpression logic in iodef-bis.

(3) Updated section to describe the difference between events and indicators and their use in IODEF v2.

version -05 updates:

(1) Changed section title from "Restrictions in IODEF" to "Disclosure level of IODEF" and added some description

(2) Mixed "Recommended classes to implement" section with "Unnecessary Fields" section into "Minimal IODEF document" section

(3) Added description to "Decide what IODEF will be used for" section, "Implementations" section, and "Security Considerations" section

version -04 updates:

(1) Expanded on the Extensions section using Take’s suggestion.

(2) Moved Future use-cases under the Other section.

(3) CIF and APWG were consolidated in one "Implementation" section

(4) Added abstract of RFC7495 to the "External References" section

(5) Added Kathleen’s example of malware delivery URL to "Appendix"

(6) Added a little description to "Recommended classes to implement" section

version -03 updates:

(1) Added "Updates" section.

(2) Added details about the flow of information exchanges in "Inter-vendor and Service Provider Exercise" section. Also updated the usecases with more background information.

(3) Added future use-cases in the "Collective Intelligence Framework" section

(4) Updated Perl and Python references with the actual module names. Added IODEF implementation reference "implementations".

(5) Added Predicate logic section

(6) Updated Logic of watchlist of indicators section to simplify the logic and include examples.

(7) Renamed Externally defined indicators section to Indicator reference and elaborated on the use of indicator-uid and indicator-set-uid attribute use.

version -02 updates:

(1) Updated the "Logic for watchlist of indications" section to clarify the logic based on community feedback.

(2) Added "Inter-vendor and Service Provider Exercise" section.

(3) Added Appendix to include actual use-case IODEF examples.

7. Acknowledgements

8. Security Considerations

This document does not incur any new security issues, since it only talks about the usage of IODEF, which is defined in RFC 5070 [RFC5070]. Nevertheless, readers of this document SHOULD refer to the security consideration section of RFC5070 and [I-D.ietf-mile-rfc5070-bis].

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, DOI 10.17487/RFC5070, December 2007, <http://www.rfc-editor.org/info/rfc5070>.

[RFC5901] Cain, P. and D. Jevans, "Extensions to the IODEF-Document Class for Reporting Phishing", RFC 5901, DOI 10.17487/RFC5901, July 2010, <http://www.rfc-editor.org/info/rfc5901>.

[RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)", RFC 6545, DOI 10.17487/RFC6545, April 2012, <http://www.rfc-editor.org/info/rfc6545>.

[RFC6546] Trammell, B., "Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS", RFC 6546, DOI 10.17487/RFC6546, April 2012, <http://www.rfc-editor.org/info/rfc6546>.

[RFC7203] Takahashi, T., Landfield, K., and Y. Kadobayashi, "An Incident Object Description Exchange Format (IODEF) Extension for Structured Cybersecurity Information", RFC 7203, DOI 10.17487/RFC7203, April 2014, <http://www.rfc-editor.org/info/rfc7203>.

[RFC7495] Montville, A. and D. Black, "Enumeration Reference Format for the Incident Object Description Exchange Format (IODEF)", RFC 7495, DOI 10.17487/RFC7495, March 2015, <http://www.rfc-editor.org/info/rfc7495>.

9.2. Informative References

[APWG] "APWG", <http://apwg.org/>.

[CIF] "CIF", <http://csirtgadgets.org/collective-intelligence-framework/>.

[I-D.ietf-mile-implementreport] Inacio, C. and d. daisu-mi@nc.u-tokyo.ac.jp, "MILE Implementation Report", draft-ietf-mile-implementreport-06 (work in progress), October 2015.

[I-D.ietf-mile-rfc5070-bis] Danyliw, R., "The Incident Object Description Exchange Format v2", draft-ietf-mile-rfc5070-bis-18 (work in progress), March 2016.

[implementations] "Implementations on IODEF", <http://siis.realmv6.org/implementations/>.

Appendix A. Inter-vendor and Service Provider Exercise Examples

Below some of the incident IODEF example information that was exchanged by the vendors as part of this proof-of-concept Inter-vendor and Service Provider Exercise.

