Internet Engineering Task Force (IETF)
Request for Comments: 5901
Category: Standards Track
ISSN: 2070-1721

P. Cain
The Cooper-Cain Group, Inc.
D. Jevans
The Anti-Phishing Working Group

July 2010

Extensions to the IODEF-Document Class for Reporting Phishing

Abstract

This document extends the Incident Object Description Exchange Format (IODEF) defined in RFC 5070 to support the reporting of phishing events, which is a particular type of fraud. These extensions are flexible enough to support information gleaned from activities throughout the entire electronic fraud cycle -- from receipt of the phishing lure to the disablement of the collection site. Both simple reporting and complete forensic reporting are possible, as is consolidating multiple incidents.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5901.

Table of Contents

1. Introduction

Deception activities, such as receiving an email purportedly from a bank requesting you to confirm your account information, are an expanding attack type on the Internet. The terms "phishing" and "fraud" are used interchangeably in this document to characterize broadly-launched social engineering attacks in which an electronic identity is misrepresented in an attempt to trick individuals into revealing their personal credentials (e.g., passwords, account numbers, personal information, ATM PINs, etc.). A successful phishing attack on an individual allows the phisher (i.e., the attacker) to exploit the individual's credentials for financial or other gain. Phishing attacks have morphed from directed email messages from alleged financial institutions to more sophisticated lures that may also include malware.

This document defines a data format extension to the Incident Object Description Exchange Format (IODEF) [RFC5070] that can be used to describe information about a phishing or other type of fraudulent incident. Sections 2 and 3 of this document provides an overview of the terminology and process of a phishing event. Section 4 introduces the high-level report format and how to use it. Sections 5 and 6 describe the data elements of the fraud extensions. The appendices include an XML schema for the extensions and a few example fraud reports.

The extensions defined in this document may be used to report the social engineering victim lure, the collection site, credential targeted ("spear") phishing, broad multi-recipient phishing, and other evolving Internet-based fraud attempts. Malware and other malicious software included within the lure may also be included within the report.

1.1. Why a Common Report Format Is Needed

To combat the rise in malicious activity on the Internet, service providers and investigative agencies are sharing more and more network and event data in a coordinated effort to identify perpetrators and compromised accounts, coordinate responses, and prosecute attackers. As the number of data-sharing parties increases, the number of party-specific tools, formats, and definitions multiply rapidly until they overwhelm the investigative and coordination abilities of those parties.

By using a common format, it becomes easier for an organization to engage in this coordination as well as correlation of information from multiple data sources or products into a cohesive view. As the number of data sources increases, a common format becomes even more important, since multiple tools would be needed to interpret the different sources of data. A big win in a common format is the ability to automate many of the analysis tasks and significantly speed up the response and prosecution activities.

1.2. Processing of Exchanged Data Not Defined

While the intended use of this specification is to facilitate data sharing between parties, the mechanics of this sharing process and its related political challenges are out of scope for this document.

1.3. Relation to the INCH IODEF Data Model

Instead of defining a new report format, this document defines an extension to [RFC5070]. The IODEF defines a flexible and extensible format and supports a granular level of specificity. These phishing and fraud extensions reuse subsets of the IODEF data model and, where appropriate, specify new data elements. Leveraging an existing specification allows for more rapid adoption and reuse of existing tools in organizations. For clarity, and in order to eliminate duplication, only the additional structures necessary for describing the exchange of phishing and e-crime activity are provided.

2. Terminology Used in This Document

Since many people use different but similar terms to mean the same thing, we use the following terminology in this document.

a. Phishing

   The overall process of identifying victims, contacting them via a lure, causing a victim to send a set of private credentials to a collection site, and storing those credentials is called phishing.

b. Fraud Event

   A fraud event is the combination of phishing and subsequent fraudulent use of the private credentials.

c. Lure

A lure is the decoy used to trick a victim into performing some activity, such as providing their private credentials. The lure relies on social engineering concepts to convince the victim that the lure is genuine and its instructions should be followed. A lure includes a pointer or link to a collection site.

d. Collection Site

The website, email box, SMS number, phone number, or other place where a phished victim sends their private credentials for later fraudulent use by a criminal.

e. Credentials

A credential is data that is transferred or presented to establish either a claimed identity or the authorizations of a system entity. Many websites require a user name and password -- combined, they are a credential -- to access sensitive content.

f. Message

Although primarily email, a lure can be transported via any messaging medium, such as instant messages, Voice over IP (VoIP), or text via an SMS service. The term "message" is used as a generic term for any of these transport mediums.

2.1. Requirements Language

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. Interesting Fraud Event Data

Before defining the structure of the IODEF extensions, we identify the "interesting" data in phishing and other fraudulent activities.

3.1. The Elements of a Phishing/Fraud Event

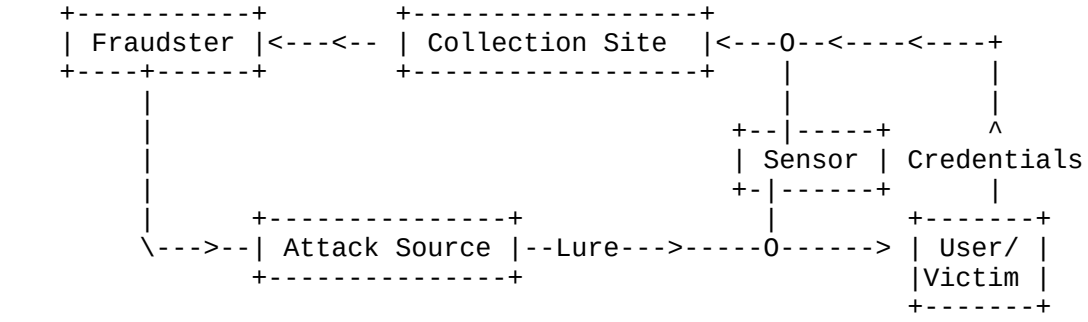

Figure 3.1. The Components of Internet Fraud

Internet-based phishing and fraud activities are normally comprised of at least six components:

1. The phisher, fraudster, or party perpetrating the fraudulent activity. Most times this party is not readily identifiable.

2. The attack source -- the source of the phishing email, virus, trojan, or other attack -- is masked in an enticing manner.

3. The lure used to trick the victim into responding.

4. The user, victim, or intended target of the fraud or phish.

5. The credentials, personal data, or other information the victim has surrendered to the phisher.

6. The collection site, where the victim sends their credentials or personal data if they have been duped by the lure of the phisher. This may be a website, mailbox, phone operator, or database.

If we take a holistic view of the attack, there are some additional components:

- The sensor -- the means by which the phish is detected. This element may be an intrusion detection system, firewall, filter, email gateway, or human analyst.

- A forensic or archive site (not pictured), where an investigator has copied or otherwise retained the data used for the fraud attempt or credential collection.

3.1.1. Fraudulent Activity Extensions to the IODEF-Document

Fraud events are reported in a fraud activity report, which is an instance of an XML IODEF-Document Incident element with added EventData and AdditionalData elements. The additional fields in the EventData specific to phishing and fraud are enclosed in a PhraudReport XML element. Fraudulent activity may include multiple emails, instant messages, or network messages, scattered over various times, locations, and methodologies. The PhraudReport within an EventData may include information about the email header and body, details of the actual phishing lure, correlation to other attacks, and details of the removal of the web server or credential collector. As a phishing attack may generate multiple reports to an incident team, multiple PhraudReports may be combined into one EventData structure, and multiple EventData structures may be combined into one incident report. One IODEF incident report may record one or more individual phishing events and may include multiple EventData elements.

This document defines new extension elements for the EventData IODEF XML elements and identifies those required in a PhraudReport. The appendices contain sample fraud activity reports and a complete schema.

The IODEF Extensions defined in this document comply with Section 4, "Extending the IODEF Format" in [RFC5070].

3.2. Useful Data Items in a Fraud Event

There are a number of subtle and non-obvious data to capture from a fraud event that make the event analysis and correlation with other events more useful. These data can be grouped into categories:

3.2.1. Data about the Lure

If a lure was presented as part of the fraud event, this category includes the original received lure, the means by which the lure was received (e.g., email, phone, or SMS), and the source addresses that sent the lure. Other useful data includes DNS data about the lure source, identification of any accompanying malware, and the brand name defrauded.

3.2.2. Credential Collection Site Data

The collection site contains victim identifications, along with copies of data supplied by the victims, such as account names or numbers, passwords, dates of birth, etc. This category of useful data includes these credentials, along with information about the

collection site itself, such as its type, site DNS data, DNS registrant data, and site physical location. The location and registrant information is particularly important if law enforcement assistance is expected. Additionally, an entire site archive can be gathered to allow a collector on a shared website to be disabled without impacting other users.

3.2.3. Detection Information

This is a non-obvious data category and contains data on how the lure or collection site was detected. Understanding how the lure was detected allows us to design and implement better detection systems.

3.2.4. Analysis Output

In an environment where time is critical, it is imperative that analysis from one party can be reliably explained to and shared with other investigative parties. This grouping includes data that an investigator found interesting or could be useful to others.

4. Fraud Activity Reporting via IODEF-Documents

A fraud activity report is an instance of an XML IODEF-Document with additional extensions and usage guidance, as specified in Section 4 of this document. These additional extensions are implemented through the PhraudReport XML element.

As described in the following subsections, reporting fraud activity has three primary components: choosing a report type, a format for the data, and how to check the correctness of the format.

4.1. Fraud Report Types

There are three actions relating to reporting phishing events. First, a reporter may *create* and exchange a new report on a new event. Secondly, a reporter may *update* a previously exchanged report to indicate new collection sites, site takedown information, or related activities. Lastly, a reporter may have realized that the report is in error or contains significant incorrect data and that the prudent reaction is to *delete* the report.

