Network Working Group B. Feinstein
Internet-Draft G. Matthews
Expires: November 21, 2001 Harvey Mudd College
J. White
MITRE Corporation
May 23, 2001

The Intrusion Detection Exchange Protocol (IDXP)
draft-ietf-idwg-beep-idxp-02

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 the Intrusion Detection Message Exchange Format (IDMEF) [2], a companion document of the Intrusion Detection Exchange Format (IDWG) working group of the IETF.

Table of Contents

1. Introduction 3
1.1 Purpose 3
1.2 Profiles 3
1.3 Terminology 3
2. The Model 5
2.1 Connection Provisioning 5
2.2 Data Transfer 7
2.3 Trust Model 8
3. The IDXP Profile 9
3.1 IDXP Profile Overview 9
3.2 IDXP Profile Identification and Initialization 9
3.3 IDXP Profile Message Syntax 10
3.4 IDXP Profile Semantics 10
3.4.1 The IDXP-GREETING Element 10
3.4.2 The OPTIONS Element 12
3.4.3 The IDMEF-MESSAGE Element 12
4. The IDXP DTD 13
5. Reply Codes 15
6. Fulfillment of IDWG Communications Protocol Requirements . . 16
7. Security Considerations 19
7.1 Use of the TUNNEL Profile 19
7.2 Use of Underlying Security Profiles 19
References 20
Authors' Addresses 20
A. IANA Considerations 22
B. History of Significant Changes 23
B.1 Significant Changes Since beep-idxp-01 23
B.2 Significant Changes Since beep-idxp-00 23
C. Acknowledgements 24
Full Copyright Statement 25

1. Introduction

IDXP is specified, in part, as a Blocks Extensible Exchange Protocol (BEEP) [6] "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 6 for a discussion of how IDXP fulfills the IDWG communication protocol requirements. See Section 7 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 [5]

The Simple Authentication and Security Layer (SASL) Family of Profiles (see Section 4.1 of [6])

The TLS Profile (see Section 3.1 of [6])

1.3 Terminology

Throughout this memo, the terms "analyzer" and "manager" are used in the context of the Intrusion Detection Message Exchange Requirements [7]. In particular, Section 3.2 of [7] defines the meaning of a collection of intrusion detection terms.

The terms "peer", "initiator", "listener", "client", and "server" are used in the context of BEEP [6]. In particular, Section 2.1 of the BEEP framework memo discusses the roles that a BEEP peer may perform.

Note that the terms "endpoint" and "proxy" are specific to IDXP, and do 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 endpoints. Endpoints can exist only in pairs, and these pairs communicate over a single BEEP session with one or more BEEP channels opened for transferring data. Endpoints are either managers or analyzers, as defined in Section 3.2 of [7].

The relationship between analyzers and managers is potentially many-to-many. I.e., an analyzer might communicate with many managers; similarly, a manager might communicate with many analyzers. Likewise, the relationship between different managers is potentially many-to-many, so that a manager can receive the alerts sent by a large number of analyzers by receiving them through intermediate managers. Analyzers are not permitted to establish IDXP exchanges with other analyzers.

An ID entity wishing to establish IDXP communications with another ID entity does so by initiating a BEEP session. A BEEP security profile offering the required security properties should initially be negotiated (see Section 7 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 an ID entity 'initial' initiates an IDXP exchange with the entity 'final'.

```
    initial                                           final
       ---------------- xport connect[1] ------------------>
      <-------------------- greeting ---------------------->
      <-------------start security profile[2] ------------->
      <-------------------- greeting ---------------------->
      <------------------ start IDXP[3] ------------------->
```

Notes:

[1] 'initial' initiates a transport connection to 'final', 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 endpoints 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 endpoints in the department onto a company-wide manager endpoint.

A BEEP tuning profile should be used to create an application-layer tunnel that transparently forwards data over a chain of proxies. The TUNNEL profile [5] SHOULD be used for this purpose; see [5] for more detail concerning the options available to setup an application-layer tunnel using TUNNEL, and see Section 7.1 for a discussion of TUNNEL related security considerations. TUNNEL MUST be offered as a tuning profile for the creation of application-layer tunnels. Once a tunnel has been created a BEEP security profile offering the required security properties would be negotiated, followed by negotiation of the IDXP profile.

