How to Share Transaction Fraud (Thraud) Report Data

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Abstract

This document describes a data-format and protocol for defining and exchanging Transaction Fraud (Thraud) Report Data. It extends the INCH WG’s IODEF XML [IODEF] incident reporting format. Both inbound (Thraud Reports) and outbound (Thraud Watchlists) mechanisms are presented. This work has been endorsed by the Initiative for Open AuTHenticaion [OATH].
Table of Contents

1. Introduction............................................... 3
2. Requirements Terminology.................................. 4
3. The Elements of Transaction Fraud Activity.............. 4
4. Thraud Activity Reporting via an IODEF-Document Incident... 5
5. Fraud Report Class Definitions............................ 7
  5.1. The FraudEventPayment class........................... 8
  5.1.1. PayeeName......................................... 9
  5.1.2. PayeeAddressLine1.................................. 9
  5.1.3. PayeeAddressLine2.................................. 9
  5.1.4. PayeeCity.......................................... 9
  5.1.5. PayeePostalCode.................................... 9
  5.1.6. PayeeCountry...................................... 9
  5.1.7. PayeeAmount....................................... 9
  5.2. The FraudEventTransfer class.......................... 9
  5.2.1. BankID............................................. 10
  5.2.2. AccountID.......................................... 10
  5.2.3. AccountType....................................... 10
  5.2.4. TransferAmount.................................... 10
  5.3. The FraudEventIdentity class.......................... 10
  5.3.1. Component.......................................... 10
  5.3.2. EmailAddress...................................... 10
  5.3.3. UserID............................................. 11
  5.4. The FraudEventOther class............................. 11
  5.4.1. OtherEventType.................................... 12
  5.4.2. OtherEventDescription............................. 12
  5.5. The PayeeAmount class................................ 12
  5.5.1. Class contents.................................... 13
  5.5.2. Currency.......................................... 13
  5.6. The TransferAmount class................................ 13
  5.6.1. Class contents.................................... 13
  5.6.2. Currency.......................................... 13
  5.7. The AccountType class................................ 13
  6. IODEF Required Classes................................... 14
     6.1. Mandatory contents.................................. 14
     6.2. Optional contents.................................. 15
  6.3. Fraud Event Signature Report.......................... 16
  7. IODEF Extensions.......................................... 16
     7.2. purpose attribute.................................. 16
  8. Security Considerations.................................. 17
     8.2. Thraud Data Authenticity and Integrity............. 17
     8.3. Thraud Data Confidentiality and Privacy............ 17
  8.4. Data Protection During Transit......................... 17
  9. IANA Considerations...................................... 18
  10. Conclusion................................................ 18
  11. Acknowledgements......................................... 18
  12. References................................................ 18
1. Introduction

Financial institutions and merchants are confronted with online fraud attacks targeted against their customers through various means. Today there is no standardized data format and open protocol that organizations and law enforcement can use to share known/suspicious transaction fraud data.

There is a clear opportunity for creating an open framework to enable organizations, using a variety of fraud monitoring products, to contribute information related to known or suspicious fraud activity. The very same framework should also formalize mechanisms for distributing correlated updates to all participating members.

This internet draft introduces a data format to facilitate interoperability and exchange of transaction-related fraud data.

More specifically, this document describes a data format and protocol for defining and exchanging Transaction Fraud (Thraud) Report Data. It extends the IODEF XML [IODEF] incident reporting Format.

Any verified organization can contribute online fraud attack records via openly defined protocols. A verified organization is one that has been authenticated and is authorized to provide fraud related data. Verification procedures and the specific requirements for authorization are outside the scope of this specification.

The specific focus of this document is the proposal of an XML schema definition for Fraud Reports and the protocols by which they can be exchanged. The document proposes a definition for both inbound (Thraud Reports) and outbound (Thraud Watchlists) mechanisms.