The three types of reports are denoted through the use of the ext-purpose attribute of an Incident element. A new report contains an empty or a "create" ext-purpose value; an updated report contains an ext-value value of "update"; a request for deletion contains a "delete" ext-purpose value. Note that this is actually an advisory marking for the report originator or recipient, as operating procedures in a report life cycle are very environment specific.

4.2. Fraud Report XML Representation

The IODEF Incident element ([RFC5070], Section 3.2) is summarized below. It and the rest of the data model presented in Section 4 is expressed in Unified Modeling Language (UML) syntax as used in the IODEF specification. The UML representation is for illustrative purposes only; elements are specified in XML as defined in Appendix A.

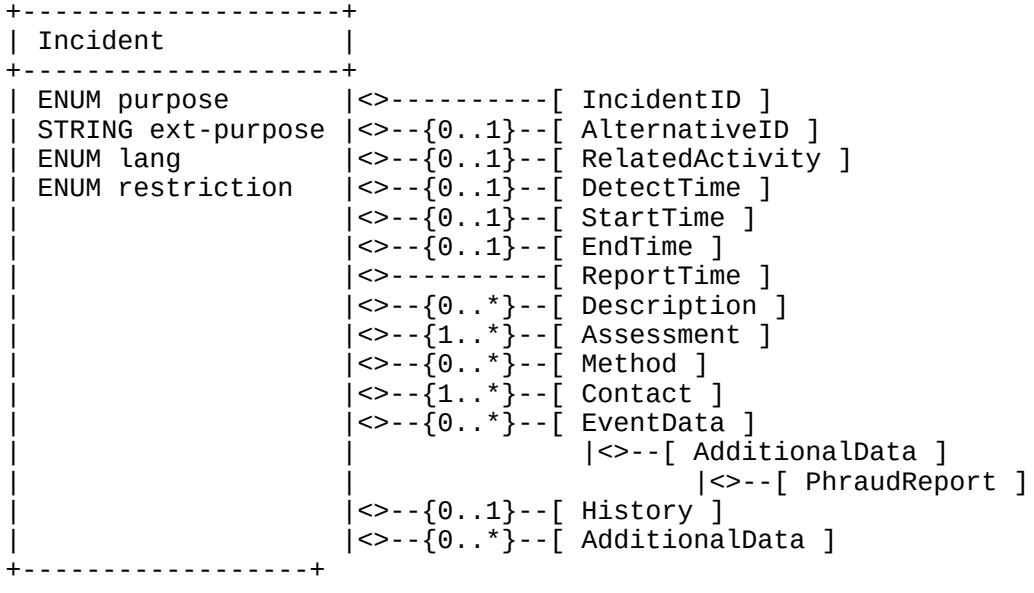

Figure 4.1. The IODEF XML Incident Element (Modified)

A fraud activity report is composed of one iodef:Incident element that contains one or more related PhraudReport elements embedded in the iodef:AdditionalData element of iodef:EventData. The PhraudReport element is added to the IODEF using its defined extension procedure documented in Section 5 of [RFC5070].

One IODEF-Document may contain information on multiple incidents with information for each incident contained within an iodef:Incident element ([RFC5070], Section 3.12).

4.3. Syntactical Correctness of Fraud Activity Reports

The fraud activity report MUST pass XML validation using the schema defined in [RFC5070] and the extensions defined in Appendix A of this document.

5. PhraudReport Element Definitions

A PhraudReport consists of an extension to the Incident.EventData.AdditionalData element with a dtype of "xml". The elements of the PhraudReport will specify information about the six components of fraud activity identified in Section 3.1. Additional forensic information and commentary can be added by the reporter as necessary to show relation to other events, to show the output of an investigation, or for archival purposes.

5.1. PhraudReport Structure

A PhraudReport element is structured as follows. The components of a PhraudReport are introduced in functional grouping, as some parameters are related and some elements may not make sense individually.

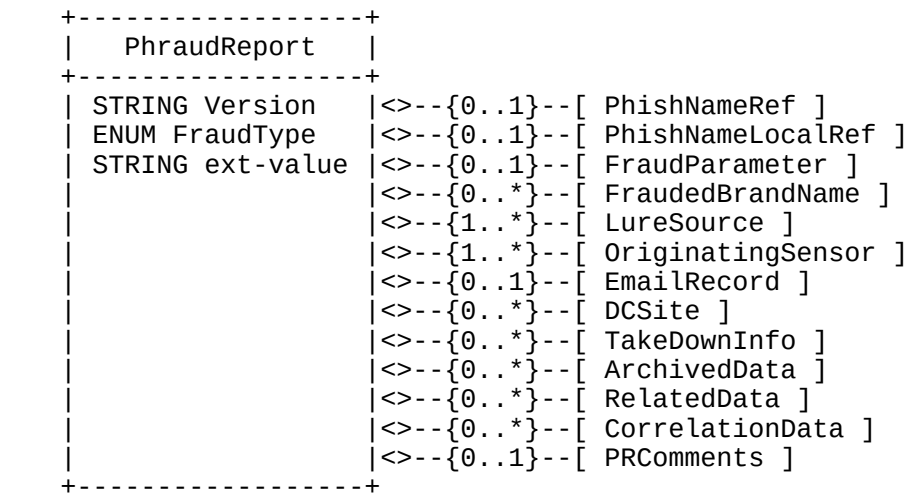

Figure 5.1. The PhraudReport Element

Relevant information about a phishing or fraud event is encoded into six components as follows:

a. The PhishNameRef and PhishNameLocalRef elements identify the fraud or class of fraud.

b. The LureSource element describes the source of the attack or phishing lure, including host information and any included malware.

c. The DCSite element describes the technical details of the credential collection site.

d. The OriginatingSensor element describes the means of detection.

The RelatedData, ArchivedData, and TakeDownInfo fields allow optional forensics and history data to be included.

A specific phish/fraud activity can be identified using a combination of the FraudType, FraudParameter, FraudedBrandName, LureSource, and PhishNameRef elements.

5.2. Reuse of IODEF-Defined Elements

Elements, attributes, and parameters defined in the base IODEF specification were used whenever possible in the definition of the PhraudReport XML element. This specification does not introduce any new variable types or encodings to the IODEF data model, but extends the IODEF Contact and System elements.

The data model schema contains a copy of the iodef:System element. Although we would like to just extend the System element, it is defined in RFC 5070 with an unable-to-extend anonymous type, so we copied the element, named its underlying type, and then generated the extension to it.

Note: Elements that are imported from the base IODEF specification are prefaced with an "iodef" XML namespace and are noted with the section defining that element in [RFC5070]. Each element in a PhraudReport is used as described in the following sections.

5.3. Element and Attribute Specification Format

The following sections describe the components of a PhraudReport XML element. Each description is structured as follows.

1. A terse XML-type identifier for the element or attribute.

2. An indication of whether the element or attribute is REQUIRED or optional. Mandatory items are noted as REQUIRED. If not specified, elements are optional. Note that when optional elements are included, they may REQUIRE specific sub-elements.

3. A description of the element or attribute and its intended use.

Elements that contain sub-elements or enumerated values are further sub-sectioned. Note that there is no "trickle-up" effect in elements. That is, the required elements of a sub-element are only populated if the sub-element is used.

5.4. Version Attribute

REQUIRED. STRING. The version shall be the value 0.06, to be compliant with this document.

5.5. FraudType Attribute

REQUIRED. One ENUM. The FraudType attribute describes the type of fraudulent activity described in this PhraudReport. The FraudType chosen determines the value of the FraudParameter filed. This field contains one of the following values:

1. phishing. The FraudParameter should be the subject line of the phishing lure email or value of a lure IM or VoIP message. This type is a standard phishing lure, usually sent as email, and is intended to exploit the recipient’s credentials for financial gain.

2. recruiting. The FraudParameter is the subject line of the recruit, or mule, email or message.

3. malware distribution. The FraudParameter is the email subject line of the phishing email. This type of email phish does not pose a risk of financial loss to the recipient, but lures the recipient to an infected site.

4. fraudulent site. This identifies a known fraudulent site that does not necessarily send spam but is used to show lures. The FraudParameter may be used to identify the website.

5. dnsspoof. This choice does not have a related FraudParameter. This value is used when a DNS system component responds with an untrue IP address for the requested domain name due to either cache poisoning, ID spoofing, or other manipulation of the DNS system.

6. archive. There is no required FraudParameter for this choice, although the FraudParameter of the original phish could be entered. The data archived from the phishing server is placed in the ArchivedData element.

7. other. This is used to identify not-yet-enumerated fraud types.

8. unknown. This choice may have an associated FraudParameter. It is used to cover confused cases.

9. ext-value. This choice identifies an unidentified FraudType. The FraudType should be captured in the ext-value attribute.

5.5.1. ext-value Attribute

OPTIONAL. This STRING may be populated with a FraudType that has not been predefined.

5.5.2. FraudParameter Element

Zero or one value of iodef:MLStringType. The contents of this element are dependent on the FraudType choice. It may be an email subject line, VoIP lure, link in an IM message, or a web URL. Note that some phishers add a number of random characters onto the end of a phish email subject line for uniqueness; reporters should delete those characters before insertion into the FraudParameter field.

5.6. PhishNameRef Element

Zero or one value of iodef:MLStringType. The PhishNameRef element is the common name used to identify this fraud event. It is often the name agreed upon by involved parties or vendors. Using this name can be a convenient way to reference the activity when collaborating with other parties, the media, or engaging in public education.