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

```
initial            proxy1              proxy2              final
  -- xport connect -->
 <---- greeting ----->
  -- start TUNNEL --->
                      - xport connect[1] ->
                     <----- greeting ----->
                      --- start TUNNEL --->
                                           --- xport connect -->
                                          <----- greeting ----->
                                           --- start TUNNEL --->
                                          <----- <ok>[2] ------
                     <------- <ok> -------
 <------ <ok> -------
 <------------------------- greeting -------------------------->
 <------------------ start security profile ------------------->
 <------------------------- greeting -------------------------->
 <------------------------ start IDXP ------------------------->
```

Notes:

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

[2] 'final' acknowledges the request from 'proxy2' to start TUNNEL, and this acknowledgment propagates back to 'initial' so that a TUNNEL application-layer tunnel is established from 'initial' to 'final'.

2.2 Data Transfer

Between a pair of ID entities communicating over a BEEP session, zero or more BEEP channels may be open using the IDXP profile. If necessary, additional BEEP sessions may be provisioned 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.

Endpoints assume the role of client or server on a per-channel basis, with one acting as the client and the other as the server. An endpoint's role of client or server is determined independent of whether the endpoint 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 endpoint sends data to a manager endpoint. E.g.,

```
        +----------+                          +---------+
        |          |                          |         |
        |          |****** BEEP session ******|         |
        |          |                          |         |
        | Analyzer | ----- IDXP profile ----> | Manager |
        |          |                          |         |
        |          |**************************|         |
        |          |                          |         |
        +----------+                          +---------+
```

Note that the arrowhead for the BEEP channel using the IDXP profile points from client to server.

Use of multiple BEEP channels in a BEEP session facilitates categorization and prioritization of data sent between IDXP endpoints. 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 3.4.2 for more detail on how to

incorporate categorization and/or prioritization into channel creation.

```
+---------+                                            +---------+
|         |                                            |         |
|         |*************** BEEP session ***************|         |
|         |                                            |         |
| Manager |  -- IDXP profile, network-based alerts --->| Manager |
|         |                                            |         |
|   M1    |  ---- IDXP profile, host-based alerts ---->|   M2    |
|         |                                            |         |
|         |  ------ IDXP profile, other alerts ------->|         |
|         |                                            |         |
|         |********************************************|         |
|         |                                            |         |
+---------+                                            +---------+
```

2.3 Trust Model

In our model, trust is placed exclusively in the endpoints. Proxies are always assumed to be untrustworthy. A BEEP security profile is used to establish end-to-end security between pairs of IDXP endpoints, doing away with the need to place trust in any intervening proxies. Only after successful negotiation of the underlying security profile are IDXP endpoints to be trusted. Only BEEP security profiles offering at least the protections required by Section 6 of [7] should be used to secure a BEEP session containing channels using the IDXP profile. See Section 3 of [6] for the registration of the TLS profile, an example of a BEEP security profile meeting the requirements of Section 6 of [7]. See Section 6 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. The TUNNEL profile SHOULD be used to provision a BEEP session running the IDXP profile over an application-layer tunnel. 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 Section 2.1 and Section 7.

The IDXP profile supports two 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 "Options" element is used to convey optional channel parameters between peers during the exchange of "IDXP-Greeting" elements.

- 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://www.idxp.org/profile

in the BEEP "profile" element during channel creation.

During channel creation, the corresponding "profile" element in the BEEP "start" element may contain an "IDXP-Greeting" element. 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 server. 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.

3.3 IDXP Profile Message Syntax

BEEP messages in the profile may have a MIME Content-Type [3] of text/xml, text/plain, or application/octet-stream. The syntax of the individual elements is specified in Section 4 and [2]'s Section 5.

3.4 IDXP Profile Semantics

Each BEEP peer issues the "IDXP-Greeting" element using "MSG" messages. The "IDXP-Greeting" element may contain an "Options" sub-element, conveying optional channel parameters. Each BEEP peer then issues "ok" in "RPY" messages or "error" in "ERR" messages. (See Section 2.3.1 of [6] for the definitions of the "error" and "ok" elements.) 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 an 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 includes the role of the peer on the channel (client or server) and the Uniform Resource Identifier (URI) [1] of the peer. Additionally, the "IDXP-Greeting" element may include a combination of the fully qualified domain name (see [4]) and IP address of the peer. The IP address chosen should be the IP address associated with the underlying transport protocol carrying the channel. An "Options" sub-element may be present. A peer may disregard the "Options" sub-element, if desired.