A.1. Malware

In this test, malware information was exchanged using RID and IODEF. The information included file hashes, registry setting changes and the C&C servers the malware uses.

```
<?xml version="1.0" encoding="UTF-8"?>
            <iodef:IODEF-Document xmlns:ds="
                http://www.w3.org/2000/09/xmldsig#"
                xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.41">
      <iodef:Incident purpose="reporting">
        <iodef:ReportID name="EXAMPLE CSIRT">
           189234
        </iodef:ReportID>
        <iodef:ReportTime>
           2013-03-07T16:14:56.757+05:30
        </iodef:ReportTime>
        <iodef:Description>
           Malware and related indicators identified
        </iodef:Description>
        <iodef:Assessment occurrence="potential">
          <iodef:Impact severity="medium" type="info-leak">
             Malware with Command and Control Server
             and System Changes
          </iodef:Impact>
        </iodef:Assessment>
        <iodef:Contact role="creator" type="organization">
          <iodef:ContactName>EXAMPLE CSIRT</iodef:ContactName>
          <iodef:Email>emccirt@emc.com</iodef:Email>
        </iodef:Contact>
        <iodef:EventData>
          <iodef:Method>
            <iodef:Reference>
              <iodef:ReferenceName>Zeus</iodef:ReferenceName>
              <iodef:URL>
                  http://www.threatexpert.com/report.aspx?
                  md5=e2710ceb088dacdcb03678db250742b7
              </iodef:URL>
            </iodef:Reference>
          </iodef:Method>
          <iodef:Flow>
            <iodef:System category="watchlist-source">
              <iodef:Node>
                <iodef:Address category="ipv4-addr">
                   192.168.2.200
                </iodef:Address>
                <iodef:Address category="site-uri">
                   http://zeus.556677889900.com/log-bin/
```

```
                        lunch_install.php?aff_id=1&amp;amp;
                        lunch_id=1&amp;amp;maddr=&amp;amp;
                        action=install
                    </iodef:Address>
                    <iodef:NodeRole attacktype="c2-server"/>
                  </iodef:Node>
                </iodef:System>
              </iodef:Flow>
              <iodef:Record>
                <iodef:RecordData>
                  <iodef:HashData>
                    <ds:Reference>
                      <ds:DigestMethod Algorithm="
                         http://www.w3.org/2001/04/xmlenc#sha1"/>
                      <ds:DigestValue>
                           MHg2NzUxQTI1MzQ4M0E2N0Q4NkUwRjg0NzYwRj
                           YxRjEwQkJDQzJFREZG</ds:DigestValue>
                    </ds:Reference>
                  </iodef:HashData>
                  <iodef:HashData>
                    <ds:Reference>
                      <ds:DigestMethod Algorithm="
                          http://www.w3.org/2001/04/xmlenc#md5"/>
                      <ds:DigestValue>
                          MHgyRTg4ODA5ODBENjI0NDdFOTc5MEFGQTg5NTE
                          zRjBBNA==
                      </ds:DigestValue>
                    </ds:Reference>
                  </iodef:HashData>
                  <iodef:WindowsRegistryKeysModified>
                    <iodef:Key registryaction="add_value">
                      <iodef:KeyName>
                          HKLM\Software\Microsoft\Windows\
                          CurrentVersion\Run\tamg
                      </iodef:KeyName>
                      <iodef:Value>
                         ?\?\?%System%\wins\mc.exe\?\??
                      </iodef:Value>
                    </iodef:Key>
                    <iodef:Key registryaction="modify_value">
                      <iodef:KeyName>HKLM\Software\Microsoft\
                         Windows\CurrentVersion\Run\dqo
                      </iodef:KeyName>
                      <iodef:Value>"\"\"%Windir%\Resources\
                          Themes\Luna\km.exe\?\?"
                      </iodef:Value>
                    </iodef:Key>
                  </iodef:WindowsRegistryKeysModified>
```

```
          </iodef:RecordData>
        </iodef:Record>
      </iodef:EventData>
      <iodef:EventData>
        <iodef:Method>
          <iodef:Reference>
            <iodef:ReferenceName>Cridex</iodef:ReferenceName>
            <iodef:URL>
                http://www.threatexpert.com/report.aspx?
                md5=c3c528c939f9b176c883ae0ce5df0001
            </iodef:URL>
          </iodef:Reference>
        </iodef:Method>
        <iodef:Flow>
          <iodef:System category="watchlist-source">
            <iodef:Node>
              <iodef:Address category="ipv4-addr">
                 10.10.199.100
              </iodef:Address>
              <iodef:NodeRole attacktype="c2-server"/>
            </iodef:Node>
            <iodef:Service ip_protocol="6">
              <iodef:Port>8080</iodef:Port>
            </iodef:Service>
          </iodef:System>
        </iodef:Flow>
        <iodef:Record>
          <iodef:RecordData>
            <iodef:HashData>
              <ds:Reference>