5.7. PhishNameLocalRef Element

Zero or one value of iodef:MLStringType. The PhishNameLocalRef element describes a local name or Unique-IDentifier (UID) that is used by various parties before a commonly agreed-upon term is adopted. This field allows a cross-reference from the submitting organization’s system to a central repository.

5.8. FraudedBrandName Element

Zero or more values of iodef:MLStringType. This is the identifier of the recognized brand name or company name used in the phishing activity (e.g., XYZ Semiconductor Corp).

5.9. LureSource Element

REQUIRED. One or more values. The LureSource element describes the source of the PhraudReport lure. It allows the specification of IP addresses, DNS names, domain registry information, and rudimentary support for the files that might be downloaded or registry keys modified by the crimeware.

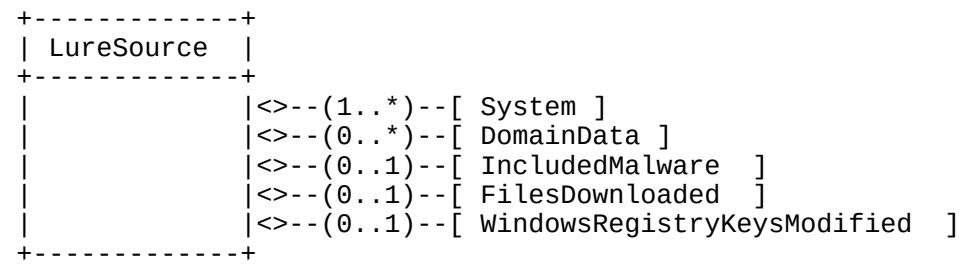

Figure 5.2. The LureSource Element

5.9.1. System Element

REQUIRED. One or more values of the iodef:System ([RFC5070], Section 3.15). The system element describes a particular host involved in the phishing activity. If the real IP address can be ascertained, it should be populated. A spoofed address may also be entered, and the spoofed attribute SHALL be set.

Multiple System elements may be used to identify the DNS name and IP address(es) of the lure source.

5.9.2. DomainData Element

Zero or more element values. The DomainData element describes the registration, delegation, and control of a domain used to source the lure and can identify the IP address associated with the System element URI. Capturing the domain data is very useful when investigating or correlating events.

The structure of a DomainData element is as follows:

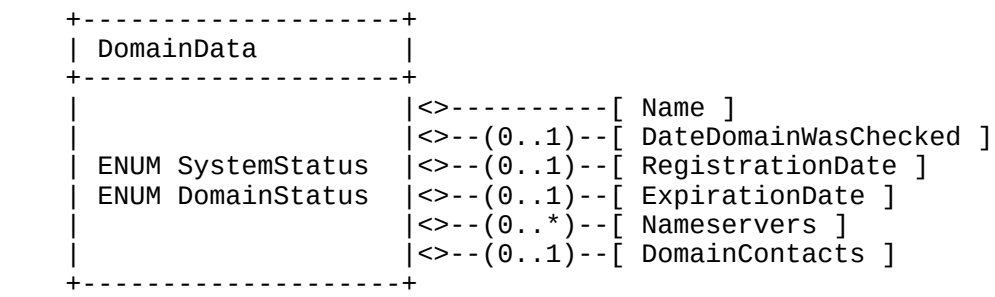

Figure 5.3. The DomainData Element

5.9.2.1. Name Element

REQUIRED. One value of iodef:MLStringType. The Name element contains the host DNS name used in this event. Its value should be the complete DNS host address; e.g., if an event targeted www.example.com, the value would be www.example.com.

5.9.2.2. DateDomainWasChecked Element

Zero or one value of DATETIME. This element includes the timestamp of when this domain data was checked and entered into this report, as many phishers modify their domain data at various stages of a phishing event.

5.9.2.3. RegistrationDate Element

Zero or one value of DATETIME. The RegistrationDate element shows the date of registration for a domain.

5.9.2.4. ExpirationDate Element

Zero or one value of DATETIME. The ExpirationDate element shows the date the domain will expire.

5.9.2.5. Nameservers Element

Zero or more values. These fields hold nameservers identified for this domain. Each entry is a sequence of DNSNameType and iodef:Address pairs, as specified below.

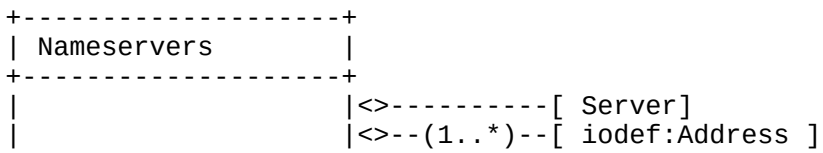

Figure 5.4. The Nameservers Element

The use of one Server value and multiple Address values is used to note multiple IP addresses associated with one DNS entry for the domain nameserver.

5.9.2.5.1. Server Element

One value of iodef:MLStringType. This field contains the DNS name of the domain nameserver.

5.9.2.5.2. iodef:Address Element

One or more values of iodef:Address. This field lists the IP address(es) associated with this Server element.

5.9.2.6. DomainContacts Element

REQUIRED. Choice of either a SameDomainContact or one or more Contact elements. The DomainContacts element allows the reporter to enter contact information supplied by the registrar or returned by whois queries. For efficiency of the reporting party, the domain contact information may be marked to be the same as another domain already reported using the SameDomainContact element.

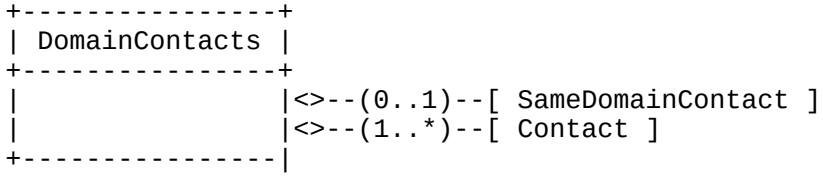

Figure 5.5. The DomainContacts Element

5.9.2.6.1. SameDomainContact Element

REQUIRED. One iodef:MLStringType. The SameDomainContact element is populated with a domain name if the contact information for this domain is identical to that name in this or another report. Implementors are cautioned to only use this element when the domain contact data returned by a registrar or registry is identical.

5.9.2.6.2. Contact Element

REQUIRED. One or more iodef:Contact elements. This element reuses and extends the iodef:Contact elements for its components. Each component may have zero or more values. If only the role attribute and the ContactName component are populated, the same (identical) information is listed for multiple roles.

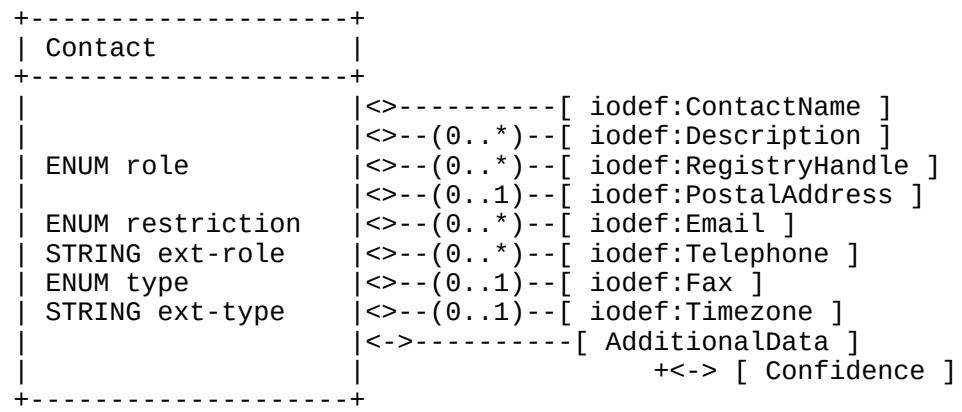

Figure 5.6. The Contact Element

Each Contact has optional attributes to capture the sensitivity and role for which the contact is listed. Elements reused from [RFC5070] are not discussed in this document.

5.9.2.6.2.1. Confidence Element

REQUIRED. ENUM. The Confidence element describes a qualitative assessment of the veracity of the contact information. This attribute is an extension to the iodef:Contact element and is defined in this document. There are five possible Confidence values, as follows.

1. known-fraudulent. This contact information has been previously determined to be fraudulent, as either non-existent physical information or containing real information not associated with this domain registration.

2. looks-fraudulent. The contact information has suspicious information included.

3. known-real. The contact information has been previously investigated or determined to be correct.

4. looks-real. The contact information does not arouse suspicion but has not been previously validated.

5. unknown. The reporter cannot make a value judgment on the contact data.

5.9.2.6.2.2. ext-role Attribute

REQUIRED. ENUM. The ext-role attribute is extended from the iodef: ext-role attribute with values identified in RFC 3982 [RFC3982]. The ext-value value of the role attribute should be used, with the ext-role attribute value chosen from one of the following values:

1. billingContacts

2. technicalContacts

3. administrativeContacts

4. legalContacts

5. zoneContacts

6. abuseContacts

7. securityContacts

8. otherContacts

9. hostingProvider. This contact is the hosting provider of this server. Although not in RFC 3982, it is useful in investigations to note where the server is located and who operates it. Load-balanced, multicast, or anycast servers may have multiple hostingProvider contact entries.

5.9.3. SystemStatus Attribute

REQUIRED. ENUM. The SystemStatus attribute assesses a system's involvement in this event. The value is chosen from this list:

1. spoofed. This domain or system did not participate in this event, but its address space or DNS name was simply used by another party.

2. fraudulent. The system is operated with fraudulent intentions, e.g., the domain name is a homophone.

3. innocent-hacked. The system was compromised by a third party and used in this event.