An "IDXP-Greeting" element may be sent by either peer at any time. The peer receiving the "IDXP-Greeting" responds 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" could be rejected (with an "error" element) if the security that has been negotiated is inadequate, if the authenticated peer does not have authorization to connect as the specified type or to serve in the specified role, or if a required optional parameter was not present.

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

```
I: MSG 0 10 . 1592 210
I: Content-Type: text/xml
I:
I: <start number='1'>
I:   <profile uri='http://www.idxp.org/profile'>
I:     <![CDATA[ <IDXP-Greeting uri='http://example.com/alice'
I:        role='client' /> ]]>
I:   </profile>
I: </start>
I: END
L: RPY 0 10 . 1865 111
L: Content-Type: text/xml
L:
L: <profile uri='http://www.idxp.org/profile'>
L:   <![CDATA[ <ok /> ]]>
L: </profile>
L: END
L: MSG 0 11 . 1976 88
L: Content-Type: text/xml
L:
L: <IDXP-Greeting uri='http://example.com/bob' role='server' />
L: END
I: RPY 0 11 . 1802 34
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 208
I: Content-Type: text/xml
I:
I: <start number='1'>
I:   <profile uri='http://www.idxp.org/profile'>
I:     <![CDATA[ <IDXP-Greeting uri='http://example.com/eve'
I:        role='client' /> ]]>
I:   </profile>
I: </start>
I: END
L: RPY 0 10 . 1592 204
L: Content-Type: text/xml
L:
L: <profile uri='http://www.idxp.org/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 OPTIONS Element

The "Options" element contains any optional channel parameters listed in a peer's "IDXP-Greeting" element. Optional channel parameters may include categorization of alert data sent on a channel, or relative priority of data sent on a particular channel. Peers may ignore "Options" elements they receive within the "IDXP-Greeting" elements sent by remote peers. Arbitrary content may be nested within the "Options" element to convey optional channel parameters during IDXP channel greeting.

3.4.3 The IDMEF-MESSAGE Element

The "IDMEF-Message" element carries the information to be exchanged between the peers. See Section 5 of [2] for the definition of this element.

4. The IDXP DTD

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

```
<!--
DTD for the IDXP Profile, as of 2001-05-10

Refer to this DTD as:

  <!ENTITY % IDXP PUBLIC "-//Blocks//DTD IDXP//EN"
        "http://www.idxp.org/profile/idxp.dtd">

  %IDXP;
-->

<!-- Includes -->

  <!ENTITY % BEEP PUBLIC "-//Blocks//DTD BEEP//EN"
             "http://xml.resource.org/profiles/BEEP/beep.dtd">

  %BEEP;


  <!ENTITY % IDMEF PUBLIC "-//IETF//DTD RFCxxxx IDMEF v0.3//EN"
             "/some/path/to/the/idmef-message.dtd">

  %IDMEF;

<!--
  Profile Summary

    BEEP profile http://www.idxp.org/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-2396]       http://example.com
```



```
    a fully qualified domain name
        FQDN          See [RFC-1034]       www.example.com

    a dotted-quad IP address
        IP            1*3DIGIT "." 1*3DIGIT "."
                       1*3DIGIT "." 1*3DIGIT
                                           10.0.0.27
-->

<!ENTITY % URI      "CDATA">
<!ENTITY % FQDN     "CDATA">
<!ENTITY % IP       "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 an Options sub-element.
-->

<!ELEMENT IDXP-Greeting  (Options)>
<!ATTLIST IDXP-Greeting
          uri           %URI;                  #REQUIRED
          role          (client|server)        #REQUIRED
          fqdn          %FQDN;                 #IMPLIED
          ip            %IP;                   #IMPLIED>

<!--
  The Options element may contain any optional channel parameters.
-->

<!ELEMENT Options (ANY)>

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

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


5. 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, etc.)
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)

6. Fulfillment of IDWG Communications Protocol Requirements

The following lists the communications protocol requirements established in Section 6 of [7] and, for each requirement, describes the manner in which it is fulfilled by IDXP.