In section 3 we introduce the different components of a transaction fraud. The reporting method via an IODEF-Document is described in section 4, and we define the report elements in section 5. In the next section we describe the required elements with respect to the IODEF format and follow with security considerations in section 7.
2. Requirements 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 RFC 2119 [RFC2119].

3. The Elements of Transaction Fraud Activity

```
+-------------------------------------+
| Fraudsters                          |
| (collect & verify victim credentials|
|   via phishing, malware etc.)      |
+-------------------------------------+

recruit

< ----------------------share profits------------------------

v

+-------------------------+-------------------+-------------------+
| Fraud Executors         | Financial Organization | Financial Organization |
|                       | <-o-> Fraud Dest. Account | =>o< Account |
| --Open Dest. Account--->|                   +-------------------+-------------------------+

v

+-------------------------+-------------------+-------------------+
| Fraud Detection Sensors | Victim             | Financial Organization |
| (realtime/ offline)     |                   <-o-> Account |
+-------------------------+-------------------+-------------------+
```

Figure 1. Transaction Fraud Elements
Transaction Fraud Activities are normally composed of the following entities:

1. The Fraudsters are those who collect victim’s login credentials via various means, such as phishing, malware etc. and then verify (usually through login attempts) that those credentials are correct. At that time Fraudsters may either recruit Fraud executors directly or wholesale the collected credentials to other Fraudsters who will do the recruiting of the Fraud executors.

2. The Fraud Executors are those that actually will attempt the fraudulent transfer or payment. For fraudulent transfers, legitimate accounts at the same financial organization or a different one from the victim will be set up as the destination account for the fraudulent transfer. Alternatively a fraudulent payment attempt via check or electronic transfer to a named destination is attempted.

3. The Victims of both credential theft and transaction fraud.

4. The Financial Organization who holds either the victim/fraudster’s accounts.

5. Sensors at the Financial Organization who attempt to detect fraudulent transaction attempts, either in realtime or offline.

The intention of Thraud Reporting is to enable any organization that has detected fraud to share this information with other organizations. The receiving organization can use this information appropriately. For example, it could require manual review of transactions from ‘risky’ IP addresses that have been reported.

4. Thraud Activity Reporting via an IODEF-Document Incident

A Thraud Activity Report is an instance of an XML IODEF-Document. Generally, these reports include added EventData.AdditionalData elements. The added elements compose a ThraudRecord Element. There may be multiple ThraudRecord Elements in a Thraud Activity Report. Required information with many optional items is populated into the ThraudRecord structure to form a Thraud Activity Report. If the Thraud Activity Report describes a particular pattern of behaviour, or fraud event signature as described in section 6.3, rather than a specific fraud event, these additional elements may not be included.

There are actually two types of Thraud Activity Reports an "inbound" and an "outbound" report. The "inbound" report is from a reporting organization to describe a transaction fraud. The "outbound" report is destined to subscribers, either through a
subscription or following a query to obtain (possibly correlated) information about fraud elements. Any correlation of information that is subsequently distributed via subscription or query is outside the scope of this specification.

The primary difference in the "inbound" and "outbound" reports is the removal in the "outbound" reports of reporting organization information in order to protect confidentiality. We elaborate on this aspect in section 7, Security Considerations.

This document defines new EventData IODEF XML elements; then identifies the attributes that are required in a compliant Thraud Activity Report. It also makes use of the IODEF POSTAL type for postal addresses, the XML DECIMAL type for financial amounts and defines a "currency" type to indicate the three character currency code as per [ISO 4217]. The Appendicies contain sample Thraud Activity Reports and the complete XML schema.

The Incident element with fraud extensions is summarized below. It provides a standardized representation for commonly exchanged incident data.