4. innocent-hijacked. The IP address or domain name was deliberately hijacked via BGP or DNS and used in this event to source the lure or host the collection site.

5. unknown. No conclusions are inferred from this event.

5.9.4. DomainStatus Attribute

ENUM. The DomainStatus attribute describes the registry status of a domain at the time of the report. The following enumerated list is taken from the "domainStatusType" of [RFC3982]. An extra "unknown" value was added in case the status is indeterminable.

1. reservedDelegation

2. assignedAndActive

3. assignedAndInactive

4. assignedAndOnHold

5. revoked

6. transferPending

7. registryLock

8. registrarLock

9. other

10. unknown

5.9.5. IncludedMalware Element

Zero or one value. The IncludedMalware element allows for the identification and optional inclusion of the actual malware that was part of the lure. The goal of this element is not to detail the characteristics of the malware but rather to allow for a convenient element to link malware to a phishing campaign.

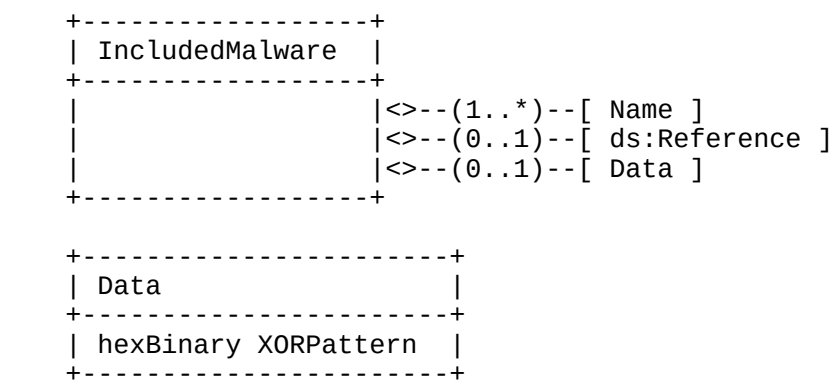

Figure 5.7. The IncludedMalware Element

5.9.5.1. Name Element

REQUIRED. One or more values of iodef:MLStringType. This field is used to identify the lure malware by its known name. Unnamed malware may be identified by a value of "unknown".

5.9.5.2. Reference Element

Zero or one value of the Reference. This optional field is used to hold the algorithm identification and value of a hash computed over the malware executable. This entire element is imported from [RFC3275]. Implementations SHOULD support the use of SHA-1 [SHA] as a DigestMethod.

5.9.5.3. Data Element

Zero or one value. The optional Data element is used to include the lure malware, which is encoded as a hexBinary type and XORed with a pattern to render it harmless.

5.9.5.3.1. XORPattern Attribute

One value of hexBinary. The Data element includes a 16-hexadecimal-character XORPattern attribute to support disabling the included malware to bypass anti-virus filters. The default value is 0x55AA55AA55AA55BB, which would be XORed with the malware datastring to recover the actual malware.

5.9.6. FilesDownloaded Element

Zero or one value of a sequence of File elements.

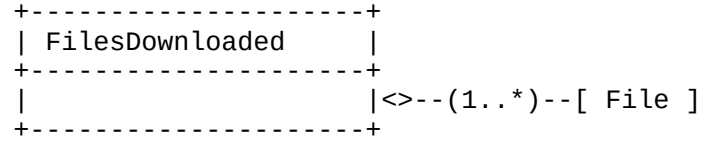

Figure 5.8. The FilesDownloaded Element

5.9.6.1. File Element

One or more values of iodef:MLStringType. The File element value is the name of a file downloaded by this lure.

5.9.7. WindowsRegistryKeysModified Element

One or more values of the Key sequence. The contents of the WindowsRegistryKeysModified element are sequences of Key elements.

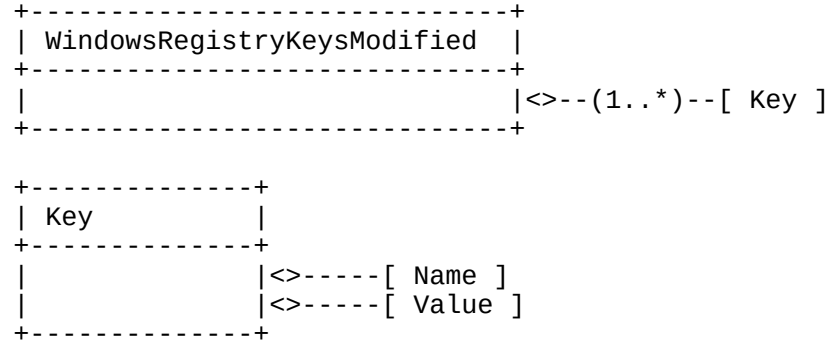

Figure 5.9. The WindowsRegistryKeysModified Element

5.9.7.1. Key Element

One or more sequences. The Key element is a sequence of Name and Value pairs representing an operating system registry key and its value. The key and value are encoded as in Microsoft .reg files [KB310516].

5.9.7.1.1. Name Element

One STRING, representing the Windows Operating System Registry Key Name. The value is encoded as in Microsoft .reg files, e.g., [HKEY_LOCAL_MACHINE\Software\Test\KeyName].

5.9.7.1.2. Value Element

One STRING, representing the value of the associated Key encoded as in Microsoft .reg files, e.g., REG_BINARY:01.

5.10. OriginatingSensor Element

REQUIRED. The OriginatingSensor element contains the identification and cognizant data of the network element that detected this fraud activity. Note that the network element does not have to be on the Internet itself (i.e., it may be a local Intrusion Detection System (IDS)), nor is it required to be mechanical (e.g., humans are allowed).

Multiple OriginatingSensor elements are allowed to support detection at multiple locations.

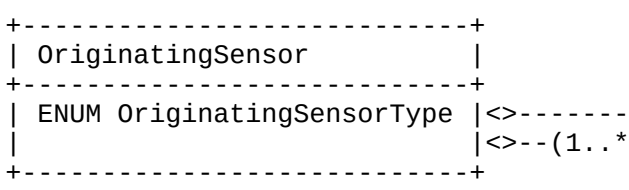

Figure 5.10. The OriginatingSensor Element

The OriginatingSensor requires a type value and identification of the entity that detected this fraudulent event.

5.10.1. OriginatingSensorType Attribute

REQUIRED. ENUM. The value is chosen from the following list, categorizing the function of this sensor:

1. web. A web server or service detected this event.

2. webgateway. A proxy, firewall, or other network gateway detected this event.

3. mailgateway. The event was detected via a mail gateway or filter.

4. browser. The event was detected at the user web interface or browser-type element.

5. ispsensor. The event was detected by an automated system in the network, such as Intrusion Detection System, Intrusion Protection System, or other Internet Service Provider device.

6. human. A non-automated system (e.g., a human, manual analysis, etc.) detected this event.

7. honeypot. The event was detected by receipt at a decoy device.

8. other. The detection was performed via a non-listed method.

5.10.2. DateFirstSeen Element

REQUIRED. DATETIME. This is the date and time that this sensor first saw this phishing activity.

5.10.3. iodef:System Element

REQUIRED. One or more values of iodef:System. This is identification information (such as the IP version, IP address, etc.) of the entity that detected this event. The ability to identify multiple detectors is supported.

5.11. The DCSite Element

Zero or more DCSite elements. The DCSite captures the type, identifier, location, and other pertinent information about the credential gathering process, or data collection site, used in the phishing incident. The data collection site is identified by four elements: the type of collector, the network location, information about its DNS domain, and a confidence factor. Further details about the domain, system, or owner of the DCSite can be inserted into the DomainData sub-element.

If the DCSite element is present, a value is required. Multiple DCSite elements are allowed to indicate multiple collection sites for a single collector. Multiple URLs pointing to the same DNS entry can be identified with multiple SiteURL elements.

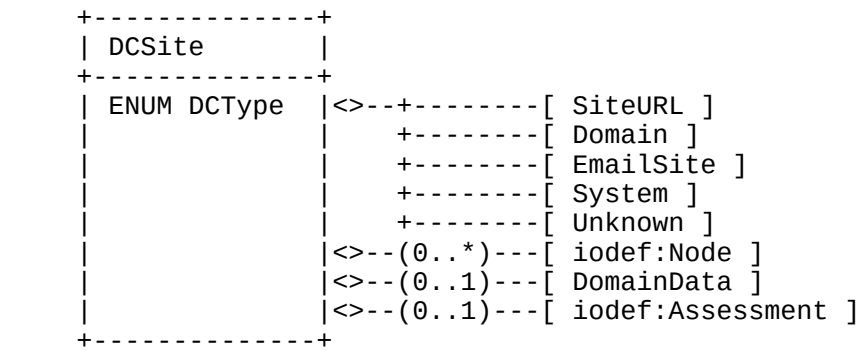

Figure 5.11. The DCSite Element

5.11.1. DCType Attribute

REQUIRED. ENUM. The DCType attribute identifies the method of data collection as determined through the analysis of the victim computer, lure, or malware. This attribute coupled with the DCSite content identifies the data collection site.

1. web. The user is redirected to a website to collect the data.

2. email. The victim sends an email with credentials enclosed.

3. keylogger. Some form of keylogger is downloaded to the victim.

4. automation. Other forms of automatic data collection, such as background Object Linking and Embedding (OLE) automation, are used to capture information on the user's machine.

5. unspecified.

5.11.2. DCSite Values

REQUIRED. The DCSite element contains the IP address, URL, email site, or other identifier of the credential or data collection site. The Domain choice may be used to identify entire "phishy" domains like those used for the RockPhish and related malware. Each DCSite element also includes a confidence attribute to convey the reporter's assessment of their confidence that this DCSite element is valid and involved with this event. The confidence value is a per-DCSite value, as multiple-site data collectors may have different confidence values.