                <ds:DigestMethod Algorithm="
                   http://www.w3.org/2001/04/xmlenc#sha1"/>
                <ds:DigestValue>
                   MHg3MjYzRkUwRDNBMDk1RDU5QzhFMEM4OTVBOUM
                   1ODVFMzQzRTcxNDFD
                </ds:DigestValue>
              </ds:Reference>
              <ds:Reference>
                <ds:DigestMethod Algorithm="
                   http://www.w3.org/2001/04/xmlenc#md5"/>
                <ds:DigestValue>MHg0M0NEODUwRkNEQURFNDMzMEE1
                   QkVBNkYxNkVFOTcxQw==</ds:DigestValue>
              </ds:Reference>
            </iodef:HashData>
            <iodef:HashData>
              <ds:Reference>
                <ds:DigestMethod Algorithm="
                    http://www.w3.org/2001/04/xmlenc#md5"/>
```

```
                    <ds:DigestValue>MHg0M0NEODUwRkNEQURFNDMzMEE
                        1QkVBNkYxNkVFOTcxQw==</ds:DigestValue>
                  </ds:Reference>
                  <ds:Reference>
                    <ds:DigestMethod Algorithm="http://www.w3.org/
                        2001/04/xmlenc#sha1"/>
                    <ds:DigestValue>MHg3MjYzRkUwRDNBMDk1RDU5QzhFME
                        M4OTVBOUM1ODVFMzQzRTcxNDFD</ds:DigestValue>
                  </ds:Reference>
                </iodef:HashData>
                <iodef:WindowsRegistryKeysModified>
                  <iodef:Key registryaction="add_value">
                    <iodef:KeyName>
                       HKLM\Software\Microsoft\Windows\
                       CurrentVersion\Run\KB00121600.exe
                    </iodef:KeyName>
                    <iodef:Value>
                       \?\?%AppData%\KB00121600.exe\?\?
                    </iodef:Value>
                  </iodef:Key>
                </iodef:WindowsRegistryKeysModified>
              </iodef:RecordData>
            </iodef:Record>
          </iodef:EventData>
          <iodef:EventData>
            <iodef:Expectation action="other"/>
              <iodef:Flow>
                <iodef:System category="source"
                      indicator-set-id="91011">
                  <iodef:Node>
                      <iodef:Address category="url"
                         indicator-uid="qrst">
                            http://foo.com:12345/evil/cc.php
                      </iodef:Address>
                  <iodef:NodeName indicator-uid="rstu">
                      evil.com
                  </iodef:NodeName>
                      <iodef:Address category="ipv4-addr"
                         indicator-uid="stuv">
                            1.2.3.4</iodef:Address>
                      <iodef:Address category="ipv4-addr"
                         indicator-uid="tuvw">
                            5.6.7.8 </iodef:Address>
                      <iodef:Address category="ipv6-addr"
                         indicator-uid="uvwx">
                             2001:dead:beef::</iodef:Address>
                   <iodef:NodeRole category="c2-server"/>
                   </iodef:Node>
```

```
      </iodef:System>
    </iodef:Flow>
    <iodef:Record>
      <iodef:RecordData indicator-set-id="91011">
        <iodef:HashData>
           <ds:Reference>
              <ds:DigestMethod Algorithm=
                  "http://www.w3.org/2001/04/xmlenc
                   #sha256"/>
               <ds:DigestValue>
                141accec23e7e5157de60853cb1e01bc3804
                2d08f9086040815300b7fe75c184
               </ds:DigestValue>
           </ds:Reference>
        </iodef:HashData>
        <iodef:WindowsRegistryKeysModified
          indicator-set-id="91011">
          <iodef:Key registryaction="add_key"
          indicator-uid="vwxy">
            <iodef:KeyName>
                HKLM\SYSTEM\CurrentControlSet\
                Services\.Net CLR
            </iodef:KeyName>
          </iodef:Key>
          <iodef:Key registryaction="add_key"
            indicator-uid="wxyz">
            <iodef:KeyName>
                HKLM\SYSTEM\CurrentControlSet\
                Services\.Net CLR\Parameters
            </iodef:KeyName>
            <iodef:Value>
                \"\"%AppData%\KB00121600.exe\"\"
            </iodef:Value>
          </iodef:Key>
          <iodef:Key registryaction="add_value"
            indicator-uid="xyza">
            <iodef:KeyName>
               HKLM\SYSTEM\CurrentControlSet\Services\
               .Net CLR\Parameters\ServiceDll
            </iodef:KeyName>
            <iodef:Value>C:\bad.exe</iodef:Value>
          </iodef:Key>
          <iodef:Key registryaction="modify_value"
            indicator-uid="zabc">
            <iodef:KeyName>
                HKLM\SYSTEM\CurrentControlSet\
                Services\.Net CLR\Parameters\Bar
            </iodef:KeyName>
```

```
                        <iodef:Value>Baz</iodef:Value>
                      </iodef:Key>
                    </iodef:WindowsRegistryKeysModified>
                  </iodef:RecordData>
                </iodef:Record>
              </iodef:EventData>
            </iodef:Incident>
          </iodef:IODEF-Document>
```

