Network Working Group
Request for Comments: 4767
Category: Experimental

B. Feinstein
SecureWorks, Inc.
G. Matthews
CSC/NASA Ames Research Center
March 2007

The Intrusion Detection Exchange Protocol (IDXP)

Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This memo describes the Intrusion Detection Exchange Protocol (IDXP), an application-level protocol for exchanging data between intrusion detection entities. IDXP supports mutual-authentication, integrity, and confidentiality over a connection-oriented protocol. The protocol provides for the exchange of IDMEF messages, unstructured text, and binary data. The IDMEF message elements are described in RFC 4765, "The Intrusion Detection Message Exchange Format (IDMEF)", a companion document of the Intrusion Detection Exchange Format Working Group (IDWG) of the IETF.

Table of Contents

1. Introduction

IDXP is specified, in part, as a Blocks Extensible Exchange Protocol (BEEP) [4] "profile". BEEP is a generic application protocol framework for connection-oriented, asynchronous interactions. Features such as authentication and confidentiality are provided through the use of other BEEP profiles. Accordingly, many aspects of IDXP (e.g., confidentiality) are provided within the BEEP framework.

1.1. Purpose

IDXP provides for the exchange of IDMEF [2] messages, unstructured text, and binary data between intrusion detection entities. Addressing the security-sensitive nature of exchanges between intrusion detection entities, underlying BEEP security profiles should be used to offer IDXP the required set of security properties. See Section 5 for a discussion of how IDXP fulfills the IDWG communications protocol requirements. See Section 11 for a discussion of security considerations.

IDXP is primarily intended for the exchange of data created by intrusion detection entities. IDMEF [2] messages should be used for the structured representation of this intrusion detection data, although IDXP may be used to exchange unstructured text and binary data.

1.2. Profiles

There are several BEEP profiles discussed, the first of which we define in this memo:

The IDXP Profile

The TUNNEL Profile [3]

The Simple Authentication and Security Layer (SASL) Family of Profiles (see Section 4.1 of [4])

The TLS Profile (see Section 3.1 of [4])

1.3. 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 BCP 14, RFC 2119 [1].

Throughout this memo, the terms "analyzer" and "manager" are used in the context of the Intrusion Detection Message Exchange Requirements [5]. In particular, Section 2.2 of [5] defines a collection of intrusion detection terms.

The terms "peer", "initiator", "listener", "client", and "server", and the characters "I", "L", "C", and "S" are used in the context of BEEP [4]. In particular, Section 2.1 of BEEP discusses the roles that a BEEP peer may perform.

The term "Document Type Definition" is abbreviated as "DTD" and is defined in Section 2.8 of the Extensible Markup Language (XML) [7].

Note that the term "proxy" is specific to IDXP and does not exist in the context of BEEP. The term "intrusion detection" is abbreviated as "ID".

2. The Model

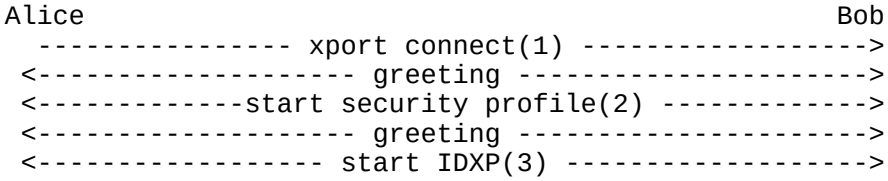
2.1. Connection Provisioning

Intrusion detection entities using IDXP to transfer data are termed IDXP peers. Peers can exist only in pairs, and these pairs communicate over a single BEEP session with one or more BEEP channels opened for transferring data. Peers are either managers or analyzers, as defined in Section 2.2 of [5].

The relationship between analyzers and managers is potentially many-to-many. That is, an analyzer MAY communicate with many managers; similarly, a manager MAY communicate with many analyzers. Likewise, the relationship between different managers is potentially many-to-many, so that a manager MAY receive the alerts sent by a large number of analyzers by receiving them through intermediate managers. Analyzers MUST NOT establish IDXP exchanges with other analyzers.

An IDXP peer wishing to establish IDXP communications with another IDXP peer does so by opening a BEEP channel, which may entail initiating a BEEP session. A BEEP security profile offering the required security properties SHOULD initially be negotiated (see Section 11 for a discussion of security considerations). Following the successful negotiation of the BEEP security profile, IDXP greetings are exchanged and connection provisioning proceeds.

In the following sequence, the peer ’Alice’ initiates an IDXP exchange with the peer ’Bob’.



Notes:

(1) 'Alice' initiates a transport connection to 'Bob', triggering the exchange of BEEP greeting messages.