The DCSite element is a choice of:

1. SiteURL. One value of iodef:MLStringType. This choice supports URIs and other web-based identifiers.

2. Domain. One value of iodef:MLStringType. This choice allows the entry of a DNS domain name.

3. EmailSite. One value of iodef:MLStringType. This choice includes an email address if the site used email communications.

4. iodef:Address. One value of iodef:Address element. This choice is used to capture the IP address of a site.

5. Unknown. One value of iodef:MLStringType. The unknown entry is used for exceptions to the preceding choices.

5.11.2.1. Confidence Attribute

One value of INTEGER. The confidence attribute is a value between 0 and 100, representing the reporter's certainty that this is a genuine phishing site. A value of 0 represents a false positive; a value of 100 signifies that the reporter has independently verified this site.

5.11.3. iodef:Node Element

Zero or more values of iodef:Node. This element is used to identify the IP address(es) or DNS names associated with the DCSite element value.

5.11.4. DomainData Element

Zero or one value of DomainData (Section 5.9.2). This element allows for the identification of data associated with the data collection site.

5.11.5. iodef:Assessment Element

Zero or one value of iodef:Assessment. This element is used to designate different confidence levels of multiple-site data collectors.

5.12. TakeDownInfo Element

Zero or more TakeDownInfo elements. This element identifies the agent or agency that performed the removal, DNS domain disablement, or ISP-blockage of the phish or fraud collector site. A PhraudReport may have multiple TakeDownInfo elements to support activities where

multiple takedown activities are involved on different dates. Note that the term "agency" is used to identify any party performing the blocking or removal, such as ISPs or private parties, and not just government entities.

The TakeDownInfo element allows one date element with multiple TakeDownAgency and Comment elements to support operations using multiple agencies.

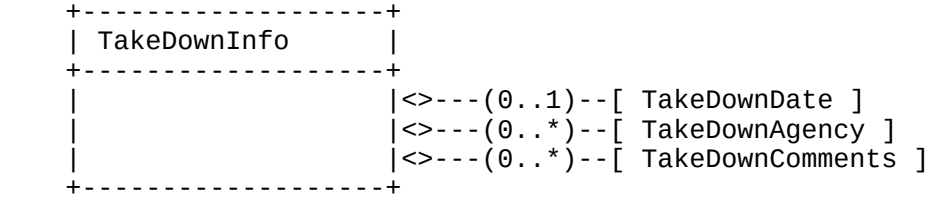

Figure 5.12. The TakeDownInfo Element

5.12.1. TakeDownDate

Zero or one value of DATETIME. This is the date and time that takedown of the collector site occurred.

5.12.2. TakeDownAgency

Zero or more iodef:MLStringType elements. This is a free-form string identifying the agency, corporation, or cooperative that performed the takedown.

5.12.3. TakeDownComments

Zero or more iodef:MLStringType elements. A free-form field to add any additional details of this takedown effort or to identify parties that assisted in the effort at an Internet Service Provider (ISP), Computer Emergency Response Team (CERT), or DNS registry.

5.13. ArchivedData Element

Zero or more values of the ArchivedData element are allowed.

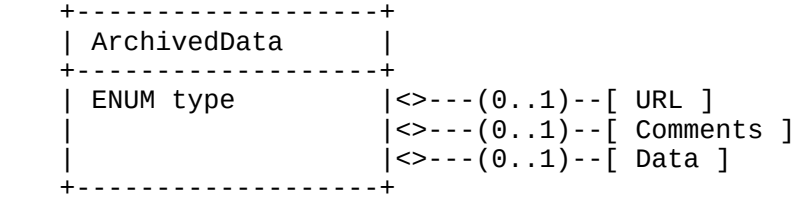

Figure 5.13. The ArchivedData Element

The ArchivedData URL element is populated with a pointer to the contents of a data collection site, base camp (i.e., development site), or other site used by a phisher. The ArchivedData Data element may also include a copy of the archived data recovered from a phishing system. This element will be populated when, for example, an ISP takes down a phisher's website and has copied the site data into an archive file.

There are four types of archives currently supported, as specified in the type field.

5.13.1. type Attribute

REQUIRED. This parameter specifies the type of site data pointed to by the ArchivedData URL element, from the following list:

1. collectionsite. The archive is a set of files from the collection site.

2. basecamp. The contents of a criminal development site are included in the archive.

3. sendersite. The archive is a set of files or data from a phishing lure sending site.

4. credentialInfo. The included archives are recovered private credentials.

5. unspecified. The archive contents do not fit into one of the above categories and will be described in the DataComments element.

5.13.2. URL Element

Zero or one value of anyURL. As the archive of an entire site can be quite large, the URL element points to an Internet-based server where the actual content of the site archive can be retrieved. Note that this element just points out where the archive is and does not include the entire archive in the report. This is the URL where the archive file is located.

5.13.3. Comments Element

Zero or one value of iodef:MLStringType. This field is a free-form area for comments on the archive and/or URL.

5.13.4. Data Element

Zero or one value of xs:Base64Binary. This field contains a base64-encoded version of the data described in the comment field above.

5.14. RelatedData Element

Zero or more values of anyURI. This element allows the listing of other websites or net sites that are related to this incident (e.g., victim site, etc.).

5.15. CorrelationData Element

Zero or more values of iodef:MLStringType. Any information that correlates this incident to other incidents can be entered here.

5.16. PRComments Element

Zero or one value of iodef:MLStringType. This field allows for any comments specific to this PhraudReport that do not fit in any other field.

5.17. EmailRecord Element

This element supports the inclusion of the actual email message received as a phishing lure. Inclusion of the actual mail message is supported by two methods: either the message may be included as one large string, or the header and body components may be dissected and included as a series of strings.

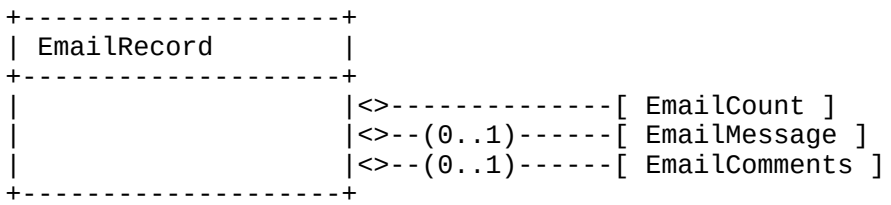

Figure 5.14. The EmailRecord Element

5.17.1. EmailCount Element

REQUIRED. INTEGER. This field enumerates the number of email messages identified in this record as detected by the reporter.

5.17.2. EmailMessage Element

Zero or one value of iodef:MLStringType. The entire SMTP mail message -- rfc822 header followed by body, as specified in [RFC5322] -- should be inserted as one large text string. In some communities, this combination is known as the message contents and full headers.

5.17.3. EmailComments Element

Zero or one value of iodef:MLStringType elements. This field contains comments or relevant data not placed elsewhere about the phishing email.

6. Mandatory IODEF and PhraudReport Elements

A report about fraud or phishing requires certain identifying information that is contained within the standard IODEF Incident data structure and the PhraudReport extensions. The following table identifies attributes required to be present in a compliant PhraudReport to report phishing or fraud. The required attributes are a combination of those required by the base IODEF element, as shown in Figure 6.1, and those required by this document, shown in Figure 6.2. Attributes identified as required SHALL be populated in conforming phishing activity reports.

A compliant IODEF PhraudReport SHALL contain the following elements and attributes:

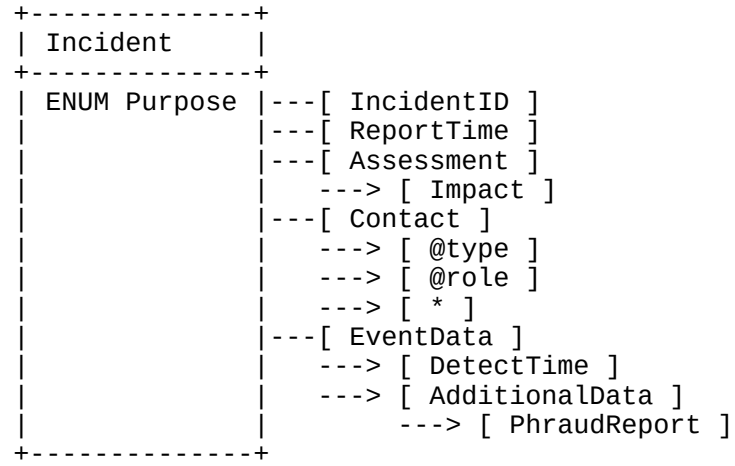

Figure 6.1. IODEF Required Classes for a PhraudReport

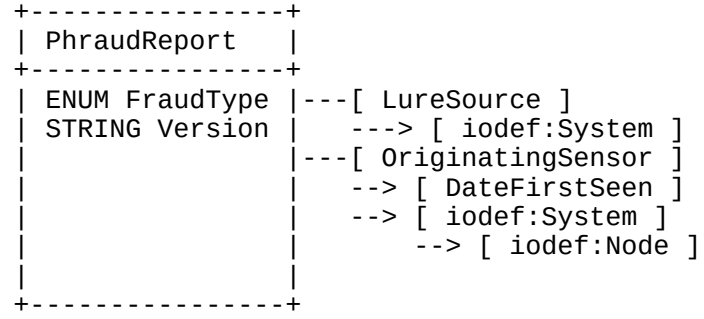

Figure 6.2. PhraudReport Required Elements

* Note that the iodef:Contact element is required, but none of its sub-elements are required. For proper XML correctness, one of the sub-elements is required; pick one.

