INCH Working Group
draft-ietf-inch-implement-01.txt
Expires: May 10, 2005

R. Danyliw
CERT Coordination Center

November 09, 2004

The Incident Object Description Exchange Format (IODEF) Implementation Guide draft-ietf-inch-implement-01

Status of this Memo

Abstract

The purpose of this Internet-Draft is to provide implementation guidelines for Computer Security Incident Response Teams (CSIRT) adopting the Incident Object Description Exchange Format (IODEF).

Table of Contents

1. Introduction

1.1 Terminology

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 RFC2119 [4].

Definitions for some of the common computer security-related terminology used in this document can be found in Section 2 of [1].

1.2 Overview

The Incident Object Description Exchange Format (IODEF) [7] is an abstract data model for representing computer security incidents exchanged by Computer Security Incident Response Teams (CSIRTs). It was designed to satisfy the requirements laid out in [1]. Practically, [7] also provides an XML DTD (soon to be Schema) implementation of the data model.

The purpose of this document is to provide additional informati
on for
IODEF implementers. Section 1 outlines the operational assumptions and context in which the IODEF will be used. Section 2 discusses general issues related to exchanging IODEF documents. The importing and exporting IODEF documents with an IHS is covered in detail in Section 3 and 4, respectively. Section 5 provides useful examples of IODEF usage for given situations.

This draft is incomplete. There are several sections marked as "TODO" either the document author has not completed the text, or the working group needs to provide clarification.

1.3 CSIRT Operations

The key function of a CSIRT is to remediate security activity in their constituency. As security events can occur across administrative domains, CSIRTs also often play a coordination role to resolve incidents. In the IODEF context, a CSIRT is defined very broadly to include almost any entity that might exchange incident information. For example, this role might be fulfilled by a single security analyst in a network operations group, a help-desk operation center, or a national-level CSIRT.

CSIRTs with responsibility over networks often build their operations around a system to manage the entire incident life-cycle. This incident handling system (IHS) must store all incident information and related communications; provide primitives to support the work-flow of the analysts; and tools to analyze the reported incident

data. It is commonly built around a ticket-tracking application adapted for a security role. The underlying data store is typically a relational database. See section 3 of [1] for additional details about the implicit operational assumptions for the IODEF.

The IODEF is agnostic to most of the internal process and technology choices made by a CSIRT. The IODEF provides no improvements for user interface issues or new analytical techniques. It is in the communications with other parties that the IODEF can have an impact on operations. The current collaboration model requires significant effort to process incident reports because of a lack of standardization in incident information. The IODEF can simplify some of this communication by providing a well-documented format and structure, published as an XML DTD, to represent the incident data already being exchanged.

The IODEF is not a replacement for all communication in and across constituencies. Any common format is useful when there is a need to inter-operate. Inside an organization, where processes can be mandated by fiat, standardization occurs implicitly. The value of IODEF comes when there is a need to exchange information across administrative domain. That said, the IODEF is not suitable for all users. The format is complex, making it unlikely that IODEF documents will be generated by hand as done with many incident reports today. Therefore, it will largely be employed only by a more sophisticated set of users that have the knowledge to reformat their incident data. In the foreseeable future, there will be a continued need to support free-form reporting mechanisms. As a middle ground, front-end tools gathering incident data (e.g., web forms) can be developed to export IODEF documents.

2. General Integration Considerations

Integrating the IODEF into a CSIRT requires changing the IHS to import and export IODEF documents. Effectively, this capability involves the ability to convert from the native IHS data format to XML, and vice versa. Since it merely introduces another representation for existing data, the underlying storage mechanism and schema of the IHS need not be changed to accommodate the IODEF. Furthermore, the use of the IODEF as the explicit storage or archive format is not recommended due to the space inefficiency of XML. Importing IODEF documents will allow a CSIRT to rapidly process newly reported incident information since the barriers of the semantics ambiguity and data normalization will not be present. Exporting IODEF documents allows a CSIRT to unambiguously share information with collaborators. This capability becomes especially relevant when coordinating with a CSIRT with whom no prior relationship might exist.