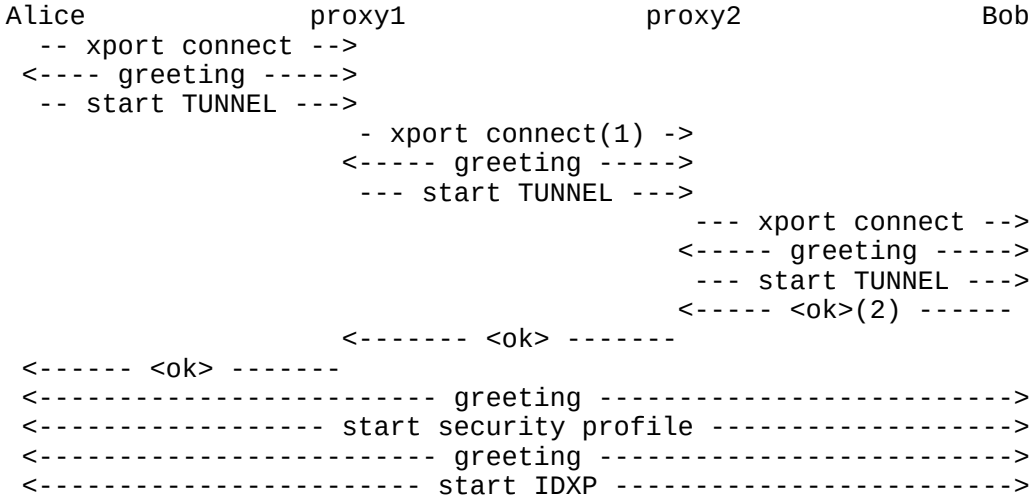
(2) Both entities negotiate the use of a BEEP security profile.

(3) Both entities negotiate the use of the IDXP profile.

In between a pair of IDXP peers may be an arbitrary number of proxies. A proxy may be necessary for administrative reasons, such as running on a firewall to allow restricted access. Another use might be one proxy per company department, which forwards data from the analyzer peers in the department onto a company-wide manager peer.

A BEEP tuning profile MAY be used to create an application-layer tunnel that transparently forwards data over a chain of proxies. The TUNNEL profile [3] SHOULD be used for this purpose; see [3] for more detail concerning the options available to set up an application-layer tunnel using TUNNEL, and see Section 11.1 for a discussion of TUNNEL-related security considerations. TUNNEL MUST be offered as a tuning profile for the creation of application-layer tunnels. The TUNNEL profile MUST offer the use of some form of SASL authentication (see Section 4.1 of [4]). Once a tunnel has been created, a BEEP security profile offering the required security properties SHOULD be negotiated, followed by negotiation of the IDXP profile.

The following sequence shows how TUNNEL might be used to create an application-layer tunnel through which IDXP would operate. A peer 'Alice' initiates the creation of a BEEP session using the IDXP profile with the entity 'Bob' by first contacting 'proxy1'. In the greeting exchange between 'Alice' and 'proxy1', the TUNNEL profile is selected, and subsequently the use of the TUNNEL profile is extended to reach through 'proxy2' to 'Bob'.



Notes:

(1) Instead of immediately acknowledging the request from 'Alice' to start TUNNEL, 'proxy1' attempts to establish use of TUNNEL with 'proxy2'. 'proxy2' also delays its acknowledgment to 'proxy1'.

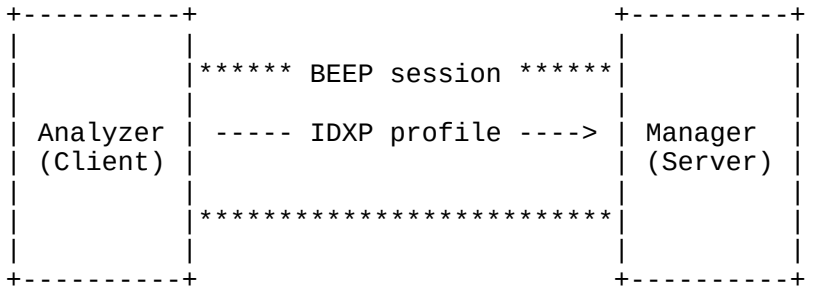
(2) 'Bob' acknowledges the request from 'proxy2' to start TUNNEL, and this acknowledgment propagates back to 'Alice' so that a TUNNEL application-layer tunnel is established from 'Alice' to 'Bob'.

2.2. Data Transfer

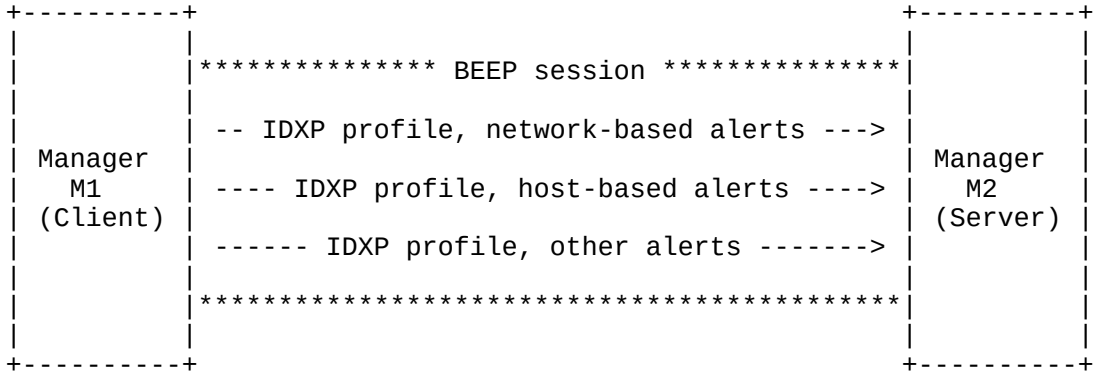
Between a pair of ID entities communicating over a BEEP session, one or more BEEP channels MAY be opened using the IDXP profile. If desired, additional BEEP sessions MAY be established to offer additional channels using the IDXP profile. However, in most situations additional channels using the IDXP profile SHOULD be opened within an existing BEEP session, as opposed to provisioning a new BEEP session containing the additional channels using the IDXP profile.