6.1. Guidance on Usage

It may be apparent that the mandatory attributes for a PhraudReport make for a quite sparse report. As incident forensics and data analysis require detailed information, the originator of a PhraudReport SHOULD include any tidbit of information gleaned from the attack analysis. Information that is considered sensitive can be marked as such using the restriction parameter of each data element.

The reporting party is encouraged to provide more than just the minimally required data elements about an event in a PhraudReport. The additional information may be volatile and not recoverable in the future, and may be useful in answering investigation questions or in performing correlation with other reported events.

7. Security Considerations

This document specifies a format for encoding a particular class of security incidents appropriate for exchange across organizations. As merely a data representation, it does not directly introduce security issues. However, it is guaranteed that parties exchanging instances of this specification will have certain concerns. For this reason, the underlying message format and transport protocol used MUST ensure the appropriate degree of confidentiality, integrity, and authenticity for the specific environment.

Organizations that exchange data using this document are URGED to develop operating procedures that document the following areas of concern.

7.1. Transport-Specific Concerns

The critical security concerns are that phishing activity reports may be falsified or the PhraudReport may become corrupt during transit. In areas where transmission security or secrecy is questionable, the application of a digital signature and/or message encryption on each report will counteract both of these concerns. We expect that each exchanging organization will determine the need, and mechanism, for transport protection.

7.2. Using the iodef:restriction Attribute

In some instances, data values in particular elements may contain data deemed sensitive by the reporter. Although there are no general-purpose rules on when to mark certain values as "private" or "need-to-know" via the iodef:restriction attribute, the reporter is cautioned not to apply element-level sensitivity markings unless they believe the receiving party (i.e., the party they are exchanging the event report data with) has a mechanism to adequately safeguard and process the data as marked. For example, if the PhraudReport element is marked private and contains a phishing collector URL in the DCSite/SiteURL element, can that URL be included within a block list distributed to other parties? No guidance is provided here except to urge exchanging parties to review the IODEF and PhraudReport documents to decide on common marking rules.

8. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688].

Registration request for the IODEF phishing namespace:

URI: urn:ietf:params:xml:ns:iodef-phish-1.0

Registrant Contact: See the "Authors' Addresses" section of this document.

XML: None.

Registration request for the IODEF phishing extension XML schema:

URI: urn:ietf:params:xml:schema:iodef-phish-1.0

Registrant Contact: See the "Authors' Addresses" section of this document.

XML: See Appendix A, "Phishing Extensions XML Schema", of this document.

9. Contributors

The extensions are an outgrowth of the Anti-Phishing Working Group (APWG) activities in data collection and sharing of phishing and other e-crimeware. (The APWG has no relationship to an IETF working group.)

This document has received significant assistance from members of the IETF INCH working group and two groups addressing the phishing problem: members of the APWG and participants in the Financial Services Technology Consortium's Counter-Phishing project. A special thanks goes to the hardy people who supplied valuable feedback after using this format to report phishing.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3275] Eastlake, D., Reagle, J., and D. Solo, "(Extensible Markup Language) XML-Signature Syntax and Processing", RFC 3275, March 2002.

[RFC3982] Newton, A. and M. Sanz, "IRIS: A Domain Registry (dreg) Type for the Internet Registry Information Service (IRIS)", RFC 3982, January 2005.

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

[SHA] National Institute of Standards and Technology, U.S. Department of Commerce, "Secure Hash Standard", FIPS 180-2, August 2002.

10.2. Informative References

[KB310516] Microsoft Corporation, "How to add, modify, or delete registry subkeys and values by using a registration entries (.reg) file", December 2007.

[RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.

[RFC5322] Resnick, P., "Internet Message Format", RFC 5322, October 2008.

Appendix A. Phishing Extensions XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="urn:ietf:params:xml:ns:iodef-phish-1.0"
           xmlns="urn:ietf:params:xml:ns:iodef-1.0"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:phish="urn:ietf:params:xml:ns:iodef-phish-1.0"
           xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.0"
           xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
     schemaLocation=
"http://www.w3.org/TR/2002/REC-xmldsig-core-20020212
            /xmldsig-core-schema.xsd"/>

  <!--
  ============================================================
  ===  Top-Level Class:  PhraudReport                     ===
  ============================================================

  It is incorporated within an
  IODEF.Incident.EventData.AdditionalData element.

  All the top-level or major elements are defined as xs:types to make
  future extension easier.

  -->

  <xs:element name="PhraudReport">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="PhishNameRef"
                type="iodef:MLStringType"/>
        <xs:element minOccurs="0" name="PhishNameLocalRef"
                type="iodef:MLStringType"/>
        <xs:element minOccurs="0" name="FraudParameter"
                    type="iodef:MLStringType"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="FraudedBrandName" type="iodef:MLStringType"/>
        <xs:element maxOccurs="unbounded" minOccurs="1"
                    name="LureSource" type="phish:LureSource.type"/>
        <xs:element maxOccurs="unbounded" minOccurs="1"
                    name="OriginatingSensor"
                    type="phish:OriginatingSensor.type"/>
        <xs:element maxOccurs="1" minOccurs="0" name="EmailRecord"
                    type="phish:EmailRecord.type"/>
```

```
        <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="DCSite"  type="phish:DCSite.type"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
                    ref="phish:TakeDownInfo"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
                    ref="phish:ArchivedData"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="RelatedData" type="xs:anyURI"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="CorrelationData" type="iodef:MLStringType"/>
        <xs:element maxOccurs="1" minOccurs="0" name="PRComments"
                    type="iodef:MLStringType"/>
      </xs:sequence>

      <xs:attribute default="1.0" name="Version" use="optional"/>

      <xs:attribute name="FraudType" type="phish:FraudType.type"
                    use="required"/>

      <xs:attribute name="ext-value" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name="FraudType.type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="phishing"/>
      <xs:enumeration value="recruiting"/>
      <xs:enumeration value="malware distribution"/>
      <xs:enumeration value="fraudulent site"/>
      <xs:enumeration value="dnsspoof"/>
      <xs:enumeration value="archive"/>
      <xs:enumeration value="other"/>
      <xs:enumeration value="unknown"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <!--
============================================================
===           End of the Top-Level Element              ===
============================================================
-->
```

```
  <!--
  ============================================================
  ===           The LureSource Element                     ===
  ============================================================
  -->

  <xs:complexType mixed="false" name="LureSource.type">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="1"
              ref="iodef:System"/>

      <xs:element minOccurs="0" maxOccurs="unbounded"
              ref="phish:DomainData"/>

      <xs:element minOccurs="0" name="IncludedMalware"
                  type="phish:IncludedMalware.type"/>

      <xs:element minOccurs="0" name="FilesDownloaded">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" name="File"
                    type="iodef:MLStringType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>