2.1 Unique Identifiers

CSIRTs track incidents by assigning each an identifier unique in context of its constituency. In the IODEF data model, this identifier is represented by the mandatory IncidentID class. It is through this identifier that an IODEF document relates to the IHS. It is possible that this same activity could be reported to another CSIRT that in turn assigns its own unique identifier. Each CSIRT associates their own identifier for the identical incident in their respective IHS. However, to provide a framework for CSIRTs to refer to each other's data, the IODEF provides the optional AlternateID class to represent another CSIRT's tracking identifier for the same activity. Combining the name of a CSIRT and the incident identifier provides a globally unique identifier for an incident.

TODO: How are IncidentIDs generated? What is the convention?

It may be possible that multiple IODEF documents are exchanged about the same incident over its lifetime. Hence, the use of the id attribute of the IODEF-Document to uniquely identify a particular IODEF document. This global identifier is only relevant for the purposes of transport (e.g., duplicate detection, retransmission). It will change with each new IODEF document, regardless of the value of the IncidentID class value.

TODO: Can an /IODEF-Document@id ever be reused? how is it generated? Is it globally unique?

2.2 Profiles

While the IODEF data model is unambiguous, the usage and the semantics of the data will need to be further refined for a community of collaborators if incident data is to be exchanged. These data exchange policies are collectively referred to as a profile. The need for profiles, in addition to the format specification, is due to the wide variety of data that might be represented in an IODEF document. A well-specified profile is the key to machine-parsing the exchanged documents.

A CSIRT may choose to publicly publish a profile for use by anyone wishing to communicate information to it. Such a model is helpful in a CSIRT with a wide constituency that might receive unsolicited reports. Likewise, in a smaller, closed community of CSIRTs, specific arrangements that apply only to the respective parties can be made. These profiles may cover organizational specific information that the parties will be sharing. Since a CSIRT may send IODEF documents to many other CSIRTS, it will have to maintain many profiles, each specific to the intended recipient.

The format for specifying a profile is outside the scope of this document and the INCH working group. However, Section 2.2.1 to Section 2.2.4 provide a cursory description of the types of information that should be included in a profile.

2.2.1 Required Data

A profile MUST specify exactly the data that will be exchanged between peers. The IODEF specification mandates the presence of certain fields, but the profile should go further to define which optional fields are required. There is no point in sending data that a peer mig
ht not be interested in collected. Likewise, insufficient information will require a follow-up communication.

Incident reports documenting different types of activity are not composed of the same type of data. For example, the data needed to describe an administrative compromise might be different from that of a policy violation. Therefore, a profile could distinguish between the different types of incidents and specify different mandatory fields for each.

2.2.2 Semantics

Given the XML DTD and the profile, the recipient of an IODEF document should be able to understand the contents. A profile MUST disambiguate the semantics of all the subjective values (see Section

3.3.3) that are exchanged. When not compromising the intent of the transformation, any sanitization performed on the data SHOULD be documented. Naming conventions for data commonly found in CSIRTs (e.g., incident numbers) SHOULD be documented.

2.2.3 Formatting

To the the degree possible, formatting conventions SHOULD be standardized to aid in machine processing of IODEF documents. This specification is especially relevant when dealing with the free-form values (see Section 3.3.4). Agreeing on a natural language for these fields is also helpful within a diverse community.

In addition the format of the content itself, the overall structure of the IODEF document should be documented. Given that the data model is non-deterministic (see Section 3.4.1), the profile SHOULD specify the required way to represent the information.

2.2.4 Transport issues

In the absence of an IODEF transport protocol, the profile MUST document how the peers will exchange the IODEF documents. The choice of protocols must take into account the properties of the XML IODEF documents. Ideally the channel between parties would provide reliable delivery, compression of the text stream, confidentiality, integrity, and authenticity. Non-repudiation may be desirable in certain situations.

The process surrounding the use of the protocol will also need to be specified. The frequency of incident data exchange SHOULD be specified (e.g., batched at pre-determined intervals, or real-time). If cryptography is used, key management issues must be addressed. Choices of trust models must be decided; will trust be based on a hierarchy (ala, a PKI) or a web-of-trust (ala, PGP).

Existing data exchange practices in CSIRT operations commonly make use of SOAP, TLS, or PGP encrypted and signed email messages.