Peers assume the role of client or server on a per-channel basis, with one acting as the client and the other as the server. A peer's role of client or server is determined independent of whether the peer assumed the role of initiator or listener during the BEEP session establishment. Clients and servers act as sources and sinks, respectively, for exchanging data.

In a simple case, an analyzer peer sends data to a manager peer. For example,



Use of multiple BEEP channels in a BEEP session facilitates categorization and prioritization of data sent between IDXP peers. For example, a manager 'M1', sending alert data to another manager, 'M2', may choose to open a separate channel to exchange different categories of alerts. 'M1' would act as the client on each of these channels, and manager 'M2' can then process and act on the incoming alerts based on their respective channel categorizations. See Section 4 for more detail on how to incorporate categorization and/or prioritization into channel creation.



2.3. Connection Teardown

An IDXP peer may choose to close an IDXP channel under many different circumstances (e.g., an error in processing has occurred). To close a channel, the peer sends a "close" element (see Section 2.3.1.3 of [4]) on channel zero indicating which channel is being closed. An IDXP peer may also choose to close an entire BEEP session by sending a "close" element indicating that channel zero is to be closed.

Section 2.3.1.3 of [4] offers a more complete discussion of the circumstances under which a BEEP peer is permitted to close a channel and the mechanisms for doing so.

It is anticipated that due to the overhead of provisioning an application-layer tunnel and/or a BEEP security profile, BEEP sessions containing IDXP channels will be long-lived. In addition, the repeated overhead of IDXP channel provisioning (i.e., the exchange of IDXP greetings) may be avoided by keeping IDXP channels open even while data is not actively being exchanged on them. These are recommendations and, as such, IDXP peers may choose to close and re-provision BEEP sessions and/or IDXP channels as they see fit.

2.4. Trust Model

In our model, trust is placed exclusively in the IDXP peers. Proxies are always assumed to be untrustworthy. A BEEP security profile is used to establish end-to-end security between pairs of IDXP peers, doing away with the need to place trust in any intervening proxies. Only after successful negotiation of the underlying security profile are IDXP peers to be trusted. Only BEEP security profiles offering at least the protections required by Section 5 of [5] should be used to secure a BEEP session containing channels using the IDXP profile. See Section 3 of [4] for the registration of the TLS profile, an example of a BEEP security profile meeting the requirements of Section 5 of [5]. See Section 5 for a discussion of how IDXP fulfills the IDWG communications protocol requirements.

3. The IDXP Profile

3.1. IDXP Profile Overview

The IDXP profile provides a mechanism for exchanging information between intrusion detection entities. A BEEP tuning profile MAY be used to create an application-layer tunnel that transparently forwards data over a chain of proxies. The TUNNEL profile [3] SHOULD be used for this purpose; see [3] for more detail concerning the options available to set up an application-layer tunnel using TUNNEL, and see Section 11.1 for a discussion of TUNNEL-related security considerations. TUNNEL MUST be offered as a tuning profile for the creation of application-layer tunnels. The TUNNEL profile MUST offer the use of some form of SASL authentication (see Section 4.1 of [4]). The TLS profile SHOULD be used to provide the required combination of mutual-authentication, integrity, and confidentiality for the IDXP profile. For further discussion of application-layer tunnel and security issues, see Sections 2.1 and 11.

The IDXP profile supports several elements of interest:

- The "IDXP-Greeting" element identifies an analyzer or manager at one end of a BEEP channel to the analyzer or manager at the other end of the channel.

- The "Option" element is used to convey optional channel parameters between peers during the exchange of "IDXP-Greeting" elements. This element is OPTIONAL.

- The "IDMEF-Message" element carries the structured information to be exchanged between the peers.

3.2. IDXP Profile Identification and Initialization

The IDXP profile is identified as

   http://idxp.org/beep/profile

in the BEEP "profile" element during channel creation.

During channel creation, "IDXP-Greeting" elements MUST be mutually exchanged between the peers. An "IDXP-Greeting" element MAY be contained within the corresponding "profile" element in the BEEP "start" element. Including an "IDXP-Greeting" element in the initial "start" element has exactly the same semantics as passing it as the first "MSG" message on the channel. If channel creation is successful, then before sending the corresponding reply, the BEEP peer processes the "IDXP-Greeting" element and includes the resulting response in the reply. This response will be an "ok" element or an "error" element. The choice of which element is returned is dependent on local provisioning of the peer.

3.3. IDXP Profile Message Syntax

BEEP messages in the profile MUST have a MIME Content-Type [8] of "text/xml", "text/plain", or "application/octet-stream". The syntax of the individual elements is specified in Section 9.1 of this document and Section 4 of [2].