      <xs:element minOccurs="0" name="WindowsRegistryKeysModified">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="Key">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Name" type="xs:string"/>
                  <xs:element name="Value" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

```
  <!--
  ===    LureSource sub-elements    ===
  -->

  <xs:complexType name="IncludedMalware.type">
    <xs:sequence>
      <xs:element name="Name"
              maxOccurs="unbounded" type="iodef:MLStringType"/>
      <xs:element minOccurs="0" ref="ds:Reference"/>
      <xs:element minOccurs="0" name="Data">
        <xs:complexType >
            <xs:simpleContent>
                  <xs:extension base="xs:hexBinary">
                      <xs:attribute default="55AA55AA55AA55BB"
                           name="XORPattern" type="xs:hexBinary"/>
                   </xs:extension>
            </xs:simpleContent>
       </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <!--
 ============================================================
 ===  The EmailRecord Element                             ===
 ============================================================
  -->

  <xs:complexType name="EmailRecord.type">
    <xs:sequence>
      <xs:element name="EmailCount" type="xs:integer"/>
      <xs:element maxOccurs="1" minOccurs="0" name="EmailMessage"
                    type="iodef:MLStringType"/>
      <xs:element maxOccurs="1" minOccurs="0" name="EmailComments"
                  type="iodef:MLStringType"/>
    </xs:sequence>
  </xs:complexType>

  <!--
 ============================================================
 ===  The Data Collection Site (DCSite) Info Element     ===
 ============================================================
  -->

  <xs:complexType name="DCSite.type">
    <xs:sequence>
      <xs:choice>
        <xs:element name="SiteURL">
```

```
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="iodef:MLStringType">
                <xs:attribute ref="phish:confidence"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>

        <xs:element name="Domain">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="iodef:MLStringType">
                <xs:attribute ref="phish:confidence"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>

        <xs:element name="EmailSite">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="iodef:MLStringType">
                <xs:attribute ref="phish:confidence"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>

        <xs:element name="System">
         <xs:complexType id="SystemType">
            <xs:sequence>
              <xs:element ref="iodef:Address"/>
            </xs:sequence>
            <xs:attribute ref="phish:confidence"/>
         </xs:complexType>
        </xs:element>

        <xs:element name="Unknown">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="iodef:MLStringType">
                <xs:attribute  ref="phish:confidence"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:choice>
```

```
      <xs:element ref="iodef:Node" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element minOccurs="0" ref="phish:DomainData"/>
      <xs:element minOccurs="0" ref="iodef:Assessment"/>
    </xs:sequence>

    <xs:attribute name="DCType" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="web"/>
          <xs:enumeration value="email"/>
          <xs:enumeration value="keylogger"/>
          <xs:enumeration value="automation"/>
          <xs:enumeration value="unspecified"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

  <!--
================================================
==== The Domain Data Element used in System =====
================================================
-->

  <xs:element name="DomainData">
    <xs:complexType id="DomainData.type">
      <xs:sequence>
        <xs:element maxOccurs="1"
                    name="Name" type="iodef:MLStringType"/>
        <xs:element maxOccurs="1" minOccurs="0"
                    name="DateDomainWasChecked" type="xs:dateTime"/>
        <xs:element maxOccurs="1" minOccurs="0" name="RegistrationDate"
                    type="xs:dateTime"/>
        <xs:element maxOccurs="1" minOccurs="0" name="ExpirationDate"
                    type="xs:dateTime"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
                   name="Nameservers">
          <xs:complexType id="Nameservers.type">
            <xs:sequence>
              <xs:element name="Server" type="iodef:MLStringType"/>
              <xs:element ref="iodef:Address" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice id="DomainContacts" maxOccurs="1" minOccurs="0">
          <xs:element name="SameDomainContact"
                      type="iodef:MLStringType"/>
```

```
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="1"
                        ref="iodef:Contact"/>
          </xs:sequence>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="SystemStatus">
        <xs:simpleType id="SystemStatus.type">
          <xs:restriction base="xs:string">
            <xs:enumeration value="spoofed"/>
            <xs:enumeration value="fraudulent"/>
            <xs:enumeration value="innocent-hacked"/>
            <xs:enumeration value="innocent-hijacked"/>
            <xs:enumeration value="unknown"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>

      <xs:attribute name="DomainStatus">
        <xs:simpleType id="DomainStatus.type">
          <xs:restriction base="xs:string">
            <xs:enumeration value="reservedDelegation"/>
            <xs:enumeration value="assignedAndActive"/>
            <xs:enumeration value="assignedAndInactive"/>
            <xs:enumeration value="assignedAndOnHold"/>
            <xs:enumeration value="revoked"/>
            <xs:enumeration value="transferPending"/>
            <xs:enumeration value="registryLock"/>
            <xs:enumeration value="registrarLock"/>
            <xs:enumeration value="other"/>
            <xs:enumeration value="unknown"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="Confidence">
    <xs:simpleType>
      <xs:restriction base="xs:nonNegativeInteger">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="100"/>
       </xs:restriction>
     </xs:simpleType>
  </xs:element>
```

```
<xs:attribute name="confidence">
  <xs:simpleType>
    <xs:restriction base="xs:nonNegativeInteger">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>

  <!--
=========================================================================
= ext-role Values for use within the DomainContact Contacts Element =
=========================================================================
-->

  <xs:simpleType name="ext-role">
    <xs:restriction base="xs:string">
      <xs:enumeration value="billingContacts"/>
      <xs:enumeration value="technicalContacts"/>
      <xs:enumeration value="administrativeContacts"/>
      <xs:enumeration value="legalContacts"/>
      <xs:enumeration value="zoneContacts"/>
      <xs:enumeration value="abuseContacts"/>
      <xs:enumeration value="securityContacts"/>
      <xs:enumeration value="otherContacts"/>
      <xs:enumeration value="hostingProvider"/>
    </xs:restriction>
  </xs:simpleType>

  <!--
================================================
===  The OriginatingSensor Data Element       ===
================================================
-->

  <xs:complexType name="OriginatingSensor.type">
    <xs:sequence>
      <xs:element name="DateFirstSeen" type="xs:dateTime"/>
      <xs:element maxOccurs="unbounded" minOccurs="1"
                  ref="iodef:System"/>
    </xs:sequence>

    <xs:attribute name="OriginatingSensorType" use="required">
      <xs:simpleType id="OriginatingSensorType.type">
        <xs:restriction base="xs:NMTOKENS">
          <xs:enumeration value="web"/>
          <xs:enumeration value="webgateway"/>
          <xs:enumeration value="mailgateway"/>
```

```
          <xs:enumeration value="browser"/>
          <xs:enumeration value="ispsensor"/>
          <xs:enumeration value="human"/>
          <xs:enumeration value="honeypot"/>
          <xs:enumeration value="other"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

  <!--
=======================================================
===            The TakeDown Data Structure          ===
=======================================================
-->

  <xs:element name="TakeDownInfo" type="phish:TakeDownInfo.type"/>

  <xs:complexType name="TakeDownInfo.type">
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="0" name="TakeDownDate"
                  type="xs:dateTime"/>

      <xs:element maxOccurs="unbounded" minOccurs="0"
              name="TakeDownAgency"  type="iodef:MLStringType"/>

      <xs:element maxOccurs="unbounded" minOccurs="0"
              name="TakeDownComments"  type="iodef:MLStringType"/>
    </xs:sequence>
  </xs:complexType>

  <!--
=========================================================
===         The ArchivedData Element                  ===
=========================================================
-->
  <xs:element name="ArchivedData" type="phish:ArchivedData.type"/>

  <xs:complexType name="ArchivedData.type">
    <xs:sequence>
      <xs:element minOccurs="0" name="URL" type="xs:anyURI"/>
      <xs:element minOccurs="0" name="Comments"
              type="iodef:MLStringType"/>
      <xs:element maxOccurs="1" minOccurs="0" name="Data"
                  type="xs:base64Binary"/>
    </xs:sequence>
```

```
    <xs:attribute name="type" use="required">
      <xs:simpleType id="ArchivedDataType.type">
        <xs:restriction base="xs:NMTOKENS">
          <xs:enumeration value="collectionsite"/>
          <xs:enumeration value="basecamp"/>
          <xs:enumeration value="sendersite"/>
          <xs:enumeration value="credentialInfo"/>
          <xs:enumeration value="unspecified"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

</xs:schema>
```

Appendix B. Example Virus Report

This section shows a received electronic mail message that included a virus in a zipped attachment and a report that was generated for that message.

B.1. Received Email

```
From: support@example.com
Sent: Friday, June 10, 2005 3:52 PM
To: someone@example.com
Subject: Account update

To:          someone@example.com
Date:      Sun, 10 June 2005 3:52:44 +0200


We would like to inform you that we have released a new version of our
Customer Form.  This form is required to be completed by all customers.

Please follow these steps:

1.Open the form at http://www.example.com/customerservice/cform.php
<http://www.2.example.com/customerservice/cform.php
        &amp;amp;email=(someone@example.com)> .
2.Follow given instructions.

Thank you,
Our Support Team
```

B.2. Generated Report

NOTE: Some wrapping and folding liberties have been applied to fit it into the margins.

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document lang="en-US"
  xmlns:phish="urn:ietf:params:xml:ns:iodef-phish-1.0"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
  xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.0">
<Incident purpose="reporting" ext-purpose="create">
  <IncidentID name="example.com">PAT2005-06</IncidentID>
  <ReportTime>2005-06-22T08:30:00-05:00</ReportTime>
  <Description>This is a test report from actual data.
   </Description>
  <Assessment>
    <Impact type="social-engineering"/>
    <Confidence rating="high"/>
  </Assessment>
  <Contact role="creator" type="person">
    <ContactName>patcain</ContactName>
    <Email>pcain@coopercain.com</Email>
  </Contact>
  <EventData>
    <DetectTime>2005-06-21T18:22:02-05:00</DetectTime>
    <AdditionalData dtype="xml">
    <phish:PhraudReport FraudType="phishing">
      <phish:FraudParameter>
       Subject: Account Update
      </phish:FraudParameter>
      <phish:FraudedBrandName>Cooper-Cain
      </phish:FraudedBrandName>
      <phish:LureSource>
        <System category="source">
          <Node>
           <Address>192.0.2.18</Address>
          </Node>
        </System>
        <phish:IncludedMalware>
          <phish:Name>W32.Mytob.EA@mm</phish:Name>
        </phish:IncludedMalware>
      </phish:LureSource>
      <phish:OriginatingSensor OriginatingSensorType="human">
        <phish:DateFirstSeen>2005-06-10T15:52:11-05:00
        </phish:DateFirstSeen>
        <System>
          <Node>
            <Address>192.0.2.13</Address>
```

```
          </Node>
        </System>
      </phish:OriginatingSensor>
      <phish:EmailRecord>
        <phish:EmailCount>1</phish:EmailCount>
        <phish:EmailMessage>
Return-path: &lt;support@example.com&gt;
 Envelope-to: someone@example.com
Delivery-date: Fri, 10 Jun 2005:52:11-0400
Received: from dsl18-2-0-192.dsl.example.net([192.0.2.18]
 helo=example.com) by mail06.example.com esmtp (Exim) id
 1DgpXy-0002Ua-IR for someone@example.com;,
 10 Jun 2005 15:52:10-0400
From: support@example.com
To: someone@example.com
Subject: Account Update
Date: Fri, 10 Jun 2005 12:52:00 -0700
MIME-Version: 1.0
Content Type: text/plain;
        charset="Windows-1251"
X-Priority: 3MSMail-Priority: Normal
X-EN-OrigIP: 192.0.2.18
EN-OrigHost: dsl18-2-0-192.dsl.example.net
Spam-Checker-Version: SpamAssassin 3.0.2 (2004-11-16)
 on.example.net
X-Spam-Level: ***** X-Spam-Status: No,
 score=5.6 required=6.0 tests=BAYES_95,CABLEDSL,HTML_20_30,
 HTML_MESSAGE,MIME_HTML_ONLY,MISSING_MIMEOLE,
 NO_REAL_NAME,
 PRIORITY_NO_NAME autolearn=disabled version=3.0.2

From:support@example.com
Sent: Friday, June 10, 2005 3:52 PM
Subject: Account update

To:          someone@example.com
Date:      Sun, 10 June 2005 3:52:44 +0200


We would like to inform you that we have released a new version of our
Customer Form.  This form is required to be completed by all customers.