A.2. Malware Delivery URL

This example indicates malware and related URL for file delivery.

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="2.00"
        xmlns="urn:ietf:params:xml:ns:iodef-2.0"
        xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<iodef:Incident purpose="reporting">
      <iodef:IncidentID name="csirt.example.com">
          189801
      </iodef:IncidentID>
      <iodef:RelatedActivity>
          <iodef:URL>http://zeus.556677889900.example.com/log-bin/lunch_install.
php?aff_id=1&amp;lunch_id=1&amp;maddr=&amp;action=install
          </iodef:URL>
      </iodef:RelatedActivity>
      <iodef:ReportTime>2012-12-05T12:20:00+00:00</iodef:ReportTime>
      <iodef:GenerationTime>2012-12-05T12:20:00+00:00</iodef:GenerationTime>
      <iodef:Description>Malware and related indicators</iodef:Description>
          <iodef:Assessment occurrence="potential">
              <iodef:SystemImpact severity="medium" type="breach-privacy">Malwar
e with C&amp;C </iodef:SystemImpact>
           </iodef:Assessment>
                   <iodef:Contact role="creator" type="organization">
                <iodef:ContactName>example.com CSIRT
                </iodef:ContactName>
                <iodef:Email>contact@csirt.example.com</iodef:Email>
            </iodef:Contact>
            <iodef:EventData>
            <iodef:Flow>
<iodef:System category="source">
                  <iodef:Node>
                    <iodef:Address category="ipv4-addr">192.0.2.200</iodef:Addre
ss>
                    <iodef:NodeRole category="www"/>
                 </iodef:Node>
              </iodef:System>
            </iodef:Flow>
        </iodef:EventData>
    </iodef:Incident>
</IODEF-Document>
```

A.3. DDoS

The DDoS test exchanged information that described a DDoS including protocols and ports, bad IP addresses and HTTP User-Agent fields. The IODEF version used for the data representation was based on [I-D.ietf-mile-rfc5070-bis]

```
 <?xml version="1.0" encoding="UTF-8"?>
 <IODEF-Document version="1.00" lang="en"
     xmlns="urn:ietf:params:xml:ns:iodef-1.41"
     xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.41"
```

```
     xmlns:iodef-sci="urn:ietf:params:xml:ns:iodef-sci-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <iodef:Incident purpose="reporting" restriction="default">
          <iodef:IncidentID name="csirt.example.com">
              189701
          </iodef:IncidentID>
          <iodef:StartTime>2013-02-05T00:34:45+00:00</iodef:StartTime>
          <iodef:DetectTime>2013-02-05T01:15:45+00:00</iodef:DetectTime>
          <iodef:ReportTime>2013-02-05T01:34:45+00:00</iodef:ReportTime>
          <iodef:description>DDoS Traffic Seen</iodef:description>
             <iodef:Assessment occurrence="actual">
                 <iodef:Impact severity="medium" type="dos">
                 DDoS Traffic</iodef:Impact>
                 <iodef:Confidence rating="numeric">90
                 </iodef:Confidence>
             </iodef:Assessment>
             <iodef:Contact role="creator" type="organization">
                 <iodef:ContactName>Dummy Test</iodef:ContactName>
                 <iodef:Email>contact@dummytest.com</iodef:Email>
             </iodef:Contact>
             <iodef:EventData>
                <iodef:Description>
                    Dummy Test sharing with ISP1
                </iodef:Description>
             <iodef:Expectation action="other"/>
             <iodef:Method>
                <iodef:Reference>
                   <iodef:ReferenceName>
                       Low Orbit Ion Cannon User Agent
                   </iodef:ReferenceName>
                   <iodef:URL>
                      http://blog.spiderlabs.com/2011/01/loic-ddos-
                      analysis-and-detection.html
                   </iodef:URL>
                   <iodef:URL>
                      http://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon
                   </iodef:URL>
                </iodef:Reference>
             </iodef:Method>
             <iodef:Flow>
                <iodef:System category="watchlist-source" spoofed="no">
                   <iodef:Node>
                     <iodef:Address category="ipv4-addr">
                        10.10.10.104</iodef:Address>
                   </iodef:Node>
                   <iodef:Node>
                     <iodef:Address category="ipv4-addr">
```