3.4. IDXP Profile Semantics

Each BEEP peer issues the "IDXP-Greeting" element using "MSG" messages. The "IDXP-Greeting" element MAY contain one or more "Option" sub-elements, conveying optional channel parameters. Each BEEP peer then issues "ok" in "RPY" messages or "error" in "ERR" messages. (See Section 2.3.1 of [4] for the definitions of the "error" and "ok" elements.) An "error" element MAY be issued within

a "RPY" message when piggy-backed within a BEEP "profile" element. See Section 3.4.1 for an example of an "error" element being issued within a "RPY" message. Based on the respective client/server roles negotiated during the exchange of "IDXP-Greeting" elements, the client sends data using "MSG" messages. Depending on the MIME Content-Type, this data may be an "IDMEF-Message" element, plain text, or binary. The server then issues "ok" in "RPY" messages or "error" in "ERR" messages.

3.4.1. The IDXP-Greeting Element

The "IDXP-Greeting" element serves to identify the analyzer or manager at one end of the BEEP channel to the analyzer or manager at the other end of the channel. The "IDXP-Greeting" element MUST include the role of the peer on the channel (client or server) and the Uniform Resource Identifier (URI) [6] of the peer. In addition, the "IDXP-Greeting" element MAY include the fully qualified domain name (see [9]) of the peer. One or more "Option" sub-elements MAY be present.

An "IDXP-Greeting" element MAY be sent by either peer at any time. The peer receiving the "IDXP-Greeting" MUST respond with an "ok" (indicating acceptance), or an "error" (indicating rejection). A peer's identity and role on a channel and any optional channel parameters are, in effect, specified by the most recent "IDXP-Greeting" it sent that was answered with an "ok".

An "IDXP-Greeting" may be rejected (with an "error" element) under many circumstances. These include, but are not limited to, authentication failure, lack of authorization to connect under the specified role, the negotiation of an inadequate cipher suite, or the presence of a channel option that must be understood but was unrecognized.

For example, a successful creation with an embedded "IDXP-Greeting" might look like this:

```
I: MSG 0 10 . 1592 187
I: Content-Type: text/xml
I:
I: <start number='1'>
I:   <profile uri='http://idxp.org/beep/profile'>
I:     <![CDATA[ <IDXP-Greeting uri='http://example.com/alice'
I:       role='client' /> ]]>
I:   </profile>
I: </start>
I: END
L: RPY 0 10 . 1865 91
```

```
L: Content-Type: text/xml
L:
L: <profile uri='http://idxp.org/beep/profile'>
L:   <![CDATA[ <ok /> ]]>
L: </profile>
L: END
L: MSG 0 11 . 1956 61
L: Content-Type: text/xml
L:
L: <IDXP-Greeting uri='http://example.com/bob' role='server' />
L: END
I: RPY 0 11 . 1779 7
I: Content-Type: text/xml
I:
I: <ok />
I: END
```

A creation with an embedded "IDXP-Greeting" that fails might look like this:

```
I: MSG 0 10 . 1776 185
I: Content-Type: text/xml
I:
I: <start number='1'>
I:   <profile uri='http://idxp.org/beep/profile'>
I:     <![CDATA[ <IDXP-Greeting uri='http://example.com/eve'
I:       role='client' /> ]]>
I:   </profile>
I: </start>
I: END
L: RPY 0 10 . 1592 182
L: Content-Type: text/xml
L:
L: <profile uri='http://idxp.org/beep/profile'>
L:   <![CDATA[
L:     <error code='530'>'http://example.com/eve' must first
L:       negotiate the TLS profile</error> ]]>
L: </profile>
L: END
```

3.4.2. The Option Element

If present, the "Option" element MUST be contained within an "IDXP-Greeting" element. An individual "IDXP-Greeting" element MAY contain one or more "Option" sub-elements. Each "Option" element within an "IDXP-Greeting" element represents a request to enable an IDXP option on the channel being negotiated. See Section 4 for a complete description of IDXP options and the "Option" element.

3.4.3. The IDMEF-Message Element

The "IDMEF-Message" element carries the information to be exchanged between the peers. See Section 4 of [2] for the definition of this element.

4. IDXP Options

IDXP provides a service for the reliable exchange of data between intrusion detection entities. Options are used to alter the semantics of the service.

The specification of an IDXP option MUST define

- the identity of the option;
- what content, if any, is contained within the option; and
- the processing rules for the option.

An option registration template (see Section 7) organizes this information.

An "Option" element is contained within an "IDXP-Greeting" element. The "IDXP-Greeting" element itself MAY contain one or more "Option" elements. The "Option" element has several attributes and contains arbitrary content:

- the "internal" and the "external" attributes, exactly one of which MUST be present, uniquely identify the option;
- the "mustUnderstand" attribute, whose presence is OPTIONAL and whose default value is "false", specifies whether the option, if unrecognized, MUST cause an error in processing to occur; and
- the "localize" attribute, whose presence is OPTIONAL, specifies one or more language tokens, each identifying a desirable language tag to be used if textual diagnostics are returned to the originator.