2.3 Updating Incident Data

TODO: The working group needs to decide how to perform updates, if at all

3. Importing and Processing Considerations

Upon receiving an IODEF document, the IHS must process it so that normal work-flow processes can be applied. The task of importing an IODEF document involves extracting the relevant data and storing it in the IHS.

3.1 Processing Algorithm

The processing of IODEF documents can be generalized into five steps. Depending on the specific exchange protocols used, and the particular IHS, certain steps may not be necessary.

- Accepting the document. IODEF documents will likely be exchanged over a cryptographically secured medium. Prior to even examining the document, it may need to be decrypted. If the underlying transport does not already handling the issues of key management, the IHS must provide the correct decryption key. It may be necessary to use external information or properties of the document itself to determine the proper key to use. When possible, the authenticity of the document from the sender must also be verified (e.g., via public key signatures). If confidentiality and integrity are implemented via XML-Encryption and XML-Signature, validation of the XML MUST occur prior to this step.

- Structural Validation. To confirm proper formatting, the document must be examined with an XML parser to check that it is both well-formed and valid according to the IODEF DTD. If XML extensions are used in the AdditionalData or RecordItem classes, then the appropriate DTDs must be used. There are numerous free and commercial parsers for virtual all programming languages. Given a properly formatted XML document, any additional constraints on element and attribute cardinality imposed by a data sharing profile SHOULD be applied.

- Data Validation. Verification of properly formatted XML document does not guarantee that the data itself is valid. Therefore, the individual element and attribute values must be checked that they conform to the data types specified by the data model (e.g., was an alphanumeric string value specified for a numeric TCP port). A profile might also impose formatting restrictions on the data.

- Semantic Validation. Given properly formatted data, its semantics must be verified. This validation may be dictated by the profile specification (e.g., a particular value range), or from external information (e.g., are the timestamps in the future). Discerning the reason why this incident report was sent to the CSIRT and the

expected response is key at this phase.

- Transformation and Storage. Since IODEF documents must be imported, transformations may need to be applied to the data to convert it to the native representation of the IHS. Once natively formatted, the necessary invocation must be made to write into the data store (e.g., SQL statements).

- Post-Processing. In the course of processing IODEF documents useful meta-data can be generated. This meta-information is often useful for work-flow, debugging and audit purposes. An IHS might note in a transaction logs when a document arrived, its size and source. Implementers can chose to store this information external to the IHS, or extend its underlying data store. Furthermore, when a new incident report arrives certain triggers in the IHS may need to be invoked to signal the presence of this new information. Automated analysis tools may be invoked to correlate the new information with the existing data set. Work-flow processes or tasking priorities may need to be adjusted.

There will be instances where an IODEF document that cannot be processed is sent to a CSIRT. This situation may be due to invalid XML formatting, or a semantic misunderstanding. The CSIRT must decide how to handle this occurrence. The range of possibilities includes silently dropping these documents to manual intervention by an analyst. The IHS SHOULD keep at least cursory logs of these types of documents. It will allow for debugging, as well as, a way to identify denial of service activity.

3.2 IDMEF relationship

In constructing the IODEF data model, certain classes were reused from the IDMEF. Therefore, their description was not included in the specification. Instead referencing the IDMEF specification is required. The relevant portions of the IDMEF DTD were copied outright into the IODEF DTD for completeness. For a list of classes from the IDMEF, see the "IDMEF" column of Appendix 1.

3.3 Types of Data

Given that the IODEF is implemented in XML, the entire document is a single text string wit
h an encoding specified in the leading XML
processing instruction. The data model enforces a structure onto this string by segmenting distinct data into XML elements and attributes each of which is typed. However, the base data type only provides a formatting specification. A richer understanding of the data items found in the IODEF is necessary to process the data in an

automated way. It is useful to segregate the various data item based

on the different approaches that will be needed to process them. The IODEF data model has items that have the following types of values: enumerated, structured, subjective, free-form, and extensions. Appendix 1 associates a data type with every data item of the IODEF data model.

3.3.1 Enumerated Values

An enumerated value is a specified list of possible values for a given data element. They are used when the domain of possible values is fixed. Practically speaking, all enumerated values in the IODEF are XML attributes. For a list of data items that are enumerated values, see those marked as "ENUM" in Appendix 1.