For "inbound" reports it contains a unique identifier that is name spaced qualified by the domain name of the reporting organization. In "outbound" reports it contains an opaque unique identifier to protect privacy of data sources. The data elements in this document are expressed in Unified Modeling Language (UML) syntax [UML].

```
+-------------------+
| Incident          |
+-------------------+

| ENUM purpose      |<-----------------[ IncidentID ] |
| STRING ext-purpose|<--(0..1)--[ AlternativeID ] |
| ENUM lang         |<--(0..1)--[ RelatedActivity ] |
| ENUM restriction  |<--(0..*)--[ Description ] |
|                   |<--(1..*)--[ Assessment ] |
|                   |<--(0..*)--[ Method ] |
|                   |<--(0..1)--[ DetectTime ] |
|                   |<--(0..1)--[ StartTime ] |
|                   |<--(0..1)--[ EndTime ] |
|                   |-----------------[ ReportTime ] |
|                   |<--(1..*)--[ Contact ] |
|                   |<--(0..*)--[ Expectation ] |
|                   |<--(0..1)--[ History ] |
|                   |<--(1..*)--[ EventData ] |

| |-----------------+ AdditionalData |
| |                     --> ThraudRecord (added)
```

Figure 2. IODEF and Thraud Reporting
A Thraud Activity Report is composed of one IODEF Incident element, containing one or more EventData elements that contain one or more ThraudRecord elements.

This document describes ThraudRecord elements for the EventData. AdditionalData element containing transaction fraud-related information that does not map to existing Incident or EventData attributes.

One Incident report may contain information on multiple incidents. After the report identification information listed in the Incident element, each individual transaction fraud event is detailed within a single EventData structure.

5. Fraud Report Class Definitions

A fraud report consists of an extension to the Incident.EventData.AdditionalData Element. The contents of the extension are associated with the dtype value ‘xml’. The components of the fraud report identify and capture information related to payment fraud, transfer fraud, identity fraud and other types of fraud.

A payment fraud report is structured as follows.

```
+--------------------------+
 | EventData.AdditionalData |
+--------------------------+
 | ENUM dtype (xml)         |<>--------[ FraudEventPayment ]
+--------------------------+
```

Figure 3. The FraudEventPayment extension

A funds transfer fraud report is structured as follows.

```
+--------------------------+
 | EventData.AdditionalData |
+--------------------------+
 | ENUM dtype (xml)         |<>--------[ FraudEventTransfer ]
+--------------------------+
```

Figure 4. The FraudEventTransfer extension

An identity fraud report is structured as follows.
An other fraud report is structured as follows. This is a general report included as a placeholder for corner cases.

---
<table>
<thead>
<tr>
<th>EventData.AdditionalData</th>
</tr>
</thead>
</table>
| ENUM dtype (xml)         |<>------[ FraudEventOther ]
|                          |
---

Figure 6. The FraudEventOther extension

5.1. The FraudEventPayment class

The FraudEventPayment class is used to report the payee instructions for a fraudulent payment or fraudulent payment attempt. Fraudsters sometimes use the same payee instructions (including the amount) for multiple fraudulent payment attempts. By reporting the payment instructions used in the fraud, other institutions may be able to stop future fraudulent payment attempts to the same payee.

The structure of the FraudEventPayment class is as follows:

---
<table>
<thead>
<tr>
<th>FraudEventPayment</th>
</tr>
</thead>
</table>
|                   |<>--{(0..1)}--[ PayeeName ]
|                   |<>--{(0..1)}--[ PostalAddress]
|                   |<>--{(0..1)}--[ PayeeAmount]
---

Figure 7. The FraudEventPayment class

The components of the FraudEventPayment class are described below. At least one component must be present.
5.1.1. PayeeName

Zero or one value of STRING. The name of the payee.

5.1.2. PostalAddress

Zero or one instance of type POSTAL. A postal address is represented by the POSTAL data type. This data type is a string whose format is documented in Sections 2.23 of [RFC 4519]. It defines a postal address as a free-form multi-line string separated by the "$" character.

The POSTAL data type is implemented as an "xs:string" in the schema.