The value of the "internal" attribute is the IANA-registered name for the option. If the "internal" attribute is not present, then the value of the "external" attribute is a URI or URI with a fragment-identifier. Note that a relative-URI value is not allowed.

The "mustUnderstand" attribute specifies whether the peer may ignore the option if it is unrecognized. If the value of the "mustUnderstand" attribute is "true", and if the peer does not

recognize the option, then an error in processing has occurred. When absent, the value of the "mustUnderstand" attribute is defined to be "false".

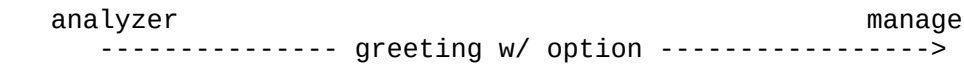
4.1. The channelPriority Option

Section 8.3 contains the IDXP option registration for the "channelPriority" option. This option contains a "channelPriority" element (see Section 9.2).

By default, IDXP does not place any requirements on how peers should manage multiple IDXP channels. The "channelPriority" option provides a way for peers using multiple IDXP channels to request relative priorities for each channel. When sending an "IDXP-Greeting" element during the provisioning of an IDXP channel, the originating peer MAY request that the remote peer assign a priority to the channel by including an "Option" element containing a "channelPriority" element.

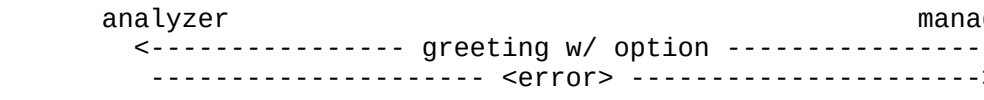
The "channelPriority" element has one attribute named "priority", of range 0..2147483647. This attribute is REQUIRED. Not coincidentally, this is the maximum range of possible BEEP channel numbers. 0 is defined to represent the highest priority, with relative priority decreasing as the "priority" value ascends.

For example, during the exchange of "IDXP-Greeting" elements during channel provisioning, an analyzer successfully requests that a manager assign a priority to the channel:




```
C: MSG 1 17 . 1984 165
C: Content-Type: text/xml
C:
C: <IDXP-Greeting uri='http://example.com/alice' role='client'>
C:   <Option internal='channelPriority'>
C:     <channelPriority priority='0' />
C:   </Option>
C: </IDXP-Greeting>
C: END
S: RPY 1 17 . 2001 7
S: Content-Type: text/xml
S:
S: <ok />
S: END
```

For example, during the exchange of "IDXP-Greeting" elements during channel provisioning, a manager unsuccessfully requests that an analyzer assign a priority to the channel:




```
S: MSG 1 17 . 1312 194
S: Content-Type: text/xml
S:
S: <IDXP-Greeting uri='http://example.com/bob' role='server'>
S:   <Option internal='channelPriority' mustUnderstand='true'>
S:     <channelPriority priority='2147483647' />
S:   </Option>
S: </IDXP-Greeting>
S: END
C: ERR 1 17 . 451 68
C: Content-Type: text/xml
C:
C: <error code='504'>'channelPriority' option was unrecognized</error>
C: END
```

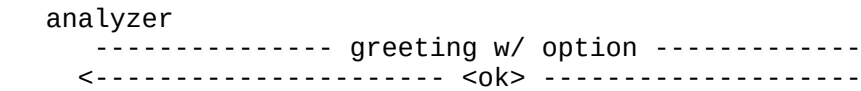
4.2. The streamType Option

Section 8.4 contains the IDXP option registration for the "streamType" option. This option contains a "streamType" element (see Section 9.3).

By default, IDXP provides no explicit method for categorizing channels. The "streamType" option provides a way for peers to request that a channel be categorized as a particular stream type. When sending an "IDXP-Greeting" element during the provisioning of an IDXP channel, the originating peer MAY request that the remote peer assign a stream type to the channel by including an "Option" element containing a "streamType" element.

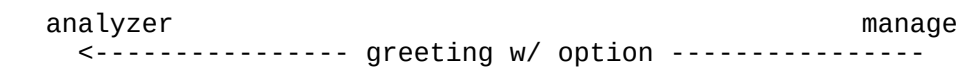
The "streamType" element has one attribute named "type", with the possible values of "alert", "heartbeat", or "config". This attribute is REQUIRED. A value of "alert" indicates that the channel should be categorized as being used for the exchange of ID alerts. A value of "heartbeat" indicates that the channel should be categorized as being used for the exchange of heartbeat messages such as the "Heartbeat" element (see Section 4 of [2]). A value of "config" indicates that the channel should be categorized as being used for the exchange of configuration messages.