Please follow these steps:

1.Open the form at http://www.example.com/customerservice/cform.php
&lt;http://www.2.example.com/customerservice/cform.php
        &amp;email=(someone@example.com)> .
2.Follow given instructions.
```

```
 Thank you,
 Our Support Team
            </phish:EmailMessage>
          </phish:EmailRecord>
        </phish:PhraudReport>
      </AdditionalData>
      </EventData>
    </Incident>
 </IODEF-Document>
```

Appendix C. Sample Phishing Report

A sample report generated from a received electronic mail phishing message in shown in this section.

C.1. Received Lure

```
Return-path: <service@example.com>
Envelope-to: pcain@example.com
Delivery-date: Tue, 13 Jun 2006 05:37:22 -0400
Received: from mail15.example.com ([10.1.1.161]
 helo=mail15.example.com)
 by mailscan38.example.com with esmtp (Exim)
 id 1Fq5Kr-0005wU-LT for pcain@example.com; Tue, 13 Jun 2006
 05:37:21 -0400
Received: from [192.0.2.61] (helo=TSI)
by mail15.example.com with
 esmtp (Exim) id 1Fq5Bj-0006dv-6b
for pcain@example.com; Tue, 13 Jun 2006 05:37:21 -0400
Received: from User ([192.0.2.157]) by TSI with
 Microsoft SMTPSVC(5.0.2195.6713);
Tue, 13 Jun 2006 02:24:30 -0400
Reply-To: <nospam@example.org>
From: "company"<service@example.com>
Subject: * * * Update & Verify Your Example Company Account * * *
Date: Tue, 13 Jun 2006 02:36:34 -0400
MIME-Version: 1.0
Content-Type: text/html; charset="Windows-1251"
Content-Transfer-Encoding: 7bit
X-Priority: 1
X-MSMail-Priority: High
X-Mailer: Microsoft Outlook Express 6.00.2600.0000
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2600.0000
Bcc:
Message-ID: <TSIlYbvhBISmT6QcWY90000085f@TSI>
X-OriginalArrivalTime: 13 Jun 2006 06:24:30.0218 (UTC)
FILETIME=[072A66A0:01C68EB2]
X-EN-OrigSender: service@example.com
```

```
X-EN-OrigIP: 192.0.2.1
X-EN-OrigHost: unknown

Company<http://www.example.com/images/company_logo.gif>
 <http://www.example.com/images/pixel.gif>
 <http://www.example.com/images/pixel.gif>
 <http://www.example.com/images/pixel.gif>
Account Update Request

Dear Example. member:,

You are receiving this notification because company is required by
law to notify you, that you urgently need to update your online
account statement, due to high risks of fraud intentions.

The updating of your example account can be done at any time by
clicking on the link shown below
http://www.example.com/cgi-bin/webscr?cmd=_login-run
<http://192.0.2.41:8080/.cgi-bin/.webscr/.secure-
login/%20/%20/.payp
al.com/index.htm>

Once you log in, update your account information.
After updating your account, click on the History sub tab of your
Account Overview page to see your most recent statement.

If you need help with your password, click the Help link that is at
the upper righthand side of the company website.  To report errors
in your statement or make inquiries, click the Contact Us link in the
footer on any page of the company website, call our Customer Service
center at (999) 555-0167, or write us at:

Company, Inc.
P.O. Box 0
Anytown, MA 00000

Sincerely,

Big Example Company

 <http://www.example.com/images/dot_row_long.gif>
```

C.2. Phishing Report

```
  <?xml version="1.0" encoding="UTF-8"?>
  <IODEF-Document xmlns:phish="urn:ietf:params:xml:ns:iodef-phish-1.0"
      xmlns="urn:ietf:params:xml:ns:iodef-1.0"
      xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.0" lang="en-US">
   <Incident purpose="mitigation" ext-purpose="create"
    restriction="private">
    <IncidentID name="example.com">CC200600000002</IncidentID>
    <ReportTime>2006-06-13T21:14:56-05:00</ReportTime>
    <Description>This is a sample phishing email received report.
          The phish was actually received as is.</Description>
    <Assessment>
     <Impact severity="high" type="social-engineering"/>
     <Confidence rating="numeric">85</Confidence>
    </Assessment>
    <Contact role="creator" type="person">
     <ContactName>patcain</ContactName>
     <Email>pcain@example.com</Email>
    </Contact>
    <EventData>
     <DetectTime>2006-06-13T05:37:21-04:00</DetectTime>
     <AdditionalData dtype="xml">
      <phish:PhraudReport FraudType="phishing">
       <phish:FraudParameter>
         * * * Update &amp;amp; Verify Your Company Account * * *
       </phish:FraudParameter>
       <phish:FraudedBrandName>company</phish:FraudedBrandName>
       <phish:LureSource>
        <System category="source">
         <Node>
          <Address>192.0.2.4</Address>
         </Node>
        </System>
       </phish:LureSource>
       <phish:OriginatingSensor OriginatingSensorType="mailgateway">
       <phish:DateFirstSeen>
              2006-06-13T05:37:22-04:00</phish:DateFirstSeen>
        <System>
         <Node>
          <NodeRole category="mail"/>
         </Node>
        </System>
       </phish:OriginatingSensor>
```

```
      <phish:EmailRecord>
       <phish:EmailCount>1</phish:EmailCount>
       <phish:EmailMessage>
  Return-path: &lt;service@example.com>
  Envelope-to: pcain@example.com
  Delivery-date: Tue, 13 Jun 2006 05:37:22 -0400
  Received: from mail15.example.com ([10.1.1.161]
   helo=mail15.example.com)
   by mailscan38.example.com with esmtp (Exim)
   id 1Fq5Kr-0005wU-LT for pcain@example.com; Tue, 13 Jun 2006
   05:37:21 -0400
  Received: from [192.0.2.61] (helo=TSI)
  by mail15.example.com with
   esmtp (Exim) id 1Fq5Bj-0006dv-6b
  for pcain@example.com; Tue, 13 Jun 2006 05:37:21 -0400
  Received: from User ([192.0.2.157]) by TSI with
   Microsoft SMTPSVC(5.0.2195.6713);
  Tue, 13 Jun 2006 02:24:30 -0400
  Reply-To: &lt;nospam@example.org>
  From: "company"&lt;service@example.com>
  Subject: * * * Update &amp; Verify Your Example Company Account * * *
  Date: Tue, 13 Jun 2006 02:36:34 -0400
  MIME-Version: 1.0
  Content-Type: text/html; charset="Windows-1251"
  Content-Transfer-Encoding: 7bit
  X-Priority: 1
  X-MSMail-Priority: High
  X-Mailer: Microsoft Outlook Express 6.00.2600.0000
  X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2600.0000
  Bcc:
  Message-ID: &lt;TSIlYbvhBISmT6QcWY90000085f@TSI>
  X-OriginalArrivalTime: 13 Jun 2006 06:24:30.0218 (UTC)
  FILETIME=[072A66A0:01C68EB2]
  X-EN-OrigSender: service@example.com
  X-EN-OrigIP: 192.0.2.1
  X-EN-OrigHost: unknown

  &lt;img src="http://www.example.com/images/company_logo.gif"&gt;
  &lt;img src="http://www.example.com/images/pixel.gif"&gt;
  &lt;img src="http://www.example.com/images/pixel.gif"&gt;
  &lt;img src="http://www.example.com/im/pixel.gif"&gt;
  Account Update Request

  Dear Example. member:,
  You are receiving this notification because company is required by
  law to notify you, that you urgently need to update your online
  account statement, due to high risks of fraud intentions.
```

```
  The updating of your example account can be done at any time by
  clicking on the link shown below
  &lt;a href="http://192.0.2.41:8080/.cgi-bin/.webscr/.secure-
  login/%20/%20/.example.com/index.htm">
  http://www.example.com/cgi-bin/webscr?cmd=_login-run &lt;/a>

  Once you log in,update your account information.
  After updating your account click on the History sub tab of your
  Account Overview page to see your most recent statement.

  If you need help with your password, click the Help link which is at
  the upper right hand side of the company website.  To report errors in
  your statement or make inquiries, click the Contact Us link in the
  footer on any page of the company website, call our Customer Service
  center at (999) 555-0167, or write us at:

  Company, Inc.
  P.O. Box 0
  Anytown, MA 00000

  Sincerely,

  Big Example Company

   &lt;img src="http://www.example.com/images/dot_row_long.gif">
  </phish:EmailMessage>
       </phish:EmailRecord>
       <phish:DCSite DCType="web">
        <phish:SiteURL>http://190.0.2.41:8080/.cgi-bin/.webscr/.secure-
           login/%20%20/.example.com/index.htm</phish:SiteURL>
        <phish:DomainData DomainStatus="assignedAndActive"
          SystemStatus="unknown">
         <phish:Name>bad.example.com</phish:Name>
         <phish:DateDomainWasChecked>2006-06-14T13:05:00-05:00
         </phish:DateDomainWasChecked>
         <phish:RegistrationDate>
                   2000-12-13T00:00:00</phish:RegistrationDate>
         <phish:Nameservers>
          <phish:Server>ns1.example.net</phish:Server>
          <Address>192.0.2.18</Address>
         </phish:Nameservers>
        </phish:DomainData>
       </phish:DCSite>
      </phish:PhraudReport>
     </AdditionalData>
    </EventData>
   </Incident>
  </IODEF-Document>
```

Authors' Addresses

Patrick Cain
The Cooper-Cain Group, Inc.
P.O. Box 400992
Cambridge, MA 02140
USA

EMail: pcain@coopercain.com

David Jevans
The Anti-Phishing Working Group
5150 El Camino Real, Suite A20
Los Altos, CA 94022
USA

EMail: dave.jevans@antiphishing.org