```
                        10.10.10.106</iodef:Address>
                                    </iodef:Node>
                    <iodef:Node>
                      <iodef:Address category="ipv4-net">
                       172.16.66.0/24</iodef:Address>
                       </iodef:Node>
                    <iodef:Node>
                      <iodef:Address category="ipv6-addr">
                       2001:db8:dead:beef::</iodef:Address>
                                   </iodef:Node>
    <iodef:Service ip_protocol="6">
       <iodef:Port>1337</iodef:Port>
       <iodef:Application user-agent="Mozilla/5.0 (Macintosh; U;
              Intel Mac OS X 10.5; en-US; rv:1.9.2.12) Gecko/
              20101026 Firefox/3.6.12">
       </iodef:Application>
    </iodef:Service>
               </iodef:System>
                <iodef:System category="target">
                   <iodef:Node>
                     <iodef:Address category="ipv4-addr">
    10.1.1.1</iodef:Address>
                  </iodef:Node>
                  <iodef:Service ip_protocol="6">
                    <iodef:Port>80</iodef:Port>
                  </iodef:Service>
                </iodef:System>
                <iodef:System category="sensor"><iodef:Description>
                    Information provided in FLow class instance is from
                    Inspection of traffic from network tap
                </iodef:Description></iodef:System>
              </iodef:Flow>
          </iodef:EventData>
      </iodef:Incident>
 </IODEF-Document>
```

A.4. Spear-Phishing

The Spear-Phishing test exchanged information that described a Spear-Phishing email including DNS records and addresses about the sender, malicious attached file information and email data. The IODEF version used for the data representation was based on [I-D.ietf-mile-rfc5070-bis].

```
 <?xml version="1.0" encoding="UTF-8"?>
 <IODEF-Document version="1.00" lang="en"
     xmlns="urn:ietf:params:xml:ns:iodef-1.41"
     xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.41"
```

```
     xmlns:iodef-sci="urn:ietf:params:xml:ns:iodef-sci-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <iodef:Incident purpose="reporting">
          <iodef:IncidentID name="csirt.example.com">
             189601
          </iodef:IncidentID>
          <iodef:StartTime>2013-01-04T08:01:34+00:00</iodef:StartTime>
          <iodef:StopTime>2013-01-04T08:31:27+00:00</iodef:StopTime>
          <iodef:DetectTime>2013-01-04T08:06:12+00:00</iodef:DetectTime>
          <iodef:ReportTime>2013-01-04T09:15:45+00:00</iodef:ReportTime>
          <iodef:description>
             Zeus Spear Phishing E-mail with Malware Attachment
          </iodef:description>
             <iodef:Assessment occurrence="potential">
                 <iodef:Impact severity="medium" type="info-leak">
                 Malware with Command and Control Server and System
                 Changes</iodef:Impact>
             </iodef:Assessment>
             <iodef:Contact role="creator" type="organization">
                 <iodef:ContactName>example.com CSIRT
                 </iodef:ContactName>
                 <iodef:Email>contact@csirt.example.com</iodef:Email>
             </iodef:Contact>
             <iodef:EventData>
               <iodef:Description>Targeting Defense Contractors,
                  specifically board members attending Dummy Con
               </iodef:Description>
             <iodef:Expectation action="other"/>
             <iodef:Method>
                <iodef:Reference indicator_uid="1234">
                   <iodef:ReferenceName>Zeus</iodef:ReferenceName>
                   </iodef:Reference>
             </iodef:Method>
             <iodef:Flow>
                <iodef:System category="source">
                   <iodef:Node>
                     <iodef:Address category="url">
                           http://www.zeusevil.com</iodef:Address>
                     <iodef:Address category="ipv4-addr">
                           10.10.10.166</iodef:Address>
                    <iodef:Address category="as">
                           225</iodef:Address>
                     <iodef:Address category="ext-value"
                       ext-category="as-name">
                       EXAMPLE-AS - University of Example"
                       </iodef:Address>
                     <iodef:Address category="ext-value"
```