5.1.3. PayeeAmount

Zero or one instances of type AmountType.

5.2. The FraudEventTransfer class

The FraudEventTransfer class is used to report the payee instructions for a fraudulent funds transfer or fraudulent funds transfer attempt. Fraudsters sometimes use the same payee instructions (including the amount) for multiple fraudulent funds transfer attempts. By reporting the funds transfer instructions used in the fraud, other institutions may be able to stop future fraudulent funds transfer attempts to the same payee.
Figure 8. The FraudEventTransfer class

The components of the FraudEventTransfer class are described below. At least one component must be present.

5.2.1. BankID

Zero or one value of STRING. The destination bank routing transit ID or other Financial Institution (FI) id.

5.2.2. AccountID

Zero or one value of STRING. The destination primary account number.

5.2.3. AccountType

Zero or one instance of type AccountTypeType.

5.2.4. TransferAmount

Zero or one instance of type AmountType.

5.3. The FraudEventIdentity class

The FraudEventIdentity class is used to report a fraudulent impersonation or fraudulent impersonation attempt. By reporting the impersonation event, other potential victims may be able to detect future fraudulent impersonation attempts.

+-------------------------+
|                         |
|FraudEventIdentity       |
+-------------------------+
                      |<>--{0..*}-->[ IdentityComponent]
+-------------------------+

Figure 9. The FraudEventIdentity class

The components of the FraudEventIdentity class are described below. At least one component must be present.

5.3.1. IdentityComponent

Zero to many instances of type iodef:ExtensionType. This specification defines two extensions: EmailAddress and UserID.

5.3.1.1 EmailAddress
In reporting an identity fraud event, the reporting institution MAY include the victim’s email address. This SHALL be achieved by adding an element of type iodef:ExtensionType in the IdentityComponent element. The data type of the extension contents SHALL be xs:string. It SHALL contain the email address of the intended fraud victim.

The IdentityComponent@type attribute SHALL be set to the value "string".

The IdentityComponent@meaning attribute SHALL be set to the value "victim email address".

5.3.1.2 UserID

In reporting an identity fraud event, the reporting institution MAY include the victim’s user id. This SHALL be achieved by adding an element of type iodef:ExtensionType in the IdentityComponent element. The data type of the extension contents SHALL be xs:string. It SHALL contain the user id of the intended fraud victim.

The IdentityComponent@type attribute SHALL be set to the value "string".

The IdentityComponent@meaning attribute SHALL be set to the value "victim user id".

5.4. The FraudEventOther class

The FraudEventOther class is used to report fraudulent events other than those detailed above, such as new event types that emerge and become problematic. This class enables such events to be reported, using this same specification and format, even though the specific characteristics of such events have not yet been formally structured and formatted. By reporting the details of these unspecified event types, other institutions may be able to stop future fraudulent activity.

The structure of the FraudEventOther class is as follows:

```
+--------------------------+
|FraudEventOther           |
+--------------------------+
   +--------------------------+
     |OtherEventType            |
     +--------------------------+
     |PayeeName                  |
     +--------------------------+
     |PostalAddress              |
```
Figure 10. The FraudEventOther class

Many of the components of the FraudEventOther class are also components of the FraudEventPayment or FraudEventTransfer class. Their use in the FraudEventOther class is identical. Therefore the descriptions are not duplicated here. The components that are unique to the FraudEventOther class are described below.

5.4.1. OtherEventType

One or more values of STRING. A name that classifies the activity.

5.4.2. OtherEventDescription

Zero or one values of STRING. A free form textual description of the activity.

5.5. The PayeeAmount class

The PayeeAmount class is used to report the amount of the payment fraud, which may be common across a set of related fraud attempts.

Figure 11. The PayeeAmount class

The components of the PayeeAmount class are described below.
5.5.1. Class contents

Required DECIMAL. The amount of the payment.

5.5.2. Currency

Required STRING. The three letter currency code.