In general, processing enumerated types is straightforward, since all possible values are known. However, there are certain enumerated values that provide an escaping mechanism whereby an unspecified value may be represented. This technique involves setting the XML attribute to "other", and encoding the desired value in the PCDATA of the XML element of the attribute. For a full list of such special attributes, see those that are marked as "ENUM+OTHER" in Appendix 1.

In order to allow updates, all enumerated lists in the IODEF are registered and maintained by IANA. However, in a closed community, it would not be uncommon to extend an enumerated list by adding community-relevant data values. This addition would have to be noted in the exchange profile, and the corresponding portion of the XML DTD updated. Otherwise, if an unrecognized value is provided for an enumerated value, it should be treated as an invalid XML document.

3.3.2 Structured Values

Structured values are the bulk of the IODEF data model. These are data items that have a well-defined format, and unambiguous semantics. The IHS will be able to parse and interpret them with little or no transformation. The specific format of the data is either dictated outright by the data type of the class, or by other information present in the IODEF documents (e.g., the value of an attribute). For a full list of structured values, see the data items marked as "STRUCTURED" in Appendix 1.

3.3.3 Subjective Values

Subjective values are identical to structured values with regard to formatting, but they have ambiguous semantics. Interpretation of subjective values requires further specification from a pre-negotiated profile. Since profiles between sites can vary, the semantics of the same value can depend of the profile used. For a

full list of the subjective values, see the data items marked as "SUBJECTIVE" in Appendix 1.

3.3.4 Free-form Values

Free-form values are text strings with no pre-defined structure used to represent natural language descriptions. These values often represent information too disparate or complex to easily specify. Machine parsing free-form text values to extract meaning is extremely difficult. Peers may agree in a profile to apply structure to these fields whereby imposing formatting constraints. However, short of such additional information, the IHS SHOULD extract these values and store them unmodified for later review by a human analyst.

Free-form values can support a variety of natural languages. Hence, associated with each element are two attributes that aid in disambiguating the formatting. While the encoding of the entire XML document is specified in the initial XML processing instruction, free-form XML elements have an attribute named "encoding" that specifies the language encoding to apply to that element. Furthermore, enumerated XML attribute named "lang" allows the specification of the natural language of the element.

Due to the processing complexity, free-form values SHOULD be used sparingly, favoring instead the existing structured data model to represent information.

For a full list of free-form values, see the data items marked "FREEFORM" in Appendix 1.

3.3.5 Extensions

No standardization effort can represent all data elements of interest to its entire community. Hence, the IODEF includes well-defined ways to extend itself. The AdditionalData and RecordItem classes allow arbitrary data to be included in the document. The inclusion of these extensions and the semantics of this information MUST be documented in a profile. If XML extensions are used, then the appropriate DTD will need to be passed to the parser.

3.4 Structure of the Data

3.4.1 Non-deterministic

The IODEF data model is non-deterministic in that two of the elements are recursive defined: Contact, and EventData. The implication of such a structure is that fixed XPaths to information might not always

be valid. Making use of the recursion allows for a logical grouping of information that eliminated redundancy. The following is a simple example of this concept.

TODO: include recursion examples

3.4.2 Document unique idents

TODO: what are they used for? to what classes do they apply?

4. Export Considerations

In order to share information with other CSIRTs, incident information must be exported from the IHS to the IODEF.

4.1 Processing Algorithm

Once the intended recipient of an incident is identified, the process to export and transmit this party an IODEF document can be generalized into four steps.

- Query the Incident. Initially, all the relevant data about the incident that will be encoded in the IODEF document must be extracted from the IHS. Based on negotiated profiles, different, or varying level of detail, might be shared about the same incident with different parties. Certain CSIRTs might receive all the information about an incident; another might only receive a subset. If there is any sensitive data that must be sanitized, or classes of information to be filtering for certain parties, the appropriate policy should be enforced.

- Reformat the Data. The internal representation of the incident data in the IHS may be quite different that the one used in the IODEF. This transformation may entail explicit reformatting of the individual data items into the IODEF data types. Furthermore, subjective values, if stored in a different way, may need to be remapped onto their equivalents in the IODEF data model and as specified in a data profile (e.g., the profile might specify a priorities from 1 to 4, with 1 being the lowest, but the IHS uses a scheme, were 4 is the lowest). Finally, during this conversion process, implementers must consider XML encoding issues. There are several special characters (see XML reference) that must be escaped. If binary data
is included in the AdditionalData or
RecordData classes, it must also be escaped. If whitespace is part of a data the xml:preserve attribute must be set correctly in the relevant XML element. A value of "preserve" for this attribute will require the IHS to treat any whitespace as significant. Otherwise, the default value of "default" allows the IHS to treat the whitespace as it likes.