For example, during the exchange of "IDXP-Greeting" elements during channel provisioning, an analyzer successfully requests that a manager assign a stream type to the channel:




```
C: MSG 1 21 . 1963 155
C: Content-Type: text/xml
C:
C: <IDXP-Greeting uri='http://example.com/alice' role='client'>
C:   <Option internal='streamType'>
C:     <streamType type='alert' />
C:   </Option>
C: </IDXP-Greeting>
C: END
S: RPY 1 21 . 1117 7
S: Content-Type: text/xml
S:
S: <ok />
S: END
```

For example, during the exchange of "IDXP-Greeting" elements during channel provisioning, a manager unsuccessfully requests that an analyzer assign a stream type to the channel:




```
S: MSG 1 21 . 1969 176
S: Content-Type: text/xml
S:
S: <IDXP-Greeting uri='http://example.com/bob' role='server'>
S:   <Option internal='streamType' mustUnderstand='true'>
S:     <streamType type='config' />
S:   </Option>
S: </IDXP-Greeting>
S: END
C: ERR 1 21 . 1292 63
C: Content-Type: text/xml
C:
C: <error code='504'>'streamType' option was unrecognized</error>
C: END
```

5. Fulfillment of IDWG Communications Protocol Requirements

The following lists each of the communications protocol requirements established in Section 5 of [5] and, for each requirement, describes the manner in which it is fulfilled. IDXP itself does not fulfill each of the communications protocol requirements, but instead relies on the underlying BEEP protocol and a variety of BEEP profiles.

5.1. Reliable Message Transmission

"The [protocol] MUST support reliable transmission of messages." See Section 5.1 of [5].

IDXP operates over BEEP, which operates only over reliable connection-oriented transport protocols (e.g., TCP). In addition, BEEP peers communicate using a simple request-response protocol, which provides end-to-end reliability between peers.

5.2. Interaction with Firewalls

"The [protocol] MUST support transmission of messages between ID components across firewall boundaries without compromising security." See Section 5.2 of [5].

The TUNNEL profile [3] MUST be offered as an option for creation of application-layer tunnels to allow operation across firewalls. The TUNNEL profile SHOULD be used to provide an application-layer tunnel. The ability to authenticate hosts during the creation of an application-layer tunnel MUST be provided by the mechanism chosen to create such tunnels. A firewall may therefore be configured to authenticate all hosts attempting to tunnel into the protected network. If the TUNNEL profile is used, SASL (see Section 4.1 of [4]) MUST be offered as a mechanism by which hosts can be authenticated.

5.3. Mutual Authentication

"The [protocol] MUST support mutual authentication of the analyzer and the manager to each other." See Section 5.3 of [5].

IDXP supports mutual authentication of the peers through the use of an appropriate underlying BEEP security profile. The TLS profile and members of the SASL family of profiles (see Section 4.1 of [4]) are examples of security profiles that may be used to authenticate the identity of communicating ID components. TLS MUST be offered as a mechanism to provide mutual authentication, and TLS SHOULD be used to provide mutual authentication.

5.4. Message Confidentiality

"The [protocol] MUST support confidentiality of the message content during message exchange. The selected design MUST be capable of supporting a variety of encryption algorithms and MUST be adaptable to a wide variety of environments." See Section 5.4 of [5].

IDXP supports confidentiality through the use of an appropriate underlying BEEP security profile. The TLS profile is an example of a security profile that offers confidentiality. The TLS profile with the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite MUST be offered as a mechanism to provide confidentiality, and TLS with this cipher suite SHOULD be used to provide confidentiality. The TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite uses ephemeral Diffie-Hellman (DHE) with DSS signatures for key exchange and triple DES (Data Encryption Standard) (3DES) and cipher-block chaining (CBC) for encryption. Stronger cipher suites are optional.

5.5. Message Integrity

"The [protocol] MUST ensure the integrity of the message content. The selected design MUST be capable of supporting a variety of integrity mechanisms and MUST be adaptable to a wide variety of environments." See Section 5.5 of [5].

IDXP supports message integrity through the use of an appropriate underlying BEEP security profile. The TLS profile and members of the SASL family of profiles (see Section 4.1 of [4]) are examples of security profiles that offer message integrity. The TLS profile with the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite MUST be offered as a mechanism to provide integrity, and TLS with this cipher suite SHOULD be used to provide integrity. The TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite uses the Secure Hash Algorithm (SHA) for integrity protection using a keyed message authentication code. Stronger cipher suites are optional.

5.6. Per-Source Authentication

"The [protocol] MUST support separate authentication keys for each sender." See Section 5.6 of [5].

IDXP supports separate authentication keys for each sender (i.e., per-source authentication) through the use of an appropriate underlying BEEP security profile. The TLS profile is an example of a security profile that supports per-source authentication through the mutual authentication of public-key certificates. TLS MUST be offered as a mechanism to provide per-source

authentication, and TLS SHOULD be used to provide per-source authentication.

5.7. Denial of Service

"The [protocol] SHOULD resist protocol denial-of-service attacks." See Section 5.7 of [5].

IDXP supports resistance to denial of service (DoS) attacks through the use of an appropriate underlying BEEP security profile. BEEP peers offering the IDXP profile MUST offer the use of TLS with the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite, and SHOULD use TLS with that cipher suite. To resist DoS attacks it is helpful to discard traffic arising from a non-authenticated source. BEEP peers MUST support the use of authentication in conjunction with any mechanism used to create application-layer tunnels. In particular, the use of some form of SASL authentication (see Section 4.1 of [4]) MUST be offered to provide authentication in the use of the TUNNEL profile. See Section 7 of [3] for a discussion of security considerations in the use of the TUNNEL profile.