- The [protocol] MUST support reliable transmission of messages.

  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.

- The [protocol] MUST support transmission of messages between ID components across firewall boundaries without compromising security.

  The TUNNEL profile [5] 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 [6]) MUST be offered as a mechanism by which hosts can be authenticated.

- The [protocol] MUST support mutual authentication of the analyzer and the manager to each other.

  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 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 athentication.

- 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.

  IDXP supports confidentiality through the use of an appropriate underlying BEEP security profile. The TLS profile is an example 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 (3DES) and cipher-block chaining (CBC) for encryption. Stronger cipher suites are optional.

- 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.

  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 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.

- The [protocol] SHOULD be able to ensure non-repudiation of the origin of IDMEF messages.

  IDXP supports non-repudiation of message origin through the use of an appropriate underlying BEEP security profile. The TLS profile is an example of a security profile that offers non-repudiation of message origin through the authentication of public-key certificates. TLS MUST be offered as a mechanism to provide non-repudiation of message origin, and TLS SHOULD be used to provide non-repudiation of message origin.

- The [protocol] SHOULD resist protocol denial of service attacks.

  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 MUST be offered to provide authentication in the use of the TUNNEL profile. See Section 7 of [5] for a discussion of security considerations in the use of the TUNNEL profile.

- The [protocol] SHOULD resist malicious duplication of messages.

  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.

7. Security Considerations

The IDXP profile is a profile of BEEP. In BEEP, transport security, user authentication, and data exchange are orthogonal. Refer to Section 8 of [6] 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 6 for a discussion of how IDXP fulfills the IDWG communications protocol requirements.

See Section 2.3 for a discussion of the trust model.

7.1 Use of the TUNNEL Profile

See Section 6 for IDXP's requirements on application-layer tunneling and the TUNNEL profile specifically. See Section 7 of [5] for a discussion of the security considerations inherent in the use of the TUNNEL profile.

7.2 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 6 of [7]. 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 6 for a discussion of how IDXP fulfills the IDWG communications requirements through the use of an underlying security profile.

References

[1] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[2] Curry, D. and H. Debar, "Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition", February 2001, <draft-ietf-idwg-idmef-xml-03 (work in progress)>.

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

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

[5] New, D., "The TUNNEL Profile Registration", February 2001, <draft-ietf-idwg-beep-tunnel-01 (work in progress)>.

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

[7] Wood, M. and M. Erlinger, "Intrusion Detection Message Exchange Requirements", February 2001, <draft-ietf-idwg-requirements-05 (work in progress)>.

Authors' Addresses

Benjamin S. Feinstein
Harvey Mudd College

EMail: me@benfeinstein.net
URI: http://benfeinstein.net/

Gregory A. Matthews
Harvey Mudd College

EMail: gmatthew@cs.hmc.edu
URI: http://www.cs.hmc.edu/

John C. C. White
MITRE Corporation

EMail: jccw@mitre.org
URI: http://www.mitre.org/

Appendix A. IANA Considerations

The IANA adds the following to its registry of BEEP profiles, upon approval of this document by the IESG:

Profile identification: http://www.idxp.org/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 of this document.
Message semantics: See Section 3.4 of this document.
Contact information: See the "Authors' Addresses" appendix of this document.

Appendix B. History of Significant Changes

B.1 Significant Changes Since beep-idxp-01

Added new MUST and SHOULD language for use of TLS and TUNNEL profiles.

Modified the "IDXP-Greeting" element to include an "Options" sub-element.

Changed IDXP profile URI.

B.2 Significant Changes Since beep-idxp-00

Added Section 6, describing how IDXP fulfills the communication protocol requirements of the IDWG.

Moved IDXP profile registration to Appendix A.

Clarified the role that underlying BEEP security profiles must play.

Clarified how IDMEF messages fit into IDXP.

Clarified how the IDXP profile channels and BEEP sessions interact.

Made terminology clarifications and changes for overall consistency.

Appendix C. Acknowledgements

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

Acknowledgement

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