- Set Expectations. When sending an IODEF document to another CSIRT, the intent behind this communication and the desired handling of the information should be document. The enumerated attributed "purpose" of the Incident class should be set to convey the reason why this IODEF document was sent to the CSIRT. Likewise the Expectation class MUST be set to encapsulate the expectation the sender has for the recipient. Judicious use of

the restriction attribute on the various data items will also

allow the sender to convey how they would request their data to be used. With all of these settings, the recipient is free to ignore this information.

- Transmission. Given a valid XML document, the proper exchange protocol, as specified in the profile associated with the recipient, will be used to send the document. In the absence of published profile for a recipient, out-of-band mechanisms MUST be used to contact the party to make arrangements.

5. Representation Examples

6. Acknowledgments

7. Appendix 1

The following is a list of all elements and attributes of the IODEF and their associated datatypes.

	Element Name	Attribute	Datatype	Type	IDMEF
1	AdditionalData	----------	ANY	FREEFORM	Y
		type		ENUM	STRUCTURED
		meaning	STRING	FREEFORM	
2	Address	PCDATA	STRING	FREEFORM	
		vlan-num		INTEGER	FREEFORM
		vlan-name		STRING	FREEFORM
		category		ENUM	STRUCTURED
3	AlternativeID	----------	----------	----------	
4	Analyzer	----------	----------	----------	
		Y			
		version	STRING	FREEFORM	Y
		osversion		STRING	FREEFORM
		Y			
		ostype	STRING	FREEFORM	Y
		model		STRING	FREEFORM
		Y			
		manufacterer	STRING	FREEFORM	Y
		class		STRING	FREEFORM
		Y			
		analyzerid		STRING	FREEFORM
		Y			
5	arg		PCDATA	STRING	FREEFORM
					Y
6	Assessment	----------	----------	----------	
7	Classification	----------		----------	----------
		origin		ENUM + O	STRUCTURED
8	command	PCDATA	STRING	FREEFORM	Y
9	Confidence	PCDATA		REAL	SUBJECTIVE
		Y			
		rating		ENUM	SUBJECTIVE
		Y			
10	Contact	----------	----------	----------	
		contacttype		ENUM	STRUCTURED

```
                                contactrole             ENUM
STRUCTURED
   11   Counter          PCDATA          INTEGER STRUCTURED
                                type                    ENUM
STRUCTURED
                                meaning         STRING  FREEFORM
   12   DateTime                 PCDATA          DATETIME        STRUCTURED
   13   Description              PCDATA          STRING  FREEFORM
                                transform               ENUM
STRUCTURED
                                preserve                ENUM
STRUCTURED
                                lang                    ENUM
STRUCTURED
   14   DetectTime               PCDATA          DATETIME        STRUCTURED
   15   Email                    PCDATA          EMAIL           STRUCTURED
   16   EndTime          PCDATA          DATETIME        STRUCTURED
   17   env                      PCDATA          STRING  FREEFORM
Y
   18   EventData                ------------    ----------      ------------
```