5.8. Message Duplication

"The [protocol] SHOULD resist malicious duplication of messages." See Section 5.8 of [5].

IDXP supports resistance to malicious duplication of messages (i.e., replay attacks) through the use of an appropriate underlying BEEP security profile. The TLS profile is an example of a security profile offering resistance to replay attacks. The TLS profile with the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite MUST be offered as a mechanism to provide resistance against replay attacks, and TLS with this cipher suite SHOULD be used to provide resistance against replay attacks. The TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite uses cipher-block chaining (CBC) to ensure that even if a message is duplicated the cipher-text duplicate will produce a very different plain-text result. Stronger cipher suites are optional.

6. Extending IDXP

The specification of IDXP options (see Section 4) is the preferred method of extending IDXP. In order to extend IDXP, an IDXP option SHOULD be documented in an RFC and MUST be registered with the IANA (see Section 7). IDXP extensions that cannot be expressed as IDXP options MUST be documented in an RFC.

7. IDXP Option Registration Template

When an IDXP option is registered, the following information is supplied:

Option Identification: specify the NMTOKEN or the URI that authoritatively identifies this option.

Contains: specify the XML content that is contained within the "Option" element.

Processing Rules: specify the processing rules associated with the option.

Contact Information: specify the postal and electronic contact information for the author(s) of the option.

8. Initial Registrations

8.1. Registration: The IDXP Profile

Profile identification: http://idxp.org/beep/profile

Messages exchanged during channel creation: "IDXP-Greeting"

Messages starting one-to-one exchanges: "IDXP-Greeting", "IDMEF-Message"

Messages in positive replies: "ok"

Messages in negative replies: "error"

Messages in one-to-many exchanges: none

Message syntax: see Section 3.3

Message semantics: see Section 3.4

Contact information: see the "Authors' Addresses" section of this memo

8.2. Registration: The System (Well-Known) TCP Port Number for IDXP

Protocol Number: 603

Message Formats, Types, Opcodes, and Sequences: see Section 3.3

Functions: see Section 3.4

Use of Broadcast/Multicast: none

Proposed Name: Intrusion Detection Exchange Protocol

Short name: idxp

Contact Information: see the "Authors' Addresses" section of this memo

8.3. Registration: The channelPriority Option

Option Identification: channelPriority

Contains: channelPriority (see Section 9.2)

Processing Rules: see Section 4.1

Contact Information: see the "Authors' Addresses" section of this memo

8.4. Registration: The streamType Option

Option Identification: streamType

Contains: streamType (see Section 9.3)

Processing Rules: see Section 4.2

Contact Information: see the "Authors' Addresses" section of this memo

9. The DTDs

9.1. The IDXP DTD

The following is the DTD defining the valid elements for the IDXP profile.

```
<!--
DTD for the IDXP Profile

Refer to this DTD as:

  <!ENTITY % IDXP PUBLIC "-//IETF//DTD RFC 4767 IDXP v1.0//EN">

  %IDXP;
-->
```

```
<!-- Includes -->

  <!ENTITY % BEEP PUBLIC "-//IETF//DTD BEEP//EN">

  %BEEP;


  <!ENTITY % IDMEF-Message PUBLIC
                            "-//IETF//DTD RFC 4765 IDMEF v1.0//EN">

  %IDMEF;

<!--
  Profile Summary

    BEEP profile http://idxp.org/beep/profile

    role       MSG               RPY      ERR
    ====       ===               ===      ===
    I or L     IDXP-Greeting     ok       error
    C          IDMEF-Message     ok       error
-->

<!--
  Entity Definitions

        entity        syntax/reference     example
        ======        ================     =======
    an authoritative identification
        URI           see RFC 3986 [6]     http://example.com

    a fully qualified domain name
        FQDN          see RFC 1034 [9]     www.example.com
-->

<!ENTITY % URI      "CDATA">
<!ENTITY % FQDN     "CDATA">

<!--
  The IDXP-Greeting element declares the role and identity of
  the peer issuing it, on a per-channel basis. The
  IDXP-Greeting element may contain one or more Option
  sub-elements.
-->
```

```
<!ELEMENT IDXP-Greeting  (Option*)>
<!ATTLIST IDXP-Greeting
          uri            %URI;                #REQUIRED
          role           (client|server)      #REQUIRED
          fqdn           %FQDN;               #IMPLIED>

  <!--
    The Option element conveys an IDXP channel option.
    Note that the %LOCS entity is imported from the BEEP Channel
    Management DTD.
  -->

<!ELEMENT Option (ANY)>
<!ATTLIST Option
          internal       NMTOKEN              ""
          external       %URI;                ""
          mustUnderstand (true|false)         "false"
          localize       %LOCS;               "i-default">

  <!--
    The IDMEF-Message element conveys the intrusion detection
    information that is exchanged.  This element is defined in the
    idmef-message.dtd
  -->

<!-- End of DTD -->
```