5.6. The TransferAmount class

The TransferAmount class is used to report the amount of the funds transfer fraud, which may be common across a set of related fraud attempts.

```
+-------------------+
| TransferAmount    |
+-------------------+
|                   |
|  DECIMAL          |
|                   |
|  STRING currency  |
+-------------------+
```

Figure 12. The TransferAmount class

The components of the TransferAmount class are described below.

5.6.1. Class contents

Required DECIMAL. The amount of the funds transfer.

5.6.2. Currency

Required STRING. The three letter currency code.

5.7. The AccountType class

The AccountType class is used to report the type of the destination account.

```
+-------------------+
| AccountType       |
+-------------------+
| ENUM (brokerage   |
|   checking        |
|   corporate       |
|   mortgage        |
```

How to Share Transaction Fraud (Thraud) Report Data       February 2008

Figure 13. The AccountType class

Required ENUMERATION. Enumerated values are: 'brokerage', 'checking', 'corporate', 'mortgage', 'retirement' and 'saving'.

6. IODEF Required Classes

This section identifies elements of the IODEF Incident class in a compliant fraud report.

Figure 14. IODEF Required classes

6.1. Mandatory contents
A compliant IODEF fraud report SHALL contain data items as described below:

Purpose - See Section 7.1.

IncidentID - As defined by IODEF.

ReportTime - As defined by IODEF.

Assessment.Impact - As defined by IODEF.

Assessment.Confidence - As defined by IODEF.

Contact.Email - A contact email address for the reporting institution.

Contact.ContactName - The name of the reporting institution. In case the reporting institute has consolidated reports from other institutions, elements of this class MAY contain the name of the consolidator, instead of the original reporting institution.

EventData.DetectTime - The date and time at which the fraud or fraud attempt was detected. This data item is mandatory for specific fraud event reports. However it is optional for fraud event signature reports described in 6.3.

EventData.AdditionalData - contains one or more ThraudRecord events.

6.2. Optional contents

A compliant IODEF fraud report SHOULD contain data items as described below.

Contact.Contact.ContactName - The name of the reporting fraud analyst.

Contact.Contact.Email - The email address of the reporting fraud analyst.

Contact.Contact.Telephone - The telephone number of the reporting fraud analyst.

EventData.Flow.System.Service.Application - Information about the software used by the attacker, including the type and version of operating system, communication and application software.

EventData.Flow.System.Node.Address.vlan-num - The IPv4 or IPv6 address or subnet mask, depending upon the accompanying value of the ‘Address@category’ attribute.
EventData.Flow.System.Node.Location - The name and address of the owner of the DNS domain from which the fraud or fraud attempt was executed.


6.3. Fraud Event Signature Report

A fraud event signature report conveys information about the behavior associated with fraudulent events, rather than reporting the specific events themselves. An example of a fraud event signature might include a customer performing a wire transfer of over $5,000.00 views email address and wire transfer within a single session, has changed email address within the past 2 weeks and performed at least 2 bill payments to the same payee within a single week. Sharing fraud event signature information enables recipients to detect similar behavior in their own systems.

A fraud event signature report contains data items as shown below:

Purpose - Includes value "reporting".

IncidentID - As defined by IODEF.

ReportTime - As defined by IODEF.

Assessment.Impact - Includes the severity attribute.

Method.Reference.ReferenceName - A name for the specific fraud event signature.

Method.URL - A URL that represents the detailed definition of the fraud event signature.

Method.Description - A brief description of the behavior covered by the fraud event signature.

Contact.Email - A contact email address for the reporting institution.

Contact.ContactName - The name of the reporting institution.

7. IODEF Extensions

7.2. purpose attribute

The following additional values are defined for the IODEF.Incident@purpose attribute.
How to Share Transaction Fraud (Thraud) Report Data  February 2008

Add - The identified fraud event SHOULD be added to the corpus by the recipient.

delete - The identified fraud event SHOULD be deleted from the corpus by the recipient.