19	Expectation		----------	----------	----------	
		priority		ENUM		
SUBJECTIVE						
		category		ENUM		
STRUCTURED						
20	Fax		PCDATA	PHONE	STRUCTURED	
21	Flow		----------	---------	----------	
22	History	----------		---------	----------	
		type		ENUM+O	STRUCTURED	
23	HistoryItem		----------	----------	----------	
24	Impact	PCDATA	STRING	FREEFORM		Y
		type		ENUM+O		
STRUCTURED	Y					
		severity		ENUM		
SUBJECTIVE	Y					
		lang		ENUM		
STRUCTURED	Y					
		completion		ENUM		
STRUCTURED	Y					
25	Incident		----------	----------	----------	
		purpose		ENUM+O	STRUCTURED	
26	IncidentID		PCDATA	UID	SUBJECTIVE	
		name		GUID		
STRUCTURED						
27	IODEF-Document	----------	----------	----------		
		version		STRING	STRUCTURED	
28	Location		PCDATA	STRING	FREEFORM	
		lang		ENUM		
STRUCTURED						
29	Method	----------	----------	----------		
30	MonetaryImpact	PCDATA	REAL		STRUCTURED	
		severity		ENUM		
SUBJECTIVE						
		currency		ENUM		
STRUCTURED						
31	Name		PCDATA	STRING	STRUCTURED	
		transform		ENUM		
STRUCTURED						
		preserve		ENUM		
STRUCTURED						
		lang		ENUM		
STRUCTURED						
32	name		PCDATA	STRING	FREEFORM	
		lang		ENUM		
STRUCTURED						
33	Node		----------	----------	----------	
		category		ENUM		
STRUCTURED						

```
   34   NodeRole               PCDATA          STRING  FREEFORM
                               lang                    ENUM
STRUCTURED
                               category                ENUM+O  STRUCTURED
   35   path                   PCDATA          STRING  FREEFORM
Y
   36   pid                    PCDATA          STRING  FREEFORM
Y
   37   port                   PCDATA          INTEGER STRUCTURED
Y
   38   portlist               PCDATA          PORTLIST
STRUCTURED             Y
   39   PostalAddresss  PCDATA          POSTAL  STRUCTURED
                               lang                    ENUM
STRUCTURED
   40   Process         ------------    ----------      ------------    Y
                               ident                   STRING  FREEFORM
   41   Record          ------------    ----------      ------------
   42   RecordData              ------------    ----------      ------------
                               ident                   STRING  FREEFORM
   43   RecordItem              PCDATA          ANY             FREEFORM
```

```
                                dtype                   ENUM
STRUCTURED
   44   RegistryHandle  PCDATA          STRING  FREEFORM
                                type                    ENUM
STRUCTURED
   45   RelatedActivity ------------    ----------      ------------
   46   ReportTime              PCDATA          DATETIME        STRUCTURED
   47   Service         ------------    ----------      ------------
                                ip_version              INTEGER STRUCTURED
                                ip_protocol             INTEGER STRUCTURED
   48   StartTime               PCDATA          DATETIME        STRUCTURED
   49   System          -----------             ----------      ------------
                                spoofed         ENUM            STRUCTURED
                                interface               STRING  FREEFORM
                                category                ENUM
STRUCTURED
   50   Telephone               PCDATA          PHONE           STRUCTURED
   51   TimeImpact              PCDATA          REAL            STRUCTURED
                                unit                    ENUM
STRUCTURED
                                severity                ENUM
SUBJECTIVE
                                metric          INTEGER STRUCTURED
   52   TimeZone                PCDATA          STRING  STRUCTURED
   53   url                     PCDATA          STRING  STRUCTURED
Y
```

Figure 1

8. References

8.1 Normative References

[1] Demchenko, Y., Hiroyuki, H. and G. Keeni, "Requirements for Format for Incident Report Exchange", RFC XXX, September 2003.

[2] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Second Edition)", , October 2000, <http://www.w3.org/TR/2000/REC-xml-20001006>.

[3] World Wide Web Consortium, "Extensible Stylesheet Language (XSL) Version 1.0", , October 2001, <http://www.w3.org/TR/xsl/>.

[4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[5] Alvestrand, H., "Tags for the Identification of Languages", RFC 3066, January 2001.

[6] Curry, D. and H. Debar, "Intrusion Detection Message Exchange Format", RFC XXX, January 2003.

[7] Meijer, J., Danyliw, R. and Y. Demchenko, "Intrusion Detection Message Exchange Format", RFC XXX, January 2003.

[8] Eastlake 3rd, D., Reagle, J. and D. Solo, "(Extensible Markup Language) XML-Signature Syntax and Processing", RFC 3275, March 2002.

[9] Imamura, T., Dillaway, B. and E. Simon, "XML Encryption Syntax and Processing, W3C Recommendation", December 2002, <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>.

Author's Address

Roman Danyliw
CERT Coordination Center
4500 Fifth Ave.
Pittsburgh, PA 15213
USA
EMail: rdd@cert.org

Full Copyright Statement