9.2. The channelPriority Option DTD

The following is the DTD defining the valid elements for the channelPriority option.

```
  <!--
  DTD for the channelPriority IDXP option, as of 2002-01-08

  Refer to this DTD as:

    <!ENTITY % IDXP-channelPriority PUBLIC
      "-//IETF//DTD RFC 4767 IDXP-channelPriority v1.0//EN">

    %IDXP-channelPriority;
  -->

  <!--
    Entity Definitions

          entity        syntax/reference     example
          ======        ================     =======
```

```
       a priority number
            PRIORITY      0..2147483647        1
     -->

   <!ENTITY % PRIORITY          "CDATA">

   <!ELEMENT channelPriority    EMPTY>
   <!ATTLIST channelPriority
             priority           %PRIORITY    #REQUIRED>

   <!-- End of DTD -->
```

9.3. The streamType DTD

The following is the DTD defining the valid elements for the streamType option.

```
     <!--
     DTD for the streamType IDXP option, as of 2002-01-08

     Refer to this DTD as:

       <!ENTITY % IDXP-streamType PUBLIC
         "-//IETF//DTD RFC 4767 IDXP-streamType v1.0//EN">

       %IDXP-streamType;
     -->

     <!--
       Entity Definitions

             entity        syntax/reference                example
             ======        ================                =======
        a stream type
             STYPE         (alert | heartbeat | config)    "alert"
     -->

   <!ENTITY % STYPE        (alert|heartbeat|config)>

   <!ELEMENT streamType    EMPTY>
   <!ATTLIST streamType
             type          %STYPE    #REQUIRED>

   <!-- End of DTD -->
```

10. Reply Codes

This section lists the three-digit error codes the IDXP profile may generate.

code	meaning
421	Service not available (e.g., the peer does not have sufficient resources)
450	Requested action not taken (e.g., DNS lookup failed or connection could not be established. See also 550.)
454	Temporary authentication failure
500	General syntax error (e.g., poorly-formed XML)
501	Syntax error in parameters (e.g., non-valid XML)
504	Parameter not implemented
530	Authentication required
534	Authentication mechanism insufficient (e.g., cipher suite too weak, sequence exhausted)
535	Authentication failure
537	Action not authorized for user
550	Requested action not taken (e.g., peer could be contacted, but malformed greeting or no IDXP profile advertised)
553	Parameter invalid
554	Transaction failed (e.g., policy violation)

11. Security Considerations

The IDXP profile is a profile of BEEP. In BEEP, transport security, user authentication, and data exchange are orthogonal. Refer to Section 9 of [4] for a discussion of this. It is strongly recommended that those wanting to use the IDXP profile initially negotiate a BEEP security profile between the peers that offers the required security properties. The TLS profile SHOULD be used to provide for transport security. See Section 5 for a discussion of how IDXP fulfills the IDWG communications protocol requirements.

See Section 2.4 for a discussion of the trust model.

11.1. Use of the TUNNEL Profile

See Section 5 for IDXP's requirements on application-layer tunneling and the TUNNEL profile specifically. See Section 7 of [3] for a discussion of the security considerations inherent in the use of the TUNNEL profile.

11.2. Use of Underlying Security Profiles

At present, the TLS profile is the only BEEP security profile known to meet all of the requirements set forth in Section 5 of [5]. When securing a BEEP session with the TLS profile, the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite offers an acceptable level of security. See Section 5 for a discussion of how IDXP fulfills the IDWG communications requirements through the use of an underlying security profile.

12. IANA Considerations

The IANA registered "idxp" as a TCP port number as specified in Section 8.2.

The IANA maintains a list of:

IDXP options, see Section 7.

For this list, the IESG is responsible for assigning a designated expert to review the specification prior to the IANA making the assignment. As a courtesy to developers of non-standards track IDXP options, the mailing list idxp-discuss@lists.idxp.org may be used to solicit commentary.

IANA made the registrations specified in Sections 8.3 and 8.4.

13. References

13.1. Normative References

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

[2] Debar, H., Curry, D., and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)", RFC 4765, March 2007.

[3] New, D., "The TUNNEL Profile", RFC 3620, October 2003.

[4] Rose, M., "The Blocks Extensible Exchange Protocol Core", RFC 3080, March 2001.

[5] Wood, M. and M. Erlinger, "Intrusion Detection Message Exchange Requirements", RFC 4766, March 2007.

13.2. Informative References

[6] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[7] Bray, T., Paoli, J., Sperberg-McQueen, C. and E. Maler, "Extensible Markup Language (XML) 1.0 (2nd ed)", W3C REC-xml, October 2000, <http://www.w3.org/TR/REC-xml>.

[8] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[9] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

14. Acknowledgements

The authors gratefully acknowledge the contributions of Darren New, Marshall T. Rose, Roy Pollock, Tim Buchheim, Mike Erlinger, John C. C. White, and Paul Osterwald.

Authors' Addresses

Benjamin S. Feinstein
SecureWorks, Inc.
PO Box 95007
Atlanta, GA 30347
US

Phone: +1 404 327-6339
Email: bfeinstein@acm.org
URI: http://www.secureworks.com/

Gregory A. Matthews
CSC/NASA Ames Research Center

EMail: gmatthew@nas.nasa.gov
URI: http://www.nas.nasa.gov/

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.