Modify - The identified elements of the fraud event SHOULD be replaced in the corpus by the supplied values. Where no corresponding element exists in the corpus, the element SHOULD be added to the corpus by the recipient.

8. Security Considerations

This document focuses on the data format for exchanging transaction fraud data. The critical security concerns are the validity of submitted information (Thraud Reports) and outbound information (Thraud Watchlists) sent upon a query, as well as the protection of contributors privacy when sharing the data.

8.2. Thraud Data Authenticity and Integrity

The Thraud Reports SHOULD be digitally signed. This would guarantee the origin and integrity of the submitted information. A possible method is to use the optional IODEF XML signature as defined in [XMLSIG]. The potential recipients of Thraud Reports SHOULD be able to verify these digital signatures.

8.3. Thraud Data Confidentiality and Privacy

The Thraud Reports MAY be encrypted when stored for future usage. Contributors of Thraud Reports might not be willing to disclose fraudulent transactions attached to their name. A simple mechanism MUST enable the query of any data to return a valid response without disclosing the unique Identifier of a specific organization.

We suggest to use an opaque identifier for each report in order to index privately the reports. A hash function (e.g. SHA-1) MAY be used to generate the opaque identifier from the organization name:

\[
\text{OpaqueIdentifier} = \text{SHA-1}(<\text{IncidentID Field}>)
\]

The OpaqueIdentifier can be used to reference the report without disclosing the full organization identity.

8.4. Data Protection During Transit

In addition to protecting thraud data when stored, the data also needs to be protected during transit. This specification does not mandate a particular transport protocol for transmitting thraud data. However, the protocol must enable the thraud data to be
digitally signed and encrypted during transit. It is recommended that commonly used secure protocols, such as HTTPS, SSL and SOAP over EV SSL be used.

9. IANA Considerations

This document has no actions for IANA.

10. Conclusion

This internet draft introduced Transaction Fraud (Thraud) reporting mechanisms to enable the sharing of Fraud data. Based on the IODEF-document format, the proposed extension should facilitate interoperability and provide increased security for online applications.

11. Acknowledgements

We would like to thank Tim Moses for his extremely valuable contribution to completing this draft document.

12. References

12.1. Normative


12.2. Informative

[OAUTH] Initiative for Open AuTHentication
http://www.openauthentication.org


Appendix A. Fraud Extensions XML Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.openauthentication.org/thraud/protocol/v01/wd06"
xmlns:thraud="http://www.openauthentication.org/thraud/protocol/v01/wd06"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="FraudEventPayment" type="thraud:FraudEventPaymentType"/>
  <xs:element name="FraudEventTransfer" type="thraud:FraudEventTransferType"/>
  <xs:element name="FraudEventIdentity" type="thraud:FraudEventIdentityType"/>
  <xs:element name="FraudEventOther" type="thraud:FraudEventOtherType"/>
  <xs:complexType name="FraudEventPaymentType">
    <xs:sequence>
      <xs:element name="PayeeName" type="xs:string" minOccurs="0"/>
      <xs:element name="PostalAddress" type="xs:string" minOccurs="0"/>
      <xs:element name="PayeeAmount" type="thraud:AmountType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FraudEventTransferType">
    <xs:sequence>
      <xs:element name="BankID" type="xs:string" minOccurs="0"/>
      <xs:element name="AccountID" type="xs:string" minOccurs="0"/>
      <xs:element name="AccountType" type="thraud:AccountTypeType" minOccurs="0"/>
      <xs:element name="TransferAmount" type="thraud:AmountType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FraudEventIdentityType">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="IdentityComponent" type="iodef:ExtensionType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```
</xs:complexType>
<xs:complexType name="FraudEventOtherType">
  <xs:sequence>
    <xs:element name="OtherSigType" type="xs:string"
      minOccurs="1"/>
    <xs:element name="PayeeName" type="xs:string"
      minOccurs="0"/>
    <xs:element name="PostalAddress" type="xs:string"
      minOccurs="0"/>
    <xs:element name="BankID" type="xs:string"
      minOccurs="0"/>
    <xs:element name="AccountID" type="xs:string"
      minOccurs="0"/>
    <xs:element name="AccountType" type="thraud:AccountTypeType"
      minOccurs="0"/>
    <xs:element name="PayeeAmount" type="thraud:AmountType"
      minOccurs="0"/>
    <xs:element name="OtherSigDescription" type="xs:string"
      minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="AccountTypeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="brokerage"/>
    <xs:enumeration value="checking"/>
    <xs:enumeration value="corporate"/>
    <xs:enumeration value="mortgage"/>
    <xs:enumeration value="retirement"/>
    <xs:enumeration value="saving"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="AmountType">
  <xs:simpleContent>
    <xs:extension base="xs:decimal">
      <xs:attribute name="currency" type="xs:string"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
Appendix B. Example of a Thraud Report