```
                      ext-category="as-prefix">
                      172.16..0.0/16
                      </iodef:Address>
                    <iodef:NodeRole category="www"
                      attacktype="malware-distribution"/>
                  </iodef:Node>
              </iodef:System>
            </iodef:Flow>
            <iodef:Flow>
               <iodef:System category="source">
                  <iodef:Node>
                   <iodef:NodeName>mail1.evildave.com</iodef:NodeName>
                    <iodef:Address category="ipv4-addr">
                          172.16.55.6</iodef:Address>
                   <iodef:Address category="asn">
                          225</iodef:Address>
                    <iodef:Address category="ext-value"
                      ext-category="as-name">
                      EXAMPLE-AS - University of Example
                    </iodef:Address>
    <iodef:DomainData>
       <iodef:Name>evildaveexample.com</iodef:Name>
       <iodef:DateDomainWasChecked>2013-01-04T09:10:24+00:00
       </iodef:DateDomainWasChecked>
       <iodef:RelatedDNS RecordType="MX">
           evildaveexample.com MX prefernce = 10, mail exchanger
           = mail1.evildave.com</iodef:RelatedDNS>
       <iodef:RelatedDNS RecordType="A">
           mail1.evildaveexample.com
           internet address = 172.16.55.6</iodef:RelatedDNS>
       <iodef:RelatedDNS RecordType="SPF">
            zuesevil.com. IN TXT \"v=spf1 a mx -all\"
       </iodef:RelatedDNS>
    </iodef:DomainData>
                  <iodef:NodeRole category="mail"
                        attacktype="spear-phishing"/>
                  </iodef:Node>
                  <iodef:Service>
                     <iodef:EmailInfo>
                        <iodef:Email>emaildave@evildaveexample.com
                        </iodef:Email>
                        <iodef:EmailSubject>Join us at Dummy Con
                        </iodef:EmailSubject>
                        <iodef:X-Mailer>StormRider 4.0
                        </iodef:X-Mailer>
                     </iodef:EmailInfo>
                  </iodef:Service>
              </iodef:System>
```

```
            <iodef:System category="target">
               <iodef:Node>
                  <iodef:Address category="ipv4">
                    192.168.54.2</iodef:Address>
               </iodef:Node>
             </iodef:System>
          </iodef:Flow>

          <iodef:Record>
            <iodef:RecordData>
              <iodef:HashData type="file_hash"
                 indicator_uid="1234">
                 <iodef:FileName>Dummy Con Sign Up Sheet.txt
                 </iodef:FileName>
                 <iodef:FileSize>152</iodef:FileSize>
              <ds:Reference>
                 <ds:DigestMethod Algorithm=
                    "http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <ds:DigestValue>
                      141accec23e7e5157de60853cb1e01bc38042d
                      08f9086040815300b7fe75c184
                    </ds:DigestValue>
              </ds:Reference>
            </iodef:HashData>
           </iodef:RecordData>
           <iodef:RecordData>
            <iodef:HashData type="PKI_email_ds" valid="0">
              <ds:Signature>
               <ds:KeyInfo>
                  <ds:X509Data>
                     <ds:X509IssuerSerial>
                        <ds:X509IssuerName>FakeCA
                     </ds:X509IssuerName>
                     </ds:X509IssuerSerial>
                     <ds:X509SubjectName>EvilDaveExample
                     </ds:X509SubjectName>
                  </ds:X509Data>
               </ds:KeyInfo>
                  <ds:SignedInfo>
                   <ds:Reference>
                      <ds:DigestMethod Algorithm=
                        "http://www.w3.org/2001/04/xmlenc#sha256"/>
                      <ds:DigestValue>
                       352bddec13e4e5257ee63854cb1f05de48043d09f9
                       076070845307b7ce76c185
                      </ds:DigestValue>
                  </ds:Reference>
                </ds:SignedInfo>
```

```
                </ds:Signature>
             </iodef:HashData>
          </iodef:RecordData>
         </iodef:Record>
        </iodef:EventData>
     </iodef:Incident>
 </IODEF-Document>
```

Authors' Addresses

Panos Kampanakis
Cisco Systems
170 West Tasman Dr.
San Jose, CA 95134
US

Email: pkampana@cisco.com

Mio Suzuki
NICT
4-2-1, Nukui-Kitamachi
Koganei, Tokyo 184-8795
JP

Email: mio@nict.go.jp