<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document xmlns="urn:ietf:params:xml:ns:iodef-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:thraud= "http://www.openauthentication.org/fraud/protocol/v01/wd02"
xsi:schemaLocation="urn:ietf:params:xml:ns:iodef-1.0" lang="en">
<Incident purpose="reporting">
<IncidentID name="fraud.openauthentication.org">908711</IncidentID>
<ReportTime>2006-10-12T00:00:00-07:00</ReportTime>
<Assessment>
<Impact severity="high" completion="failed"/>
<Confidence rating="high"/>
</Assessment>
<Contact type="organization" role="creator">
<ContactName>Open Authentication</ContactName>
<Email>contact@example.com</Email>
</Contact>
<EventData>
<DetectTime>2006-10-12T07:42:21-08:00</DetectTime>
<Flow>
<System category="source">
<Node>
<Address category="ipv4-addr">192.0.2.53</Address>
</Node>
<Description>Source of numerous attacks</Description>
</System>
</Flow>
<AdditionalData dtype="xml">
<thraud:Fraud-Event-Transfer>
<thraud:Bank-ID>1234567</thraud:Bank-ID>
<thraud:Account-ID>3456789</thraud:Account-ID>
<thraud:Account-Type>saving</thraud:Account-Type>
<thraud:Transfer-Amount Currency="USD">10000</thraud:Transfer-Amount>
</thraud:Fraud-Event-Transfer>
</AdditionalData>
</EventData>
</Incident>
</IODEF-Document>

13. Authors’ Addresses

Primary point of contact (for sending comments and question):

David M’Raihi
VeriSign, Inc.
685 E. Middlefield Road       Phone: 1-650-426-3832
Mountain View, CA 94043 USA   Email: dmraihi@verisign.com

Other Authors’ contact information:

Sharon Boeyen,
Entrust Inc.,
1000 Innovation Drive,       Phone: 1-613-270-3181
Ottawa, ON, K2K 3E7           Email: sharon.boeyen@entrust.com

Michael Grandcolas
Grandcolas Consulting LLC.
247 Ocean Park Blvd.          Phone: 1-310-399-1747
Santa Monica, Ca 90405        Email: michael.grandcolas@hotmail.com

Siddharth Bajaj
VeriSign, Inc.
487 E. Middlefield Road       Phone: 1-650-426-3458
Mountain View, CA 94043 USA   Email: sbajaj@verisign.com

14. Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

15. Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.
Copies of IPR disclosures made to the IETF Secretariat and any 
assurances of licenses to be made available, or the result of an 
attempt made to obtain a general license or permission for the use 
of such proprietary rights by implementers or users of this 
specification can be obtained from the IETF on-line IPR repository 

The IETF invites any interested party to bring to its attention any 
copyrights, patents or patent applications, or other proprietary 
rights that may cover technology that may be required to implement 
this standard. Please address the information to the IETF at ietf-
ipr@ietf.org.